Future Generation Computer Systems 93 (2019) 33–42

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

LSTM-EFG for wind power forecasting based on sequential correlation features

Ruiguo Yu[a,c,d,e], Jie Gao[a,d,e], Mei Yu[a,c,d,e,*], Wenhuan Lu[b], Tianyi Xu[a,d,e], Mankun Zhao[a,d,e], Jie Zhang[c,d,e], Ruixuan Zhang[a,d,e], Zhuo Zhang[c,d,e]

[a] *School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China*
[b] *School of Computer Software, College of Intelligence and Computing, Tianjin University, Tianjin, China*
[c] *Tianjin International Engineering Institute, Tianjin University, Tianjin, China*
[d] *Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China*
[e] *Tianjin Key Laboratory of Advanced Networking, Tianjin, China*

HIGHLIGHTS

- An improved Long Short-Term Memory network is proposed for wind power forecasting.
- This improved model enhances effect of forget-gate and optimize convergence speed.
- A new wind feature extraction method is proposed to optimize the forecasting effect.
- The results show this method can increase accuracy of 18.3% than those of the others.

ARTICLE INFO

Article history:
Received 15 June 2018
Received in revised form 3 September 2018
Accepted 24 September 2018
Available online 16 October 2018

Keywords:
Wind power forecasting
Deep learning
LSTM
Clustering

ABSTRACT

Amid the gradual increase of wind power generation, how to relieve the pressure of peak load and frequency regulation to the power system by wind power forecasting to make it run steadily becomes a key issue. Due to the continuous development of the field of artificial intelligence, neural network, as a machine learning technology, has shown a good predictive effect in time series data forecasting. Long-term short-term Memory is a kind of time recursive neural network, which is suitable for processing and predicting events with relatively Long intervals and delays in time series. This paper proposes an improved Long Short-Term Memory-enhanced forget-gate network model, abbreviated as LSTM-EFG, used to forecasting wind power. Based on the correlation, the features data of turbine groups in certain distance are filtered to further optimize the forecasting effect on wind power by clustering. The results show that the method with Spectral Clustering has an higher accuracy with an increase of 18.3% than those of the other forecasting models, and at the same time the convergence process has been sped up.

1. Introduction

Due to the increasingly serious environmental pollution, the development technology of new energy sources (such as wind energy, solar energy, nuclear energy, etc.) is gradually developing and becoming more and more mature. The development of renewable energy can effectively reduce greenhouse gas emissions and alleviate environmental pollution. Fig. 1 shows the total amount of renewable energy installed in the first seven countries between 2004 and 2014 [1]. Wind energy, as a new type of clean and renewable green power, has been widely developed and utilized in the world, and has dominated long-term energy plans in some countries [2]. According to the latest report provided by Global WIND ENERGY COUNCIL [3], more than 52 GW of clean, emissions-free wind power was added in 2017, bringing total installations to 539 GW globally. However, wind power have provided people with clean energy and have brought severe challenges to the safe and stable operation of the power system at the same time, resulting from its volatility and intermittency. Precise wind power forecasting can relieve the pressure of peak load and frequency regulation to the power system, which is of great significance to the integration of wind power into the power grid. At the same time, it can also solve such problems as unit commitment (UC) and economic dispatch (ED) [4], dynamic balanced reserve [5], and energy storage optimization [6].

At present, extensive and in-depth researches on wind power forecasting have been conducted in the related fields and have

* Corresponding author at: School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China.

E-mail addresses: rgyu@tju.edu.cn (R. Yu), gaojie@tju.edu.cn (J. Gao), yumei@tju.edu.cn (M. Yu), wenhuan@tju.edu.cn (W. Lu), tianyi.xu@tju.edu.cn (T. Xu), zmk@tju.edu.cn (M. Zhao), tjuzhangj@tju.edu.cn (J. Zhang), zrx_6566@tju.edu.cn (R. Zhang), 1812755792@qq.com (Z. Zhang).

https://doi.org/10.1016/j.future.2018.09.054
0167-739X/© 2018 Elsevier B.V. All rights reserved.

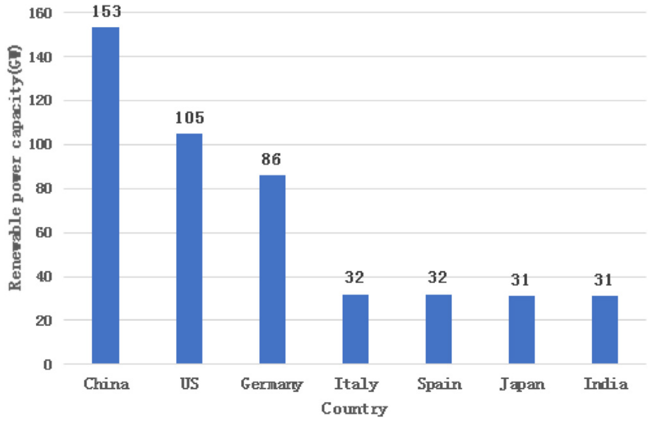


Fig. 1. The total amount of renewable energy installed between 2004 and 2014.

Table 1
Classification of wind forecasting.

Time horizon	Range	Applications
Short-term	≤ 6 h	• Electricity market clearing • Regulation actions economic load dispatch planning • Load increment/Decrement decisions
Long-term	≥ 72 h	• Unit commitment decisions • Reserve requirement decisions • Maintenance scheduling to obtain optimal operating cost

achieved remarkable progress [7–18]. Existing methods can be divided into three categories: (1) physical methods; (2) statistical methods; (3) machine learning methods. The physical methods are based on numerical weather prediction (NWP) [19] and the information of topography and geomorphology around the wind farm. After the establishment of hydrodynamics and thermodynamic models, the wind speed and force can be calculated and then mapped to the final output power through the power function [20]. This method can reflect the situation of atmospheric motion without a large amount of historical data, which is suitable for forecasting electric power before establishing a wind power plant.

Compared with the physical method, the other two methods need a large amount of wind turbine historical data collected to build forecasting models, and can be divided into two categories according to whether NWP is used as input. The forecasting model using NWP as input is a long-term model with a forecasting range of up to 72 h or more. And the forecasting model only based on historical data can achieve a forecasting range of 6 h, which is known as a short-term model. The specific classification and application scenarios are shown in Table 1. The statistical methods aim to directly describe the nonlinear relationship between wind speed and electric power by analyzing the statistical laws of wind speed [19]. The commonly used statistical methods are based on time series models [7]. Regression moving average (ARMA) model [8] is the most popular type of time-serialization-based methods for predicting future wind speed or power value. Furthermore, autoregressive integral moving average (ARIMA) [9] and kalman filter method [10] et al. are also commonly used. Whats more, with the gradual development of the field of artificial intelligence, the application of machine learning becomes more and more perfect. Machine learning methods are designed to use artificial intelligence algorithms that implicitly describe nonlinear and highly complex relationships between input data (wind-force, wind speed, etc.) and output data (electric power), with the existing methods such as SVR, KNN, RNN, LSTM [11–18] and so on.

This paper proposes an optimized LSTM model, namely, Long Short-Term Memory-enhanced forget-gate (LSTM-EFG). This model enhances the effect of forget-gate and changes the activation function to optimize convergence speed. In addition, a new wind power feature extraction method is proposed in this paper. According to the correlation between sequential features of turbines, the "adjacent" turbines are filtered for target turbine through clustering. And related features are extracted from their sequential data called sequential correlation features.

The structure of this paper is as follows: Section 2 introduces the work of wind power forecasting. Section 3 introduces the principle of circulatory neural network and the improvement method of LSTM model proposed in this paper. Section 4 introduces the method of extracting time series feature. Section 5 is the experimental part of this paper, introduces the data source and makes comparative analysis on different experimental results. Section 6 is the conclusion.

2. Related work

As the development of neural networks, LSTM network has effectively avoided the problem of the gradient vanishment in the conventional RNN training process due to its own special structure design. Hui Liu et al. [12] and Jie Chen et al. [13] used LSTM to predict wind speed. Among them, Hui Liu et al. proposed a novel wind speed multistep prediction model by combining Variational Mode Decomposition (VMD), Singular Spectrum Analysis (SSA), LSTM network and Extreme Learning Machine (ELM), in which, the LSTM network is used to complete the forecasting for the low-frequency sub-layers obtained by the VMD-SSA; Jie Chen et al. proposed a novel method called Ensem LSTM based on LSTM, Support Vector Regression Machine (SVRM) and extreme optimization algorithm (EO) used ensemble learning to make relevant predictions. In the field of wind power forecasting. Erick et al. [14] proposed an architecture using LSTM blocks instead of the hidden units in the Echo State Network and used a quantile regression in order to obtain a robust estimate of the expected target, then compared them with the result provided by the Wind Power Forecasting System developed by the Danish Technical University. Qu Xiaoyun et al. [15] used numerical weather prediction (NWP) data and reduced the dimension of input variables of LSTM forecasting model based on principal component analysis and compared it with BP neural network and support vector machine (SVM) model, which indicated that LSTM forecasting model would be endowed with a higher forecasting accuracy and greater engineering application potential. Yao Cheng et al. [16] proposed Power LSTM, a power demand forecasting model based on Long Short-Term Memory neural network and calculate the feature significance and compact our model by capturing the features with the most important weights. Zhu Qiaomu et al. [17] proposed a wind power forecasting method based on LSTM. Firstly, the distance analysis method is used to screen variables with high correlation degree with wind power, so as to reduce the scale and complexity of data. Then, the LSTM network is used to model the dynamic time of multi-variable time series, and finally the prediction of wind power is realized. At present, the models used in existing methods are based on the standard LSTM network structure [18]. Considering that forget-gate and output activation are the most critical components of the LSTM model, two peepholes are added to the standard LSTM forgotten gates and output layer gates. Furthermore, in order to make full use of forget-gate, we use all-one matrix to process the output of forget-gate and take the processed results as input values

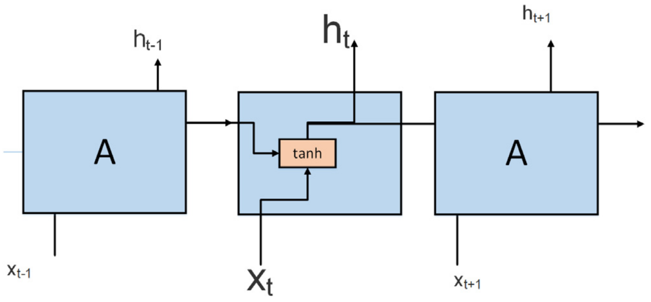


Fig. 2. Duplicate modules in standard RNN contain a single layer.

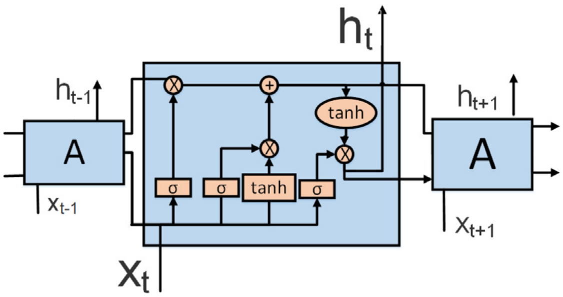


Fig. 3. The repeating module in the LSTM contains four interacting layers.

for data updates. Thanks to its smoother and slower decreasing rate which can be conducive to a more efficient learning, the softsign function is used in this paper instead of the tanh function. In addition, the existing feature selection methods take all the turbines within a certain distance into consideration, which not only has a large computational cost and a slow speed, but also causes an increase in the forecasting errors. This paper proposes a correlation-based feature data filtering method that filters the most similar turbine feature data to the target turbine by clustering sequential correlation features, and then realizes a more accurate and more efficient forecasting of wind power.

3. LSTM-EFG

In this paper, we have improved the structure of traditional LSTM and proposed a new structure model, namely, LSTM-EFG, which could enhance the effect of forget-gate and improve the convergence speed of the network.

3.1. LSTM model

RNN is the general term of two artificial neural networks: recurrent neural network and recursive neural network. The inter-neuronal connections of time-recurrent neural networks form a matrix, while structural recurrent neural networks recursively construct more complex deep networks using similar neural network structures. RNN generally refers to time recurrent neural networks. Time-recurrent neural networks can describe dynamic time behavior and unlike the feedforward neural network accepting input from a particular structure, the RNN cyclically passes states in its own network, thus accepting a wider range of time series structure inputs.

LSTM (Long Short-Term Memory) is a kind of time recursive neural network and a special kind of RNN that can learn to rely on information for a long time. Besides, it is suitable for processing and predicting important events with relatively long intervals and delays in time series. LSTM was proposed by Hochreiter and Schmidhuber [21] and recently improved and popularized by Alex Graves. On many issues, LSTM has achieved considerable success and has been widely used. In this experiment, two peepholes are added in the structures to predict wind energy more accurately.

The LSTM is an improvement on RNN, it contains a processor that determines whether information is useful or not, in which the working part is named as a cell. There are three doors in a cell: the input layer door, the forget-gate, and the output layer door. Input layer door and forget-gate both work on the state of cells. But the role of the input gate is to selectively record new information into the cell state, while forget-gate is aimed at selectively forgetting information about cell states. The output layer gate acts on the hidden layer to output information.

RNN is a network that contains cycles and has a form of chain of repetitive neural network modules. In a standard RNN, the repeating module has a very simple structure, such as a tanh layer, as shown in Fig. 2. Where, X_t is the input information, h_t represents output information, and A is the neural network module.

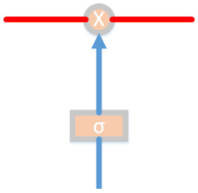


Fig. 4. LSTM "door" structure.

RNN can be used to connect the previous information to the current task, but as the position interval increases, the learning ability of RNN decreases. In order to avoid long-term dependence problems, LSTM network appears. LSTM has the same chain structure as RNN, but repeated modules have a different structure. Unlike the single neural network layer, there are four, interacting in a very special way, as shown in Fig. 3.

In Fig. 3, each black line transmits an entire vector from the output of one node to the input of another node. The circles represent the operations of pointwise, such as the sum of vectors, while the rectangles and ellipse are the neural network layer learned. The lines that come together represent the connection of the vector, and the lines that come apart represent the content being copied and distributed to different locations.

The key to LSTM is the cell state, with horizontal lines running across the top of the graph. The red line at the top of Fig. 6. The cell state is similar to a conveyor belt, which runs directly over the entire chain with only a few linear interactions. It is easy for information to circulate on it and stay the same.

LSTM removes or adds information to the cell state through the structure of the gate. A door is a way of allowing information to pass selectively. They include a sigmoid layer and a pointwise multiplication operation, as shown in Fig. 4.

There are three gates in the LSTM shown in Fig. 3 to protect and control cell state.

The LSTM has a variant that also allows the door layer to receive input from the cell state by adding three "peephole connections". The structural model is shown in Fig. 5.

3.2. LSTM-EFG model

In this paper, LSTM is improved in the following four aspects: (1) adding two peepholes (f-o), (2) changing the activation function tanh into softsign, (3) deleting the input-gate, (4) subtracting the value previously output by the forget-gate in the way of the all-1 matrix, and then turning to update. The improved model can maximize the effect of the forget-gate and increase the convergence

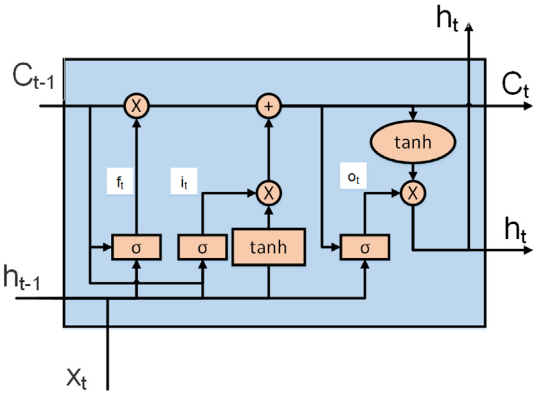


Fig. 5. LSTM variant.

rate of the algorithm. And then LSTM-EFG model is used for wind power forecasting which is proved to have the best performance according to the experimental results. The model structure used in this paper is illustrated in Fig. 6. The entire network includes an input layer, a hidden layer, and an output layer, which are fully connected (blue arrow) to each other. In the input layer, the input (i.e., each white square) dimension is the number of features that enter the hidden layer. The horizontally schematic portion in Fig. 6(b) represents five LSTM-EFG layers included in the hidden layer, and each LSTM-EFG layer is represented by a longitudinal schematic portion in Fig. 6(b). The red arrow represents C, H is consistent with Fig. 6(a). The black arrow represents the H output of the previous LSTM-EFG layer and serves as the input for the next LSTM-EFG layer.

3.3. The calculation steps of LSTM-EFG model

The calculation steps of LSTM-EFG model used in this paper are as follows.

1. Determine which information is discarded from the cell state. This decision is made through the forget-gate, which reads the cell states $C_{(t-1)}$, $h_{(t-1)}$ and x_t, and outputs *a* value between 0 and 1 to the cell state $C_{(t-1)}$. The 1 means "complete reservation" and 0 means "completely abandoned". This section is shown in Fig. 6 with blue lines and the equation of f_t is shown in (1).

$$f_t = sigmoid(f + W_f * C_{t-1}). \tag{1}$$

2. Determine which information is stored in the cell state. This process could be divided into two parts. First, there is a full 1 matrix that subtracts the output value of the previous forget-gate and this determines what value we will update accordingly. Then, the activation function softsign will create a new candidate values vector C_t^1, the candidate values will be added to the state. This section is shown in Fig. 6 with black lines and the equation of C_t^1 is shown in (2).

$$C_t^1 = softsign(W_c * [h_{t-1}, x_t] + b_c). \tag{2}$$

3. Update the old cell status $C_{(t-1)}$ to C_t. We multiply the old state $C_{(t-1)}$ with f_t to discard the information we need to discard. Then we add $(1-f_t)*C_t^1$ and get a new cell state C_t. The equation of C_t is shown in (3).

$$C_t = f_t * C_{t-1} + (1 - f_t) * C_t^1. \tag{3}$$

4. Determine the output value. This output will be based on our cellular state, but it is also a filtered version. First, we run a sigmoid layer to determine which part of the cell's state will be output. Then, we process the cell state through softsign (get a value between −1 to 1) and multiply it with the output value of sigmoid. And finally, we will only output the part which we need.

This section is shown in Fig. 6 with yellow lines. The equation of o_t is shown in (4). The final expression of output h_t is shown in (5).

$$o_t = sigmoid(o + W_o * C_t). \tag{4}$$

$$h_t = o_t * softsign(C_t). \tag{5}$$

4. Sequential correlation features extraction

For wind power forecasting task of this article, if all the turbines data of the wind farm is used to predict the output power of the target turbine, not only the parameters are large, but also the operation speed is slow, and most of the turbines that are too far away from target turbine do not help the forecasting. Therefore, a feature data filtering method based on time series correlation is proposed in this paper. Firstly, find out all the neighboring turbines within a certain distance around the target turbine. And then use different methods to filter out the sequential correlation features based on their correlation with the target turbine. Finally, the target turbine feature and the sequential correlation features are used together as the input data of the LSTM-EFG, so as to forecast the wind power output value of the target turbine after 90 min, as shown in Fig. 7. The time interval in our experiment is 30 min and the flow chart is shown in Fig. 8.

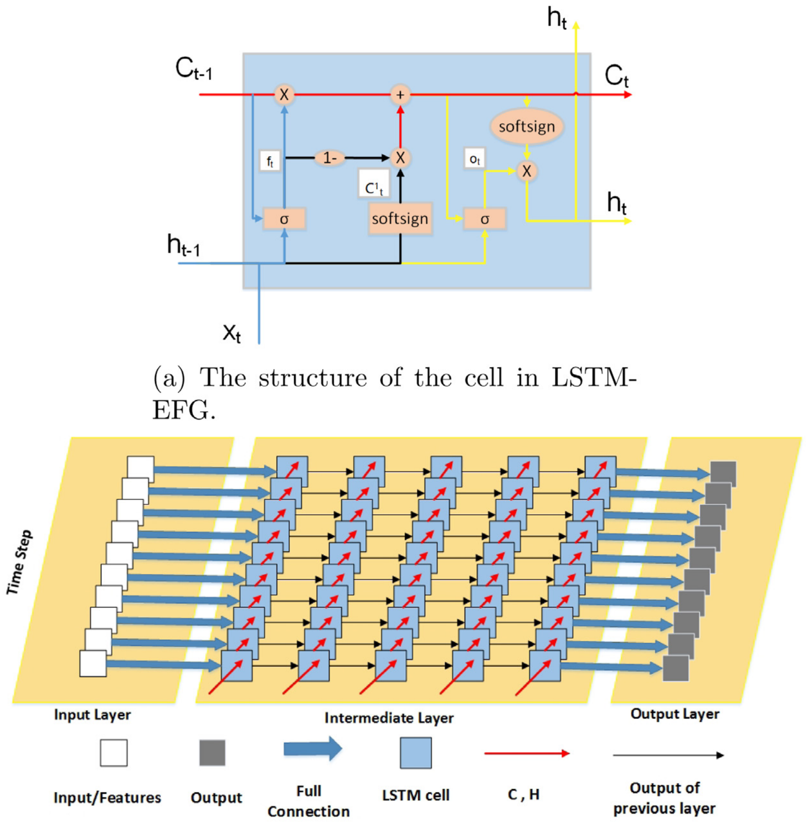
In this paper, there are many methods for feature screening. For example, euclidean distance, K-Means, Spectral Clustering, Agglomerative Clustering and Birch. Among them, based on the feature selection method of the euclidean distance, the neighboring turbines in a certain range are selected according to the euclidean distance between the time series based on the target turbine and the target turbine. Other methods are clustering methods, a typical unsupervised learning algorithm, which is mainly used to automatically group similar samples into one category. In the clustering algorithm, the samples are divided into different categories according to the similarities between the samples. Different clustering results are obtained for different similarity calculation methods.

4.1. K-means

The K-Means algorithm was originally proposed by J. Macqueen in 1967 [22]. The idea of the algorithm is very simple. Predetermine the constant K. The constant K means the number of the final clusters. First randomly select the starting point as the centroid and calculate the similarity between each sample and the centroid (i.e. Euclidean distance), and sort sample points into the most similar class. Then, recalculate the centroid of each class (i.e. the class center) and repeat the process, knowing that the centroid no longer changes. Finally, the category to which each sample belongs and the centroid of each class are determined. What k-means does is minimizes the function

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|x^{(i)} - \mu_k\|^2. \tag{6}$$

where r_ik is 1 when data point i is classified into cluster k, otherwise it is 0. $x^{(i)}$ is the ith sample. μ_k is the centroid of the k-th cluster. The algorithm steps are as follows:



(a) The structure of the cell in LSTM-EFG.

(b) The structure of LSTM-EFG.

Fig. 6. LSTM-EFG recurrent neural network structure in this paper.

1. Randomly select k cluster centroids points as $\mu_1, \mu_2, \ldots, \mu_k \in R^n$
2. Repeat the process until convergence {
   For each sample i, calculate what class it should belong to

$$c^{(i)} := \arg\min_{j} \|x^{(i)} - \mu_j\|^2 \tag{7}$$

For each class j, recalculate the centroid of the class

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\}x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}} \tag{8}$$

}

4.2. Spectral clustering

Spectral clustering [23] is an algorithm that evolved from graph theory and was later widely used in clustering. Its main idea is to treat all data as points in space and use edges to connect these points. The value of the edge weight between two points that are farther away is lower, and the value of the edge weight between two points that are closer together is higher. By cutting the graph composed of all data points, the sum of edge weights between different subgraphs is minimized, and the sum of the edge weights within the subgraph is maximized, so as to achieve the purpose of clustering. Graph is expressed as an adjacency matrix in the form of W, where w_{ij} is the weight of node i to node j. If the two nodes are not connected, the weight is zero. Let A and B be two subsets of Graph (without intersection), then the cost function cut between the two can be formally defined as:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{9}$$

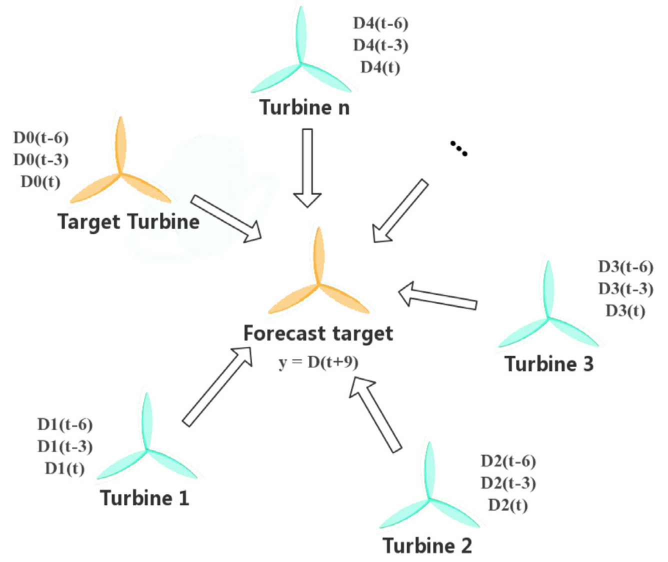


Fig. 7. Combining the data of target turbine with neighbor turbines. If n = 4, we have $(4 + 1) * 3 = 15$ groups of data for forecasting.

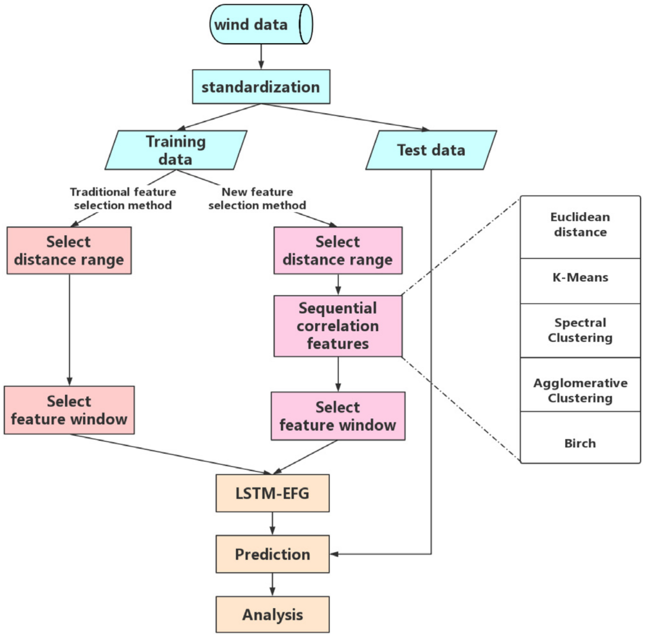


Fig. 8. Flow diagram.

Consider the simplest case first. If you divide a Graph into two parts, then minimum cut is to minimize $cut(A, \bar{A})$ (where $\bar{A}$ denotes the complement of A). However, since this often happens when isolated nodes are split, RatioCut appears:

$$RatioCut(A, B) = \frac{cut(A, \bar{A})}{|\bar{A}|} + \frac{cut(A, \bar{A})}{|A|} \tag{10}$$

and NormalizedCut:

$$NormalizedCut(A, B) = \frac{cut(A, \bar{A})}{vol(\bar{A})} + \frac{cut(A, \bar{A})}{vol(A)} \tag{11}$$

where $|A|$ represents the number of nodes in A, and $vol(A) = \sum_{i \in A} w_{ij}$. Both can be counted as a measure of the size of A. By placing such an item on the denominator, it is possible to effectively prevent the occurrence of outliers and achieve a relatively even split. The algorithm steps are as follows:

1. Construct a Graph from the data. Each node of Graph corresponds to a data point, connecting similar points, and the weights of the edges are used to represent the similarity between the data. The graph is represented in the form of an adjacency matrix, denoted as W.
2. Add each of the elements of W to get N numbers, put them on the diagonal (zero everywhere else), and form an $N \times N$ matrix, denoted as D. And let $L = D - W$.
3. Find the first k eigenvalues of L, $\{\lambda\}_{i=1}^{k}$, and the corresponding eigenvector $\{v\}_{i=1}^{k}$.
4. The k features (column) vectors are arranged together to form an $N \times k$ matrix, each of which is considered as a vector in the k-dimensional space, and clustered using the K-means algorithm. The category to which each row belongs in the clustering result is the category of the original N data points in the original Graph.

4.3. Agglomerative hierarchical clustering

Agglomerative Hierarchical Clustering (AHC), is a kind of Hierarchical Clustering method that from the bottom to top, it can calculate the distance between the different clusters based on the specified similarity or distance. The basic idea of this method is that it considers the individual item as one kind of class and then use different methods to merge them to gradually reduce the number of classes until it comes to the one class or to the required number of classes.

And according to the different definition of similarity (distance), the Agglomerative Clustering method is divided into three kinds: Single-linkage, Complete linkage and Group business.

In Single-linkage, the distance to be compared is the minimum distance between element pairs. In Complete-linkage, the distance to be compared is the maximum distance between element pairs. In Group average, the distance to be compared is the average distance between classes and the definition of average distance is as follows: Suppose there are two classes A and B, and there are n elements in A and m elements in B. You take one element in A and one element in B, and you can get the distance between them. Then you get the sum of the distances by adding up the n ∗ m distances like this. And finally, you divide the sum of the distances by n ∗ m to get the average distance between class A and class B.

The main steps of the algorithm are as follows:

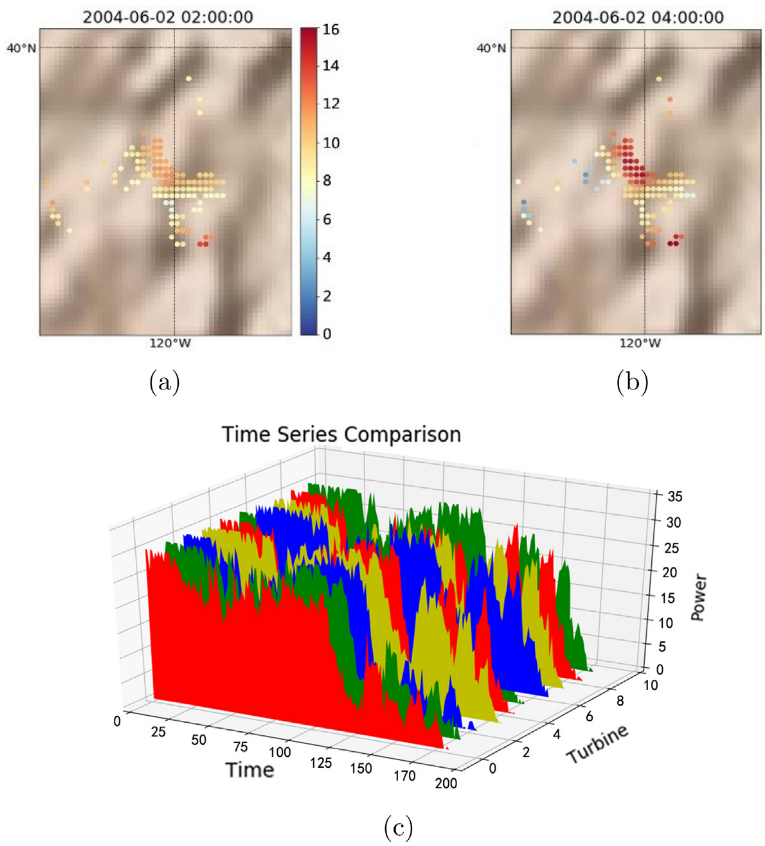
1. Classify each element as a class
2. Repeat: each round merges the smallest class with the specified distance (important to understand the specified distance)
3. Until all elements are grouped into the same category

4.4. BIRCH

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [24] is used to cluster, the large-scale data set and is a very effective, based on distance, integrated hierarchical clustering algorithm. This algorithm can cluster effectively by scanning once and can effectively deal with outliers. It uses the concepts of Clustering Feature (CF) and Clustering Feature Tree (CF Tree) to summarize Clustering description. Clustering feature tree summed up the useful information and take up the smaller space than the metadata set. It can be stored in memory, which can improve the clustering speed and scalability of the algorithm on large data sets.

The main idea of Birch algorithm is to establish a cluster feature tree that is initially stored in memory by scanning the database, and then cluster the leaf nodes of the cluster feature tree. Its core is clustering feature (CF) and clustering feature Tree (CF Tree). CF refers to the ternary group CF = (N, LS, SS), which is used to summarize the sub cluster information rather than store all data points. Where, N is the number of d-dimensional points in the cluster; LS: is the linear sum of N points; SS is the sum of squares at N points. For example, given a set of 2d points (3, 4), (2, 6), (4, 5), then: CF structure summarizes the basic information of the cluster and is highly compressed because it stores the clustering information smaller than the actual data points. Meanwhile, the three-element structure of CF makes it very easy to calculate the radius of cluster, the diameter of cluster and the distance between cluster and cluster.

CF tree is a highly balanced tree with two parameters to store the clustering features of hierarchical clustering. It involves two parameter which are branching factors and thresholds. Where, the branch factor specifies the maximum number of child nodes, that is, the maximum number of children that each non-leaf node can have. The threshold specifies the maximum diameter of the sub cluster stored in the leaf node, which affects the size of the CF tree. Changing the threshold can change the size of the tree. The CF tree is created dynamically with the insertion of data points, so the method is incremental. In fact, the construction of CF tree is a data point insertion process.



(a)

(b)

(c)

Fig. 9. (a, b) Topological structure in reno, which illustrates the power change of two time stamps, (c) Wind sequential map of seven wind turbines in reno . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Experiments and analysis

5.1. Data sources and analysis standard

Our experiment uses data from the National Renewable Energy Laboratory (NREL). The data set covers wind speed and power generation of 32,043 wind turbines with time intervals of 10 min between 2004 and 2006. In order to visualize the correlation between the wind turbines sequential data, Figs. 9(a) and 9(b) shows wind changes of reno two hours later (red is strong, blue is weak). Fig. 9(c) shows a comparison of sequential power of ten wind turbines around reno.

It can be seen that there is a strong correlation between the peak value of the turbines near target turbine and the sequential variation of the overall trend. Moreover, Mean squared error (MSE) is used for model assessment and analysis.

5.2. Forecasting by sequential correlation features based on Euclidean distance

First, the results of feature screening based on the Euclidean distance between the turbine's time series feature data are shown in Table 2. As shown. Among them, Num represents the number of turbines closest to the target turbine selected based on the Euclidean distance (including the target turbine itself). As can be seen from the table, when 15 turbines were selected based on the Euclidean distance for prediction, the error MSE of the five different wind farms was the smallest. On the contrary, when Num = 5, the prediction effect is relatively poor.

Table 2
MSE of forecasting by Euclidean distance.

Wind farm	Num	MSE
Tehachapi	5	7.0689
	10	6.7032
	15	**6.5642**
Cheyenne	5	7.4852
	10	7.2990
	15	**7.1479**
Palmsprings	5	8.5178
	10	5.8669
	15	**5.5451**
Lasvegas	5	10.3877
	10	9.7076
	15	**9.4652**
Lancaster	5	8.5072
	10	7.9219
	15	**7.6354**

5.3. Forecasting by sequential correlation features based on K-means

In this experiment, K-Means algorithm is used to cluster the timing characteristics of wind turbines, and the motor electrical characteristics data of the cluster where the target motor is located is used to predict the wind power of the target motor. The experimental results are shown in Table 3. As shown. Among them, K represents the classification number of the clustering algorithm; Num represents the number of motors of the cluster to which the target motor belongs (including the target motor itself). It can be seen from the table that, except for Lasvegas, the prediction effect is best when K = 2, and the prediction effect of other four wind farms at K = 3 is relatively good.

Table 3
MSE of forecasting by K-means.

Wind farm	K	Num	MSE
Tehachapi	2	49	**6.5630**
	3	36	6.6500
Cheyenne	2	78	**6.9178**
	3	56	6.9557
Palmsprings	2	30	5.8576
	3	12	**5.7974**
Lasvegas	2	31	**9.3605**
	3	21	9.5860
Lancaster	2	34	**7.5228**
	3	29	7.9775

Table 4
MSE of forecasting by spectral clustering.

Wind farm	K	Num	MSE
Tehachapi	2	22	**6.6290**
	3	24	6.7024
Cheyenne	2	105	**6.9661**
	3	84	7.1115
Palmsprings	2	25	5.6162
	3	18	**5.5311**
Lasvegas	2	31	9.4486
	3	23	**9.2824**
Lancaster	2	29	**7.3950**
	3	8	8.0539

Table 5
MSE of forecasting by agglomerative clustering.

Wind farm	K	Num	MSE
Tehachapi	2	44	6.6160
	3	44	**6.5325**
Cheyenne	2	98	**6.9061**
	3	45	7.6662
Palmsprings	2	30	5.8259
	3	9	**5.7726**
Lasvegas	2	27	**9.5711**
	3	27	9.6435
Lancaster	2	36	7.5007
	3	36	**7.4352**

Table 6
MSE of forecasting by Birch.

Wind farm	K	Num	MSE
Tehachapi	2	44	**6.3648**
	3	44	6.4977
Cheyenne	2	98	**6.9896**
	3	45	7.8200
Palmsprings	2	30	5.9393
	3	9	**5.7240**
Lasvegas	2	27	9.8086
	3	27	**9.7298**
Lancaster	2	36	7.5778
	3	36	**7.5020**

5.4. Forecasting by sequential correlation features based on spectral clustering

In this experiment, we use the Spectral Clustering algorithm to cluster the timing characteristics of the wind turbine, and use the motor characteristic data in the cluster where the target motor is located to predict the wind power of the target motor. The experimental results are shown in Table 4. As shown. among them. As can be seen from the table, there are 3 wind farms that have achieved good predictions at K = 2, and Palmsprings and Lasvegas have better predictions at K = 3.

5.5. Forecasting by sequential correlation features based on agglomerative clustering

In this experiment, we use the Agglomerative Clustering algorithm to cluster the timing characteristics of the wind turbines and use the motor characteristic data in the cluster where the target motor is located to predict the wind power of the target motor. The experimental results are shown in Table 5. As can be seen from Table 5, there are 3 wind farms that have achieved good predictions at K = 3, and Cheyenne and Lasvegas have better predictions at K = 2. The best predictor is Palmsprings, which has a MSE value of 5.7726.

5.6. Forecasting by sequential correlation features based on Birch

In this experiment, we use the Birch algorithm to cluster the timing characteristics of the wind turbines and use the motor characteristic data in the cluster where the target motor is located to predict the wind power of the target motor. The experimental results are shown in Table 6. As can be seen from Table 6, there are 3 wind farms that have achieved good predictions at K = 3, and Cheyenne and Tehachapi have better predictions at K = 2. The best predictor is Palmsprings, which has a MSE value of 5.7240.

5.7. Analysis of predicted performance between LSTM-EFG and others

Five different turbines (tehachapi, cheyenne, palmsprings lasvegas, lancaster) are selected in the experiment to predict the wind power output function. And we compare and analyze the forecasting results of five different methods: SVR, KNN, LSTM, LSTM-EFG with different feature windows and LSTM-EFG based on timing correlation features and cluster methods. The experimental results are shown in Table 7. In Table 7, the best results are shown in bold, the second best is underlined.

On the analysis of Table 7, the MSE values of the forecasting results using LSTM-EFG model alone are better than KNN algorithm. The MSE values of LSTM-EFG model based on timing correlation features are better than SVR algorithm. In addition, the effect will not be worse than that of the traditional LSTM method under this condition and would be much better in most cases. The predicted results of LSTM-EFG model based on timing correlation features are superior to those obtained by LSTM-EFG model. Spectral Clustering uses the similarity matrix of the sample data to perform feature decomposition to obtain the eigenvectors for clustering. It can be seen that it is independent of the sample feature and only related to the number of samples, and has high computational efficiency. therefore, the LSTM-EFG model using Spectral Clustering algorithms for the features get the best results in three locations, and get the second best results in another site. Taken together, this method has the best prediction effect, the optimal MSE value is 5.5311.

These results suggest that the way of feature selection based on timing correlation and the improved LSTM structure in the paper are critical to advance the predicting performance. And among the results, the forecasting accuracy of turbines in palmsprings increased most significantly, which is 18.30% higher than SVR, 16.84% higher than KNN and 13.10% higher than LSTM.

Table 7
Predicted performance of LSTM-EFG and others.

		Tehachapi	Cheyenne	Palm springs	Las Vegas	Lancaster
SVR		6.9915	7.2196	6.7699	9.8445	8.1765
KNN(k=25)		7.7479	7.6173	6.6512	10.861	8.5924
LSTM		6.9322	7.3676	6.3650	9.8769	8.1031
LSTM-EFG	Feature window=3	6.7884	7.3002	6.3413	9.5638	7.9193
	Feature window=4	7.1005	7.5017	6.5049	9.8395	8.0105
	Feature window=5	6.7683	7.3646	6.4134	10.0206	7.7004
	ED	6.5642	7.1479	5.5451	9.4652	7.6354
	K-Means	6.5630	6.9178	5.7974	9.3605	7.5228
	Spectral Clustering	6.6290	6.9661	**5.5311**	**9.2824**	**7.3950**
	Agglomerative Clustering	6.5325	**6.9061**	5.7726	9.5711	7.4352
	Brich	**6.3648**	6.9896	5.7240	9.7298	7.5020

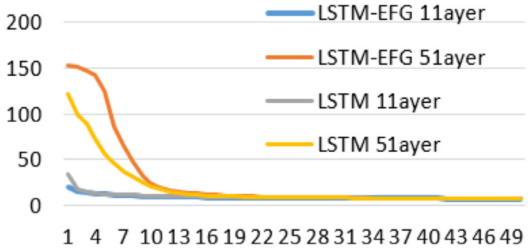


Fig. 10. Convergence speed of LSTM-EFG and LSTM.

5.8. Analysis of convergence speed between LSTM-EFG and LSTM

In this part, we compared and analyzed the convergence rates of LSTM-EFG and LSTM when they are respectively 1 layer and 5 layers. The results are shown in Fig. 10.

From the Fig. 10, we can see that the convergence speed of LSTM-EFG model is faster than LSTM model because the LSTM-EFG model tends to be stable after the 10th round of training, while the LSTM model tends to be stable after the 13th round of training. And After the 13th round of training the convergence speed of two models are similar to each other until the 50th round of training. So, LSTM-EFG model has a higher convergence speed, which is even faster than traditional LSTM at some time.

6. Conclusion

In this paper, we propose an improved LSTM-EFG model based on LSTM, which adds two peepholes (f-o), replaces activation function tanh with softsign, at the same time, removes input-gate in the traditional LSTM, subtracts the output of the forget-gate by the full 1 matrix and finally uses the result as the input value of the data update. The improved model LSTM-EFG enhances the effect of forget-gate and accelerates the convergence process. Meanwhile, this paper also put forward a kind of temporal dependencies feature extraction method combined with the cluster methods, which select the most similar characteristics data with target turbine within a certain distance based on the temporal correlation and clustering methods.

The experimental results show that the MSE value obtained by LSTM-EFG is lower than that of the existing methods (LSTM, SVR and KNN), which indicates that our method has a better forecasting performance. Besides, by using Spectral Clustering method to get the temporal correlation characteristics can make prediction the most effect. At the same time, the LSTM-EFG model used in this paper is also faster than the traditional LSTM. The method and model proposed in this paper can better predict the power of wind power at a certain time, so as to relieve the pressure of peak and frequency regulation of power system and make full use of wind energy.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61877043, No. 61877044 and No. 91746205), and Key Project for Science and Technology Support from Key R&D Program of Tianjin (Grant No. 18YFZCGX00960).

References

[1] REN, Renewables 2015 - global status report, Environ. Policy Collect. (2015).
[2] J. Hossain, World wind resource assessment report, 2014.
[3] G.W.E. Council, Global wind report, 2018.
[4] A. Botterud, Z. Zhou, J. Wang, J. Sumaili, H. Keko, J. Mendes, R.J. Bessa, V. Miranda, Demand dispatch and probabilistic wind power forecasting in unit commitment and economic dispatch: a case study of illinois, IEEE Trans. Sustainable Energy 4 (1) (2013) 250–261.
[5] N. Menemenlis, M. Huneault, A. Robitaille, Computation of dynamic operating balancing reserve for wind power integration for the time-horizon 1c48 hours, IEEE Trans. Sustainable Energy 3 (4) (2012) 692–702.
[6] H. Bludszuweit, J.A. Dominguez-Navarro, A probabilistic method for energy storage sizing based on wind power forecast uncertainty, IEEE Trans. Power Syst. 26 (3) (2011) 1651–1658.
[7] R. Billinton, H. Chen, R. Ghajar, A sequential simulation technique for adequacy evaluation of generating systems including wind energy, IEEE Trans. Energy Convers. 11 (4) (1996) 728–734.
[8] S. Rajagopalan, S. Santoso, Wind power forecasting and error analysis using the autoregressive moving average modeling, Power Energy Soc. Gen. Meet. .pes.ieee (2009) 1–6.
[9] K. Yunus, T. Thiringer, P. Chen, ARIMA-Based frequency-decomposed modeling of wind speed time series, IEEE Transactions on Power Systems 31 (4) (2016) 2546–2556.
[10] E.A. Bossanyi, Short-term wind prediction using kalman filters, Wind Eng. 9 (1985).
[11] N.A. Treiber, J. Heinermann, O. Kramer, Wind power prediction with machine learning, 2016.
[12] H. Liu, X. Mi, Y. Li, 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. 159 (2018) 54–64.
[13] J. Chen, G. Zeng, W. Zhou, W. Du, K. Lu, Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization, Energy Convers. Manage. 165 (2018) 681–695.
[14] E. Lpez, C. Valle, H. Allende, E. Gil, H. Madsen, Wind power forecasting based on echo state networks and long short-term memory, Energies 11 (3) (2018) 526.
[15] X. Qu, X. Kang, C. Zhang, S. Jiang, X. Ma, Short-term prediction of wind power based on deep long short-term memory, in: Power and Energy Engineering Conference, 2016, pp. 1148–1152.
[16] Y. Cheng, C. Xu, D. Mashima, V.L.L. Thing, Y. Wu, PowerLSTM: Power demand forecasting using long short-term memory neural network, 2017.
[17] Z. Qiaomou, L. Hongyi, W. Ziqi, C. Jinfu, W. Bo, Short-Term wind power forecasting based on LSTM, Power Syst. Technol. (2017).
[18] F.A. Gers, J. Schmidhuber, Recurrent nets that time and count, in: Ieee-Inns-Enns International Joint Conference on Neural Networks, 2000, p. 3189.
[19] B. Ernst, B. Oakleaf, M.L. Ahlstrom, M. Lange, Predicting the wind, Power Energy Mag. IEEE 5 (6) (2007) 78–89.
[20] S. Li, D.C. Wunsch, E.A. O'Hair, M.G. Giesselmann, Using neural networks to estimate wind turbine power generation, in: Power Engineering Society Winter Meeting, 2001, p. 977 vol.3.
[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[22] J. Macqueen, Some methods for classification and analysis of multivariate observations, in: Proc. of Berkeley Symposium on Mathematical Statistics and Probability, 1966, pp. 281–297.
[23] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Math. J. 23 (23) (1973) 298–305.
[24] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: A New Data Clustering Algorithm and Its Applications, Kluwer Academic Publishers, 1997.

Ruiguo Yu received the bachelor's degree in computer software from Tianjin University, China. In addition, he received the M.S. degree and the Ph.D. degree in computer application technology from Tianjin University, China. Now he works as an associate professor in School of Computer Science and Technology in Tianjin University. His current research interests include recommended algorithm research and application, text feature extraction and clustering, network information retrieval and public opinion monitoring. He has participated in a number of projects including Natural Fund projects.

Jie Gao received the bachelor's degree from School of Ren'ai in Tianjin University, China, in 2013, and the master's degree in School of Computer Science and Technology from Tianjin University, China, in 2016. From 2016, she works an Assistant Engineer in School of Computer Science and Technology in Tianjin University. Her research interests are Network and Data Mining.

Mei Yu received Ph.D. degree in computer application technology from Tianjin University. She is currently a professor engaged in computer networks, data mining, database in Tianjin University. As the coach of the Tianjin University ACM-ICPC team, she led the team winning a number of awards in the Asian Regional Contest of ACM International Collegiate Programming Contest, and obtaining the world finals twice. She serves as the instructor of Tianjin University IT discipline innovation and entrepreneurship training base, responsible for the base construction.

Wenhuan Lu is an associate professor at School of Computer Software in Tianjin University. She received her Master Degree at Tianjin University, P.R. China, and Ph.D. Degree at Japan Advanced Institute of Science and Technology, Japan, in 2004 and 2007, respectively. She has worked as postdoctoral researcher at JAIST. Afterwards, she joined School of Computer Software, Tianjin University, P.R. China in 2010.

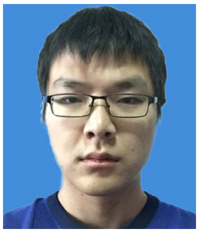
Dr. Lu is working in knowledge modeling with semantic technology applied to knowledge-based system. She also work on audio security and AI in data science in recent years.

Tianyi Xu received the bachelor's degree from School of Electronic Engineering and Automation in Tianjin University, China, in 2012, and the master's degree in School of Computer Science and Technology from Tianjin University, China, in 2015. From 2015, he works as an Assistant Engineer in School of Computer Science and Technology in Tianjin University. His research interests are Network and Data Mining.

Zhao Mankun, master's degree, currently working at the computer experimental teaching center of Tianjin University, mainly engaged in machine learning, network information retrieval and computer image processing, etc. His current research interests include recommended algorithm research and application, text feature extraction and clustering, network information retrieval and public opinion monitoring.

Zhang Jie, a graduate student in computer science at the school of Tianjin International Engineering Institute, Tianjin University. At present, she mainly work on block chain technology and data mining technology.

Zhang Ruixuan, a master's student at Tianjin University, received the bachelor's degree from Northeast Normal University in 2017. At present, she is mainly engaged in machine learning, and computer image processing, which is mainly about medical image recognition, classification and image target detection.



Zhuo Zhang, graduated from the school of computer science and technology of Tianjin University, and now is studying for a master' degree in Tianjin International Engineering Institute of Tianjin University. When he was a undergraduate student, he got some experience of programming contest, and his main research area was in images and graphics. And now he is in a laboratory for studying the deep learning.