Future Generation Computer Systems 113 (2020) 145–157

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Superposition Graph Neural Network for offshore wind power prediction

Mei Yu [a,c,d], Zhuo Zhang [b,c,d], Xuewei Li [a,c,d,*], Jian Yu [a,c,d], Jie Gao [a,c,d], Zhiqiang Liu [a,c,d], Bo You [b,c,d], Xiaoshan Zheng [b,c,d], Ruiguo Yu [a,c,d,*]

[a] College of Intelligence and Computing, Tianjin University, China
[b] Tianjin International Engineering Institute, Tianjin University, China
[c] Tianjin Key Laboratory of Cognitive Computing and Application, China
[d] Tianjin Key Laboratory of Advanced Networking, China

ARTICLE INFO

Article history:
Received 18 November 2019
Received in revised form 2 June 2020
Accepted 17 June 2020
Available online 20 June 2020

Keywords:
Wind power
Prediction
Graph neural network

ABSTRACT

Wind power prediction plays an important role in its utilization. Currently, in machine learning methods and other traditional methods, the prediction is always based on the time series of data nodes, and sometimes wind turbines near the predicted nodes are also applied. These methods have limitations in the utilization of the spatial features of the entire wind farm, and can only be used to predict a single wind turbine. Offshore wind farm data is more difficult to predict due to the more dispersed distribution of wind turbines and the intermittent nature of offshore winds. We proposed a data integration method, which can connect all wind turbines in a certain range of wind farms by their geographical locations and other related information to form a graph(one type of data structure), then superimpose these graphs in a certain period of time. Then, we proposed the SGNN(Superposition Graph Neural Network) for feature extraction, which can maximize the use of spatial and temporal features for prediction. In the four offshore wind farms used in experiments, the mean square error (MSE) of the method is reduced by 9.80% to 22.53% compared with current-advanced methods, and the prediction stability of the method has also been greatly improved.

1. Introduction

At present, the rapid economic growth has led to high energy consumption and a significant increase in greenhouse gas emissions. Due to the large reserves of renewable energy and its pollution-free nature, more and more countries have turned to renewable energy in industry in the past few decades. Compared with traditional energy sources such as gas and coal combustion, wind power is endowed with incomparable advantages in environmental protection. Wind energy is essentially a conversion of solar energy. It is rich in reserves and is one of the best alternatives to fossil energy. With the advancement of wind power technology, Wind power has become a renewable energy that can be exploited and utilized on a large scale [1]. Global wind power capacity reached 539,291MW by the end of 2017, and this capacity is 51,402MW more than the capacity in 2016. Fig. 1 shows the global wind power capacity and annual additions from 2007 to 2017.

Compared with onshore wind power, offshore wind power is more productive and has a rapid development [2]. The terrain of the land is fluctuating, which has a slowing effect on the wind speed, while the sea level is stable and the wind resistance is low. The offshore wind speed is about 20% higher than that on land, and the output power of wind power is proportional to the cube of the wind speed. Therefore, the offshore wind power generation under the same power generation capacity is about 70% higher than that on land. The frequency of change in sea wind direction is also lower, and offshore wind turbines do not need to consider land occupation problems and noise pollution. For coastal areas, the use of offshore wind power can avoid a waste caused by long-distance transmission. However, due to the influence of wind speed and direction, the randomness and fluctuation of wind turbines are inevitable, and the impact on the power grid is large, which brings severe challenges to the security and stability of power system operation. The nature of offshore wind power are more complicated, such as salt spray corrosion, wave load, sea ice collision, typhoon damage and other constraints. Accurate wind power prediction can enhance the controllability of wind power, ensure the stable operation of the grid, and improve the ability of the grid to accept wind power [3]. Compared to onshore wind power prediction, for offshore wind power prediction, more

* Corresponding authors at: College of Intelligence and Computing, Tianjin University, China.
E-mail addresses: lixuewei@tju.edu.cn (X. Li), rgyu@tju.edu.cn (R. Yu).

https://doi.org/10.1016/j.future.2020.06.024
0167-739X/

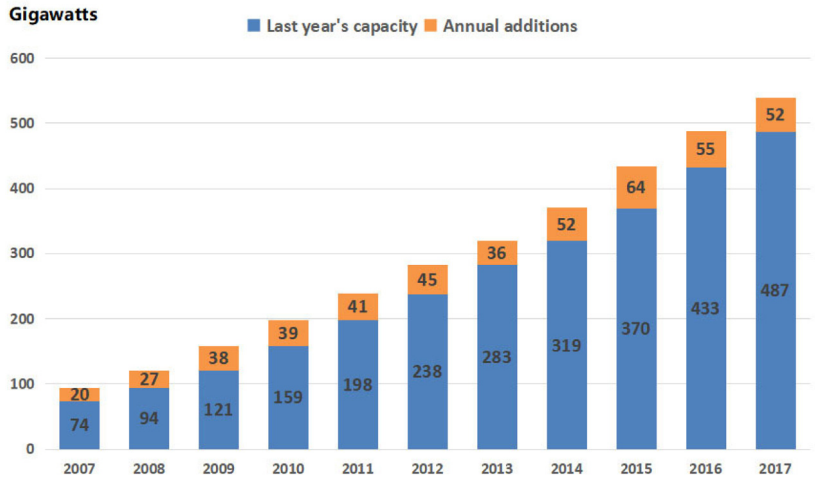


Fig. 1. Global wind power capacity and annual additions from 2007 to 2017 [4].

difficulties lie in fewer environmental factors, more dispersed distribution of wind turbines, and fewer adjacent spatial nodes for a single wind turbine which can be referred to during the process of prediction.

At present, smart grid is a topic of great concern in recent years, and wind power forecasting technology is conductive to smart grid [5,6]. There have been a lot of researches on wind power prediction, including physical methods [7,8], statistical methods [9,10] and machine learning methods. Physical models forecast wind power by using physical and meteorological variables, including numerical weather prediction (NWP) and environmental temperature. It is suitable for long-term planning rather than short-term forecasting [11]. The statistical models assume that there are linear relationships between wind energy data series and forecast them, while these models cannot correctly forecast nonlinear wind power. In machine learning methods, support vector machine regression (SVR) [12,13], k-nearest neighbor regression (kNN) [14,15] and multi-layer perceptron neural network (MLP) [16,17] are used to model wind speed or power time series. Machine learning methods actually simplify wind power prediction, but fail to improve the prediction accuracy in recent years.

Each model has disadvantages and advantages. Therefore, researchers began to propose hybrid models, which integrated the advantages of different models, and have achieved some good results [18,19]. These are improvements in actual prediction methods. But in fact, for wind farms, the changes in the space environment in which it is located should also be considered. It is believed that the wind has its rules for time and space changes, so it is necessary to refer to historical statistics in time order to better make wind power prediction. Besides, some current studies have referred to some environmental conditions around wind turbines as spatial statistics, and other methods also use data from other adjacent wind turbines or wind towers. These methods are all used for the prediction of a single wind turbine, which cannot make full use of the spatial features of wind farms, so when it comes to offshore wind farms with a more dispersed distribution of wind turbines, the effect will be worse. Finding features that can simultaneously express the temporal and spatial state of wind farms is the key to improve the prediction accuracy.

Based on the above situation, this paper proposes a new data organization form of wind farm, which connects wind turbines in a certain range based on their geographical locations so as to form a graph. This graph can represent the direct spatial correlation between wind turbines, and later can be used to extract spatial features. Then, the encapsulated data structure representing the spatial and temporal features of wind farms can be obtained by superimposing spatial maps at different time nodes in a certain time range. Finally, feature extraction and prediction are carried out in the way of Superposition Graph Neural Network (SGNN) structure proposed in this paper, which maximizes the utilization of spatial and temporal features of wind farms. The experimental results show that within the selected four offshore wind farms, the mean square error (MSE) drops by 9.80% to 22.53% compared with current-advanced methods.

The innovations of this paper are listed in the following three aspects:

1. Wind turbine data nodes are connected based on their geographic locations to obtain their spatial correlation. And then the temporal features are expressed by overlapping the time series. The historical data of the wind farm is encapsulated and can represent the temporal and spatial variations of the entire wind farm. The results of this step are then applied for the input of the network model.
2. The Superposition Graph Neural Network structure is proposed, which refers to the feature transfer method of graph neural network and the superposition structure of general convolutional neural network on feature channel [20]. It can extract the spatial and temporal features at the same time of wind farm data for prediction. The size of the network is related to the size of the wind farm.
3. The method in this paper can predict all data nodes in a certain range at the same time, and can also be used in researches such as classification and regression of other irregularly distributed data nodes. The method proposed is versatile.

2. Related work

In this part we introduced some machine learning methods that can predict wind power. In fact, most wind power plants use these methods. The biggest difference between these methods and the method in this paper is that they can only predict a single node. We also introduced the graph neural network, which is the basis of the method in this paper.

2.1. Common prediction methods

Wind power prediction has good short-term predicting effect under the guidance of machine learning. Researchers can use regression models or neural networks to map historical time

series data to the data at the predicting node in the future. The commonly used methods include SVR, kNN and LSTM [21,22], etc.

SVR has perfect mathematical foundation and performs best among a large number of regression analysis methods. After normalizing the hyperplane, all the data turn to be the closest to the hyperplane to achieve the regression. This process can be summarized as finding the parameter that satisfy the constraints of (1) and minimize the value of (2). In these two equations, C and ε are empirical parameters, ξ_i and ξ_i^* are called relaxation factors, and w and b represent hyperplanes. SVR method has been proved to be one of the best methods in wind power prediction in many literatures.

$$\begin{cases} y_i - \langle w, X_i \rangle - b \le \varepsilon + \xi_i \\ \langle w, X_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{1}$$

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2}$$

kNN is the simplest machine learning model based on similarity metric and it still has a good performance in practice. As the similarity between the two vectors is negative correlated with the distance between them, the similarity can be represented by distance. The k vectors in a set with the minimum distance from the target vector x can be called k nearest neighbors of x. If $\aleph_k(x)$ denotes the subscript of k nearest neighbors of x in the training set, then the prediction result $p(x)$ of kNN model for x is generated by Eq. (3), and the function f can use arithmetic average method, weighted average method or other methods that are more complex.

$$p(x) = f_{i \in \aleph_k(x)}(y_i) \tag{3}$$

LSTM algorithm, known as long short-term memory, is a special form of RNN (Recurrent neural network). RNN is the general term of a series of neural networks capable of processing sequential data. LSTM adds different gates (input gate, forgetting gate and output gate) on the basis of RNN to make the weight of self-circulation constantly change, while avoiding the problem of vanishing-gradient or gradient expansion. The calculation method in LSTM cell is shown in (4), where f_t represents forgetting gate, i_t represents input gate, o_t represents output gate, C_t represents the cell state at time t, x_t represents input at time t, and h_t represents output of cell at time t.

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ C_t = f_t * C_{t-1} + i_t * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ h_t = o_t * \tanh(C_t) \end{cases} \tag{4}$$

The above methods essentially use sequential data, providing no additional processing method for the data containing spatial information, which cannot adequately reflect the spatial and temporal changes of air in the region for wind power prediction.

2.2. Introduction of graph

Graph is a nonlinear data structure in computer science that describes some of the data elements and their interrelationships. It has many examples in real life, such as maps, transportation networks, social networks, and so on.

A graph consists of a finite non-empty set of vertices and a set of edges between vertices, usually expressed as Eq. (5), where G represents a graph, V is a set of vertices in graph G, and E is a set of edges in graph G. Edges can be directed or undirected, depending on whether there is a directional dependency between the vertices.

$$\begin{cases} G = (V, E) \\ V = (v_0, v_1, \ldots, v_n) \\ E = (vi, vj)\ G \text{ is an undirected graph} \\ E = \langle vi, vj \rangle\ G \text{ is a directed graph} \end{cases} \tag{5}$$

If there is no directionality in all the edges of a graph structure, then this graph is called an undirected graph. If the edges are directional in the graph, then the graph is called a directed graph. For directed graphs, the order of the two vertices is required when expressing the edges. Fig. 2 shows the two types of graphs.

During the actual application and modeling process, the vertices and edges in the graph often contain different kinds of information. The information in the vertices can be used to represent the attributes of the vertices. The information in the edge will have side weights in addition to the direction. For example, in a map, vertices can represent cities, and weights can represent distances between cities.

A common way of storing a graph is the adjacency matrix. The adjacency matrix uses two arrays to represent a graph: a one-dimensional array is used to store information about each vertex; a two-dimensional array (i.e., an adjacency matrix) is used to store edge or arc information in the graph. For the graph $G = (V, E)$, the adjacency matrix *matrix* is a square matrix of $|V| * |V|$, assuming $1 <= i, j <= |V|$, if *matrix*[i][j] is equal to zero, indicating that there is no edge between vertex i and vertex j; conversely, if *matrix*[i][j] is not zero, it means that there is an edge between vertex i and vertex j, and the value of *matrix*[i][j] is the weight of the edge.

For the wind energy prediction task in this paper, the vertex can represent a wind turbine, which contains various properties of the wind turbine and the output power value, and the degree of correlation between the wind turbines can be represented by the weight of the edge between the wind turbines. Since the two wind turbines have different degrees of influence on each other, the weights of the edges in the two directions will be different. Therefore, it can be considered that the graph composed of the wind farm wind turbine is a weighted directed graph.

2.3. Graph Neural Network

The network structure in this paper refers to the idea of Graph Neural Network (GNN). The concept of graph neural network was firstly put forward in 2009 [23]. GNN extends the existing neural network to deal with the data represented in the graph and can learn and represent unstructured data well, which is also widely used in structured scenarios [24]. It can effectively model the relationships or interactions between objects in the system, while giving the deep learning model a certain degree of causal reasoning ability. Because of its good performance and ability to explain, GNN has become a common graph analysis method.

A graph neural network is a neural network that runs directly on the graph structure. In the graph, each node is defined by both its features and related nodes. In the process of deep learning, information is transmitted and integrated between nodes. After multi-level calculation, the information retained in each node can be used for classification and regression of nodes [25,26]. The computational process of GNN can be abstracted into (6) and (7), in which $x_n(t)$ denotes the state of node n in the iteration t. In (6), f_w denotes the process of state transformation in the local region, l_n, $l_{co[n]}$, $x_{ne[n]}$ and $l_{ne[n]}$ being the attributes of vertex n, the attribute of associated edges, the state and attribute of adjacent vertex respectively. Eq. (7) represents the local output function $g(w)$, with $o_n(t)$ representing the local output result. What is

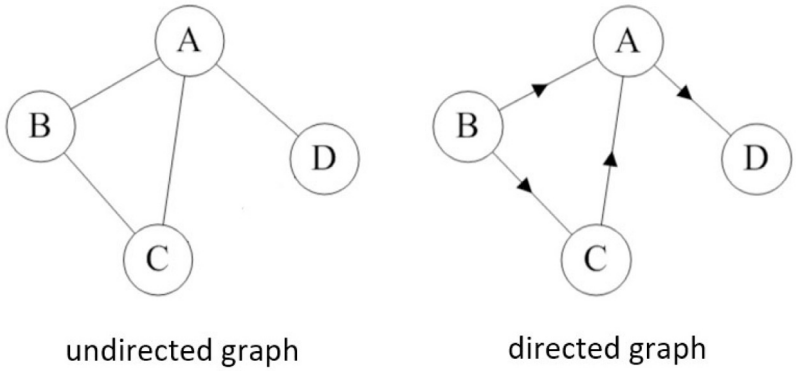


Fig. 2. Two types of graphs.

more, this function should be designed according to the actual task requirements.

$$x_n(t+1) = f_w(l_n, l_{co[n]}, x_{ne[n]}(t), l_{ne[n]}) \tag{6}$$

$$o_n(t) = g_w(x_n(t), l_n) \tag{7}$$

The graph neural network is suitable for an end-to-end learning framework, and can be used for predictive tasks. However, there are some disadvantages in GNN. If there are too many GNN layers, it will cause over-smoothing problems. The number of neighbors of a node increases so that the representation of the final node tends to be consistent. Also, GNN cannot currently handle dynamic graphs. Changes to the base graph will cause the entire network to be rebuilt and retrained.

The research on related variants of GNN models is a hot topic in the field. These variants have improved the performance of GNN in some aspects. These variants are mainly divided into graph types, GNN training methods, and improved network propagation methods [27]:

1. In terms of graph types, in addition to common undirected graphs and directed graphs, based on the problem that needs to be solved, the graph itself can also contain more information, such as the type of edges and the attributes of nodes. Such as RGCN [28], this method is used for link prediction and entity classification tasks in the knowledge graph, that is, the recovery of edges and nodes in a graph. In relationship representation, different entities have different types of relationships. RGCN depends on the type and direction of edges. Different types of local structures are used to extract feature information, and finally all results are integrated.
2. The training method of GNN can also be improved. Such as self-training and co-training, use the results sampled from the trained model and repeat the training, or assist the training of other models by the trained model. We can also adjust the transmission method of feature information during training. For example, FASTGCN [29] samples between layers of the network instead of transmitting the information of the full graph. This method speeds up the training of the network.
3. In terms of network propagation methods, referring to the research on convolutional neural networks, it is also possible to adjust the structure of GNN, such as adding short-cut connections, adding attention mechanism to graph neural networks in order to extract features more effectively [30], or adding gating structure to distinguish information that needs to be retained or to be discarded [31]. Changing the network structure is to deal with different tasks, not that more changes are better. The SGNN proposed in this paper is also an improvement of GNN based on network propagation.

Amid the rapid development of GNN, data processing methods and network structures based on GNN have been widely used in various kinds of fields [32,33], such as social networks, knowledge graph, recommendation systems, the study of molecular structure in chemistry [26,34,35]. For wind farms, wind turbines and wind towers and other wind measurement equipment can be considered as nodes in a graph. It is difficult to find a uniform rule in spatial distribution of these nodes. So we consider connecting these data nodes to get a fixed graph structure. And then, it will be effective to process this graph in the method based on GNN.

3. Proposed method

The data related to wind power can be easily integrated in space according to their geographic locations, but it is difficult for traditional machine learning methods to fully learn these spatial features. Graph neural network provides a good idea. It does not require the specific spatial distribution of data itself, and can take advantage of the links between the data nodes, so being suitable for extracting the spatial features of wind farms. For the processing of time series, the channel superposition method can be used. In this chapter, the data integration mode and model structure are introduced in detail, including the idea of extracting spatio-temporal features through a graph convolutional network, and an effective end-to-end prediction method.

3.1. Graph structure and SGNN input

Most methods of wind power prediction use simple linear time series as data features. Because of the correlation of wind turbines geographical location, there are some methods for predicting wind turbines based on wind turbines and related environmental factors. Fig. 3 shows that how the current wind turbine can be predicted by other wind turbines located adjacent to each other. This idea is derived from [36] and can be used as the basis for the construction of the graph in this paper. These types of methods can use the temporal and spatial characteristics of wind turbines to some extent and have achieved some results. However, the properties of the wind itself can also be treated as data for the overall environment. It can describe a variety of information and trends in a certain area and space, and can play a very important role in wind power prediction. Based on the above ideas, each node can make full use of the spatial characteristics of the entire region.

When sorting out the motor data nodes in shore wind farm, we connect the adjacent nodes in space according to their geographical locations. The input graph is generated by using the

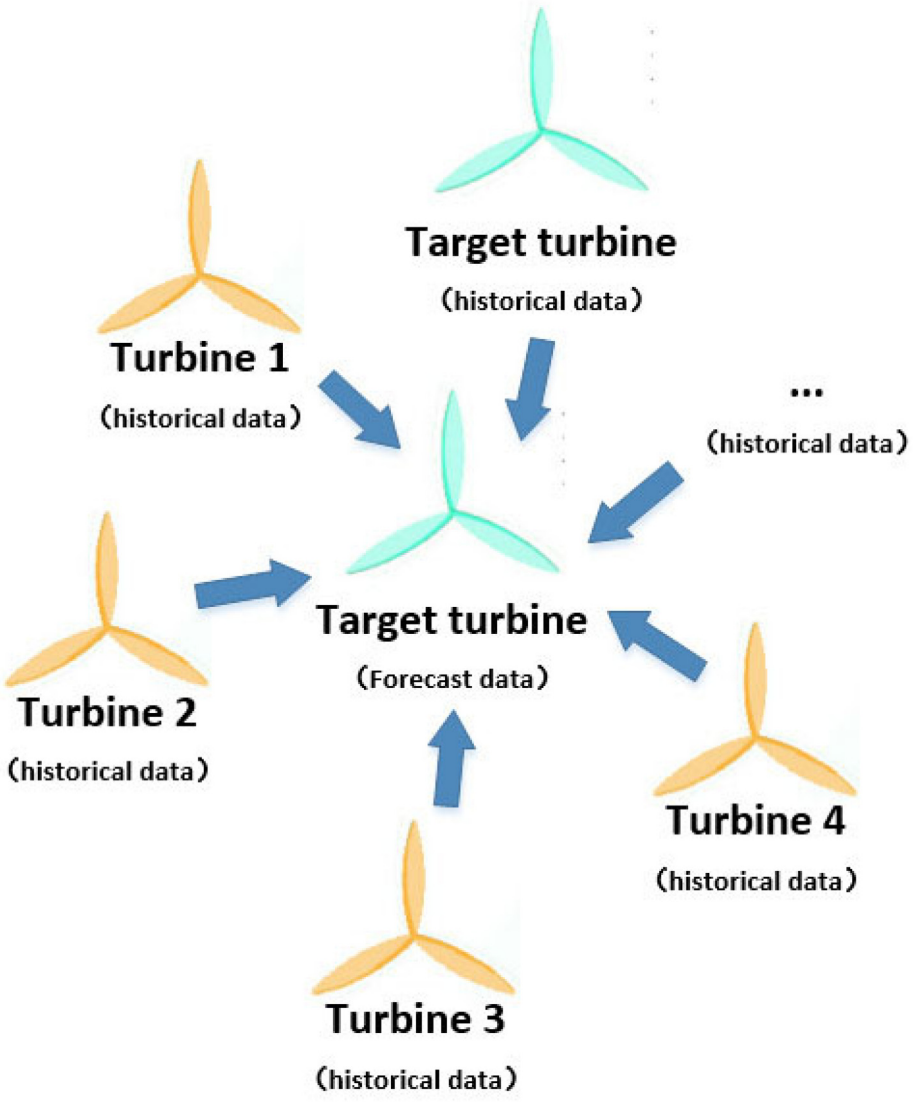


Fig. 3. One basic idea of prediction using space information [36].

pseudocode in Algorithm 1. Whether this method of generating graphs is the most effective is not yet evaluated, which is the direction we can conduct research later, but this method can quickly generate a usable connected graph. In practical applications, if the selected area is large, the entire picture does not need to be fully connected. We can cut the graph according to actual needs, or adjust the algorithm for generating the graph.

For the constructed two-dimensional mesh graph, it is superpositioned on the time series as the input of the subsequent SGNN, as shown in Fig. 4. After the final feature integration, it is shown in the two-dimensional mesh graph structure in space and in the number of feature channels after superposition in time. This structure already contains the spatial and temporal features required for prediction, and as a superpositioned graph structure, information transfer between nodes is facilitated.

The expression of layer structure is shown in (8). x_l represents the output at lth layer, x_l^i represents the ith channel in x_l, $p_l^i(j)$ represents the jth node in x_l^i, and n_l represents the total number of channels in the lth layer, m represents the total number of wind turbines in the area. For the input layer, $l = 1$, and n_1 is the time series length of the input.

$$\begin{cases} x_l^i = [p_l^i(1) \dots p_l^i(m)]_{connection}, i = 1, 2 \dots n_l \\ x_l = [x_l^1 \dots x_l^{n_l}]_{superposition} \end{cases} \tag{8}$$

After that, the input value and the value of the label are normalized and mapped to a decimal between [0, 1]. This step prevents the network to be subject to excessive differences in the calculation, so as to increase the accuracy of the weight parameters of the network input layer and the output layer, and accelerates the initial training process. Eq. (9) represents the normalization method, where x_{max} represents the upper limit of the value (refer to the real wind power output and the data set), and $\boldsymbol{X}_{in}$ is processed to get $\boldsymbol{X}_{out}$.

$$\boldsymbol{X}_{out} = \frac{\boldsymbol{X}_{in}}{x_{max}} \tag{9}$$

3.2. SGNN structure

The Superposition Graph Neural Network(SGNN) structure proposed in this paper refers to the simple GNN method when extracting the spatial features of wind farms, that is, when the value of the current node is calculated through each layer of network, the data of itself and its adjacent nodes are in consideration.

Algorithm 1: Method for generating graph of the wind farm

Input:
L, the threshold distance length;
m, the number of wind power data nodes in the wind farm;
A zero matrix G with a size of (m, m);
$p(1) \ldots p(m)$ as the turbines, with their longitude, latitude, and ids from 1 to m.
Output:
Matrix G,used to store the structure of the graph.

Function:
for *each two nodes* $p(i)$ *and* $p(j)$ **do**
  if *Distance between* $p(i)$ *and* $p(j)$ *is less than* L **then**
    $G(i, j)$ and $G(j, i)$ are set to 1.
  end
end
for $p(i)$, *i from* 1 *to* $m - 1$ **do**
  for $p(j)$, *j from* $i + 1$ *to* m **do**
    if $G(i, j) == 1$ ***and*** $p(i)$*'s id is not equal to* $p(j)$*'s id* **then**
      Set $p(j)$'s id equal to $p(i)$'s id.
    end
  end
end
Count different id numbers, then get different clusters $d(1) \ldots d(n)$.
if $n == 1$ **then**
  return G;
else
  for *each two clusters* $d(i)$ *and* $d(j)$ **do**
    Calculate the distance between the inner nodes of $d(i)$ and $d(j)$. Get the closest distance between the two clusters.
  end
  Sort the closest distance between all clusters from small to large, get set C.
  for *each distance from* C **do**
    The distance is from $d(i)$ and $d(j)$.
    if $d(i)$ *and* $d(j)$ *is not in the same cluster* **then**
      Combine $d(i)$ and $d(j)$. Get $p(i)$ and $p(j)$, as the two nodes in two clusters at the closest distance.
      $G(i, j)$ and $G(j, i)$ are set to 1. $n = n - 1$.
    end
    if $n == 1$ **then**
      return G;
    end
  end
end

In terms of time features, some ideas of convolution layers are used, that is to say, feature extraction and integration are carried out to the maximum extent by adjusting the number of feature channels output. The final result is superimposed with the input of the current layer as the output of the current layer.

The calculation method of SGNN layer is shown in (10). $p_l(i)$ represents the value of all channels of node i at lth network layer. D represents all nodes adjacent to node d (including node d itself), and W_l is the weight. The size of W_l is (n_l, n_l). C_l is the output after the local computation. O_l indicates some post-processing and optimization methods after lth layer.

$$\begin{cases} p_l(d) = \sum_{i \in D} W_{l-1}(i) p_{l-1}(i) \\ C_l = [p_l(1) \ldots p_l(m)]_{connection} \\ x_l = O_l(C_l + x_{l-1}) \end{cases} \tag{10}$$

The superposition graph convolution layer is shown in Fig. 5. The experiment in this paper contains two SGNN layers. Each layer structure is characterized by using convolution of graph to extract spatial features, and extracting temporal features by layer superposition which encourages feature reuse and alleviates the problem of vanishing-gradient.

In each layer, independent local calculations are required for each node, which is the most computationally intensive part of the algorithm. At the same time, the current calculation results are superpositioned with the current layer input, which also means that as the overall number of layers increases, the network will expand exponentially. This is why we have less network layers.

The overall structure of SGNN consists of input layer, multiple graph convolution layers, fully connected layer and output layer. The task of the fully connected layer is to map the features previously extracted to each node in the graph by the full connection, and then take the node value as the prediction result. In the experiment of this paper, all valid data collection nodes need to be predicted, so the output size of the network is the same as the initial input size. The size of the output layer in actual application is determined by the number of data nodes that need to be predicted. If only a few wind turbines need to be predicted, the input is still the data in the entire area, and then the output layer size is adjusted and only these wind turbines are used as output labels for training.

In SGNN structure, batch normalization layer is added after each graph convolution layer to accelerate training speed and indirectly improve training performance [37]. In the deep neural network, the activation input value before the nonlinear transformation is gradually shifted or changed as the network depth is deepened or during the training process. The essential reason why the training convergence is slow is that the overall distribution gradually approaches the upper and lower limits of the value interval of the nonlinear activation function, so this leads to the disappearance of the gradient of the low-level neural network in the back propagation. Batch normalization is a standardization method which can make the distribution of any neuron input value of each layer of neural network into a standard normal distribution with a mean of zero and a variance of one. In this way, the activation input value will fall in the sensitive area of the nonlinear activation function, avoiding the disappearance of the gradient and speeding up the training convergence.

Considering that wind power prediction is a non-linear prediction, we can use the ReLU (Rectified Linear Unit) [38] as the activation function, so the SGNN layer output will be processed by Eq. (11). ReLU can make the output of a part of neurons zero, which causes the sparseness of the network and reduces the interdependence of parameters. Therefore, the over-fitting problem is alleviated.

$$ReLU(\boldsymbol{x}) = max(0, \boldsymbol{x}) \tag{11}$$

Owing to that graph convolution layer calculates parameters independently for each data node, the whole network will produce a large number of parameters, easily leading to over-fitting in training. So the dropout [39,40] processing is carried out in each graph convolution layer. The dropout method is show in Eq. (12). x indicates the output of the current layer, and the Bernoulli function produces a random zero–one distribution, indicating that the network randomly removes some nodes to prevent over-fitting.

$$\begin{cases} r_j^{(l)} \sim Bernoulli(p) \\ \tilde{\boldsymbol{x}}^{(l)} = \boldsymbol{r}^{(l)} * \boldsymbol{x}^{(l)} \end{cases} \tag{12}$$

The structure of the whole network is shown in Fig. 6. In the network, the number of internal SGNN layers can be adjusted according to requirements. If there are fewer data nodes in the wind farm, the number of layers can be increased appropriately.

The computational complexity of SGNN is shown in Eq. (13). This shows two SGNN layer and one fully connected layer. F is the length of the feature window. P is the total number of wind turbines, and E is the number of wind turbines connected to

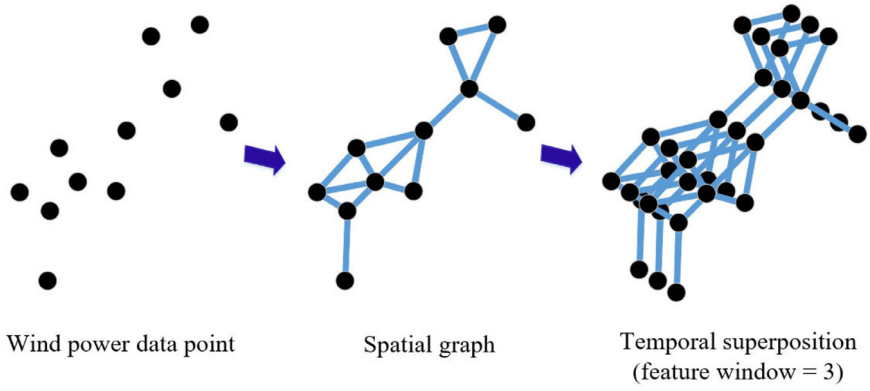


Fig. 4. The structure sketch of SGNN input with few nodes and no intersecting edges drawn (for example).

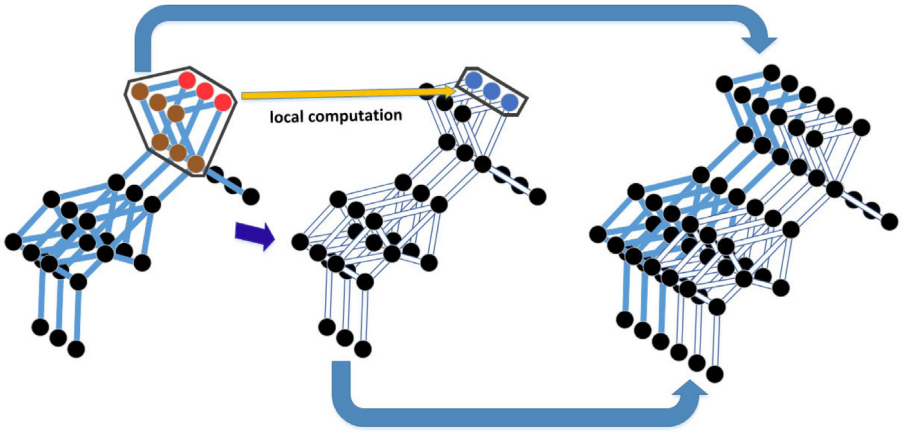


Fig. 5. Superposition graph convolution layer.

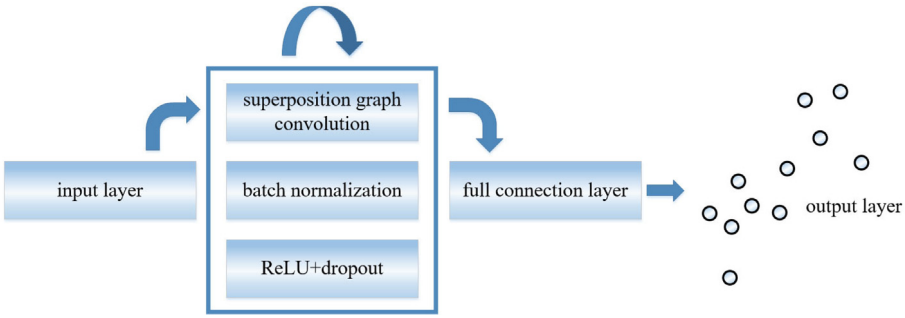


Fig. 6. SGNN overall structure.

each. Adjustment of the number of SGNN network layers does not change the order of magnitude of the computational complexity. No bias is added during the calculation, so space complexity is the same as time complexity.

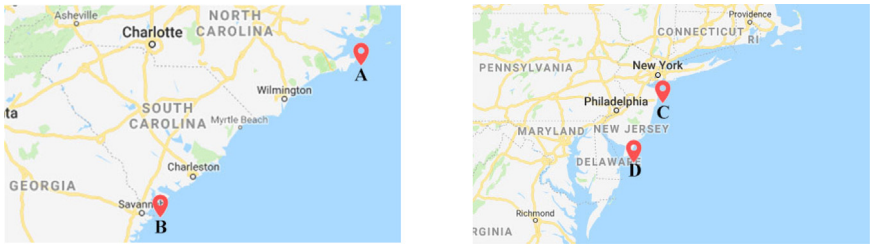
$$Time \sim O((F^2 + (2F)^2)\sum_{i=1}^{P} E_i + 4FP^2) \tag{13}$$

4. Experimental results

4.1. Data sets

The data used in the experiment are collected from the open wind power data set on NREL [41] from 2010 to 2011. The data in 2010 are used for training, and that in 2011 are used for testing. Four offshore wind farms A, B, C and D, located on the east coast of the United States, are selected in the experimental area. The specific locations are shown in Fig. 7. The geographic range of offshore wind farm A is from (34.585, −76.424) to (35.054, −75.865). That of offshore wind farm B is from (31.600, −80.570) to (32.122, −80.097). That of offshore wind farm C is from (39.815, −73.929) to (40.210, −73.580). That of offshore wind farm D is from (38.673, −74.650) to (39.011, −74.348). The sampling interval is 10 min, and the output power of the wind turbine ranges from 0 to 16 MW, which is also the range we directly use in prediction.

When selecting experimental data, the four offshore wind farms are not the sparsest or densest distributed areas of wind turbines in the optional area. This is because for the method proposed in this paper, the size of the graph directly affects the algorithm calculation time, and we only need to evaluate whether the algorithm is valid. In the early experiments involved in this



Wind farm A and B. Wind farm C and D.

Fig. 7. The locations of the offshore wind farms.

paper, we have found that the accuracy of the prediction is not really related to the distribution density of wind turbines in the region, it depends more on local environmental factors.

For subsequent experiments, the horizon predicted in this experiment is 30 min, which equals to three time intervals in the data, and feature window represents the number of historical intervals retrieved forward. The description of the horizon and feature window is shown in Fig. 8.

4.2. Evaluation criteria

Accuracy is the most important factor to measure the effect of wind power prediction. The main error evaluation indicators are mean square error (MSE) and root mean square error (RMSE), RMSE being the square root of MSE. The evaluation standard used in this paper is MSE. Its calculation method is shown in (14), in which t represents the time step, $prediction_i$ the predicted value of the node, and $real_i$ the real value of the node.

$$MSE = \frac{1}{t}\sum_{1}^{t}(prediction_i - real_i)^2 \tag{14}$$

In the evaluation of the results, since we predicted the wind turbines in the entire area, it is not suitable to display only the prediction results of a certain wind turbine and its prediction curve. We will study the prediction error in the entire region at the same time. In the method used for comparison, we can only do single node prediction, and we need to summarize all the errors. We will show the maximum and minimum values of the error in the entire region, as well as the distribution of the error, to determine whether the prediction results will be within a suitable range. At the same time, we also arrange the prediction errors according to their geographical location to visually display the different geographical locations and the impact of the density of data nodes in the region on the prediction results.

4.3. Network training

One of the most important factors in any deep learning model is the training data set label. The optimizer selected by the network is the momentum optimizer. The loss function of the network is the average MSE of the entire wind farm, which is Eq. (15), where m represents the number of valid data nodes in the current training wind farm.

$$loss = \frac{1}{m}\sum_{1}^{m}MSE_i \tag{15}$$

Algorithm 2 shows the overall training and testing process. As the number of iterations increases during the training process, the network learning rate will gradually decrease[42]. The final training time is positively correlated with the size of the wind farm and the number of SGNN layers.

SGNN is not much parameter sensitive, mainly because the network does not have too much deformation and layers. If the number of layers of the basic SGNN is increased, the initial convergence of the network will be slower and the risk of over-fitting will increase. During the training, the parameters of each layer of the network are adjusted at the same time. When the network is fixed, remaining hyperparameters are only the dropout layer retention coefficient, the percent decrease in learning rate of network, and the number of iterations. Between each complete training, we adjust the dropout layer retention coefficient(0.3–0.8, with interval 0.05) by verifying the degree of over-fitting, and the learning rate decreases gradually as the number of iterations increases. The best results are gradually obtained.

Algorithm 2: Overall training and testing process

Input:
Wind farm data, including the geographic location of all points in the entire wind farm area and the output power at different historical moments;
IT_{max}, the maximum number of iterations;
SGNN structure;
W, randomly initialized network weight.
Output:
predicted result and predicted error.
Network model and its internal weights W.

```
Function:
Calculating the graph structure G of the wind farm;
Normalize wind power data;
Integrate data to get SGNN input through G;
Integrate data to get corresponding labels.
for range of IT_max do
    Calculate network through input;
    Calculate loss through network output and labels;
    Do back propagation algorithm through optimizer,
    update network weight W.
end
Calculate network output;
Do anti-normalization for the network output, get predicted
result;
Save network model;
Calculate prediction error.
```

4.4. Results and analysis

This section compares and explains the prediction results, error distribution, and training time. We mainly verify the prediction accuracy, prediction result stability, and prediction efficiency of SGNN. These are also the main concerns of wind farm operators in wind power prediction.

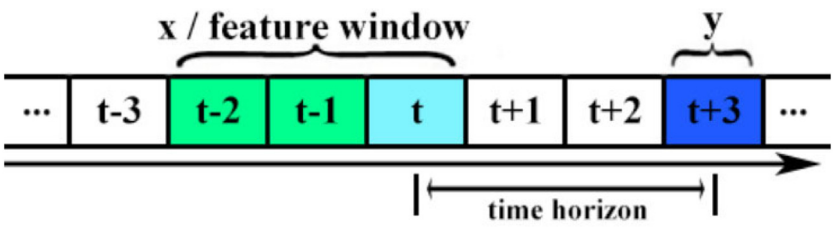


Fig. 8. Section of the temporal series. The pattern x is mapped to label y. Here shows the time horizon and feature window.

Table 1
MSE of offshore wind farm A.

FW		kNN	SVR	LSTM	SGNN
3	MAX	2.907	2.042	2.160	2.019
	MIN	1.753	1.678	1.862	1.318
	AVE	1.886	1.828	1.996	1.513
4	MAX	2.412	2.045	2.175	1.927
	MIN	1.760	1.681	1.860	1.255
	AVE	1.889	1.826	2.005	1.444
5	MAX	2.492	2.045	2.182	1.908
	MIN	1.776	1.679	1.853	1.262
	AVE	1.902	1.824	2.012	1.434
6	MAX	2.344	2.044	2.249	1.872
	MIN	1.788	1.679	1.882	1.244
	AVE	1.915	1.823	2.019	1.409
7	MAX	2.386	2.043	2.190	1.873
	MIN	1.809	1.678	1.893	1.245
	AVE	1.940	1.823	2.030	1.412

Table 2
MSE of offshore wind farm B.

FW		kNN	SVR	LSTM	SGNN
3	MAX	1.834	1.861	2.006	1.815
	MIN	1.568	1.497	1.675	1.258
	AVE	1.657	1.622	1.807	1.463
4	MAX	1.838	1.851	1.985	1.828
	MIN	1.571	1.487	1.699	1.211
	AVE	1.661	1.614	1.812	1.419
5	MAX	1.914	1.839	1.989	1.788
	MIN	1.579	1.482	1.680	1.184
	AVE	1.672	1.608	1.820	1.392
6	MAX	1.943	1.829	2.043	1.768
	MIN	1.585	1.477	1.690	1.177
	AVE	1.683	1.603	1.822	1.380
7	MAX	1.871	1.822	2.017	1.772
	MIN	1.603	1.477	1.681	1.175
	AVE	1.694	1.600	1.829	1.378

Table 3
MSE of offshore wind farm C.

FW		kNN	SVR	LSTM	SGNN
3	MAX	2.488	1.925	2.051	1.979
	MIN	1.521	1.513	1.673	1.163
	AVE	1.647	1.655	1.810	1.412
4	MAX	2.253	1.918	2.043	1.972
	MIN	1.547	1.498	1.697	1.208
	AVE	1.660	1.640	1.816	1.429
5	MAX	2.341	1.915	2.119	1.929
	MIN	1.576	1.491	1.707	1.116
	AVE	1.689	1.630	1.824	1.357
6	MAX	2.326	1.909	2.098	1.919
	MIN	1.593	1.490	1.693	1.117
	AVE	1.726	1.626	1.831	1.362
7	MAX	2.379	1.901	2.077	1.892
	MIN	1.615	1.488	1.693	1.114
	AVE	1.747	1.623	1.841	1.331

The comparative experimental method of this method is kNN, SVR and LSTM mentioned in Section 2. When we use these methods to predict each wind turbine, we extract the information of the surrounding wind turbines to improve its prediction accuracy. The reference range is taken from the mesh graph constructed in Section 3.1, and the data contained in the nodes directly connected with the prediction nodes are used to assist the prediction. This step can be considered as extracting spatial information around each data node.

In the kNN method, the k value is set to 10, and the connected points are equally weighted. The SVR uses an RBF kernel with a kernel coefficient of 0.0001 and a penalty parameter of 100. We only use a single-layer LSTM structure to compare predictions, and have adjusted parameters to deal with wind energy prediction tasks. When using these methods for comparison, we have adjusted the parameters as much as possible to obtain the best results.

The experimental results of four offshore wind farms under different length of feature windows are shown in Tables 1, 2, 3, 4. For the methods of kNN, SVR and LSTM, it is necessary to predict all nodes in the selected range before calculating the MSE. As can be seen from the table, compared with other methods, the error obtained by using SGNN for prediction is greatly reduced. Compared with the best performing SVR method among other methods, the prediction error of SGNN has decreased by 9.80% to 22.53%.

By comparing the results under different feature windows, it can be found that SGNN makes full use of the temporal features of the wind farm. With the increase of the forward reference time as the feature window, the prediction accuracy can be gradually improved. This should depend on the SGNN superpositioning these features when processing the features, and still retain the features of the entire model operation process until the last fully connected layer. Common neural networks always continuously extract and filter features and keep only the last part. In comparison, just using the integrated features is more effective for tasks with unequal input and output (such as image classification and segmentation, the input is an image, and the output is a specific value or a binary image), but for end-to-end tasks like wind energy prediction that output accurate values, neural networks should retain features as much as possible. During data preprocessing and network design, we may distinguish features, but avoid over-filtering features.

In most prediction methods, as the feature window increases, there are too many reference data available for short-term wind energy prediction. That may cause over-fitting or the model cannot effectively converge. Here only the wind energy data is used for prediction, which causes little effect. If we add reference data with different dimensions (such as some environmental parameters), this problem will be more serious. These prediction methods usually get the best results when the feature window reaches 3 or 4, and then as the feature window increases, the prediction accuracy no longer improves significantly. For SGNN, due to the network's feature retention ability and simultaneous

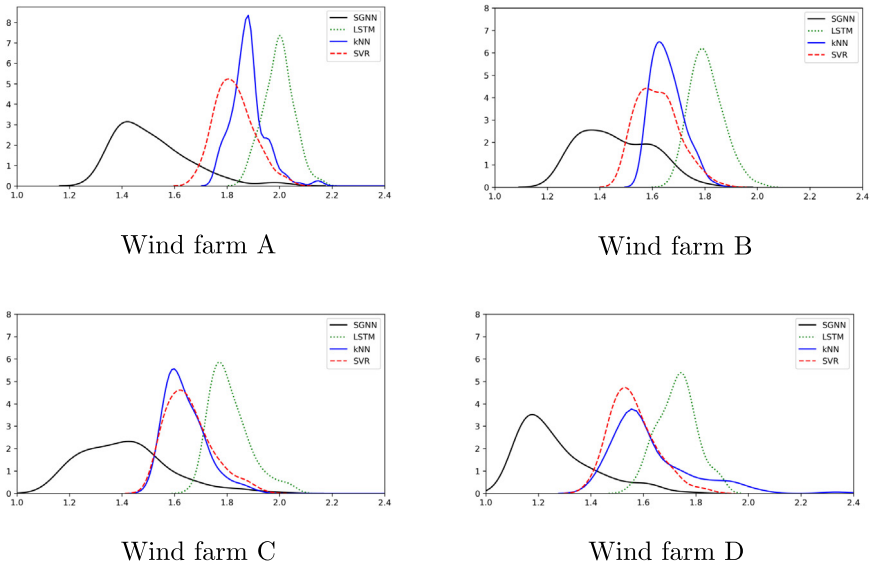


Fig. 9. Predictive error distribution for each method of the offshore wind farms.

Table 4
MSE of offshore wind farm D.

FW		kNN	SVR	LSTM	SGNN
3	MAX	3.639	1.845	1.893	1.762
	MIN	1.426	1.400	1.550	1.105
	AVE	1.661	1.560	1.728	1.267
4	MAX	3.934	1.839	1.915	1.744
	MIN	1.439	1.387	1.571	1.088
	AVE	1.669	1.548	1.730	1.256
5	MAX	3.565	1.828	1.918	1.720
	MIN	1.453	1.373	1.586	1.086
	AVE	1.655	1.541	1.740	1.250
6	MAX	2.431	1.817	2.050	1.671
	MIN	1.440	1.363	1.582	1.059
	AVE	1.620	1.535	1.746	1.220
7	MAX	3.501	1.814	1.993	1.692
	MIN	1.462	1.362	1.576	1.070
	AVE	1.677	1.532	1.757	1.229

training of the entire wind farm, by expanding the feature window, although the network will become complicated, it can still improve the prediction accuracy effectively.

Fig. 9 shows the distribution curves of predictive error of four offshore wind farms in different methods, in which the horizontal coordinate represents MSE value, vertical coordinate represents probability density, and different curves represent probability density distributions of different methods. On the whole, the prediction errors of the various methods are similar in distribution. It can be clearly seen from the figures that the prediction error probability of SGNN is more in the place where the value is smaller, indicating that the method not only greatly improves the prediction accuracy, but also guarantees the stability of the prediction. In the figure, the SGNN method has an even distribution and a large distribution width, because we directly adopted MSE as the training loss and did not set different weights for different wind turbines in this experiment. At the same time, for the prediction of offshore wind farms, some of the wind turbines always lack sufficient reference data.

Fig. 10 shows the prediction error distribution of SGNN on wind farm A, which is displayed based on the approximate distribution of geographical locations. The blue in the figure represents less error and the red represents larger error. It can be seen from the figure that the prediction result is affected by the spatial distribution of the wind turbine. Near the edge of the entire prediction area, the prediction error will be larger, while the prediction error of the wind turbine in densely distributed areas, especially the central area, is less. The difficulty to process edge information is a flaw in both the graph neural network and most of its variant network. For this result, it can also be considered that there is more spatial information in the dense area, and the SGNN model can fully learn the rule of wind power changes in this area. For the edge region, less information is available, but information in other farther regions can still be obtained through information dissemination of the SGNN model. Limited by the reference historical data actually available in the region, there is currently no effective means to solve this problem, but the prediction results in the edge region are still greatly improved compared with the traditional methods.

Table 5 shows the time required for each method to be trained once. Here we also show the program running environment and the number of iterations of LSTM and SGNN used in this experiment. For other methods, the prediction of wind turbine is done one by one, then the training time of SVR and LSTM is much longer. Despite this, kNN is still the fastest. SGNN maintains a good efficiency due to the simultaneous prediction of the entire offshore wind farm. Because the parameters of each method (especially LSTM and SGNN) need to be adjust sometimes, the actual training time is longer.

4.5. Industrial application

In practical industrial applications, the method in this paper can guide the allocation and storage strategy of wind power generation by predicting the amount of wind energy generated. The method can simultaneously utilize any effective wind measurement node in the area. In the prediction, for the problem that the prediction result on the edge of the entire region is worse than the central region, only the results close to the central region can be adopted. For prediction nodes on the edge, predictions can be made by integrating with other methods. Although, in fact, even if the method has poor results at the nodes on the edge, it is more accurate than other methods. At the same time, in actual use, we can consider increasing the wind energy data collection node to improve the prediction accuracy.

For the wind forecasting problem of offshore wind turbines, the method is not limited by the actual distribution and sparsity

Table 5
Training time of each method for wind farm A with a feature window length of 3.

Hardware	Intel Xeon Gold 6126		TESLA V100 16G	
Method	kNN	SVR	LSTM (100 iterations)	SGNN (200 iterations)
Training time (sec)	847	33325	106557	10122

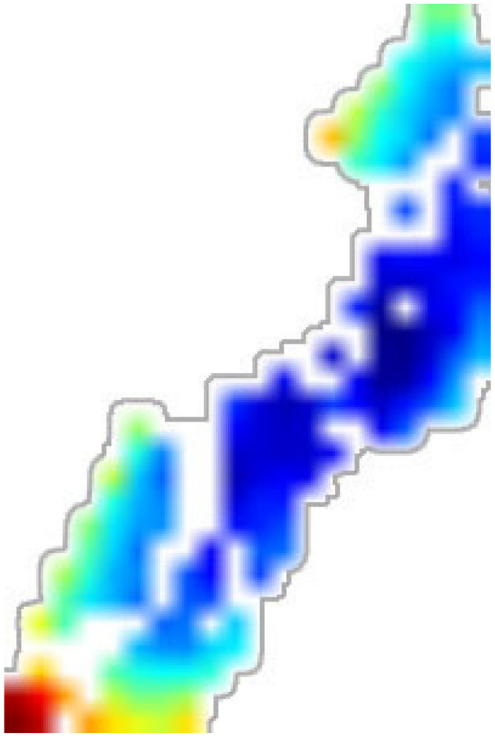


Fig. 10. Spatial predictive error distribution of SGNN of the offshore wind farm A (fw = 3).

of wind turbines when considering the reference factors, and can be used for actual applications through one training of historical data. After adding new forecasting nodes such as wind turbines, wind towers, etc., we can consider whether to reconstruct the distribution graph for training according to our requirements.

4.6. Limitations of SGNN

Although SGNN can handle most offshore wind power prediction tasks, it still has limitations:

1. We can only use a fixed graph for training at a time. After adding a new wind turbine or removing a wind turbine, we need to rebuild the graph and construct the input of SGNN. At the same time, the structure of SGNN also changes accordingly.
2. For offshore wind farms that are overly densely distributed (less often), we need to reconsider the method of constructing the graph of wind farms in order to avoid the structure of SGNN being too complicated, or only crop the wind farm into multiple parts before making predictions.
3. In practical applications, sometimes it is not necessary to predict the entire wind farm, and at this time SGNN is not suitable in terms of efficiency. If high prediction accuracy is not required, operators can also choose some less prediction accurate but very efficient prediction methods (such as KNN and some common regression equations)

5. Conclusion and future work

In this paper, the wind turbine data nodes on offshore wind farm are connected according to their geographical locations, and then feature extraction and wind energy prediction are carried out by using the proposed SGNN method, which effectively utilizes the spatial and temporal features contained in the wind farm data and finally achieves good results.

In our future work, there are also some parts that can be improved. We can evaluate and integrate environmental data such as temperature, humidity, and pressure, then adjust the structure of the current network and use these data as input to the network. Based on principle and target task, SGNN can be applied to other prediction of irregular distribution point clouds, such as solar energy forecasting. If the data sets contain elevation information, we can also consider the prediction of 3D point cloud data.

In preprocessing, an efficient graph construction method can speed up the efficiency of neural networks in extracting spatial information. However, we only propose a feasible construction method without using more theory about machine learning or computer graphics, which would remain to be studied in our future work. The structure of SGNN in this paper refers to the idea of GNN. As a fast-developing deep learning method, GNN has accumulated a lot of related researches, thus providing abundant ideas for model selection and optimization in our future research.

CRediT authorship contribution statement

Mei Yu: Conceptualization, Validation, Project administration, Funding acquisition. **Zhuo Zhang:** Conceptualization, Methodology, Software, Writing original draft, Writing - review & editing. **Xuewei Li:** Validation, Resources, Writing - review & editing. **Jian Yu:** Validation, Resources, Supervision. **Jie Gao:** Validation, Resources, Supervision. **Zhiqiang Liu:** Conceptualization, Validation, Writing - review & editing. **Bo You:** Validation, Investigation. **Xiaoshan Zheng:** Investigation, Data curation. **Ruiguo Yu:** Conceptualization, Validation, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61976155 and No. 61877043), and Key Project for Science and Technology Support from Key R&D Program of Tianjin, China (Grant No. 18YFZCGX00960).

References

[1] Y. Zhao, Y. Lin, L. Zhi, X. Song, Y. Lang, S. Jian, A novel bidirectional mechanism based on time series model for wind power forecasting, Appl. Energy 177 (2016) 793–803, http://dx.doi.org/10.1016/j.apenergy.2016.03.096.

[2] Y. Wang, J. Wang, X. Wei, A hybrid wind speed forecasting model based on phase space reconstruction theory and Markov model: A case study of wind farms in northwest China, Energy 91 (2015) 556–572, http://dx.doi.org/10.1016/j.energy.2015.08.039.

[3] D. Liu, Y. Huang, J. Guo, Analysis and prediction on wind power in provincial grid, in: 2011 IEEE Power Engineering and Automation Conference, Vol. 1, IEEE, 2011, pp. 307–310, http://dx.doi.org/10.1109/PEAM.2011.6134862.

[4] R. COMMUNITY, Renewables 2018 Global Status Report, 2018.

[5] S. Maharjan, Q. Zhu, Z. Yan, S. Gjessing, T. Basar, Dependable demand response management in the smart grid: A Stackelberg game approach, IEEE Trans. Smart Grid 4 (1) (2013) 120–132, http://dx.doi.org/10.1109/TSG.2012.2223766.

[6] Z. Yan, Y. Rong, M. Nekovee, L. Yi, S. Gjessing, Cognitive machine-to-machine communications: Visions and potentials for the smart grid, IEEE Netw. 26 (3) (2012) 6–13, http://dx.doi.org/10.1109/MNET.2012.6201210.

[7] P. Lyncha, The origins of computer weather prediction and climate modeling, J. Comput. Phys. 227 (7) (2008) 3431–3444, http://dx.doi.org/10.1016/j.jcp.2007.02.034.

[8] J. Zhao, Z.H. Guo, Z.Y. Su, Z.Y. Zhao, X. Xiao, F. Liu, An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed, Appl. Energy 162 (19) (2016) 808–826, http://dx.doi.org/10.1016/j.apenergy.2015.10.145.

[9] K. Bhaskar, S.N. Singh, AWNN-assisted wind power forecasting using feed-forward neural network, IEEE Trans. Sustain. Energy 3 (2) (2012) 306–315, http://dx.doi.org/10.1109/tste.2011.2182215.

[10] F. Ziel, C. Croonenbroeck, D. Ambach, Forecasting wind power – Modeling periodic and non-linear effects under conditional heteroscedasticity, Appl. Energy 177 (2016) 285–297, http://dx.doi.org/10.1016/j.apenergy.2016.05.111.

[11] J.P.S. Catalao, H.M.I. Pousinho, V.M.F. Mendes, Hybrid intelligent approach for short-term wind power forecasting in Portugal, IET Renew. Power Gener. 5 (3) (2011) 251–257, http://dx.doi.org/10.1049/iet-rpg.2009.0155.

[12] Moreover, Support vector regression based on grid-search method for short-term wind power forecasting, J. Appl. Math. 2014 (1–4) (2014) 1–11, http://dx.doi.org/10.1155/2014/835791.

[13] K. Chen, J. Yu, Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach, Appl. Energy 113 (2014) 690–705, http://dx.doi.org/10.1016/j.apenergy.2013.08.025.

[14] W. Yi, M. Song, J. Wang, A combined AR-kNN model for short-term wind speed forecasting, in: Decision and Control, 2016, http://dx.doi.org/10.1109/CDC.2016.7799245.

[15] E. Mangalova, E. Agafonov, Wind power forecasting using the k-nearest neighbors algorithm, Int. J. Forecast. 30 (2) (2014) 402–406, http://dx.doi.org/10.1016/j.ijforecast.2013.07.008.

[16] L. Hui, H.Q. Tian, Y.F. Li, Comparison of new hybrid FEEMD-MLP, FEEMD-ANFIS, Wavelet Packet-MLP and Wavelet Packet-ANFIS for wind speed predictions, Energy Convers. Manage. 89 (2015) 1–11, http://dx.doi.org/10.1016/j.enconman.2014.09.060.

[17] W.C. Yeh, Y.M. Yeh, P.C. Chang, Y.C. Ke, V. Chung, Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization, Int. J. Electr. Power Energy Syst. 55 (2) (2014) 741–748, http://dx.doi.org/10.1016/j.ijepes.2013.10.001.

[18] X. Yuan, C. Chen, Y. Yuan, Y. Huang, Q. Tan, Short-term wind power prediction based on LSSVM–GSA model, Energy Convers. Manage. 101 (2015) 393–401, http://dx.doi.org/10.1016/j.enconman.2015.05.065, URL http://www.sciencedirect.com/science/article/pii/S0196890415005300.

[19] J.-Z. Wang, Y. Wang, P. Jiang, The study and application of a novel hybrid forecasting model – A case study of wind speed forecasting in China, Appl. Energy 143 (2015) 472–488, http://dx.doi.org/10.1016/j.apenergy.2015.01.038, URL http://www.sciencedirect.com/science/article/pii/S0306261915000446.

[20] G. Huang, Z. Liu, V.D.M. Laurens, K.Q. Weinberger, Densely Connected Convolutional Networks, 2016, http://dx.doi.org/10.1109/CVPR.2017.243.

[21] X. Qu, X. Kang, Z. Chao, J. Shuai, X. Ma, Short-term prediction of wind power based on deep long short-term memory, in: Power and Energy Engineering Conference, 2016, http://dx.doi.org/10.1109/APPEEC.2016.7779672.

[22] R. Yu, J. Gao, M. Yu, W. Lu, T. Xu, M. Zhao, J. Zhang, R. Zhang, Z. Zhang, LSTM-EFG for wind power forecasting based on sequential correlation features, Future Gener. Comput. Syst. 93 (2019) 33–42, http://dx.doi.org/10.1016/j.future.2018.09.054.

[23] S. Franco, G. Marco, T. Ah Chung, H. Markus, M. Gabriele, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80, http://dx.doi.org/10.1109/TNN.2008.2005605.

[24] M.M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond Euclidean data, IEEE Signal Process. Mag. 34 (4) (2017) 18–42, http://dx.doi.org/10.1109/MSP.2017.2693418.

[25] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2016.

[26] S. KT, F. Arbabzadah, S. Chmiela, M. KR, A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks, Nature Commun. 8 (13890) (2017) 13890, http://dx.doi.org/10.1038/ncomms13890.

[27] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, 2018, arXiv preprint arXiv:1812.08434.

[28] M. Schlichtkrull, T.N. Kipf, P. Bloem, R.v.d. Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, 2017, arXiv preprint arXiv:1703.06103.

[29] J. Chen, T. Ma, C. Xiao, FastGCN: Fast learning with graph convolutional networks via importance sampling, in: International Conference on Learning Representations, 2018.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.

[31] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, in: International Conference on Learning Representations, 2016.

[32] D. Muratore, M. Hagenbuchner, F. Scarselli, A.C. Tsoi, Sentence extraction by graph neural networks, in: International Conference on Artificial Neural Networks, 2010.

[33] R. Li, M. Tapaswi, R. Liao, J. Jia, S. Fidler, Situation Recognition with Graph Neural Networks, 2017, http://dx.doi.org/10.1109/ICCV.2017.448.

[34] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: International Conference on Neural Information Processing Systems, 2015.

[35] S. Kearnes, K. Mccloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Comput. Aided Mol. Des. 30 (8) (2016) 1–14, http://dx.doi.org/10.1007/s10822-016-9938-8.

[36] N.A. Treiber, J. Heinermann, O. Kramer, Wind power prediction with machine learning, in: J. Lässig, K. Kersting, K. Morik (Eds.), Computational Sustainability, Springer International Publishing, Cham, 2016, pp. 13–29, http://dx.doi.org/10.1007/978-3-319-31858-5_2.

[37] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015.

[38] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: International Conference on Artificial Intelligence & Statistics, 2011.

[39] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, Comput. Sci. 3 (4) (2012) 212–223.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[41] C. Draxl, B.M. Hodge, A. Clifton, J. McCaa, Overview and Meteorological Validation of the Wind Integration National Dataset Toolkit, 2015, http://dx.doi.org/10.2172/1214985.

[42] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back propagating errors, Nature 323 (1986) 533–536, http://dx.doi.org/10.1038/323533a0.

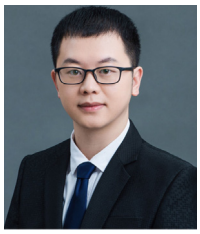
Mei Yu received Ph.D. degree in computer applicaton technology from Tianjin University. She is currently a professor engaged in computer networks, data mining, database in Tianjin University. She serves as the instructor of Tianjin University IT discipline innovation and entrepreneurship training base, responsible for the base construction.

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.

Xuewei Li received a Ph.D. in computer applicaton technology from Tianjin University. Her research directions include image processing, computer vision, artificial intelligence and so on, mainly including image segmentation, image enhancement, object detecton, and tracking. She has published more than 10 papers and conference papers in domestic and foreign journals and served as a reviewer of the top international journal APPLIED ENERGY.

Jian Yu received the Ph.D. degree from Tianjin University, China, in 2010. Now he works as a senior engineer in College of Intelligence and Computing, Tianjin University, China.

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.



Zhiqiang Liu received the M.S. degree in Computer Science and Technology from Tianjin University, China, in 2018. He has published articles in Applied Energy, a top academic journal, and has published four SCI papers. Now he is focuses on deep learning and assistant medicine.

Bo You received the bachelors degree from North University of China in 2017. He currently pursuing the master degree in Tianjin International Engineering Institute of Tianjin University. His current research include deep learning, images and graphics.

Xiaoshan Zheng received the bachelor's degree in Internet of Things Engineering from Central China Normal University, China, in 2017. Now he is focuses on deep learning and its applicaton in the field of new energy.

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 applicaton technology from Tianjin University, China. Now he works as an associate professor in College of Intelligence and Computing, Tianjin University, China. He now is supported by National Natural Science Foundation of China. Now his research is focused on deep learning, image processing and educational technology.