

A Prediction Model for Ultra-Short-Term Output Power of Wind Farms Based on Deep Learning

Y. S. Wang, J. Gao, Z. W. Xu, J. D. Luo, L. X. Li

Yongsheng Wang

1. College of Computer and Information Eng., Inner Mongolia Agricultural University, Hohhot 010018, China
 2. Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application for Agriculture and Animal Husbandry, Hohhot 010018, China
 3. College of Data Science and Application, Inner Mongolia University of Technology, Hohhot 010080, China
 4. Inner Mongolia Autonomous Region Eng. & Technology Research Center of Big Data Based Software Service, Hohhot 010080, China
- wangys@imut.edu.cn

Jing Gao*

1. College of Computer and Information Eng., Inner Mongolia Agricultural University, Hohhot 010018, China
 2. Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application for Agriculture and Animal Husbandry, Hohhot 010018, China
- *Corresponding author: gaojing@imau.edu.cn

Zhiwei Xu

1. College of Data Science and Application, Inner Mongolia University of Technology, Hohhot 010080, China
 2. Inner Mongolia Autonomous Region Eng. & Technology Research Center of Big Data Based Software Service, Hohhot 010080, China
 3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
- xuzhiwei2001@ict.ac.cn

Jidong Luo

Haohan Data Technology Co., Ltd, Beijing 100080, China
11034161@qq.com

Leixiao Li

1. College of Data Science and Application, Inner Mongolia University of Technology, Hohhot 010080, China
 2. Inner Mongolia Autonomous Region Eng. & Technology Research Center of Big Data Based Software Service, Hohhot 010080, China
- llxhappy@126.com

Abstract

The output power prediction of wind farm is the key to effective utilization of wind energy and reduction of wind curtailment. However, the prediction of output power has long been a difficulty faced by both academia and the wind power industry, due to the high stochasticity of wind energy. This paper attempts to improve the ultra-short-term prediction accuracy of output

power in wind farm. For this purpose, an output power prediction model was constructed for wind farm based on the time sliding window (TSW) and long short-term memory (LSTM) network. Firstly, the wind power data from multiple sources were fused, and cleaned through operations like dimension reduction and standardization. Then, the cyclic features of the actual output powers were extracted, and used to construct the input dataset by the TSW algorithm. On this basis, the TSW-LSTM prediction model was established to predict the output power of wind farm in ultra-short-term. Next, two regression evaluation metrics were designed to evaluate the prediction accuracy. Finally, the proposed TSW-LSTM model was compared with four other models through experiments on the dataset from an actual wind farm. Our model achieved a super-high prediction accuracy 92.7% as measured by d_MAE , an evidence of its effectiveness. To sum up, this research simplifies the complex prediction features, unifies the evaluation metrics, and provides an accurate prediction model for output power of wind farm with strong generalization ability.

Keywords: wind power, output power, ultra-short-term prediction, deep learning (DL), long short-term memory (LSTM) model.

1 Introduction

The output power of wind turbines is very unstable, due to the stochasticity and volatility of wind energy. The grid-connection of a massive amount of wind power poses a huge challenge to the operation and dispatching of the power system and the security of the grid [4, 7]. Against this backdrop, it is very meaningful to predict the output power of wind farms in a future period. Accurate predictions help to rationalize dispatch and maintenance plans, and improve the utilization of wind power and wind energy [15, 18].

By time scale, the output power prediction of wind farms falls into long-term prediction, medium-term prediction, and ultra-short-term prediction. The final category refers to the rolling forecast of the output power of wind farms in the coming hours. If predicted accurately, the data on ultra-short-term output power can be used to ease the pressure on frequency adjustment, and reduce the capacity of spinning reserve, making the power system and power supply more reliable [10, 23].

For decades, both the industry and academia have probed deep into the output power prediction of wind farms. Three types of prediction methods have been developed with stable performance [9]: physical modelling [3], statistical modelling [1] and intelligent computing [2].

To obtain the output power, physical modelling derives the output power curve of wind farms through hydrodynamic and thermodynamic analyses on the results of numerical weather prediction (NWP) and the surface and spatial correlation data around the wind farms. This prediction strategy, involving complex models, numerous empirical parameters, and massive data on terrain and meteorology, is faced with heavy computing loads and slow updates. Therefore, physical modelling only applies to medium to long-term prediction. In this paper, the predictions of physical models are used for comparative analysis in experiments.

Based on time series of output power and wind speed, statistical modelling forecasts future output power by mapping the input features to the time series of output power, in the light of historical data only. The common methods of statistical modeling include the Kalman filter [28], stochastic time series method [12, 14], and support vector machine [6, 8, 20]. This prediction strategy cannot always make accurate predictions, owing to the difficulty in modelling, complexity of parameters and poor ability of generalization.

Intelligent computing is increasingly popular in the output power prediction of wind farms, thanks to the development of computer hardware and software and artificial intelligence (AI). Intelligent algorithms like wavelet analysis and genetic algorithm (GA) have all been introduced to output power prediction [11, 26]. Under different principles, these algorithms extract data features with varied structural designs. The applications of intelligent computing have enriched the theories on output power prediction of wind farms. However, the prediction effects fall short of the expectations of wind power enterprises, as the parameters are too complex and randomly initialized.

With the boom of deep learning (DL) [5, 17], many researchers have attempted to apply the DL in intelligent computing. The multilayered structure of deep neural networks (DNNs) can fit complex nonlinear mappings, and effectively prevent vanishing gradient [13, 16]. Hence, the DNNs have clear advantages in handling massive samples and nonlinear data. Xue et al. [24] successfully

combined the gated recurrent units (GRU), an improved version of long short-term memory (LSTM), and convolutional neural network (CNN) into a DL network. Nevertheless, DL networks should not be directly adopted to predict output power of wind farms, because the output power is affected by multiple constantly-changing factors. Otherwise, the DL networks will have poor prediction accuracy and generalization ability, and even fail to converge.

Drawing on the merits of the above prediction methods, this paper aims to develop an output power prediction strategy for wind farms, which can overcome the existing problems in the prediction task with its strong generalization ability and high forecast accuracy. For this purpose, the authors put forward a DL prediction model for output power prediction of wind farm, based on the LSTM and time sliding window (TSW). The proposed model is denoted as TSW-LSTM. Firstly, the data from multiple sources (e.g. meteorology and historical power) were fused and cleaned. Then, the TSW was introduced to set up an input dataset of wind power time series, and extract the cyclic features of output power. After that, a DL network model was constructed based on the LSTM for ultra-short-term prediction of output power in wind farm. The proposed model was verified through comparative experiments on the actual dataset of a wind farm. The results show that our model achieved the accuracy of 92.7%, as measured by d_MAE .

This research makes two major contributions:

(1) The novel concept of TSW was introduced to construct the dataset. With the aid of the TSW, the original small dataset was expanded in size, such that data features could be extracted as much as possible. Thanks to the robustness of the extracted features, the proposed TSW-LSTM prediction model boasts strong generalization ability and high prediction accuracy.

(2) There is no unified, intuitive criterion for regression-based output power predictions of wind farms. To solve the problem, two new performance metrics were designed for statistical regression, namely, statistical distribution of maximum relative error (s_MRE), and the mean distribution difference of mean absolute value (d_MAE). The two metrics are suitable for communication in the wind power industry.

The remainder of this paper is organized as follows: Section 2 fully explains our research method from the perspectives of overall framework, data preprocessing, dataset construction, TSW-LSTM model building, and evaluation of model performance; Section 3 verifies the performance of our model in output power prediction of wind farms, details the sources and features of experimental data, and introduces the experimental process, including constructing dataset, setting up evaluation criteria, designing experiments, and comparative analysis of experimental results; Section 4 wraps up the research by explaining the causes of good prediction effects of our TSW-LSTM model.

2 Methodology

2.1 Overall framework

The output power prediction of wind farms is essentially mapping a set of input series to a set of output series. The key issue lies in the generation of a series of predicted output powers. As shown in Figure 1, this paper designs a six-step prediction approach:

Step 1. Fusion and cleaning of multi-source data. The data on meteorology, turbine state and power were sampled from a wind farm. The sampling intervals were unified and the data sampled at the same time were stitched together. Then, the missing values were imputed by multiple linear regression (MLR), and the outliers were corrected through piecewise linear interpolation (PLI), creating the initial dataset.

Step 2. Dimension reduction and standardization. The main factors affecting the output power were identified through principal component analysis (PCA), aiming to reduce the dimensionality of the data, while retaining most of the effective features. Next, the data were standardized through discretization, normalization and one-hot coding, producing a discrete dataset of zeros and ones that facilitates machine learning.

Step 3. Dataset construction based on the TSW. The time cycles of historical data (i.e. meteorological data, turbine state data and power data) were identified and extracted. On this basis, the TSW was introduced to set up a training set and a test set.

Step 4. Construction of the DL model. Based on the LSTM, a DL model was constructed to predict the ultra-short-term output power of the wind farm. Since it is capable of processing the dataset generated by the TSW, the proposed model is denoted as the TSW-LSTM. The model adopts a multilayered neural network, including LSTM layers, fully-connected layers, etc.

Step 5. Model training and optimization. The DL model was trained by the training set. The series of influencing factors (e.g. meteorology and turbine state) were mapped into the series of output powers. Next, the prediction effect of the trained model was evaluated and optimized based on the test set.

Step 6. Prediction of output power series. The optimized model was applied to predict the output power series of a wind farm in a specified period in future. The prediction results were compared with the actual output power, and contrasted with the results of other prediction methods.

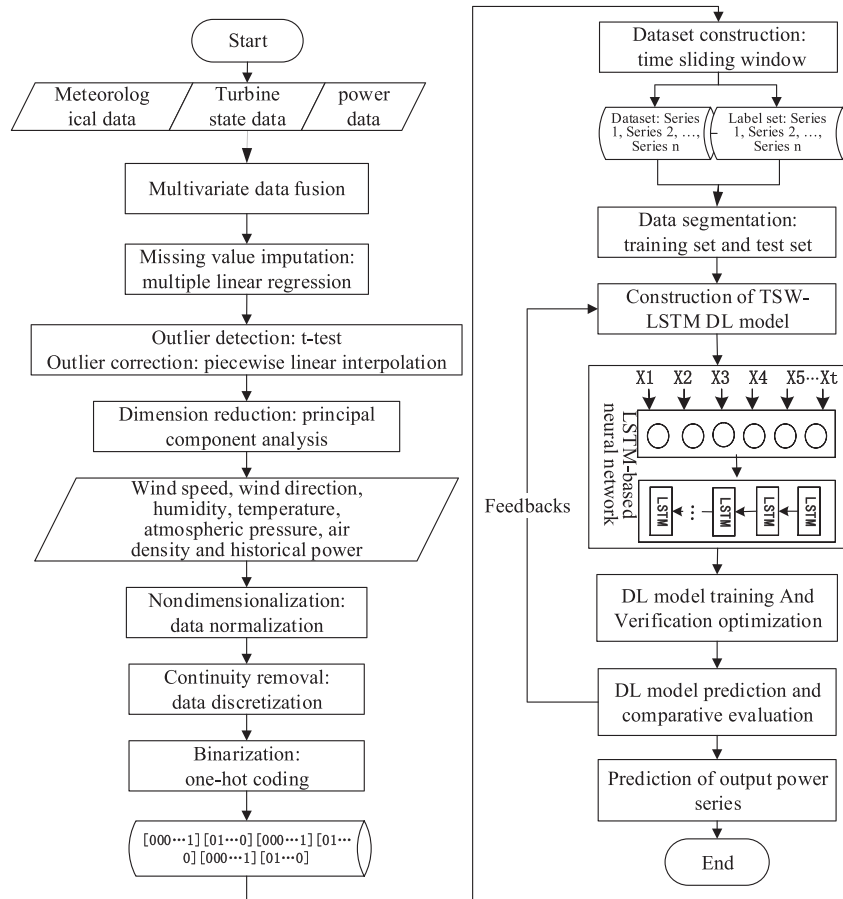


Figure 1: The workflow of output power prediction of wind farms

2.2 Data fusion and cleaning

During the operation, a wind farm generates a huge amount of data. The generated data fall in different categories, and differ in format and sampling frequency. There are often anomalies like missing values and outliers. The data entries are interconnected, involving key factors that affect turbine operations [25]. Therefore, it is necessary to fuse and clean the original data before using them in output power prediction. In this paper, the meteorological data, turbine state data and power data collected from a wind farm are fused and cleaned to create a complete initial dataset.

2.2.1 Data fusion

The output power prediction of wind farm involves three types of data: meteorological data (e.g. wind speed, wind direction, humidity, temperature, atmospheric pressure and air density), turbine

state data (e.g. engine room temperature and generator torque), and power data (e.g. rated output power, planned output power, corrected output power and actual output power).

According to the provisions of China's National Energy Administration, the sampling intervals of all types of data were unified as 15min. Then, the data sampled at the same time were stitched together, such that all three types of data are presented in the form of a unified 2D table at unified time points, forming the initial 2D dataset.

2.2.2 Data cleaning

Part of the collected data may be missing or distorted under turbine failure, transmission interruption and signal interference. The missing values and outliers affect the statistical and distribution features of the collected data. In this case, the confidence interval of the collected data will widen, and the confidence coefficient will be reduced. If the data are analyzed by DL models, the ensuing errors will suppress the prediction accuracy. Hence, the missing values must be imputed, and the outliers be corrected.

(1) Missing value imputation

Each missing value is usually directly removed, or simply padded with zeros, the previous value, the subsequent value or the mean value. In the data collected from the wind farm, the missing values are the time series of output powers. These values are distributed continuously or randomly in the collected data. The direct removal of these values will damage the time continuity and correlation of the time series. The simple padding will lower the variance of variables, and bring large covariance and correlation deviation, undermining the original data structure.

The time series of output powers are the only missing values in the collected data, while all the other data features are complete. In other words, the data missing problem has only one variable. Thus, the MLR was employed to fit and complete the missing values in the collected data. Here, the output powers are regarded as a continuous time series. It is assumed that the output power at time t_i is missing. Let $t_1, t_2, t_3, \dots, t_m$ be the m moments adjacent to time t_i , at which the output powers are known. Then, the missing output power at time t_i can be imputed by the MLR:

$$y_{ti} = \beta_0 + \beta_1 y_{t1} + \beta_2 y_{t2} + L + \beta_m y_{tm} \quad (1)$$

where, y_{ti} is the explained variable, i.e. the output power at time t_i ; $y_{tk} (k = 1, 2, \dots, m)$ are m explanatory variables, i.e. the output powers at time t_k ; $\beta_j (j = 1, 2, \dots, m)$ is the partial correlation coefficient relative to y_{tk} , i.e. the influence of y_{tk} over y_{ti} ; μ_i is a random error obeying a Gaussian distribution with a mean of 0 and a variance of σ^2 . Then, the partial correlation coefficient β_j was estimated by maximum likelihood, yielding $\hat{\beta}_j$. Substituting the estimate to formula (1), the missing value \hat{y}_{ti} can be estimated by:

$$\hat{y}_{ti} = \beta_0 + \hat{\beta}_1 y_{t1} + \hat{\beta}_2 y_{t2} + \dots + \hat{\beta}_m y_{tm} \quad (2)$$

The missing values were imputed iteratively as above, producing a complete dataset without any missing value.

(2) Outlier detection and correction

Firstly, the outliers were found out through t-test. The non-suspicious values were regarded as a normally distributed population. The mean value \bar{x} and standard deviation s of the population were computed. Meanwhile, the suspicious values were considered as a special population with a sample size of 1. If the suspicious and non-suspicious values belong to the same population, there should be no significant difference between them. The t-statistic can be defined as:

$$k = |x_d - \bar{x}| \quad (3)$$

Suppose the σ can be replaced with the standard deviation s . Then, the t-statistic can be rewritten as $k = \frac{|x_d - \bar{x}|}{s}$. If the t-statistic is greater than the threshold under the corresponding confidence, then x_d must be an outlier.

After that, the outliers were corrected through the PLI: each two adjacent nodes were connected by a straight line, forming a polyline, i.e. the PLI function $I_n(x)$ satisfying $I_n(x) = y$. In each small

interval x_i, x_{i+1} , $I_n(x)(i = 1, 2, \dots, n)$ is a linear function. $I_n(x)$ and $l_i(x)$ can be respectively expressed as:

$$I_n(x) = \sum_{i=0}^n y_i l_i(x) \tag{4}$$

$$l_i(x) = \left\{ \begin{array}{ll} \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x-x_{i+1}}{x_i-x_{i+1}}, & x \in [x_{i+1}, x_i] \\ 0, & \text{otherwise} \end{array} \right\} \tag{5}$$

The interpolation of point x was computed by $I_n(x)$, using the two nodes on the left and right of x . The computing load is independent of the number n of nodes. However, the greater the n value, the more the segments, and the smaller the interpolation error.

The outliers were iteratively processed as above, until all of them had been corrected.

2.3 Dimension reduction and standardization

Data fusion and cleaning produced the initial dataset with unified sampling frequency, complete attributes, and rational data distribution. But the dataset cannot be directly imported to the DL model. To solve the problem, the dataset was transformed into a 3D sparse matrix of zeros and ones, through PCA, discretization, normalization and one-hot coding.

2.3.1 Dimension reduction

The initial dataset reflects the historical states of the wind farm more accurately than the collected data. The dataset contains various features, most of which have little impact on the output power. If all these features are imported, the DL model will face a heavy computing load and might not converge during the training. Thus, the PCA was carried out to select the key features that affect the output power, and reduce the dimensionality of the dataset, without sacrificing the effective information [21].

The PCA mainly maps n -dimensional features to a k -dimensional space, that is, reconstruct k -dimensional features based on the original n -dimensional features. During the PCA, a set of mutually orthogonal coordinate axes were found sequentially from the original space. The first axis points to the largest variance in the original data, the second axis points to the largest variance in the plane orthogonal to the first axis, and the third axis points to the largest variance in the plane orthogonal to the first two axes. The rest can be deduced by analogy. Most variances of the original data are contained in the first k axes, while the variances of the latter axes are almost zero. To reduce the dimensionality, the first k axes that contain most variances were preserved, and the other axes that contain near-zero variances were ignored.

The PCA was implemented through eigenvalue decomposition of the covariance matrix. To begin

with, the initial dataset was rewritten as a matrix $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$, where n is the number

of features and m is the number of samples. Then, the dimensionality of matrix A , which contains m samples with n features, was reduced to k in the following steps:

Step 1. Decentralization: subtract the mean value of each column from the features in that column, creating a new matrix A .

Step 2. Calculate an $n \times n$ covariance matrix $Cov(A)$ by $Cov(A) = 1/m \cdot A^T \cdot A$. The covariance matrix of three features can be expressed as:

$$Cov(x, y, z) = \begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) \end{bmatrix} \tag{6}$$

where, the diagonal element c_{ii} is the variance of the i -th feature; any other element c_{ij} is the covariance between the i -th and j -th elements. The covariance matrix is symmetric.

Step 3. Solve the eigenvalues and eigenvectors of the covariance matrix through eigenvalue decomposition. In other words, decompose matrix A into:

$$Cov(A) = Q \Sigma Q^{-1} \quad (7)$$

where, Q is the matrix of eigenvectors of matrix $Cov(A)$; Σ is a diagonal matrix of eigenvalues.

Step 4. Sort the eigenvalues in descending order and select the top k eigenvalues. Then, take the k eigenvectors corresponding to the top k eigenvalues as row vectors, forming an eigenvector matrix P .

Step 5. Map matrix A to the new space of the k eigenvectors by $Y = PA$, marking the end of dimension reduction.

2.3.2 Data standardization

Despite dimension reduction, the features of our data still have obvious dimensional difference. If these features are directly inputted to the prediction model, the network learning will focus on the variables with a large dimensional range. To unify the dimension, the variables were normalized by min-max scaling:

$$X' = (X - X_{\min}) / (X_{\max} - X_{\min}) \quad (8)$$

where, X is the value of a feature at the current moment; X_{\min} and X_{\max} are the minimum and maximum of the feature, respectively; X' is the normalized value of the feature.

After normalization, the change trend and distribution law of each feature were observed carefully, and used to perform discretization and one-hot coding. Take the wind direction for example. The wind direction data of the wind farm obey a continuous distribution from 0° to 360° . The values of many angles are nothing but noises to DNNs. In fact, the rotation plane of the blade is adjusted automatically by the yaw system of the turbine, according to the wind direction. The output power is not greatly affected by small changes in wind direction. Therefore, the wind directions were discretized in segments to $[0, 7]$, depending on their distribution law. The discretized values were then converted into a sparse matrix of zeros and ones through one-hot coding, which promotes the training effect of the prediction model. The other features were processed in a similar manner.

2.4 TSW-based dataset construction

The processed data are insufficient to mine all the features of the wind farm. To expand the input data, the TSW algorithm was adopted to construct the dataset to be inputted to the prediction model, laying a good basis for accurate prediction. The actual output powers form a time series with a specific cycle. Extracting the cyclic features help to improve the prediction accuracy. Hence, the input dataset should cover the actual output powers.

First, the cyclic features of the actual output powers were analyzed to determine the lookback of the sliding window, such that the output power curve has basically the same phase during the sliding window. Next, the window was moved downward sequentially to segment the processed data: the first to the lookback-th entries were taken as the first sample, the second to the lookback+1-th entries as the second sample, and the rest can be deduced by analogy. In this way, the elements of the dataset and label set were obtained, creating the training set and test set. If there are L entries in the processed data, then the data were expanded by lookback times to $(L - \text{lookback} + 1) \times \text{lookback}$ entries.

Algorithm 1 shows the TSW algorithm used to construct the input dataset. The standardized 2D sparse matrix was inputted to the algorithm. The first column to the penultimate column are meteorological, turbine state and power features, while the last column are output power features. The total number of rows equals the number of entries in the collected data. The first row of the algorithm defines the width of the sliding window, i.e. lookback; the second row defines and saves the empty lists dataX and dataY for the dataset and label set; the sixth to tenth rows define an iteration of the algorithm. In each iteration i , a 2D matrix of lookback rows and one-fewer columns was taken from the input dataset and added to the dataX list, and the element on the lookback + 1 - th row in

the last column of the input dataset was added to the dataY list. At the end of the iterative process, the two lists contain the dataset and label set required for the DL prediction model.

Algorithm 1 The TSW algorithm used to construct the input dataset

- 1: Input: The processed input dataset
 - 2: Outputs: dataX (list of data imported to the DL model) and dataY (list of labels imported to the DL model).
 - 3: Start:
 - 4: Define the lookback of the TSW=3*96;
 - 5: Define empty lists dataX and dataY;
 - 6: **for all** $i = 1, 2, \dots, n, [\text{len}(\text{dataset}) - \text{lookback}]$ **do**
 - 7: Form element a based on row i to row $(i + \text{lookback})$;
 - 8: Add a to dataX;
 - 9: Add the last column of row $(i + \text{lookback})$ to dataY;
 - 10: **end for**
 - 11: End.
-

2.5 LSTM modelling

2.5.1 The LSTM network

Both the processed input dataset and the predicted output powers are time series. To handle end-to-end series, a DL model can be established based on the recurrent neural network (RNN). The cyclic feedback structure of the RNN correlates the output state at time t with the historical signals before time t , thereby enhancing the network memory and reduces parameters. In theory, the RNN can handle time series of any length. However, if the input time series or the time series to be predicted is too long, the historical information will be replaced with the more recent information, causing vanishing or exploding gradient during model training. In our experiment, the input time series has more than 13,000 entries. The RNN cannot achieve a good prediction effect on such a long input time series.

The vanishing or exploding gradient can be effectively solved by the LSTM, an extension of the RNN. The LSTM maintains the excellent structure of the RNN, and adds four new structures, namely, an input gate, a forget gate, an output gate and a memory cell. The additional structures can memorize and forget the entries in the input data series in a reasonable manner, allowing the long time series to propagate freely in the network being trained. The cyclic features of the input and predicted time series could be memorized well by the LSTM, laying a solid basis for accurate prediction [19]. The typical structure of a memory module in the LSTM network is explained in Figure 2, where $t(t = 1, 2, \dots, n)$ is time step; x_t , N_t , and h_t are the input signal, state, and output signal of the memory module, respectively; f_t , i_t and o_t are the state signals of the forget gate, input gate and output gate, respectively.

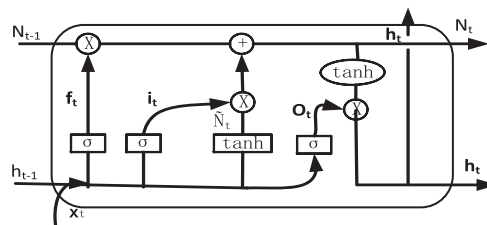


Figure 2: The structure of a memory module in the LSTM network

Let W and b be the weight and bias of each layer, respectively. The operation of the memory module can be described as follows:

The forget gate determines which input information should be discarded. This gate receives the input signal x_t of the current module and the output signal of the previous module h_{t-1} , and generates

a signal of zero or one by sigmoid (activation function):

$$f_t = \sigma(W_f \cdot [H_{t-1}, x_t] + b_f) \tag{9}$$

The generated signal undergoes point multiplication with the state signals N_{t-1} of the previous module. The resulting signal $f_t * N_{t-1}$ determines whether the state signal N_{t-1} of the previous module should be forwarded.

The input gate controls the states of x_t and x_t flowing into the current module. Based on the two signals, a signal i_t and a candidate signal N_t are generated by sigmoid (activation function) and tanh (activation function), respectively:

$$i_t = \sigma(w_i \cdot [H_{t-1}, x_t] + b_i) \tag{10}$$

$$\tilde{N}_t = \tanh(i_t (W_C \cdot [h_{t-1}, x_t] + b_C)) \tag{11}$$

The update gate derives the state N_t of the current module from the dot product N_t of the above two signals.

It can be seen that the state signal f_t of the forget gate determines whether the state of the input time series in the previous module should be memorized. If not, the state signal of the current module N_t only depends on x_t and x_t :

$$N_t = f_t \cdot N_{t-1} + i_t \cdot \tilde{N}_t \tag{12}$$

Based on x_t and x_t , the output gate computes a signal o_t by sigmoid (activation function):

$$O_t = \sigma(W_O \cdot [H_{t-1}, x_t] + b_O) \tag{13}$$

Meanwhile, a signal $\tanh(N_t)$ is derived from N_t by the tanh (activation function). The dot product between o_t and $\tanh(N_t)$ is the output signal of the current module h_t :

$$h_t = O_t \cdot \tanh(N_t) \tag{14}$$

The above description shows that the cyclic features of the input time series can be memorized for a long time in the modules of the LSTM network. The useless information can be discarded by the forget gate. Through multiple trainings, the LSTM network can theoretically fit the nonlinear relationship between the input time series and the output powers.

2.5.2 Model construction

The LSTM network was integrated with the TSW-based dataset to create a prediction model for output power of wind farm. As shown in Figure 3, the established TSW-LSTM model consists of an input layer, two fully-connected layers, three LSTM layers, a regularization layer, and a dropout layer. The structure of the model is detailed as follows:

(1) The TSW-based dataset was inputted to the fully-connected layer. The dataset exists in the form of a 3D matrix X, where the first dimension is the number of elements; each element is a 2D matrix, whose row number is the size of time window and column number is the number of input features. The number of elements on the fully-connected layer equals the number of features. All the input features were transferred to the next layer.

(2) The second layer of our model is the first LSTM layer, which contains 32 memory modules. On this layer, the input dataset was automatically learned and encoded. The correlation between meteorological data and power data was extracted, so were the cyclic features of the two types of data. All the extracted information was transferred to the next layer.

(3) The dropout layer falls between the first and second LSTM layers. This layer randomly cuts off the connections between the two LSTM layers, aiming to prevent overfitting.

(4) The second LSTM layer has 64 memory modules. On this layer, the input dataset was further learned and encoded to enhance the accuracy of nonlinear fitting.

(5) The regularization layer, falling between the second and third LSTM layers, helps to prevent overfitting.

(6) The third LSTM layer involves 96 memory modules. On this layer, the signals from the previous layers were learned for the last time, making nonlinear fitting even more accurate.

(7) The last layer of our model is a fully-connected layer, which outputs the series of predicted output powers.

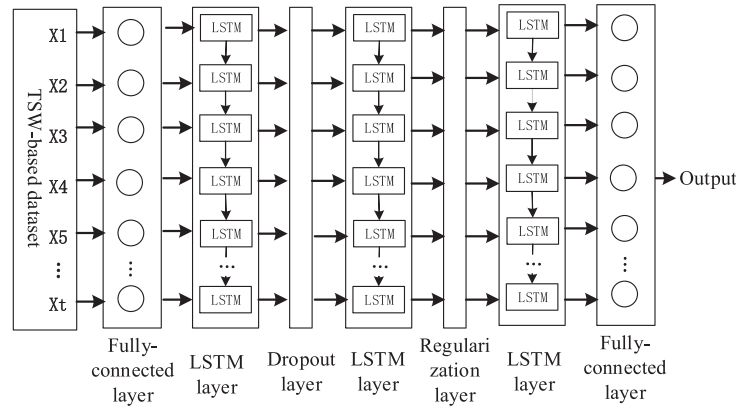


Figure 3: The structure of the TSW-LSTM model

2.6 Performance evaluation

Error metrics are often adopted to evaluate the performance of regression analysis [22, 27], such as mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE) and R-squared (R^2) score. However, the prediction accuracy of output power of wind farm should be assessed by statistical metrics of accuracy. This paper designs two intuitive evaluation metrics, according to the statistical distributions of predicted values and true values. The two metrics are named as the statistical distribution of maximum relative error (s_MRE), and the mean distribution difference of mean absolute value (d_MAE).

The s_MRE value can be computed by:

$$\lambda_i = \frac{|y_i - \hat{y}_i|}{y_i}, n = \sum_{i=1}^n \lambda_i \leq \theta, A = \left\{ \frac{\sum_{i=1}^n \left[\frac{|y_i - \hat{y}_i|}{y_i} \right] \leq \theta}{N} \right\} \% = \left(\frac{n}{N} \right) \% \quad (15)$$

where, N is the number of elements in the test set; λ_i is the ratio of the absolute difference between the i -th predicted value and the actual value to the actual value; θ is the value that can be accepted by the wind farm; n is the number of predicted values that satisfy $< \theta$; $(n/N)\%$ is the statistical accuracy A of the prediction model. After consulting with the wind farm, the θ value was set to 0.2.

The d_MAE refers to the difference between the mean distribution of the MAE and 1: the greater the d_MAE, the higher the prediction accuracy. The d_MAE value can be computed by:

$$d_MAE = 1 - \frac{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}{\bar{y}} = 1 - \frac{MAE}{\bar{y}} \quad (16)$$

3 Experimental verification

The tf.keras, a high-level application programming interface (API) of TensorFlow, was adopted as the DL framework of our modelling process. The data processing and visualization modules were called to process and display the collected data; the optimizers and regularizers modules were called to optimize our model; the callbacks module was called to dynamically adjust the learning rate; the LSTM module was called to construct an LSTM DL model based on the TSW-based dataset. The programs were compiled in python, and the experiments were conducted in a DL environment accelerated by graphics processing unit (GPU) [13, 16, 29].

3.1 Experimental dataset

Our model was trained and tested by the data from a wind farm in Inner Mongolia, China. The experimental data encompass two parts: the NWP data (historical meteorological data) and the power data of the wind farm. The NWP data were collected from January 1st to May 22nd, 2019 at a frequency of 15min by the anemometer tower on the wind farm, and corrected against the historical weather forecasts. There are 13,632 entries in the NWP data. Each entry includes 9 fields: date and time occupy 2 fields, and meteorological features (e.g. wind speed, wind direction, air density and atmospheric pressure) occupy the other 7 fields. The power data were captured from January 1st to May 21st, 2019 at a frequency of 15min by the supervisory control and data acquisition (SCADA) of the wind farm. There are 13,536 entries in the power data.

Initial observations show that the NWP data have a high quality, without any missing values or outliers. This is because the data were collected from more than one source and cross-checked. The most important column in the power data provides the actual output powers. In this column, empty elements were found at 111 time points, which might be results from SCADA faults or transmission failures. Further analysis reveals that the 42 of the 111 missing values are at the end of the data, and could be removed directly. The remaining 69 values are inside the data. The removal of the latter values would undermine the data integrity.

Through the above analysis, the NWP data were stitched with the power data. After stitching, the fields of time and date were turned into an index, which is not used for machine learning. The other features were subjected to the PCA. The features (e.g. floor height and humidity) that do not greatly affect the output power were removed, leaving 6 features (e.g. wind speed, wind direction, atmospheric pressure and air density) in 13,494 effective and continuous entries. Among them, the actual output power was empty in 69 entries. These empty items were completed through the MLR and the outliers were corrected. In the end, 1,3494 effectively stitched entries were obtained, and organized as the initial dataset.

From an intuitive point of view, when other conditions remain the same, the output power of the wind farm should have an approximately linear relationship with the wind speed. Here, the correlation between the output power and wind speed is investigated by preparing a scatterplot based on 1,920 entries collected over 20 days. As shown in Figure 4, there is no linear relationship between wind speed and output power. This means the output power is influenced by various factors, making it difficult to create a physical model. The complex nonlinear relationship should be learned automatically through the DL.

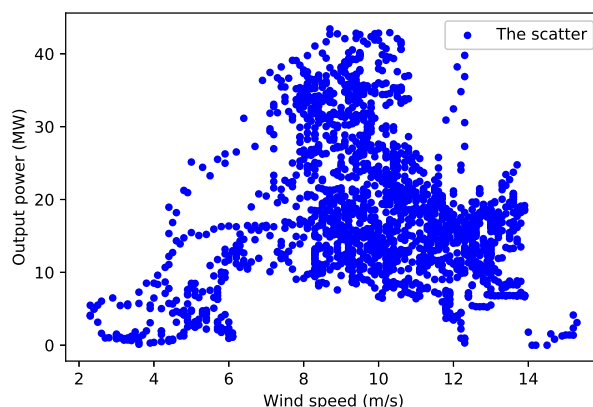


Figure 4: The scatterplot between wind speed and output power

3.1.1 Cyclic analysis

The actual output powers in the initial dataset were subjected to cyclic analysis. The samples in three consecutive days were visualized. On each day, there are 96 entries from 00: 00 to 24: 00. As

shown in Figure 5, the actual output powers on the three consecutive days had basically the same cyclic features. As a result, the cyclic features should be extracted from the actual output powers, in order to accurately predict the output power of the wind farm.

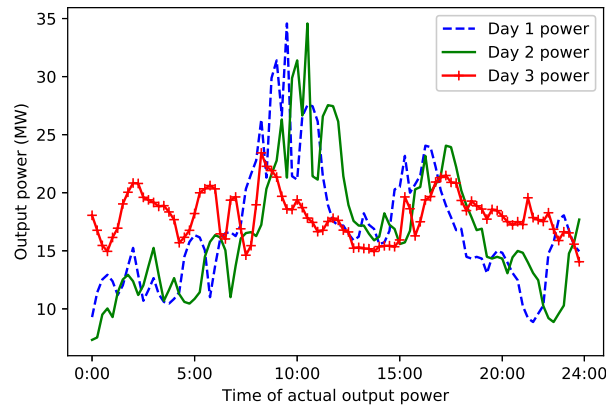


Figure 5: The cyclic features of actual output powers

3.1.2 TSW-based dataset construction

To extract the cyclic feature of output power, the power data and NWP data were fused into the input feature, and the initial dataset was divided into a dataset and a label set. The dataset is a 2D matrix of 13,494 rows and 7 columns, while the label set is a 1D matrix of 13,494 elements.

Referring to the cyclic distribution of data, the time step of the LSTM was set to 3 days, covering 96 sampling points; the lookback of the TSW was set to 288. Firstly, the first 288 entries of the dataset were selected by the sliding window to be the first element of the input dataset, and the 289-th element in the label set was taken as the corresponding label.

After that, the window was slid to the next entry in the dataset and the next element in the label set, and similar operations were performed. In the end, a 3D matrix was obtained as the input dataset. The number of elements in the input dataset equals the total number of entries in the original dataset plus one. Each element is a 2D matrix defined by a sliding window.

In our experiments, the final dataset and label set were (13,206, 288, 7) and (13,206, 1), respectively. Figure 6 explains the procedure of the TSW-based dataset construction. For simplicity, the lookback in the figure was set to 4.

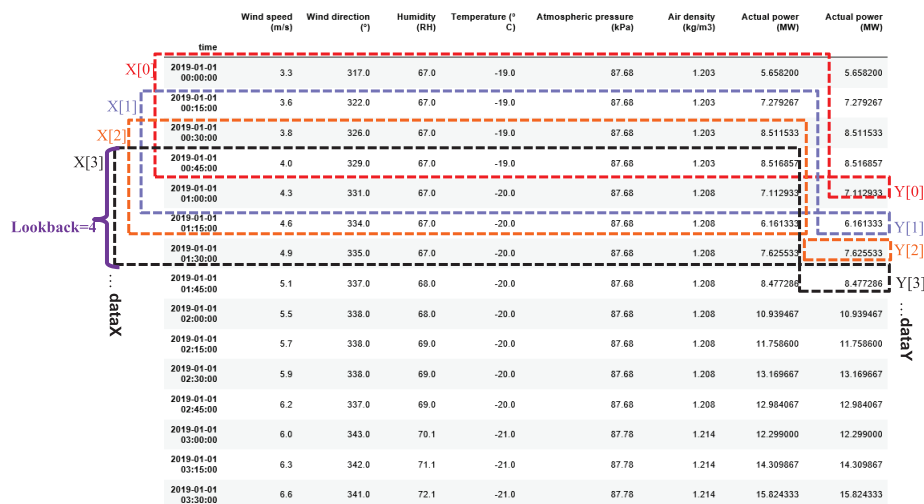


Figure 6: The procedure of the TSW-based dataset construction

3.2 Error metrics

3.2.1 MSE and RMSE

The MSE refers to the mean of the squared errors between the values predicted on the test set and the actual values. For the same dataset, the MSE is negatively correlated with the prediction effect. The MSE can be computed by:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (17)$$

where, n is the number of samples; y_i and \hat{y}_i are the actual values of and the values predicted on the samples, respectively; i is the serial number of samples.

The RMSE is the square root of the MSE. The two metrics have the same meaning. The RMSE is more suitable for computation and comparison, because the error dimension is reduced through the extraction of root. Both the MSE and RMSE increase with the number of samples. Hence, the magnitude of the two metrics is meaningless if the datasets are different. The same dataset was applied in all our experiments, so that the RMSE could be adopted to evaluate and compare the errors of different prediction models.

3.2.2 MAE

The MAE refers to the mean absolute error between the values predicted on the test set and the actual values. The MAE is negatively correlated with the prediction effect. This metric can accurately reflect the actual prediction error. Therefore, it was selected to evaluate the errors of different prediction models. The MAE can be computed by:

$$MAE = \frac{1}{n} |y_i - \bar{y}_i| \quad (18)$$

3.2.3 R² score

Despite their excellence under the same dataset, RMSE and MAE cannot measure the prediction effect if there are different dimensions. The impact of dimensional difference can be eliminated by R² score:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (19)$$

If R² score < 0, the prediction error is greater than the error of using the mean value; i.e. the model is meaningless. If R² score = 0, the numerator is equal to the denominator, and each predicted value equals the mean value, i.e. the model is still meaningless; If R² score = 1, the predicted values are equal to the actual values, i.e. the model makes error-free predictions. Thus, the closer the R² score is to 1, the better the prediction model.

3.3 Experimental design

Table 1: The settings of hyper parameters

	Number of nodes	Activation function	L1	L2	Dropout	Optimizer
Fully-connected layer	7	Rectified linear units (ReLU)	-	-	-	-
LSTM1	32	Sigmoid	0.001	0.002		Nadam
Dropout layer					0.3	
LSTM2	64	Sigmoid	0.001	0.001		Nadam
Regularization layer			0.001	0.001	0.5	
LSTM3	96	Sigmoid	0.001	0.002	0.6	Nadam
Fully-connected layer	1	Rectified linear units (ReLU)	-	-	-	-
Nadam	lr = 0.002, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-08					
reduce_lr	patience = 5, factor = 0.8, mode = "auto", verbose = 1, min_delta = 0.0001, cooldown = 0, min_lr = 0.0000001					

The tf.keras DL platform was deployed on the GPU. Then, a DL network was constructed according to the abovementioned procedure. The established network encompasses an input layer (a fully-connected layer), a hidden layer (two convolutional layers, a max-pooling layer, three LSTM layers and three dropout layers), and an output layer. There are five types of hyper parameters in the network, namely, the number of nodes in the input layer, the number of nodes in each LSTM layer, the regularization parameters L1 and L2, the dropout value, as well as the initial parameters and learning rate decay of Nesterov Adam (Nadam), a stochastic gradient descent optimizer. The hyper parameters are configured as shown in Table 1.

3.4 Experimental results

3.4.1 Experiments on our model

The TSW-LSTM model was trained for 50 iterations on the training set. The MAE loss curve (Figure 7) shows that the training and validation losses, relatively large at the start, exhibited a rapid decline. After 5 rounds of training, both losses started to gradually decrease. The decrease of training loss was relatively smooth, while that of validation loss fluctuated. Thus, the overfitting occurred in model training. Then, the model made automatic adjustments according to the preset hyper parameters, such as dropout, L1, L2 and learning rate (lr). After 30 rounds of training, the MAE losses of training and validation both slowly decreased. The two loss curves were basically horizontal after 50 rounds, indicating that the model has completely converged, the losses were minimized and the prediction accuracy was maximized.

The trained model was applied to predict the output power based on the test set. The predicted values were compared with the actual values of the test set. The predicted values and the actual values in 3 h are contrasted in Figure 8, where the x -axis is the sampling intervals of 15 min, and the y -axis is the output power (unit: MW). It can be seen that the predicted output power (4 MW) was 0.3 MW smaller than the actual output power (4.3 MW) at the 0th sampling point. In the 3 h ultra-short prediction period, the output power curve predicted by the TSW-LSTM agrees well with the actual output power curve at the wind farm, an evidence of the good effect of our model.

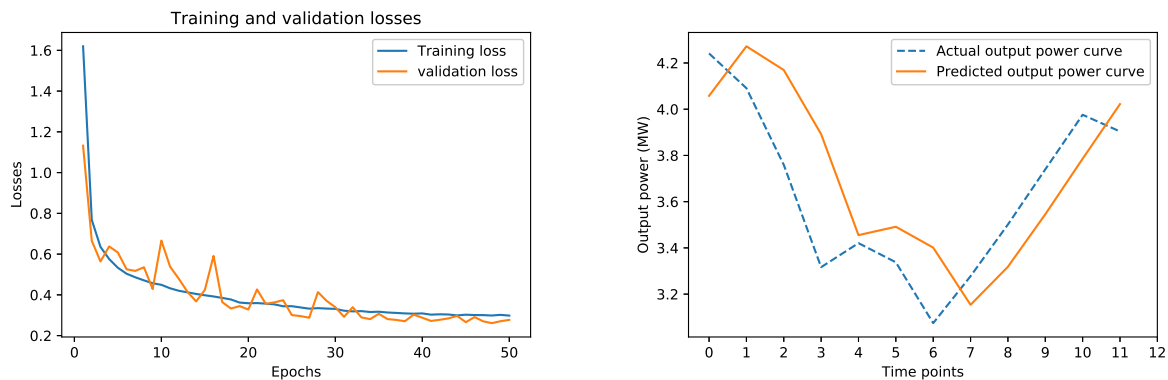


Figure 7: The training and validation losses of our model
 Figure 8: The comparison between the output power predicted by TSW-LSTM and the actual output power

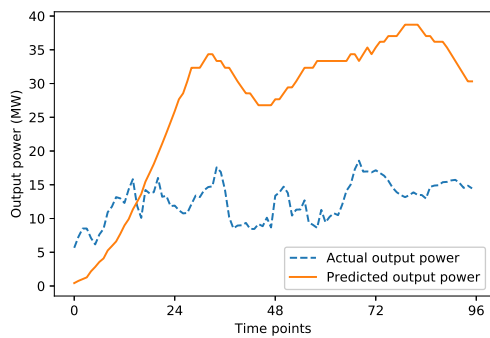
3.4.2 Contrastive experiments

To fully demonstrate its engineering value, our model was further compared with the physical model and several DL models: decision tree (DT), random forest (RF) and SVM. The four models were separately applied to the same dataset to predict the output power in the coming 24 h. The performance metrics of the models are compared in Table 2 and Figure 7. As shown in Table 2, the three machine learning models, namely, the DT, RF and SVM had similar MSEs, RMSEs and MAEs. The MAEs were all close to 20, a sign of the large gap between predicted and actual values. On

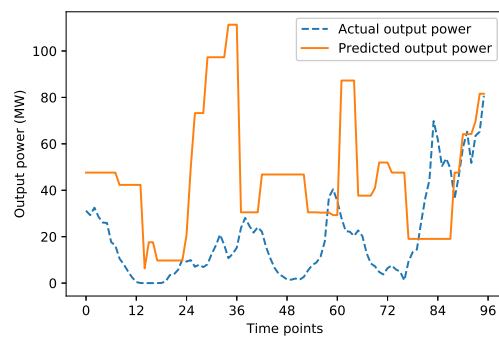
regression accuracy, the d_MAE values of the three machine learning models were about 60% and the s_MRE values were below 40%. The results show that the three models cannot realize satisfactory predictions.

Table 2: The comparison between the performance metrics of the models

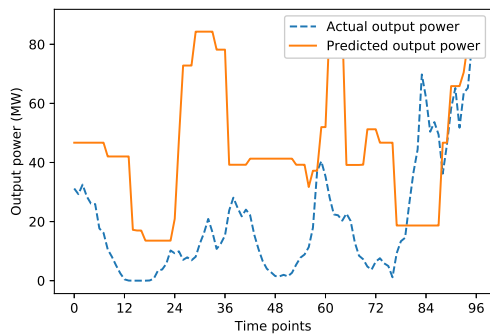
	MSE	RMSE	MAE	R ² score	d-MAE	s-MRE
Physical model	220.1	149	8.53	-	65.7%	56.3%
DT	1678.1	40.9	22.96	0.17	61.5%	37.4%
RF	1162.2	34.0	19.12	0.41	61.4%	38.6%
SVM	1218.7	35	20.86	0.38	62.9%	39.8%
TSW-LSTM	11.7	3.4	1.39	0.93	92.7%	77.9%



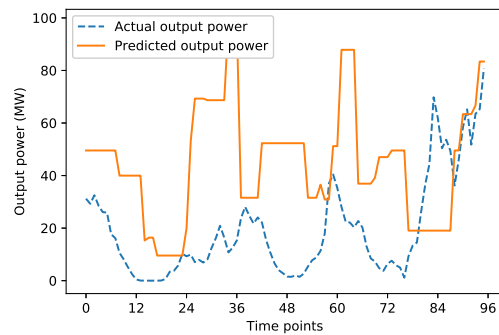
(a) Physical model prediction vs. actual output



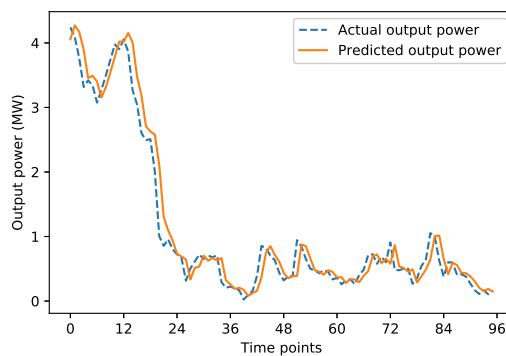
(b) DT prediction vs. actual output



(c) RF prediction vs. actual output



(d) SVM prediction vs. actual output



(e) TSW-LSTM prediction vs. actual output

Figure 9: The comparison between prediction effects

The physical model of the wind farm had an MAE of 8, much lower than that of any machine

learning model. Hence, the physical model predicted the output power more accurately than the three machine learning models. The proposed TSW-LSTM model achieved an MAE of 1.39, a d_MAE of 93% and an s_MRE of 78%. This means our model far outperformed the other four models in MAE and d_MAE, and achieved satisfactory predictions.

Figure 9 compares the predicted output power (solid line) of each model and actual output power (dotted line). It can be seen that the predicted curves of physical model, DT, RF and SVM for 24 h deviated far from the actual curve, i.e. none of the four models could fit the actual output power curve. By contrast, the predicted curve of the TSW-LSTM agrees well with the actual curve, reflecting the high prediction accuracy of our model.

4 Conclusions

This paper presents a time series prediction model based on the LSTM network: the TSW-LSTM model. The wind power data from multiple sources were fused, and processed in multiple steps into an input dataset. Under the input dataset, the output power of wind farm was predicted accurately by the proposed DL model. The main conclusions are as follows:

(1) The proposed TSW-LSTM model can effectively fit the output power curve of the wind farm, and clearly outperform the physical model of the wind farm and three machine learning models.

(2) The wind farm data come from multiple sources, and contain many missing values and outliers. This paper fuses the multi-source data, and cleans the data through MLR and PLI. The fusion and cleaning can effectively mine the data features, suppress noises, and improve prediction accuracy.

(3) The TSW-LSTM prediction model was trained by historical data and optimized repeatedly to overcome the exploding and vanishing gradients in training. The optimized model provides a desirable prediction tool.

(4) Considering the cyclic features of the power data of the wind farm, the historical power data were fused into the input dataset, and the TSW was introduced to construct the input dataset. In this way, the cyclic features were effectively extracted from the actual output power, pushing up the prediction accuracy.

Funding

This work is supported by Inner Mongolia Science and Technology Major Special Projects (2019ZD016); Natural Science Foundation of China (61462070, 61962045, 61502255, 61650205); Inner Mongolia Agricultural University Doctoral Scientific Research Fund Project (NO.BJ09-44); Natural Science Foundation of Inner Mongolia Autonomous Region (2019MS03014, 2018MS-06003, 2019MS06027); Inner Mongolia Key Technological Development Program (2019ZD015); Key Scientific and Technological Research Program of Inner Mongolia Autonomous Region (2019GG273).

Conflict of interest

Authors declare no conflict of interest.

References

- [1] Alexiadis, M.C.; Dokopoulos, P.S.; Sahsamanoglou, H.S.; Manousaridis, I.M. (1998). Short-term forecasting of wind speed and related electrical power, *Solar Energy*, 63(1), 61–68, 1998.
- [2] Brown, B.G.; Katz, R.W.; Murphy, A.H. (1984). Time series models to simulate and forecast wind speed and wind power, *Journal of climate and applied meteorology*, 23(8), 1184–1195, 1984.
- [3] Chen, Y.; Zhou, H.; Wang, W.P.; Cao, X.; Ding, J. (2011). Analysis and improvement of ultra-short-term prediction results of wind farm output power, *Power System Automation*, 35(15), 30–33, 2011.

- [4] Costa, A.; Crespo, A.; Navarro, J.; Lizcano, G.; Madsen, H.; Feitosa, E. (2008). A review on the young history of the wind power short-term prediction, *Renewable and Sustainable Energy Reviews*, 12(6), 1725–1744, 2008.
- [5] de Sousa Junior, W.T.; Montevechi, J.A.B.; Miranda, R.de.C.; Rocha, F.; Vilela, F.F. (2019). Economic Lot-Size Using Machine Learning, Parallelism, Metaheuristic and Simulation, *International Journal of Simulation Modelling*, 18(2), 205–216, 2019.
- [6] Ding, Z.Y.; Yang, P.; Yang, X.; Zhang, Z. (2012). Wind power prediction method based on sequential time clustering support vector machine, *Automation of Electric Power Systems*, 36(14), 131–135, 2012.
- [7] Ding, M.; Zhang, C.; Wang, B.; Bi, R.; Miao, L.Y.; Che, J.F. (2019). Short-term forecasting and error correction of wind power based on power fluctuation process, *Automation of Electric Power Systems*, 43(3), 2–9, 2019.
- [8] Gorur, K.; Bozkurt, M.R.; Bascil, M.S.; Temurtas, F. (2019). GKP signal processing using deep CNN and SVM for tongue-machine interface, *Traitement du Signal*, 36(4), 319–329, 2019.
- [9] Han, Z.F.; Jin, Q.M.; Zhang, Y.K.; Bai, R.Q.; Guo, K.M.; Zhang, Y. (2019). Wind power forecasting methods and new trends, *Power System Protection and Control* 47(24), 178–187, 2019.
- [10] Hong, D. Y.; Ji, T. Y.; Li, M. S.; Wu, Q. H. (2019). Ultra-short-term forecast of wind speed and wind power based on morphological high frequency filter and double similarity search algorithm, *International Journal of Electrical Power & Energy Systems*, 104, 868-879, 2019.
- [11] Kim, J.B. (2019). Implementation of artificial intelligence system and traditional system: A comparative study, *Journal of System and Management Sciences*, 9(3), 135–146, 2019.
- [12] Lee, D.; Baldick, R. (2013). Short-term wind power ensemble prediction based on Gaussian processes and neural networks, *IEEE Transactions on Smart Grid*, 5(1), 501–510, 2013.
- [13] Lee, H.Y.; Tseng, B.H.; Wen, T.H.; Tsao, Y. (2016). Personalizing recurrent-neural-network-based language model by social network, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(3), 519–530, 2016.
- [14] Li, Z.; Han, X.S.; Han, L.; Kang, K. (2010). Ultra-short-term prediction method of wind power in regional power grid, *Automation of Electric Power Systems*, 34(7), 90–94, 2010.
- [15] Liu, S.W. (2016). Study on the influence mechanism of grid connected doubly fed wind turbine on power system transient stability, *North China Electric Power University (Beijing)*, 2016.
- [16] Maragatham, G.; Devi, S. (2019). LSTM model for prediction of heart failure in big data, *Journal of medical systems*, 43(5), 111, 2019.
- [17] Meng, W.L.; Mao, C.Z.; Zhang, J.; Wen, J.; Wu, D.H. (2019). A fast recognition algorithm of online social network images based on deep learning, *Traitement du Signal*, 36(6), 575–580, 2019.
- [18] Mu, G.; Yang, M.; Wang, D.; Yan, G.; Qi, Y. (2016). Spatial dispersion of wind speeds and its influence on the forecasting error of wind power in a wind farm, *Journal of Modern Power Systems and Clean Energy*, 4(2), 265–274, 2016.
- [19] Qian, Y.S.; Shao, J.; Ji, X.X.; Li, X.R.; Mo, C.; Chen, Q.Y. (2019). Short term wind power prediction based on LSTM attention network, *Motor and control application*, 46(9), 95–100, 2016.
- [20] Sun, Y.; Zhang, M.; Chen, S.; Shi, X. (2018). A financial embedded vector model and its applications to time series forecasting, *International Journal of Computers Communications & Control*, 13(5), 881–894, 2018.

- [21] Wang, C.; Zhang, H.L.; Fan, W.H. (2018). Wind power prediction based on projection pursuit principal component analysis and coupling model, *Acta Energiæ Solaris Sinica*, 39(2), 315–323, 2018.
- [22] Wu, X.G.; Su, R.F.; Ji, Y.; Lu, Z.X. (2017). Estimation of error distribution for wind power prediction based on power curves of wind farms, *Power System Technology*, 41(6), 1801–1807, 2017.
- [23] Xue, Y.; Yu, C.; Li, K.; Wen, F.; Ding, Y.; Wu, Q.; Yang, G. (2016). Adaptive ultra-short-term wind power prediction based on risk assessment, *CSEE Journal of Power and Energy Systems*, 2(3), 59–64, 2016.
- [24] Xue, Y.; Wang, L.; Zhang, Y.F.; Zhang, N. (2019). An ultra-short-term wind power forecasting model combined with CNN and GRU networks, *Renewable Energy*, 37(3), 456–462, 2019.
- [25] Yang, M.; Sun, Y.; Sun, Z.J.; Yin, Y.L.; Han, J.F. (2014). Design and development of large-scale data management system of wind farm, *Journal of Northeast Dianli University (Natural Science Edition)*, 34(2), 27–31, 2014.
- [26] Yang, M.S.; Ba, L.; Xu, E.B.; Li, Y.; Gao, X.Q.; Liu, Y.; Li Y. (2019). Batch Optimization in Integrated Scheduling of Machining and Assembly, *International Journal of Simulation Modelling*, 18(4), 689–698, 2019.
- [27] Yao, Q.; Liu, Y.; Bai, K.; Sun, R.F.; Liu, J.Z. (2019). Research on multi index comprehensive evaluation method of wind power prediction level, *Acta Energiæ Solaris Sinica*, 40(2), 333–340, 2019.
- [28] Yu, C.; Xue, Y.C.; Wen, F.S.; Dong, Z.Y.; Wong, K.P.; Li, K (2015). An ultra-short-term wind power prediction method using offline classification and optimization, online model matching based on time series features, *Automation of Electric Power Systems*, 39(8), 5–11, 2015.
- [29] Zhao, Z.H.; Zhang, J.S.; He, P.D.; Yang, K.L.; Wang, C.C. (2019). Wind power prediction based on wide and deep neural network, *Journal of China Academy of Electronics and Information Technology*, 14(3), 307–311, 2019.



Copyright ©2020 by the authors. Licensee Agora University, Oradea, Romania.

This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.

Journal's webpage: <http://univagora.ro/jour/index.php/ijccc/>



This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).

<https://publicationethics.org/members/international-journal-computers-communications-and-control>

Cite this paper as:

Wang, Y.S.; Gao, J.; Xu, Z. W.; Luo, J. D.; Li, L. X. (2020). A prediction model for ultra-short-term output power of wind farms based on deep learning, *International Journal of Computers Communications & Control*, 15(4), 3901, 2020. <https://doi.org/10.15837/ijccc.2020.4.3901>