Applied Energy 247 (2019) 270–284

Contents lists available at ScienceDirect

Applied Energy

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

Wind speed prediction method using Shared Weight Long Short-Term Memory Network and Gaussian Process Regression

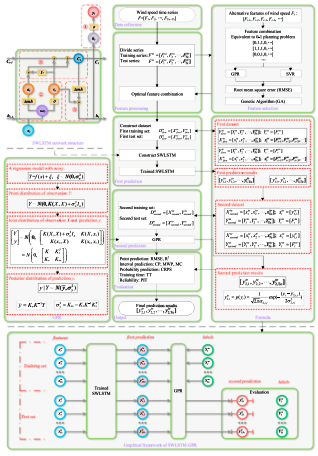
Zhendong Zhang[a], Lei Ye[b], Hui Qin[a,*], Yongqi Liu[a], Chao Wang[c], Xiang Yu[d], Xingli Yin[a], Jie Li[a]

[a] School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, China
[b] School of Hydraulic Engineering, Dalian University of Technology, Dalian, Liaoning, China
[c] Department of Water Resources, China Institute of Water Resources and Hydropower Research, Beijing, China
[d] Provincial Key Laboratory for Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang, Jiangxi, China

HIGHLIGHTS

- The training time of Long Short-Term Memory Network is reduced by sharing weights.
- A new hybrid model is proposed for wind speed probabilistic forecasting.
- Four experiments are designed to verify the performance and reliability of the model.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Wind speed prediction
Long Short-Term Memory Network
Gaussian Process Regression
Shared weight
Forecast uncertainty

ABSTRACT

Wind energy has received more and more attention around the world since it is a kind of clean, economical and renewable energy. However, the strong randomness of the wind speed makes wind power difficult to integrate into the power grid. Obtaining reliable high-quality wind speed prediction results is very important for the planning and application of wind energy. In this study, Shared Weight Long Short-Term Memory Network (SWLSTM) is proposed to decrease the number of variables that need to be optimized and the training time of Long Short-Term Memory Network (LSTM) without significantly reducing prediction accuracy. Furthermore, a new hybrid model combined SWLSTM and GPR, called SWLSTM-GPR, is proposed to obtain reliable wind speed probabilistic prediction result. SWLSTM-GPR is applied to four wind speed prediction cases in Inner Mongolia, China and compared with the state-of-the-art wind speed prediction methods from four aspects: point prediction accuracy, interval prediction suitability, probability prediction comprehensive performance and training time. The reliability test of SWLSTM-GPR guarantees that the prediction results are reliable and convincing. The experimental results show that SWLSTM-GPR can obtain high-precision point prediction, appropriate prediction

* Corresponding author at: School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail address: hqin@hust.edu.cn (H. Qin).

https://doi.org/10.1016/j.apenergy.2019.04.047
Received 14 August 2018; Received in revised form 6 March 2019; Accepted 13 April 2019

interval and reliable probabilistic prediction results with shorter training time on the wind speed prediction problems.

1. Introduction

Wind energy is a kind of clean, economical and renewable energy [1]. Wind speed is the most influential factor in wind power generation systems. However, the randomness of the wind speed is very strong, making it difficult to integrate wind power into the power grid [2]. Increasing the accuracy of wind speed prediction is conducive to the scheduling of wind power generation systems and obtaining reliable prediction uncertainty information is also beneficial to avoid risks in planning. Therefore, obtaining wind speed point prediction results with high accuracy, suitable prediction interval and probability distribution function (PDF) with high reliability is very important for the application of wind energy.

The wind speed prediction methods can be mainly divided into two categories: physical process driven method and data driven method [3]. The physical process driven method needs to collect meteorological data including humidity, temperature, pressure, wind speed, wind direction and terrain data, such as numeric weather prediction (NWP) models [4]. These models interpret the causes of wind speed generation by establishing complex mathematical physics models and simulate the wind formation process to predict wind speed [5]. The advantage of these methods is that the prediction accuracy is high and the interpretability is strong. The disadvantages are difficult data collection, complex modeling and time-consuming solution. The data driven methods use statistically relevant methods to predict wind speed from historical wind speed data, such as time series models, machine learning models and hybrid models [6]. Time series models mainly include Moving Average model (MA), Auto-regressive model (AR), Auto-regressive Moving Average model (ARMA) and their variants [7]. These time series models require data stationarity assumptions [8], whose prediction accuracy is limited due to the strong nonlinearity of wind speed. In order to deal with the nonlinearity of wind speed, many machine learning methods are used to predict wind speed, such as Gaussian Process Regression (GPR) [9], Support Vector Regression (SVR) [10], Quantile Regression (QR) [11] and Artificial Neural Networks (ANN) [12]. Studying the variants [13] of these methods, optimizing the parameters [14] of these methods, and introducing new machine learning methods [15] to predict wind speed are the research directions of this kind of methods. In order to further enhance the performance of prediction, some new hybrid methods are used to predict wind speed, include mixing different prediction methods and mixing prediction method and data processing method. ARIMA-ANN hybrid model [16], ANN and SVR hybrid model [17] are the examples of the former. The latter is the more common hybrid model, such as Empirical Model Decomposition (EMD) and machine learning models (ANN and SVR) [18], Wavelet Decomposition (WD) and ARMA [19], Two-stage Decomposition and AdaBoost-extreme learning machine [20]. These models focus on the accuracy of wind speed prediction and seldom consider the uncertainty of wind speed prediction.

In recent years, the deep learning method has received extensive attention because of its excellent performance in predicting accuracy [21]. In deep learning, Recurrent Neural Networks (RNN) is suitable for dealing with sequence problems like time series data, however it has long-term dependence problems when the sequence length becomes long [22]. Long Short-Term Memory Network (LSTM) [23] is proposed to solve this problem by adding input gates, output gates and forget gates to the RNN. There are two most important variants of LSTM, one is adding Peephole Connections to LSTM to improve prediction accuracy [24], and the other is Gated Recurrent Unit (GRU) that simplifies the gate structure of LSTM to reduce training time [25]. The variants of LSTM are mainly improved in terms of prediction accuracy and training time. In fact, LSTM has performed well in the accuracy of wind speed prediction [26], it is difficult to make a qualitative breakthrough by adding complex structure to LSTM. In contrast, the literature [27] indicates that GRU can achieve prediction accuracy close to LSTM in less time. Therefore, it is one of the focuses of this study to further reduce the number of variables that need to be optimized (NVNO) and training time of LSTM without significantly reducing prediction accuracy. The idea of this study is to propose a new method, Shared Weight Long Short-Term Memory Network (SWLSTM), which not change the gate structure of the standard LSTM, but share the weight and bias of the gate structure. Currently, most of the improvement of wind speed prediction methods focus on improving the prediction accuracy, and there are relatively few methods to obtain the prediction uncertainty information. It is another focuses of this study to propose a new hybrid method based on SWLSTM and probability prediction method to obtain prediction uncertainty information. The idea of this study is combining SWLSTM and GPR since GPR is a highly theoretical probability prediction method and can obtain reliable PDF, called SWLSTM-GPR.

In this study, a new hybrid method called SWLSTM-GPR is proposed to predict wind speed. The main contributions are outlined as follows:

(1) All the same types of weights in LSTM are shared, called SWLSTM, to decrease NVNO and training time without significantly reducing wind speed prediction accuracy.
(2) A new hybrid model combined SWLSTM and GPR, called SWLSTM-GPR, is proposed to obtain wind speed probabilistic prediction results.
(3) Four wind speed prediction cases in Inner Mongolia, China are used to test SWLSTM-GPR from five aspects: point prediction accuracy, interval prediction suitability, probability prediction comprehensive performance, forecast reliability and training time. The experimental results show that SWLSTM-GPR can obtain high-precision point prediction, appropriate prediction interval and reliable probabilistic prediction results with shorter training time.

The remainder of this paper is organized as follows. In Section 2, the implementation details of the complete SWLSTM-GPR are introduced. In Section 3, performance and reliability evaluation metrics are explained. In Section 4, SWLSTM-GPR is applied to the wind speed prediction case in Inner Mongolia, China. In Section 5, we summarize the work of this paper and give our conclusions.

2. Methodology

2.1. Shared Weight Long Short Term Memory Network (SWLSTM)

2.1.1. Forward propagation of SWLSTM

In order to reduce the number of variables that need to be optimized (NVNO) and training time of LSTM, the Shared Weight Long Short Term Memory Network (SWLSTM) is proposed. SWLSTM combines input gates, output gates and forget gates into one new gate structure called shared gates. The SWLSTM network structure is shown in the Fig. 1.

The forward propagation process and formula of SWLSTM in the t-th period are shown as follows:

(1) Calculate shared gates and information state

$$net_t = w_h \cdot h_{t-1} + w_x \cdot x_t + b \tag{1}$$

$$s_t = \sigma(net_t) = \sigma(w_h \cdot h_{t-1} + w_x \cdot x_t + b) \tag{2}$$

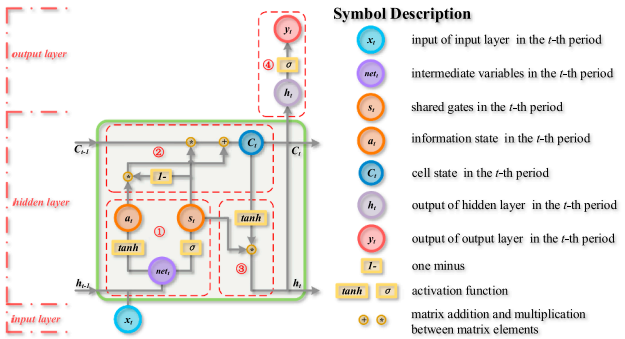


Fig. 1. Schematic diagram of SWLSTM network structure.

$$a_t = \tanh(net_t) = tanh(w_h \cdot h_{t-1} + w_x \cdot x_t + b) \tag{3}$$

(2) Update cell state

$$C_t = s_t * C_{t-1} + (1 - s_t) * a_t \tag{4}$$

(3) Calculate output of the hidden layer

$$h_t = s_t * \tanh(C_t) \tag{5}$$

(4) Output predicted value of output layer

$$y_t = \sigma(z_t) = \sigma(w_y \cdot h_t + b_y) \tag{6}$$

In the above formula and figure, x_t, s_t and a_t are the input of input layer, the shared gates and the information state in the current period, respectively. C_{t-1} and C_t represent for the cell state in the previous period and current period. h_{t-1} and h_t stand for the outputs of hidden layer in the previous period and current period. y_t is the predicted value of current period. net_t and z_t are all intermediate variables. $[w_h, w_x, b]$ and $[w_y, b_y]$ are two sets of weight variables need to be optimized. The symbol $\cdot$ indicates matrix multiplication and the symbol $*$ indicates multiplication between matrix elements. $\sigma(x)$ and $\tanh(x)$ are activation function of *Sigmoid* and *Tanh*. Its calculation formulas and its derivative formulas are as follows:

$$\begin{cases} \sigma(x) = y = \frac{1}{1+e^{-x}} \\ \sigma'(x) = y(1-y) \end{cases} \quad \begin{cases} \tanh(x) = y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \tan h'(x) = 1 - y^2 \end{cases} \tag{7}$$

2.1.2. Back propagation of SWLSTM

Because SWLSTM changed the network structure of the original LSTM, the backpropagation formula of SWLSTM needed to be re-derived. The processes and formulas of error back propagation in the t-th period are derived as follows:

(1) First, the most common squared error function is used as the target to be optimized.

$$E_t = \frac{1}{2}(y_t - Y_t)^2 \tag{8}$$

where E_t denotes the error in the t-th period. y_t and Y_t are predictions and observations in the t-th period, respectively. We minimize E_t via gradient descent by adding weight changes $[\delta w_h, \delta w_x, \delta b]$ and $[\delta w_y, \delta b_y]$ to the weights $[w_h, w_x, b]$ and $[w_y, b_y]$ using learning rate η. So the purpose of back propagation is to calculate $[\delta w_h, \delta w_x, \delta b]$ and $[\delta w_y, \delta b_y]$.

(2) Then, calculate the error of each variable in the output layer.

$$\delta y_t = \frac{\partial E_t}{\partial y_t} = y_t - Y_t \tag{9}$$

$$\delta z_t = \frac{\partial E_t}{\partial z_t} = \frac{\partial E_t}{\partial y_t}\frac{\partial y_t}{\partial z_t} = \delta y_t * [y_t * (1 - y_t)] \tag{10}$$

$$\delta w_y = \frac{\partial E_t}{\partial w_y} = \frac{\partial E_t}{\partial z_t}\frac{\partial z_t}{\partial w_y} = \delta z_t \cdot h_t \tag{11}$$

$$\delta b_y = \frac{\partial E_t}{\partial b_y} = \frac{\partial E_t}{\partial z_t}\frac{\partial z_t}{\partial b_y} = \delta z_t \cdot 1 = \delta z_t \tag{12}$$

(3) Next, calculate the error of each variable in the hidden layer.

$$\delta h_t = \frac{\partial E_t}{\partial h_t} = \begin{cases} \frac{\partial E_t}{\partial z_t}\frac{\partial z_t}{\partial h_t} = \delta z_t \cdot w_y & t = T \\ \frac{\partial E_t}{\partial z_t}\frac{\partial z_t}{\partial h_t} + \frac{\partial E_t}{\partial net_{t+1}}\frac{\partial net_{t+1}}{\partial h_t} & t \neq T \\ = \delta z_t \cdot w_y + \delta net_{t+1} \cdot w_h \end{cases} \tag{13}$$

$$\delta C_t = \frac{\partial E_t}{\partial C_t} = \begin{cases} \frac{\partial E_t}{\partial h_t}\frac{\partial h_t}{\partial C_t} = \delta h_t * s_t * [1 - tanh^2(C_t)] & t = T \\ \frac{\partial E_t}{\partial h_t}\frac{\partial h_t}{\partial C_t} + \frac{\partial E_t}{\partial C_{t+1}}\frac{\partial C_{t+1}}{\partial C_t} & t \neq T \\ = \delta h_t * s_t * [1 - tanh^2(C_t)] + \delta C_{t+1} * s_t \end{cases} \tag{14}$$

where T is the last period. δnet_{t+1} will be solved in the next step.

$$\delta a_t = \frac{\partial E_t}{\partial a_t} = \frac{\partial E_t}{\partial C_t}\frac{\partial C_t}{\partial a_t} = \delta C_t * (1 - s_t) \tag{15}$$

$$\begin{aligned} \delta s_t &= \frac{\partial E_t}{\partial s_t} = \frac{\partial E_t}{\partial h_t}\frac{\partial h_t}{\partial s_t} + \frac{\partial E_t}{\partial C_t}\frac{\partial C_t}{\partial s_t} \\ &= \delta h_t * \tanh(C_t) + \delta C_t * (C_{t-1} - a_t) \end{aligned} \tag{16}$$

$$\begin{aligned} \delta net_t &= \frac{\partial E_t}{\partial net_t} = \frac{\partial E_t}{\partial a_t}\frac{\partial a_t}{\partial net_t} + \frac{\partial E_t}{\partial s_t}\frac{\partial s_t}{\partial net_t} \\ &= \delta a_t * (1 - a_t^2) + \delta s_t * [s_t * (1 - s_t)] \end{aligned} \tag{17}$$

$$\delta w_h = \frac{\partial E_t}{\partial w_h} = \frac{\partial E_t}{\partial net_t}\frac{\partial net_t}{\partial w_h} = \delta net_t \cdot h_{t-1} \tag{18}$$

$$\delta w_x = \frac{\partial E_t}{\partial w_x} = \frac{\partial E_t}{\partial net_t}\frac{\partial net_t}{\partial w_x} = \delta net_t \cdot x_t \tag{19}$$

$$\delta b = \frac{\partial E_t}{\partial b} = \frac{\partial E_t}{\partial net_t}\frac{\partial net_t}{\partial b} = \delta net_t \cdot 1 = \delta net_t \tag{20}$$

At this point, both $[\delta w_h, \delta w_x, \delta b]$ and $[\delta w_y, \delta b_y]$ are solved. Through the above formulas, the training algorithm in deep learning can be used to optimize the weights $[w_h, w_x, b]$ and $[w_y, b_y]$ to minimize the error E_t. In this study, Adaptive Momentum Estimation method (Adam) [28] is recommended as the optimization algorithm because it has strong robustness in adaptively adjusting the learning rate of different parameters and gradually becomes the most popular neural network training algorithm.

2.1.3. Theoretical analysis of NVNO and prediction accuracy

(1) Theoretical analysis of NVNO

Suppose the number of input layer nodes is n_i which is usually equal to the dimension of the input x_t. The number of hidden layer nodes is n_h. The number of output layer nodes is n_o that is usually equal to the dimension of the observation Y_t. In time series regression, n_o is equal to one. Therefore, the shapes of the matrices w_x, w_h, b, w_y and b_y are $[n_i \times n_h]$, $[n_h \times n_h]$, $[n_h \times n_o]$, $[n_h \times n_o]$ and $[n_o \times n_o]$, respectively. The NVNO of SWLSTM is $n_h(n_i + n_h + n_o) + n_o(n_h + n_o)$. In LSTM, the calculation of the input gates, the output gates, forget gates and information states all involves the matrix similar to the weight $[w_h, w_x, b]$, so the NVNO of LSTM is $4n_h(n_i + n_h + n_o) + n_o(n_h + n_o)$. In GRU, it simplifies the network structure of LSTM and reduced three gates to new two gate structures called updated gates and reset gates. It removes bias of hidden layer. Therefore, the NVNO of GRU is reduced to $3n_h(n_i + n_h) + n_o(n_h + n_o)$. Thus, SWLSTM dramatically reduces the

number of variables that need to be optimized.

(2) Theoretical analysis of prediction accuracy

In SWLSTM, the forget gates and output gates in the LSTM are replaced by shared gates s_t and the input gates is replaced by $1 - s_t$. SWLSTM only shares all the same types of weights in the hidden layer and does not break the function of three gates in the LSTM. Therefore, the shared gates in SWLSTM reserve the functions of the three gates in the LSTM and still has the ability to discard useless historical information and keep current useful information.

Klaus Greff tested the performance of eight variants of LSTM with three classic cases, and obtained some important conclusions [29], which are (1) coupling the input and forget gates, or removing peephole connections simplified LSTM without significantly decreasing performance; (2) the forget gate and the output activation function are the

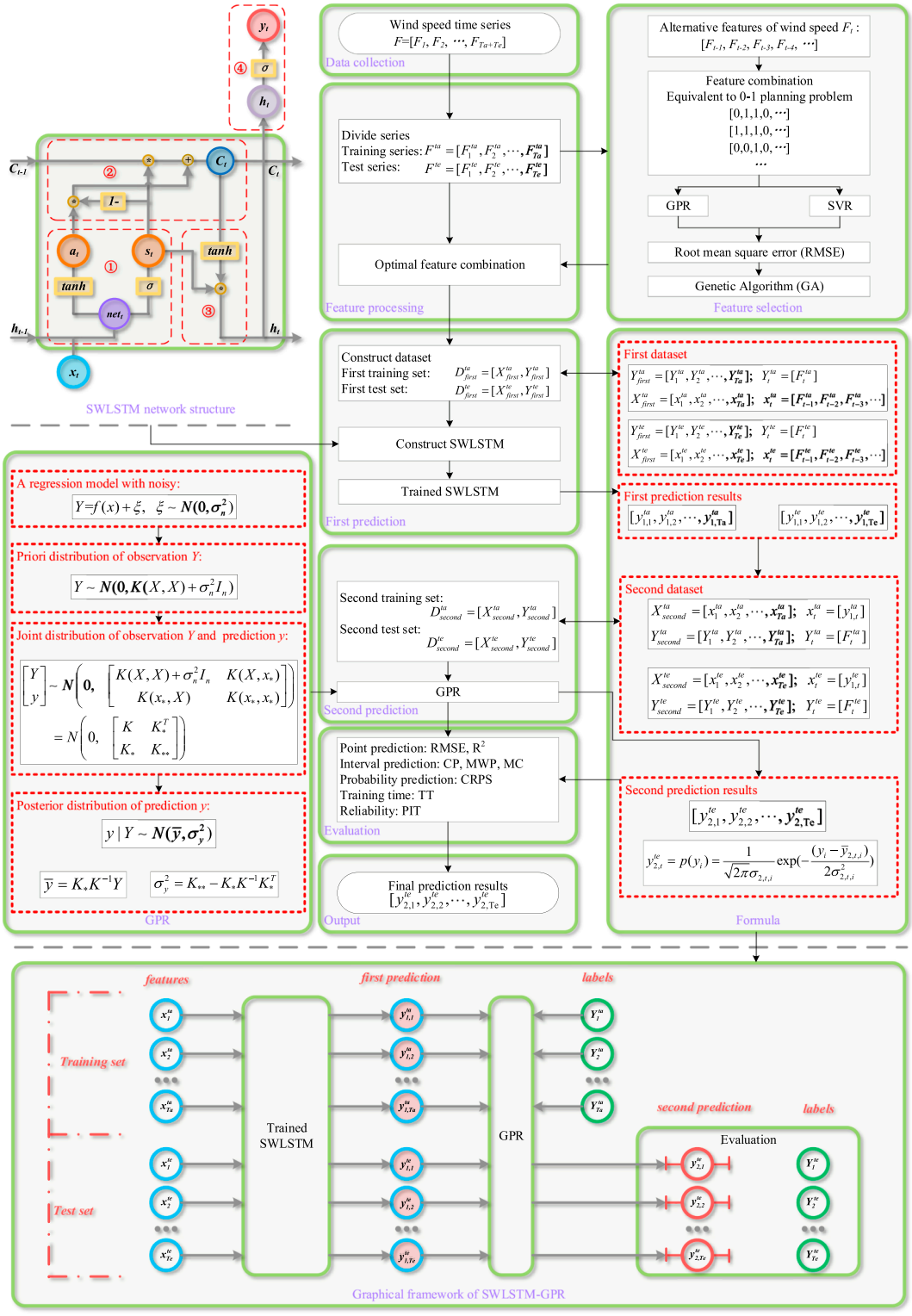


Fig. 2. Flowchart of SWLSTM-GPR.

Table 1
Statistical information of four datasets.

Datasets unit	time 1 period = 15 min	T period	Ta	Te	min m/s	mean m/s	max m/s	std
Dataset 1	04/23/2016 00:00–04/28/2016 00:00	481	385	96	1.09	9.95	20.60	4.73
Dataset 2	03/20/2016 00:00–03/27/2016 00:00	673	538	135	0.33	7.12	17.85	4.08
Dataset 3	04/11/2016 00:00–04/18/2016 00:00	673	538	135	0.74	9.05	18.83	4.75
Dataset 4	05/03/2016 07:00–05/13/2016 07:00	961	769	192	0.55	9.99	25.39	5.82

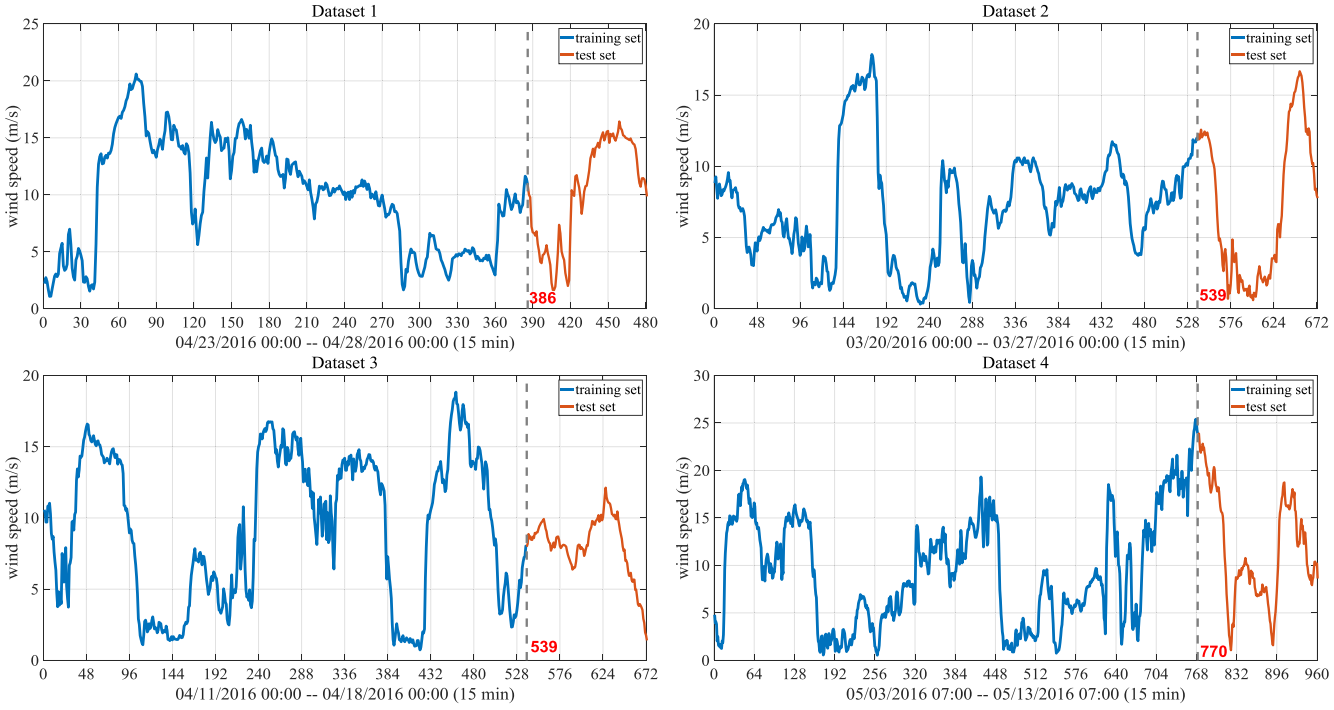


Fig. 3. Four datasets of wind speed.

Table 2
Top 3 feature combinations of the four datasets.

Dataset	Top 3	Historical wind speed										RMSE(m/s)	
		F_{t-1}	F_{t-2}	F_{t-3}	F_{t-4}	F_{t-5}	F_{t-6}	F_{t-7}	F_{t-8}	F_{t-9}	F_{t-10}	SVR	GPR
Dataset 1	1	√	√	√	√	√	√	×	√	√	×	0.795	0.799
	2	√	√	√	√	√	√	√	√	×	×	0.788	0.763
	3	√	√	√	√	√	×	×	√	×	×	0.756	0.749
Dataset 2	1	√	√	√	√	×	√	√	×	√	×	0.701	0.705
	2	√	√	√	√	√	×	×	√	√	×	0.699	0.681
	3	√	√	√	√	√	√	×	×	×	×	0.667	0.657
Dataset 3	1	√	√	√	√	√	√	×	×	×	×	0.298	0.292
	2	√	√	√	√	√	√	√	×	×	×	0.271	0.275
	3	√	√	√	√	√	√	√	√	×	×	0.252	0.246
Dataset 4	1	√	√	√	√	√	√	×	×	√	×	0.719	0.801
	2	√	√	√	√	√	×	×	×	√	×	0.686	0.784
	3	√	√	√	√	√	√	×	√	×	×	0.673	0.762

Table 3
Optimization algorithm parameter settings.

algorithm	symbol	meaning	value
Adagrad	ε	avoid dividing by zero	10^{-8}
RMSprop Nesterov	γ	momentum term parameter	0.9
Adadelta	ε	avoid dividing by zero	10^{-8}
Adam	β_1	biased first moment estimate parameter	0.9
	β_2	biased second raw moment estimate	0.999
	ε	avoid dividing by zero	10^{-8}

Table 4
Probability prediction metrics in four datasets.

model	Datasets	Dataset 1	Dataset 2	Dataset 3	Dataset 4
SWLSTM-GPR	min	0.019	0.017	0.010	0.015
	mean	0.019	0.018	0.012	0.016
	max	0.022	0.027	0.017	0.026
GPR	mean	0.029	0.021	0.022	0.020

most critical components of the LSTM block. The forget gates and input gates in the SWLSTM are replaced by s_t and $1 - s_t$, which are coupled. Activation functions *Sigmoid* and *tanh* are retained in SWLSTM. These two points indicate that the design of SWLSTM meets the conclusion of Klaus Greffe, further indicating that SWLSTM does not significantly reduce the prediction accuracy.

2.2. Gaussian Process Regression (GPR)

Gaussian Process Regression (GPR) [9] is a machine learning method based on Bayesian theory and statistical learning theory. It is suitable for complex regression problems such as high dimensionality and nonlinear. We assume a regression model with noisy as follows:

$$Y = f(X) + \xi \tag{21}$$

where Y is observation and $f(X)$ is an underlying function. We further assume noise $\xi \sim N(0, \sigma_n^2)$. Then priori distribution of the observation Y and the joint prior distribution of the observed value Y and the predicted value y can be obtained.

$$Y \sim N(0, K(X, X) + \sigma_n^2 I_n) \tag{22}$$

$$\begin{bmatrix} Y \\ y \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I_n & K(X, x_*) \\ K(x_*, X) & K(x_*, x_*) \end{bmatrix}\right) = N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{23}$$

where $K(x, x) = (\kappa_{ij})$ is a symmetric positive definite covariance matrix, whose elements κ_{ij} measure the correlation between x_i and x_j through a kernel function κ. $K(x_*, x) = K(x, x_*)^T$ is the covariance matrix between the test set x_* and training set x. $K(x_*, x_*)$ is the covariance matrix of the test set itself. I_n is an n-dimensional unit matrix. Squared exponential kernel, linear kernel and polynomial kernel are all common kernel functions. The formula of squared exponential kernel is as follows. p_1 and are adjustable parameters.

$$\kappa_{ij} = p_1 \cdot \exp\left(-\frac{(x_i - x_j)^2}{2p_2}\right) \tag{24}$$

The posterior distribution of the predicted value y is

$$y|Y \sim N(\bar{y}, \sigma_y^2) \tag{25}$$

$$\bar{y} = K_* K^{-1} Y \tag{26}$$

$$\sigma_y^2 = K_{**} - K_* K^{-1} K_*^T \tag{27}$$

Therefore, the point prediction result of GPR is $\bar{y}$ and the interval prediction result corresponding to the 95% confidence level is $[\bar{y} - 1.96\sigma_y, \bar{y} + 1.96\sigma_y]$. The probability density function of i-th predicted value is as follows:

Table 5
Parameter details of six models.

model	symbol	meaning	value	reason
SWLSTM-GPR	n_i	number of input layer nodes	–	number of feature inputs
	n_h	number of hidden layer nodes	8	common value [2, 4, 8, 10, …]
	n_o	number of output layer nodes	1	time series regression
	η	fixed learning rate	0.01	common value [0.001, 0.01, 0.05, 0.1, …]
	T	size of batch	32	common value [8, 16, 32, 50, 100, …]
	Ep	epochs of training	2000	converged
	kf	kernel function	Gaussian Function	the same as GPR
	p_1	parameter in Gaussian Function	2	the same as GPR
	p_2	parameter in Gaussian Function	1	the same as GPR
LSTM	n_i	number of input layer nodes	–	the same as SWLSTM-GPR
	n_h	number of hidden layer nodes	8	the same as SWLSTM-GPR
	n_o	number of output layer nodes	1	the same as SWLSTM-GPR
	η	fixed learning rate	0.01	the same as SWLSTM-GPR
	T	size of batch	32	the same as SWLSTM-GPR
	Ep	epochs of training	2000	the same as SWLSTM-GPR
GRU	n_i	number of input layer nodes	–	the same as SWLSTM-GPR
	n_h	number of hidden layer nodes	8	the same as SWLSTM-GPR
	n_o	number of output layer nodes	1	the same as SWLSTM-GPR
	η	fixed learning rate	0.01	the same as SWLSTM-GPR
	T	size of batch	32	the same as SWLSTM-GPR
	Ep	epochs of training	2000	the same as SWLSTM-GPR
GPR	kf	kernel function	Gaussian Function	a competitive kernel functions
	p_1	parameter in Gaussian Function	2	obtained by GA in $[-5, 5]$
	p_2	parameter in Gaussian Function	1	obtained by GA in $[-5, 5]$
SVR	kf	kernel function	Radial Basis Function	a competitive kernel functions
	C	parameter in Radial Basis Function of "sklearn"	1	obtained by GA in $[-5, 5]$
QR	kf	kernel function of "statsmodels"	Gaussian Function	a competitive kernel functions

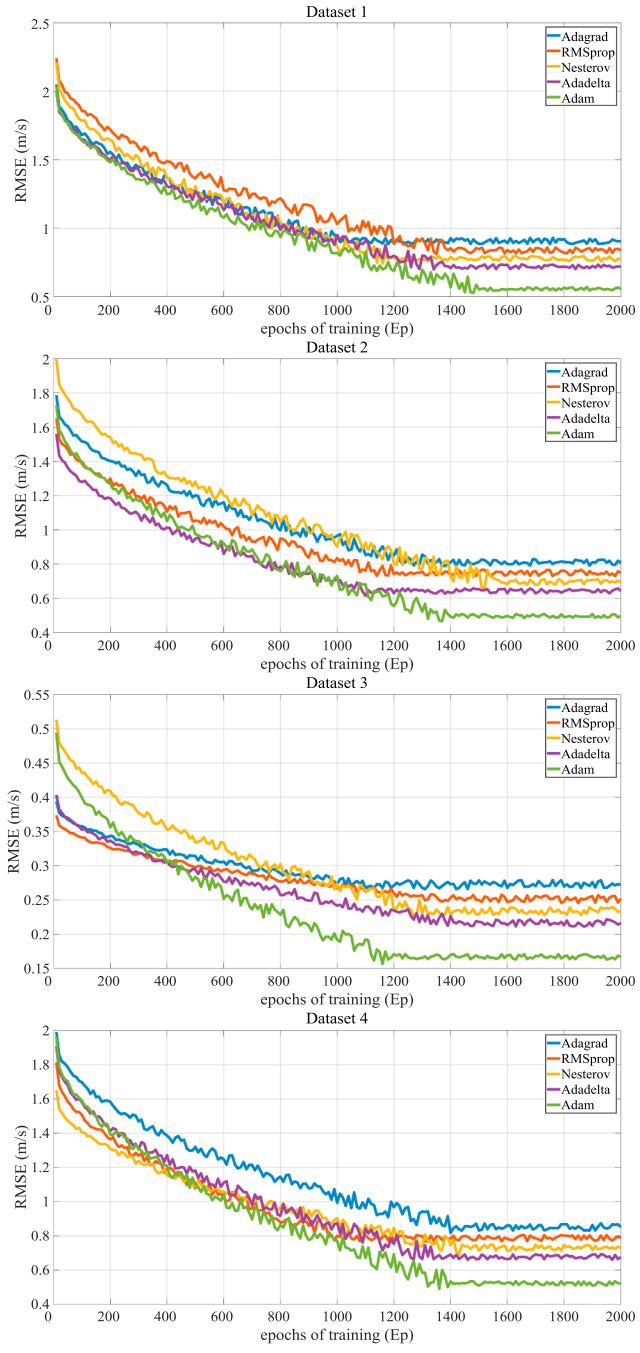


Fig. 4. Convergence curve of SWLSTM.

$$p(y_i) = \frac{1}{\sqrt{2\pi}\sigma_{yi}} \exp\left(-\frac{(y_i - \bar{y}_i)}{2\sigma_{yi}^2}\right) \tag{28}$$

2.3. SWLSTM-GPR

Since SWLSTM's point prediction accuracy is high and GPR's probability prediction results are reliable, SWLSTM and GPR are combined to obtain high-precision point prediction, high-reliability interval prediction and probability prediction. The idea for combination is to first train SWLSTM network completely, then use the training set and test set as inputs to make the first prediction with SWLSTM, finally construct GPR for the second prediction between the first predicted value and the observed value. The result of second prediction include point prediction, interval prediction and probability prediction. The first advantage of this idea is that it does not beak the point prediction accuracy of SWLSTM. Since the point prediction accuracy of SWLSTM is high, a more reliable prediction interval and PDF can be obtained when GPR is constructed between the first predicted value and the observed value, which is the second advantage of the idea.

The flowchart of SWLSTM-GPR is shown in the Fig. 2. x_t still represents feature input. y_{1t} is the first predicted value obtained by SWLSTM, which has only point prediction results. y_{2t} is the second predicted value obtained by SWLSTM-GPR which includes point prediction, interval prediction and probability prediction results. Y_t are observations. The superscripts *ta* and *te* represent the identifier of the training set and test set, respectively. *Ta* and *Te* are the number of training set and test set samples.

The complete process of SWLSTM-GPR is as follows: First, $[x_1^{ta}, x_2^{ta}, \cdots, x_{Ta}^{ta}]$ and $[Y_1^{ta}, Y_2^{ta}, \cdots, Y_{Ta}^{ta}]$ are used to train SWLSTM. Then $[x_1^{ta}, x_2^{ta}, \cdots, x_{Ta}^{ta}]$ and $[x_1^{te}, x_2^{te}, \cdots, x_{Te}^{te}]$ are input into the trained SWLSTM to get the first prediction results, which are $[y_{1,1}^{ta}, y_{1,2}^{ta}, \cdots, y_{1,Ta}^{ta}]$ and $[y_{1,1}^{te}, y_{1,2}^{te}, \cdots, y_{1,Te}^{te}]$. After that, $[y_{1,1}^{ta}, y_{1,2}^{ta}, \cdots, y_{1,Ta}^{ta}]$, $[y_{1,1}^{te}, y_{1,2}^{te}, \cdots, y_{1,Te}^{te}]$ and $[Y_1^{ta}, Y_2^{ta}, \cdots, Y_{Ta}^{ta}]$ are input together to train GPR and produce the second prediction results which are $[y_{2,1}^{te}, y_{2,2}^{te}, \cdots, y_{2,Te}^{te}]$. Finally $[y_{2,1}^{te}, y_{2,2}^{te}, \cdots, y_{2,Te}^{te}]$ and $[Y_1^{te}, Y_2^{te}, \cdots, Y_{Te}^{te}]$ are used to evaluate the performance and reliability of the model. The pseudo code is listed in the Appendix A. Algorithm 1 is the overall framework of SWLSTM-GPR. Algorithm 2 and 3 are the forward propagation and back propagation algorithm of SWLSTM, respectively. Algorithm 4 updates weights and bias using Adam optimization algorithm. Algorithm 5 is the pseudo code of GPR.

2.4. Feature selection

In order to make the model show better performance, Genetic Algorithms (GA) [30] are used to filter optimal feature combinations. First, some alternative features need to be listed. Then, these alternative features have two possibilities of being left and deleted, which corresponds to the 0–1 planning problem. Finally, various features combination are screened by GA using root mean square error (RMSE) as the fitness to obtain an optimal feature combination. It should be noted that feature selection and neural network (such as LSTM, SWLSTM) parameter training are both an iterative optimization problem. And the training of neural network is a relatively time consuming process. If the neural network method is used to calculate the RMSE, this would be a two-layer optimization problem and very time-consuming. Therefore, GPR and Support Vector Regression (SVR) are used to calculate the RMSE since they can calculate RMSE in a short time.

3. Method evaluation metric

3.1. Evaluation metric of point prediction

(1) Root mean square error (RMSE)

RMSE [31] is defined as the square root of the mean of squared error, whose formula is as follows. y_i and Y_i are prediction and observation, respectively. *Te* is the size of test set sample. The smaller the RMSE, the higher the prediction accuracy.

$$RMSE = \sqrt{\frac{1}{Te}\sum_{i=1}^{Te}(y_i - Y_i)^2} \tag{29}$$

(2) Coefficient of determination (R^2)

Coefficient of determination (R^2) [32] is the ratio of the sum of the squares of the regression to the sum of the squares of the total deviations, reflecting the proportion of the independent variable explaining the variation of the dependent variable. The closer the value of R^2 is to

Table 6
Point prediction metrics for six models in four datasets.

	Datasets	Dataset 1			Dataset 2			Dataset 3			Dataset 4		
	metric	RMSE	R^2	TT	RMSE	R^2	TT	RMSE	R^2	TT	RMSE	R^2	TT
model	unit	(m/s)		(s)	(m/s)		(s)	(m/s)		(s)	(m/s)		(s)
SWLSTM-GPR	min	0.521	0.981	15.9	0.446	0.987	21.4	0.151	0.991	22.6	0.427	0.989	32.5
	mean	0.579	0.983	17.1	0.517	0.990	23.7	0.174	0.992	24.3	0.544	0.991	34.2
	max	0.628	0.987	18.2	0.580	0.992	25.1	0.187	0.994	26.6	0.606	0.994	37.3
LSTM	min	0.483	0.978	29.0	0.469	0.986	38.2	0.168	0.989	38.1	0.457	0.989	57.5
	mean	0.558	0.985	30.3	0.532	0.989	40.4	0.194	0.991	40.4	0.526	0.992	57.7
	max	0.663	0.989	31.8	0.611	0.992	43.6	0.212	0.993	42.0	0.600	0.994	61.1
GRU	min	0.559	0.975	19.6	0.498	0.985	26.0	0.177	0.990	27.1	0.508	0.988	39.1
	mean	0.610	0.982	20.5	0.557	0.988	27.9	0.193	0.991	27.1	0.555	0.991	40.3
	max	0.710	0.985	21.3	0.629	0.991	28.3	0.202	0.992	27.8	0.628	0.992	43.3
GPR	mean	0.799	0.969	<2	0.705	0.981	<2	0.292	0.979	<2	0.801	0.981	<2
SVR	mean	0.795	0.969	<2	0.701	0.981	<2	0.298	0.978	<2	0.719	0.984	<2
QR	mean	1.084	0.942	<2	0.705	0.981	<2	0.423	0.955	<2	1.018	0.969	<2

1, the higher the point prediction accuracy. Its formula is as follows, where $\bar{Y}_i$ is the mean of observations.

$$R^2 = 1 - \frac{\sum_{i=1}^{Te}(y_i - Y_i)^2}{\sum_{i=1}^{Te}(Y_i - \bar{Y}_i)^2} \tag{30}$$

3.2. Evaluation metric of interval prediction

(1) Coverage probability (CP)

CP_α [33] is defined as the probability that the observation falls within the prediction interval under confidence level of α. Its formula is as follows. c_α is the number of samples whose observation fall within the prediction interval.

$$CP_\alpha = \frac{c_\alpha}{Te} \times 100\% \tag{31}$$

(2) Mean width percentage (MWP)

If the interval is wide enough, it is easy to satisfy the case where the CP_α is 100%. Such interval is too conservative and does not provide effective information on the uncertainty of the prediction, which makes the prediction interval have no practical value. MWP_α [33] is defined as the mean percentage of the interval width to the observation to ensure the validity of the interval, whose formula is as follows. The ideal prediction interval should have high CP_α and low MWP_α. So we define the metric MC, whose formula is as follow. It unifies CP and MWP. The smaller the MC, the better the interval prediction result.

$$MWP_\alpha = \frac{1}{Te}\sum_{i=1}^{Te}\frac{up_i - down_i}{Y_i} \tag{32}$$

$$MC_\alpha = MWP_\alpha / CP_\alpha \tag{33}$$

3.3. Evaluation metric of probability prediction

Continuous ranked probability score (CRPS) [34] is a comprehensive evaluation metric for predictive performance, which can verify deterministic, ensemble, and probabilistic forecasts. We suppose that the PDF of i-th predicted value by SWLSTM-GPR is $p(y_i)$ and its cumulative distribution function (CDF) is $F(y_i)$. The formula of CRPS is as follows, where $H(y_i - Y_i)$ is the Heaviside function. The smaller the CRPS, the better the comprehensive performance.

$$CRPS = \frac{1}{Te}\sum_{i=1}^{Te}\int_{-\infty}^{+\infty}[F(y_i) - H(y_i - Y_i)]^2 dy_i \tag{34}$$

$$F(y_i) = \int_{-\infty}^{y_i} p(x)dx \tag{35}$$

$$H(y_i - Y_i) = \begin{cases} 0 & y_i < Y_i \\ 1 & others \end{cases} \tag{36}$$

3.4. Evaluation metric of reliability

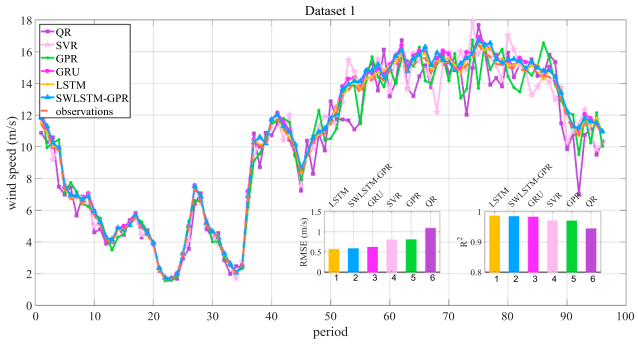
Reliability refers to the statistical consistency of predictions and observations. A uniform probability plot of the probability integral transform (PIT) values is used to assess predictive reliability [35]. PIT is calculated from CDF and observation, as follows. If the predictions are reliable, the PIT values obey uniform distribution between 0 and 1. PIT values for all test samples displayed in uniform probability plot can clearly check whether it is subject to uniform distribution.

$$PIT = F(Y_i) = \int_{-\infty}^{Y_i} p(x)dx \tag{37}$$

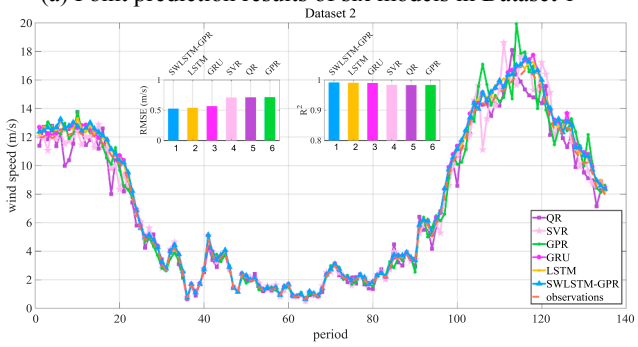
4. Case study

4.1. Case introduction

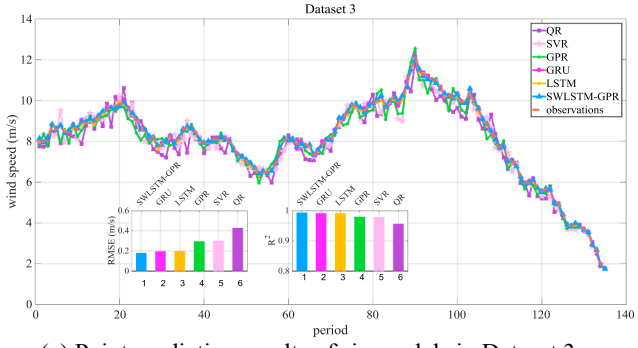
Wind speed data used in this study is gathered from the wind farm in Inner Mongolia, China. The step of wind speed data is 15 min. Four different datasets are used to test the performance of the model, whose statistical information is shown in the Table 1. Four datasets use different lengths of data, which are 5 days, 7 days, 7 days and 10 days. In the table, *T*, *Ta* and *Te* represent the size of total sample, training set sample and test set sample. The minimum, mean, maximum and standard deviation of total sample are abbreviated as min, mean, max and std. Eighty percent of each dataset is used as training set and the rest is used as test set (also known as validation set in deep learning). Four



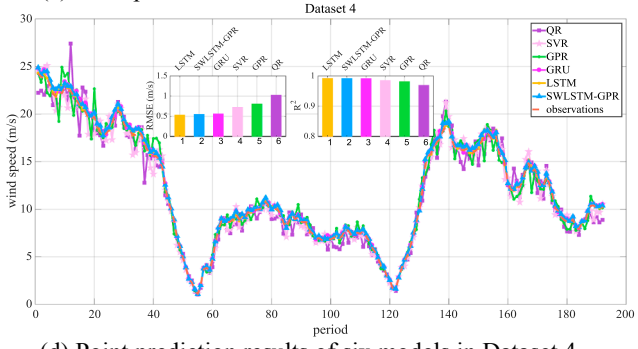
(a) Point prediction results of six models in Dataset 1



(b) Point prediction results of six models in Dataset 2



(c) Point prediction results of six models in Dataset 3



(d) Point prediction results of six models in Dataset 4

Fig. 5. Point predictions of six models in four test sets.

datasets of wind speed are shown in the Fig. 3.

In this case, there are four tasks that need to be completed.

(1) Select optimal feature combination for wind speed prediction;
(2) Verify the convergence of SWLSTM;
(3) Compare SWLSTM-GPR with the state-of-the-art wind speed prediction methods from point prediction accuracy, interval prediction suitability, probability prediction comprehensive performance and training time;
(4) Verify the reliability of SWLSTM-GPR;
(5) Show probabilistic forecast results of wind speed prediction.

4.2. Task I: select optimal feature combination

In order to make all the models participating in the comparison show better performance, the optimal feature combinations of four datasets are selected using the method introduced in Section 2.4. Historical wind speed data for the previous ten periods is used as the alternative feature. Top 3 feature combinations of the four datasets are listed in the Table 2. In the following experiments, all models use the Top 1 feature combination as input. The symbol √ indicates that the feature is selected while the symbol × indicates that the feature is deleted.

4.3. Model selection and parameter settings

Wind speed is time series data. The models commonly used in traditional machine learning to solve time series problems are GPR, SVR and QR. In deep learning, LSTM is good at solving time series problems and GRU is the variant of LSTM. Therefore, LSTM, GRU, GPR, SVR and QR are selected as comparison models with SWLSTM-GPR. Since all five models can make point predictions, they are used in point prediction comparison. Because LSTM, GRU and SVR cannot make interval prediction, GPR and QR are used in interval prediction comparison. Since GPR can get a specific PDF, it is chosen as probability prediction comparison model.

In this study, SWLSTM-GPR, LSTM, GRU and GPR are implemented in Java. SVR and QR are implemented using “sklearn”[1] framework and “statsmodels”[2] framework in python, respectively. GA is used to optimize parameters of the model. The setting of all model parameters is either optimized with GA or some common values. The same parameters are equal in these models to ensure the fairness of the comparison. The parameter details of six models are shown in the Table 5. Since the results of each run of GPR, SVR and QR are the same, these models only run once. SWLSTM-GPR, LSTM and GRU need to iterate when optimizing weights and there are random numbers in the model. The results of each run are different, so these models run 10 times and the average is taken as the final results.

4.4. Task II: verify convergence

In order to ensure the prediction accuracy of the proposed model SWLSTM, its convergence is first verified before comparison. Five commonly used optimization algorithms in deep learning are used to verify convergence: Adaptive Gradient Algorithm (Adagrad) [36], Root Mean Square Prop (RMSprop) [36], Nesterov Momentum (Nesterov) [36], Adaptive Delta Algorithm (Adadelta) [36] and Adam [28]. The parameters of these five algorithms are used the default parameters recommended in the paper [36], as shown in the Table 3. The hyperparameters of the proposed model SWLSTM are shown in the Table 5. In order to make the results of each run the same, the random number seeds of SWLSTM in convergence verification are set to be the same. RMSE is used as loss function.

The convergence curve of SWLSTM in four datasets are shown in the Fig. 4. All loss function curves are close to a horizontal line at a later stage, which indicates that setting the value of epochs to 2000 ensures that the five optimization algorithms can converge SWLSTM on all four datasets. From dataset 1 to 4, the Adam algorithm converges at approximately 1500, 1400, 1200 and 1400 epoch, respectively. In each dataset, RMSE trained by Adam is the smallest, indicating that Adam can make SWLSTM converge best in five optimization algorithms.

[1] https://scikit-learn.org/stable/.
[2] http://www.statsmodels.org/stable/index.html.

Table 7
Interval prediction metrics in four datasets.

model	Datasets / metric	Dataset 1			Dataset 2			Dataset 3			Dataset 4		
		$CP_{95\%}$	$MWP_{95\%}$	$MC_{95\%}$	$CP_{95\%}$	$MWP_{95\%}$	$MC_{95\%}$	$CP_{95\%}$	$MWP_{95\%}$	$MC_{95\%}$	$CP_{95\%}$	$MWP_{95\%}$	$MC_{95\%}$
SWLSTM-GPR	min	0.91	0.323	0.334	0.93	0.721	0.737	0.96	0.106	0.108	0.93	0.277	0.284
	mean	0.95	0.324	0.342	0.95	0.724	0.755	0.97	0.107	0.110	0.96	0.278	0.291
	max	0.97	0.326	0.358	0.99	0.726	0.778	0.99	0.108	0.112	0.98	0.279	0.300
GPR	mean	0.86	0.327	0.378	0.90	0.728	0.805	0.84	0.108	0.128	0.90	0.279	0.309
QR	mean	0.90	0.490	0.547	0.96	1.089	1.131	0.84	0.162	0.192	0.92	0.420	0.458

Therefore, the model in the later comparative experiments are trained by Adam algorithm.

4.5. Task III: compare different methods

(1) Point prediction result evaluation

The point prediction result evaluation is to verify the prediction accuracy of SWLSTM-GPR. Point prediction metrics of six models in four datasets are shown in Table 6. TT is a shorthand for training time. TT is only compared in the three models SWLSTM, LSTM and GRU. The best results in the six models are highlighted with gray fill. For prediction accuracy, LSTM has the best results in Dataset 1 and Dataset 4; SWLSTM-GPR has the best results in Dataset 2 and Dataset 3. For RMSE and R^2, the three models SWLSTM-GPR, LSTM and GRU have similar results. But SWLSTM-GPR's training time is much smaller than the other two models. As the size of samples increases, the difference in training time between the three models will increase. These metrics confirm that SWLSTM-GPR has achieved our goal of simplifying LSTM without significantly reducing prediction accuracy. Although the TT of GPR, SVR and QR are particularly small, their prediction accuracy is limited. The minimum of RMSE and the maximum of R^2 of SWLSTM-GPR are far better than that of GPR, SVR and QR in Dataset 1–4. In practical application, SWLSTM-GPR can be run several times to take the best result as the prediction result.

In order to compare the differences in point predictions more visually, the prediction results of six models in four test sets are drawn in the Fig. 5. SWLSTM-GPR, LSTM and GRU use the average of 10 results as the final prediction result. The results with smaller RMSE and larger R^2 are still closer to the observations. The histogram in the Fig. 5 is the ranking of the model on RMSE and R^2. The ranks of SWLSTM-GPR in the four datasets are [2,1,1,2]. In summary, the comprehensive point prediction performance of SWLSTM-GPR is optimal among the six models.

(2) Interval prediction result evaluation

The interval prediction result evaluation is to verify the suitability of interval obtained by SWLSTM-GPR. In interval prediction, GPR and QR are used to compare with SWLSTM-GPR. In this study, confidence level is set as 95%. Interval prediction metrics of the three models in four datasets are shown in the Table 7. In the Dataset 1–4, SWLSTM-GPR and GPR have close MWP, but SWLSTM-GPR has higher CP. The reason is that the probability prediction mechanism of SWLSTM-GPR and GPR is the same but SWLSTM-GPR has higher point prediction accuracy. Compared with QR, SWLSTM-GPR have higher CP and lower MWP in the Dataset 1, 3 and 4. In the Dataset 2, the CP of QR is more than SWLSTM-GPR while the MWP of SWLSTM-GPR is less than QR. It is difficult to know which model is better with these two metrics. The MC is proposed to solve this situation. From MC, SWLSTM-GPR is better than QR in the Dataset 2. These metrics indicate that the prediction interval obtained by SWLSTM-GPR is more suitable.

In order to observe the suitability of each model's prediction interval more visually, the prediction intervals are plotted in the Fig. 6. Yellow dot in the Fig. 6 is the point at which the observation exceeds the prediction interval of SWLSTM-GPR. Obviously, the interval of QR is relatively wide, which indicates that prediction interval obtained by QR is conservative and it increase the coverage by expanding the interval width. The interval widths of SWLSTM-GPR and GPR are very close, but the interval position of SWLSTM-GPR is more appropriate. The histogram in the Fig. 6 is the ranking of the model on CP, MWP and MC. The MC rank of SWLSTM-GPR in four datasets is [1,1,1,1]. In summary, the comprehensive interval prediction performance of SWLSTM-GPR is optimal among the three models.

(3) Probability prediction result evaluation

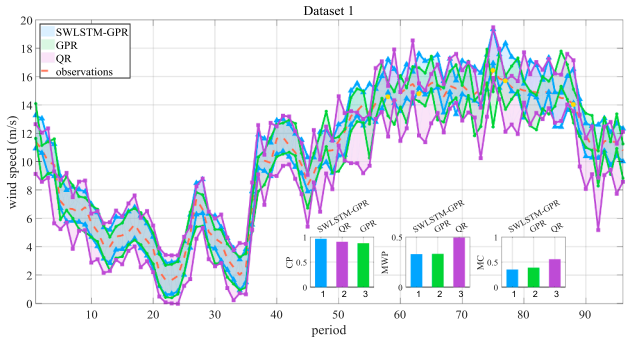
The probability prediction result evaluation is to verify the comprehensive performance of PDF obtained by SWLSTM-GPR and GPR. The probability prediction metrics in four datasets are shown in the Table 4. Since the CRPS evaluates the entire PDF, its results can include not only the results of point prediction and interval prediction, but also the overall performance of the PDF. The CRPS of SWLSTM-GPR is better than GPR in the Dataset 1–4. This result is consistent with point prediction and interval prediction.

4.6. Task IV: Verify the reliability of SWLSTM-GPR

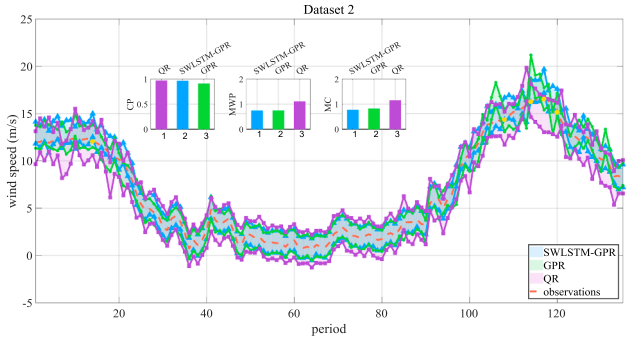
After evaluating the prediction results of SWLSTM-GPR from the three aspects of point prediction, interval prediction and probability prediction, the reliability is tested to ensure that prediction results are persuasive. We first calculate the PIT value for each observation of test set. If these PIT values are subject to a uniform distribution, the prediction results are reliable [35]. The uniform probability plot of PIT values can be drawn to clearly see whether these values are subject to uniform distribution, as shown in the Fig. 7. The PIT values of the four datasets are evenly distributed around the diagonal and its range evenly covers $[0, 1]$. All PIT points are located in the Kolmogorov 5% significance band, which indicates that predicted PDF are not excessively high or low, or excessively wide or narrow [36]. Therefore, the prediction results obtained by SWLSTM-GPR are reliable and convincing.

4.7. Task V: show probabilistic forecast results

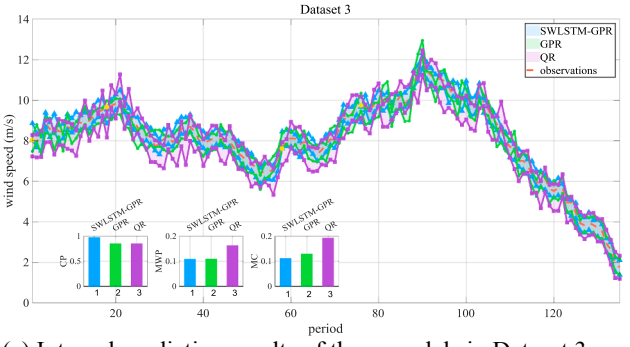
PDFs obtained by SWLSTM-GPR of six points of equidistant sampling in Dataset 4 are shown in the Fig. 8. Six probability density curves are very full, and no curve is excessively high or low, wide or narrow, which indicate the PDF obtained by SWLSTM-GPR is suitable. In period 38, 77, 115, 154, the observation lines are near the center of curve, which show these points' prediction accuracy is very high. In period 1, 192, the observation lines are far from the center, which shown these points' prediction error is a little high. In probability prediction results of test set, some observations are close to the center and other observations are off-center, which just indicates that the probabilistic



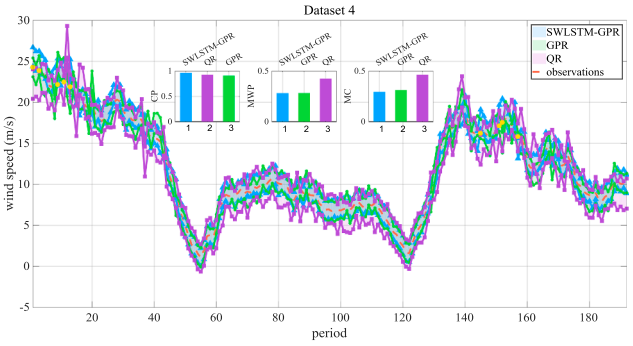
(a) Interval prediction results of three models in Dataset 1



(b) Interval prediction results of three models in Dataset 2



(c) Interval prediction results of three models in Dataset 3



(d) Interval prediction results of three models in Dataset 4

Fig. 6. Interval predictions of three models in four test sets.

forecast is reliable. If all points are at the center or far from the center, we may not be convinced of this probabilistic forecast results.

5. Conclusions and future research

Obtaining reliable high-quality wind speed prediction results is very important for the planning and application of wind energy. A new wind speed prediction method called SWLSTM-GPR is proposed to decrease

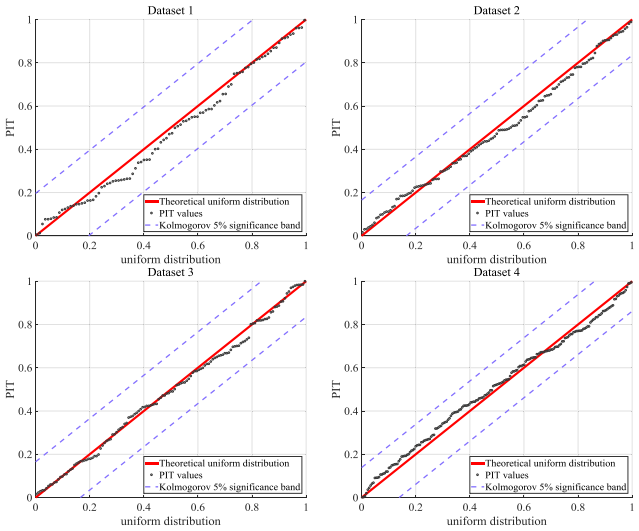


Fig. 7. Reliability test of SWLSTM-GPR in four datasets.

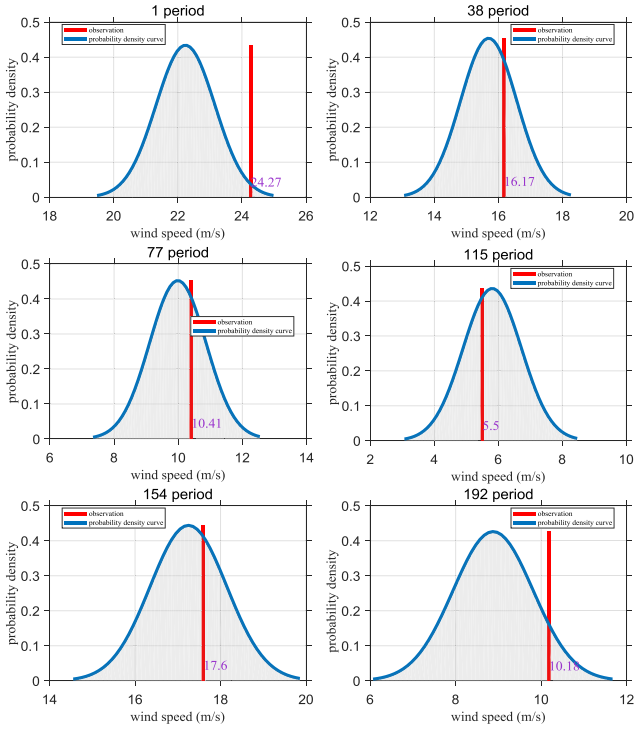


Fig. 8. PDF obtained by SWLSTM-GPR in the Dataset 4.

the training time of LSTM without significantly decrease the prediction accuracy and obtain uncertain information of probabilistic prediction. SWLSTM preserves the functionality of LSTM and reduces training time by sharing weights. Second prediction by GPR enables the model to obtain reliable PDF.

The proposed methods are applied to predict wind speed for four actual case in Inner Mongolia, China. Eight verification metrics RMSE, R^2, CP_α, MWP_α, MC_α, CRPS, PIT and TT are used to evaluate point prediction accuracy, interval prediction suitability, probability prediction comprehensive performance, forecast reliability and training time. The experimental results show that SWLSTM-GPR can obtain high-precision point prediction, appropriate prediction interval and high-performance probabilistic prediction results with shorter training time.

In fact, SWLSTM-GPR can not only be used to predict wind speed, it is very good at solving time series problems. In the complementary system integrated of hydro, wind and solar power, the most relevant

factors of hydro, wind and solar power are runoff, wind speed and solar radiation intensity respectively. All three of these factors are time series data, therefore SWLSTM-GPR can be used to solve the time series prediction problems in the complementary system.

Due to the strong volatility and randomness of wind speed, the Gaussian distribution hypothesis in GPR may not be the optimal distribution. In future research work, more distributions like Weibull, Rayleigh and Beta distribution will be used to test predictive performance.

Declaration of interest

None.

Appendix A. Pseudo codes of SWLSTM-GPR

The pseudo codes of SWLSTM-GPR can be found in a separate text:.

Algorithm 1: The overall framework of SWLSTM-GPR

input:

Ep:	epochs of training	Ta:	number of training set samples
T:	size of batch	x^{ta}:	features of training set
Y^{ta}:	labels of training set	x^{te}:	features of test set
w_h:	weight matrix, shape:[$n_h \times n_h$]	w_x:	weight matrix, shape:[$n_h \times n_i$]
b:	bias matrix, shape:[$n_h \times n_o$]	w_y:	weight matrix, shape:[$n_o \times n_h$]
b_y:	bias matrix, shape:[$n_o \times n_o$]	η:	fixed learning rate
α:	confidence level		

output:

y_2^{te}:	results of point prediction	[up, $down$]:	results of interval prediction
σ_y:	together with y_2^{te} constitutes results of probability prediction		

```
ti=0;
for e = 1:Ep
    for b = 1:T:Ta-T
        xBatch = x^ta (b:b+T, :);   yBatch = Y^ta (b:b+T,1);   % get a batch of training set
        [s, a, C, h, y] = forward(xBatch, w_h, w_x, b, w_y, b_y);
        [δW_h, δW_x, δB, δW_y, δB_y] = back(xBatch, yBatch, s, a, C, h, y);
        [w_h, w_x, b, w_y, b_y] = adam(w_h, w_x, b, w_y, b_y, δW_h, δW_x, δB, δW_y, δB_y, ti, η);
        ti = ti+1;
    end
end
[~, ~, ~, ~, y_1^ta] = forward(x^ta, w_h, w_x, b, w_y, b_y);
[~, ~, ~, ~, y_1^te] = forward(x^te, w_h, w_x, b, w_y, b_y);
[y_2^te, σ_y, up, down] = GPR(y_1^ta, Y^ta, y_1^te, α);
```

Acknowledgment

This work is supported by the National Key R&D Program of China (2017YFC0405900), the National Natural Science Foundation of China (No. 91647114, 51709275, 61703199), the Fundamental Research Funds for the Central Universities (HUST: 2016YXZD047), and special thanks are given to the anonymous reviewers and editors for their constructive comments.

Algorithm 2: $[s, a, C, h, y]$ = forward(x, w_h, w_x, b, w_y, b_y) % forward propagation of SWLSTM

input:

x: features of a batch of training set or test set

$[w_h, w_x, b]$: weight and bias

$[w_y, b_y]$: weight and bias

output:

s: shared gates of all samples in x

a: information state of all samples in x

C: cell state of all samples in x

h: output of hidden layer of all samples in x

y: output of output layer of all samples in x

1: **for** t = *1*:**length**(x)
2: $net(t) = w_h \cdot h(t\text{-}1) + w_x \cdot x(t) + b$;
3: $s(t) = \sigma(net(t))$;
4: $a(t) = \mathbf{tanh}(net(t))$;
5: $C(t) = s(t)*C(t\text{-}1) + (1\text{-}s(t))*a(t)$;
6: $h(t) = s(t)*\mathbf{tanh}(C(t))$;
7: $z(t) = w_y \cdot h(t) + b_y$
8: $y(t) = \sigma(z(t))$;
9: **end**

Algorithm 3: $[\delta W_h, \delta W_x, \delta B, \delta W_y, \delta B_y]$ = back(x, Y, s, a, C, h, y) % back propagation of SWLSTM

input:

x: features of a batch of training set or test set

Y: labels corresponding to x

s, a, C, h, y: forward propagation results corresponding to x

output:

δW_h: average gradient of W_h in batch $[x,Y]$

δW_x: average gradient of W_x in batch $[x,Y]$

δB: average gradient of B in batch $[x,Y]$

δW_y: average gradient of W_y in batch $[x,Y]$

δB_y: average gradient of B_y in batch $[x,Y]$

1: **for** t = **length**(x):t
2: $\delta y(t) = y(t) - Y(t)$;
3: $\delta z(t) = \delta y(t) * [y(t)*(1\text{-}y(t))]$;
4: $\delta w_y = \delta z(t) \cdot h(t)$;
5: $\delta b_y = \delta z(t)$;
6: **if** t == **length**(x)
7: $\delta h(t) = \delta z(t)\ w_y$;
8: $\delta C(t) = \delta h(t)*s(t)*[1\text{-}\mathbf{tanh}^2(C(t))]$;
9: **else:**
10: $\delta h(t) = \delta z(t)\ w_y + \delta net(t+1)\ w_h$;
11: $\delta C(t) = \delta h(t)*s(t)*[1\text{-}\mathbf{tanh}^2(C(t))] + \delta C(t+1)*s(t)$;
12: **end**
13: $\delta a(t) = \delta C(t)*(1\text{-}s(t))$;
14: $\delta s(t) = \delta h(t)*\mathbf{tanh}(C(t)) + \delta C(t)*[C(t\text{-}1)\text{-}a(t)]$;
15: $\delta net(t) = \delta a(t)*[1\text{-}a(t)^2] + \delta s(t)*[s(t)*(1\text{-}s(t))]$;
16: $\delta w_h = \delta net(t)\ h(t\text{-}1)$;
17: $\delta w_x = \delta net(t)\ x(t)$;
18: $\delta b = \delta net(t)$;
19: $\delta W_y \mathrel{+}= \delta w_y$; $\delta B_y \mathrel{+}= \delta b_y$; $\delta W_h \mathrel{+}= \delta w_h$; $\delta W_x \mathrel{+}= \delta w_x$; $\delta B \mathrel{+}= \delta b$;
20: **end**
21: δW_y/= **length**(x); δB_y/= **length**(x); δW_h/= **length**(x); δW_x/= **length**(x); δB/= **length**(x);

Algorithm 4: $[w_h, w_x, b, w_y, b_y]$ = adam(w_h, w_x, b, w_y, b_y, δW_h, δW_x, δB, δW_y, δB_y, ti, η) % update weights and bias using Adam optimization algorithm

input:

$[w_h, w_x, b, w_y, b_y]$:	weights and bias
$[\delta W_h, \delta W_x, \delta B, \delta W_y, \delta B_y]$:	gradients of weights and bias
ti:	current number of updates for weights and bias
η:	fixed learning rate

output:

$[w_h, w_x, b, w_y, b_y]$:	updated weights and bias

$\beta_1 = 0.9$; $\beta_2 = 0.999$; $\varepsilon = 10^{-8}$;
for $[w, \delta W]$ **in** $\{[w_h, \delta W_h], [w_x, \delta W_x], [b, \delta B], [w_y, \delta W_y], [b_y, \delta B_y]\}$
  $m_w(ti) = \beta_1 \cdot m_w(ti-1) + (1-\beta_1) \cdot \delta W$;
  $v_w(ti) = \beta_2 \cdot v_w(ti-1) + (1-\beta_2) \cdot (\delta W)^2$;
  $m'_w(ti) = m_w(ti)/(1-\beta ti\ 1)$;
  $v'_w(ti) = v_w(ti)/(1-\beta ti\ 2)$;
  $w = w - \eta \cdot m'_w(ti)/[(v'_w(ti))^{0.5} + \varepsilon]$;
end

Algorithm 5: $[y, \sigma_y, up, down]$ = GPR(x, Y, x_*, α); % Gaussian process regression

input:

x: features of training set	x_*: features of test set
Y: labels of training set	α: confidence level

output:

y: results of point prediction	$[up, down]$: results of interval prediction
σ_y: together with y constitutes results of probability prediction	

determine the kernel function, taking the square exponential kernel as an example
$[K]$=**kernel**($X1$, $X2$){
  for i=1:**length**($X1$)
    for j=1:**length**($X2$)
      num=**sum**([$X1(i,:)$-$X2(j,:)$].^2); % .^2 represents the square of each element
      $K(i,j)$=p_1***exp**[-num/(2*p_2)]
    end
  end
}
K = **kernel**(x, x);
K_* = **kernel**(x_*, x);
K_{**} = **kernel**(x_*, x_*);
$y = K_* K^{-1} Y$;
$\sigma_y = (K_{**} - K_* K^{-1} K_*^T)^{0.5}$
query the coefficient r corresponding to the confidence level α; %if α==95% then r=1.96;
$up = y + r \cdot \sigma_y$;
$down = y - r \cdot \sigma_y$;

Appendix B. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.apenergy.2019.04.047.

References

[1] Ren Y, Suganthan PN, Srikanth N. A novel empirical mode decomposition with support vector regression for wind speed forecasting. IEEE T Neur Net Lear 2016;27:1793–8.
[2] Tasnim S, Rahman A, Oo AMT, Haque ME. Wind power prediction in new stations based on knowledge of existing Stations: a cluster based multi source domain adaptation approach. Knowl-Based Syst 2018;145:15–24.
[3] Yuan X, Yuan Y, Tan Q, Lei X, Wu X. Wind power prediction using hybrid auto-regressive fractionally integrated moving average and least square support vector machine. Energy 2017;129:122–37.
[4] Zhang J, Draxl C, Hopson T, Monache LD, Vanvyve E, Hodge B. Comparison of numerical weather prediction based deterministic and probabilistic wind resource assessment methods. Appl Energy 2015;156:528–41.
[5] Allen DJ, Tomlin AS, Bale CSE, Skea A, Vosper S, Gallani ML. A boundary layer scaling technique for estimating near-surface wind energy using numerical weather prediction and wind map data. Appl Energy 2017;208:1246–57.
[6] Wang J, Li Y. Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy. Appl Energy 2018;230:429–43.
[7] Erdem E, Shi J. ARMA based approaches for forecasting the tuple of wind speed and direction. Appl Energy 2011;88:1405–14.
[8] Mauricio JA. Exact maximum likelihood estimation of stationary vector ARMA models. J Am Stat Assoc 1995;90:282–91.
[9] Zhang C, Zhang K, Wei H, Zhao X, Liu T. A Gaussian process regression based hybrid approach for short-term wind speed prediction. Energy Convers Manage 2016;126:1084–92.

[10] Chen K, Yu J. Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. Appl Energy 2014;113:690–705.
[11] Nielsen HA, Madsen H, Nielsen TS. Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts. Wind Energy 2006;9:95–108.
[12] Li G, Shi J. On comparing three artificial neural networks for wind speed forecasting. Appl Energy 2010;87:2313–20.
[13] Hu Q, Zhang S, Xie Z, Mi J, Wan J. Noise model based ν-support vector regression with its application to short-term wind speed forecasting. Neural Netw 2014;57:1–11.
[14] Ren C, An N, Wang J, Li L, Hu B, Shang D. Optimal parameters selection for BP neural network based on particle swarm optimization: a case study of wind speed forecasting. Knowl-Based Syst 2014;56:226–39.
[15] Wang L, Li X, Bai Y. Short-term wind speed prediction using an extreme learning machine model with error correction. Energy Convers Manage 2018;162:239–50.
[16] Liu H, Tian H, Li Y. Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. Appl Energy 2012;98:415–24.
[17] Khosravi A, Koury RNN, Machado L, Pabon JJG. Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. Sustain Energy Technol Assess 2018;25:146–60.
[18] Zhang C, Zhang K, Wei H, Zhao J, Liu T, Zhu T. Short-term wind speed forecasting using empirical mode decomposition and feature selection. Renew Energy 2016;96:727–37.
[19] Kiplangat DC, Asokan K, Kumar KS. Improved week-ahead predictions of wind speed using simple linear models with wavelet decomposition. Renew Energy 2016;93:38–44.
[20] Peng T, Zhou J, Zhang C, Zheng Y. Multi-step ahead wind speed forecasting using a hybrid model based on two-stage decomposition technique and AdaBoost-extreme learning machine. Energy Convers Manage 2017;153:589–602.
[21] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44.
[22] Barbounis TG, Theocharis JB, Alexiadis MC, Dokopoulos PS. Long-term wind speed and power forecasting using local recurrent neural network models. IEEE T Energy Convers 2006;21:273–84.
[23] Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput 2000;12:2451–71.
[24] Gers FA, Schmidhuber J. Recurrent nets that time and count. IEEE. 2000. p. 189–94.
[25] Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014.
[26] Liu H, Mi X, Li Y. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. Energy Convers Manage 2018;159:54–64.
[27] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. 2014.
[28] Kingma DP, Adam BaJ. A Method for Stochastic Optimization. 2014.
[29] Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J. LSTM: a search space odyssey. IEEE T Neur Net Lear 2017;28:2222–32.
[30] Guo Z, Uhrig RE. Using genetic algorithms to select inputs for neural networks. IEEE Comput Soc Press; 1992. p. 223–34.
[31] Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geosci Model Dev 2014;7:1247–50.
[32] Quinino RC, Reis EA, Bessegato LF. Using the coefficient of determination R2 to test the significance of multiple linear regression. Teach Stat 2013;35:84–8.
[33] Li R, Jin Y. A wind speed interval prediction system based on multi-objective optimization for machine learning method. Appl Energy 2018;228:2207–20.
[34] Alessandrini S, Delle Monache L, Sperati S, Cervone G. An analog ensemble for short-term probabilistic solar power forecast. Appl Energy 2015;157:95–110.
[35] Liu Y, Ye J, Ye L, Qin H, Hong X, Yin X. Monthly streamflow forecasting based on hidden Markov model and Gaussian Mixture Regression. J Hydrol 2018;561:146–59.
[36] Ruder S. An overview of gradient descent optimization algorithms. 2016.