Cluster Division in Wind Farm Based on Ensemble Modeling of Machine Learning

Fengrui Liu[a], Baitong Li[a, *], Huiyang Xie[b, *], Yikun Yin[a], Yinan Yang[c]

[a] *School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China*
[b] *School of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, Gyeonggi, Korea*
[c] *State Grid Feidong County Power Supply Company, Anhui 230000, China*

Abstract—**In order to improve the equivalence accuracy of wind farms and widen the applicability of multiple operating conditions, a method of using Blending to merge extreme gradient boosting (XGBoost) for clustering index dimensionality reduction and dynamic time warping (DTW) optimization is proposed. Density-based spatial clustering of applications with noise (DBSCAN) is a method of clustering clusters and clustering results fusion. It is used to process wind turbines' multi-dimensional time-series feature operation data to obtain accurate and effective wind farm plans for the division of aircraft clusters on site. Firstly, leverage XGBoost-Blending to select clustering indicators for dimensionality reduction; Secondly, propose a clustering method based on DBSCAN-DTW for cluster division and perform ensemble clustering; Finally, employ Matlab / Simulink to build a simulation model. Thus, a three-phase short-circuit at the grid-connection point of a wind farm is constructed, and a case study is performed under a variety of wind speed scenarios. The results verify the accuracy and wide applicability of the equivalence model after implementing this method.**

Keywords Dynamic time warping; Fusion density clustering; XGBoost; Cluster division; Wind farm equivalence

I. INTRODUCTION

Under the challenges of resource shortages and environmental pollution in the world, the scale of wind farms is gradually increasing. However, their dynamic characteristics have a great impact on the stability of the power system. It is necessary to build a simulation model that accurately reflects the dynamic characteristics of wind farms [1]. If each fan is modeled in detail, the model structure will be too complex with high dimension, increasing the complexity of the power system and the simulation time[2-3]. Therefore, how to establish a wind farm equivalent model that accurately represents the operating characteristics plays an important role in the analysis of the safe and stable operation of the system[4], but the accuracy of single-machine equivalence can not meet the demand. Hence, how to group wind turbines reasonably and effectively is the key issue of wind farm equivalence.

At present, research on wind farm clustering mainly focuses on clustering indexes and clustering methods.

As for clustering indexes, [5] regards the rotor speed as the clustering index, but the physical quantity at a fixed time point cannot represent the change of the fan's operating state; [6] and [7] exploit the data of a certain period of time to group wind turbines, so the results are suitable for the selected period, without considering the impact of faults on grouping; Considering the impact of faults on groups, [8] selects pitch angles under both steady states and fault states. The above-mentioned documents all select mechanical indicators as the clustering indicators, but under the electro-mechanical transient time scale, the inertial time constant of the mechanical system is relatively large, leading them not appropriate to only adopt mechanical characteristics. [9] and [10] opt the multi-dimensional state variables of doubly fed induction generator (DFIG) as clustering indexes, where there are redundant and strongly correlated data; [2] selects the mechanical and electrical features of DFIG as clustering indexes, and applies principal component analysis (PCA) to eliminate the strong correlation and redundancy. However, the principal components obtained by PCA are not produced by the actual system, leading to limitations for practical engineering applications. For example, if a user has prior knowledge and knows some data features, the user cannot intervene in the process, the efficiency and effect may be worse than expected.

Therefore, the cluster division indexes should be selected from multiple perspectives. When the mechanical and electrical indicators are selected simultaneously, the effectiveness of clustering results will be improved. However, the subjective selection of clustering indicators may lead to insufficiently mining the information of power units represented by clustering data. Plus, the correlation between variables and the redundancy of data also have a great impact on the clustering effect. On the one hand, the variables with strong correlation increase the workload of index acquisition and aggregation classification. On the other hand, it will cause that the clustering results mainly represent these variables, while ignoring the influence of other factors. Such a strong correlation between variables and redundant information will lead to the tendency of the cluster division results and relatively low equivalent accuracy.

As for clustering methods, clustering algorithms have been a research hotspot recently. Most clustering algorithms have extremely fast calculation speeds with significant advantages in dealing with massive data. Common clustering algorithms for wind farms include k-means clustering [11], fuzzy c-means clustering [12], spectral clustering [13], etc., in which k-means is applied the most frequently. However, k-means is sensitive to initial centroids and noise, which may

[*] Corresponding author.
E-mail address: cheryllee626@gmail.com (Baitong Li), xiehuiyang@hanyang.ac.kr (Huiyang Xie).

cause an overall offset. Plus, it requires pre-setting the *k* value, which has strong limitations. In [14], k-means is improved based on immune outlier data and sensitive initial centers to increase the accuracy of equivalence. However, it still can't deal with time series, because it can not capture temporal dependencies while requiring fixed points in space as the initial centroids and regarding each sampling point as a discrete dimensionality.

In summary, k-means has a fast processing speed, but it is easily affected by the shape of the data set, the noise data and the initial centroids. Moreover, in the wind farm, the different wind speeds will cause turbines to have different dynamic response times [15].

Addressing the above problems, this paper proposes a clustering method for the multi-dimensional time-series of each wind turbine in the wind farm. First, it innovatively introduces XGBoost and Blending for selecting clustering indicators and simplifying the model; Second, it novelly calculates the normalized path distance obtained by DTW as similarities between turbines; Third, DBSCAN is applied for clustering; Finally, the results under various industrial conditions are integrated.

The method is implemented by python, and the wind farm model with 16 DFIGs is built by Matlab / Simulink. The three-phase short circuit is set at the grid point of the wind farm. Under various wind speed scenarios, the case analysis and the clustering are carried out according to the proposed method, whose effectiveness and accuracy are verified by the response comparison between the equivalent model and the original model.

The main innovations are summarized as follows:

(1) The dynamic response time of the same model of DFIG in a wind farm is different under different wind speeds, and it will be more different considering different electrical distances, fault locations or fault types. But the premise of using traditional Euclidean distance and cosine distance in clustering algorithms is the correspondence of time points, so they can not cluster DFIG effectively. DTW algorithm is novelly introduced to calculate the integration path, so the above problems are solved through transforming time series. It is more reliable to adopt the Euclidean distance of the integration paths as the similarity index between wind turbines, and it can effectively solve the problem of missing values in practical data of wind farms, which improves the accuracy of the model.

(2) Due to the randomness of the DFIG distribution in the multi-dimensional feature space, similar DFIGs are usually irregular clusters, while traditional distance-based clustering algorithms can only form spherical clusters, which may cause similar DFIGs to be separated while clustering. Hence, such a problem increases model complexity and reduces the clustering accuracy degree; There are plenty of noise signals in the stable operation, fault occurrence, fault recovery and other time periods of wind farms under multi-dimensional features, so traditional clustering methods will reduce the clustering accuracy. Density-based DBSCAN can be utilized to obtain clusters of different shapes with low sensitivity to noise and can eliminate the influence of noise, which solves the above problems and improves the simplification and performance of the model.

(3) The operating features of the wind turbines are in high dimensions. In order to fully consider the characteristics of DFIG, the electrical and mechanical indicators are in high dimension. There may be a strong correlation between the indicators, which will reduce the clustering speed and accuracy, all the while manually selected indicators have a strong subjectivity. In this paper, XGBoost is applied to opt clustering indexes, eliminate the strong correlation between variables and filter out the noise and redundant data. It improves the calculation speed of subsequent clustering and solves the problems that PCA does not adapt to non-Gaussian distribution data and principal components can not be explained. Compared with the traditional random forest algorithm, it also significantly improves the speed and accuracy of dimension reduction; The model is ulteriorly simplified by further implementing Blending on XGBoost.

(4) In practical engineering applications, it is sometimes necessary to make the wind farm equivalent to a fixed multi-machine model, which requires the model widely applicable under multiple operating conditions. Therefore, this paper proposes a multi-machine equivalent clustering method based on a clustering ensemble of grouping results.

II. Select Clustering Indicators Based on XGBoost-Blending

In order to avoid over-fitting and dimensional disasters, it is necessary to intuitively display the importance ranking of clustering indexes to increase the running speed and remove irrelevant interference. Compared with PCA, the results of XGBoost's is more explicable, and the impact of different features on the results are visible. PCA is a simple data variance and relies on experience when selecting the number of principal components, while XGBoost can modify the process through prior knowledge.

A. Extreme Gradient Boosting Tree

XGBoost in this paper is an integrated algorithm based on the classification and regression tree (CART) model. The basic structure of CART is shown in Figure 1, and the principle is derived from [16].

The process of building CART is as follows:

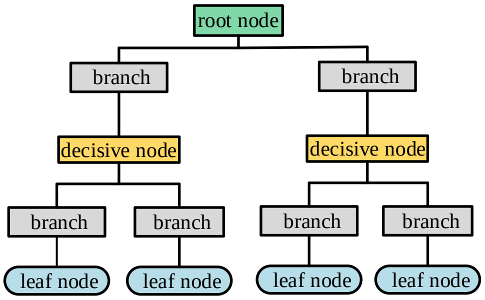


Fig.1 CART basic structure

(1) Looking for the split point: starting from the root node (sample set) traverse the features and calculate the gain of the model when the sample is divided as the threshold. (2) Division and cessation: record the corresponding feature and its gain when it reaches maximum, and divide the tree into left and right nodes according to the node value, until the gain of all split points is less than 0. (3) Results and scores: each leaf node in the last layer corresponds to a sample subset (the final classification result), and the gain of each feature can be calculated according to the previous step. Thus, the importance of each feature can be directly obtained.

The overall structure of the algorithm is as follows:

$$\hat{y}_i = \phi\left(x_i\right) = \sum_{k=1}^{K} f_k\left(x_i\right) \tag{4}$$

where *i* refers to the *i* -th sample, input this data x_i to get the estimated value $\hat{y}_\mathrm{i}$, but the function f_k trained by the *k*-th tree. The overall spatial mapping relationship is the sum of the functions generated by the *k*-th tree.

To measure the advantages and disadvantages of the function, this paper defines the objective function and transforms it into an optimization problem. Since the results of the training data are known, the goal is to make the predicted value and the true value as close as possible, but also to ensure that the model is not too complicated to avoid overfitting. The objective function is expressed as:

$$Y(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)}\right) + \Omega(f_t) + c \tag{5}$$

where L is the training error, that is, the loss function of the predicted value and the real value, which represents the matching degree of the model to the training set; y_i is the true value; $\Omega(f_t)$ the regular term represents the complexity of the model; c is a constant term.

For regression problems, MSE is the most popular loss function, as (6) shows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(t-1)}\right)^2 \tag{6}$$

XGBoost adds a regular term to the objective function to avoid overfitting. The regular term defines the L1 norm and L2 norm by constructing the parameters of the leaf nodes of the new tree:

$$L1\text{:}\ \ \Omega(w) = \lambda\|w\|_1 \tag{7}$$

$$L2\text{:}\ \ \Omega(w) = \lambda\|w\|^2 \tag{8}$$

where $\Omega(w)$ is the norm; w is the score set of the leaf nodes; λ the score of the leaf nodes can be controlled not to be too large, that is, to prevent over-fitting.

In this paper, L2 norm is used to define the regular term:

$$\Omega(f_t) = \mu W + \frac{1}{2}\lambda\|w\|^2 \tag{9}$$

where, W represents the number of leaf nodes of each tree, μ and λ both are hyper-parameters.

The algorithm in this paper adds a function for each round of training, which is to reduce the residual error. In each epoch a new model is established in the direction of the gradient of the residual reduction.

Pre-setting the model requires training for t rounds, and the process of determining the final function is as follows:

$$\begin{aligned}
&\hat{y}_i^{(0)} = 0\\
&\hat{y}_i^{(1)} = f_1\left(x_\mathrm{i}\right) = \hat{y}_i^{(0)} + f_1\left(x_\mathrm{i}\right)\\
&\hat{y}_i^{(2)} = f_1\left(x_\mathrm{i}\right) + f_2\left(x_\mathrm{i}\right) = \hat{y}_i^{(1)} + f_2\left(x_\mathrm{i}\right)\\
&\ldots\ldots\\
&\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k\left(x_\mathrm{i}\right) = \hat{y}_i^{(t-1)} + f_t\left(x_\mathrm{i}\right)
\end{aligned} \tag{10}$$

In the new round, a new function $f_t(x_i)$ is added to minimize the objective function. In the t-th round, the objective function is:

$$Y(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + c \tag{11}$$

Then it applies Taylor expansion to make an approximation：

$$f\left(x+\Delta x\right) = f\left(x\right) + f\,'\left(x\right)\Delta x + \frac{1}{2}f\,''\left(x\right)\Delta x^2 \tag{12}$$

$$g_i = \partial_{\hat{y}^{(t-1)}} L(y_i, \hat{y}^{(t-1)}), h_i = \partial^2_{\hat{y}(t-1)} L(y_i, \hat{y}^{(t-1)}) \tag{13}$$

where $f(x)$ corresponds to the newly added function; $f'(x)$ is the derivative of $f(x)$; g_i is the derivative of training error; $f''(x)$ is the second derivative of $f(x)$; h_i is the second derivative of training error.

Define:

$$G_j = \sum\nolimits_{i\in I_j} g_i \tag{14}$$

$$H_j = \sum\nolimits_{i\in I_j} h_i \tag{15}$$

The new objective function is expressed as:

$$\begin{aligned}Y(t) = \sum_{i=1}^{n}&\left[L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^{\,2}(x_i)\right]\\ &+ \Omega(f_t) + c\end{aligned} \tag{16}$$

$$\tilde{Y}(t)=\sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\mu W+\frac{1}{2}\theta\sum_{j=1}^{W}m_j^2$$
$$=\sum_{j=1}^{W}\left[(\sum_{i\in I_j}g_i)m_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\theta)m_j^2\right]+\mu W \quad (17)$$

where $\tilde{Y}(t)$ is the objective function of each lifting tree; θ is the hyperparameter of leaf node prediction score; m_j is the prediction score of leaf node; I_j is the sample space corresponding to the j th leaf node.

The above is the objective function of the t-th step, that is, the above-mentioned function is minimized during the t-th training round, which significantly speeds up the calculation, enables more rounds of calculation, and reduces the loss function more.

The purpose of each iteration of the node is to fit the residual error of the error function to the predicted value, and then build a new model on the negative gradient. To minimize the objective function, let the derivative be 0, and the minimum prediction value of the node is as follows:

$$m_j^*=-\frac{\sum_{i\in I_j}g_i}{\sum_{i\in I_j}h_i+\theta} \quad (18)$$

Substituting m_j^* into the objective function, the minimum loss value is:

$$\hat{Y}^{(t)}(s)=-\frac{1}{2}\sum_{j=1}^{W}\frac{(\sum_{i\in I_j}g_i)^2}{\sum_{i\in I_j}h_i+\theta}+\mu W \quad (19)$$

where $s(x)$ is the sample on a certain node, and m is the predicted value of the node.

B. *Tree generation*

Step 1: For the first tree, traverse each feature value of each wind turbine sample, divide the original samples into two parts by the value, and calculate the MSE of the two sets respectively. Find the minimum MSE sum of the left and right sets in all values, record the value and the corresponding feature, according to which divide the tree into left and right nodes.

Step 2: Repeat the above step until the average gain G of all possible segmentation points is negative, and the model ends.

Step 3: After endding the division, the last layer is called the child nodes of the tree (each node is a set), and the average of the current predicted values of the samples falling in this child node is the classification result.

The square error is generally used as the loss function for the general regression tree. The overall optimization can be guaranteed if each time the optimization can be achieved, according to the forward distribution. Due to the particularity of the square error, it can be deduced that only the residuals (real value-predicted value) are needed to be fitted each time, so the input of generating the second tree is the residual.

C. *Blending*

The ensemble of meta-models can not only enhance the learning effect, but also avoid excessive redundancy of the overall model. Recently, it has been widely employed in solving prediction problems, especially the stacking method[17]. However, due to the complexity of stacking, there will be a data traversal problem that training data refer to global statistics in the training process, which is not suitable for solving the dimension reduction problem.

In view of the above defect, Blending introduced in this paper has the advantages of simplicity and overcoming data traversal. The steps of XGBoost meta-model ensemble are shown in Figure 2.

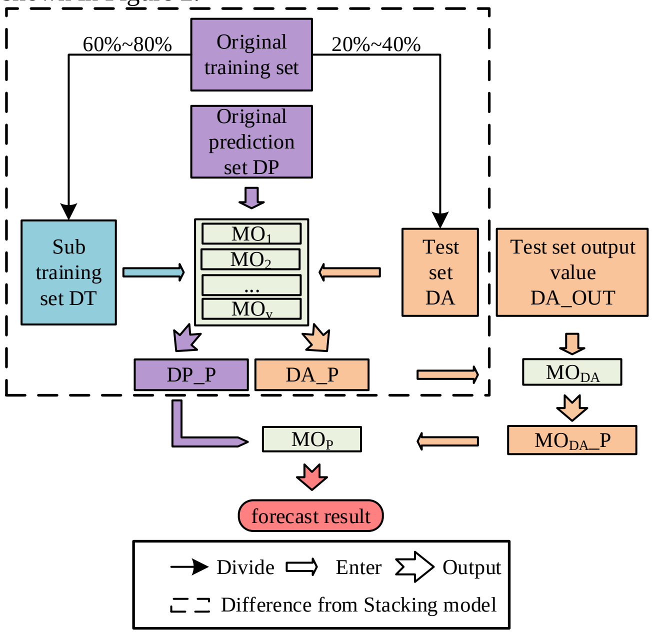


Fig.2 Schematic figure of Blending steps

(1) Segmentation of the original data set

The original training set is divided into sub-training set *DT* and test set *DA*, and the original prediction data set is defined as *DP*.

(2) Ensemble

Construct V XGBoost meta-models *MO*$_1$、*MO*$_2$、…、*MO*$_V$， exploit these meta-models to learn *DT*. After completing training, output the index contribution results of *DA_P* and *DP_P* from *DA* and *DP* on the meta-model.

Combine *DA_P* and the actual result *DA_OUT* corresponding to the original *DA* data to form a new data set. Establish a new meta-model *MO*$_{DA}$ for training, and obtain the indicator contribution degree output *MO*$_{DA}$*_P*.

A new data set is composed of *MO*$_{DA}$*_P* and *DP_P*, and construct a new meta-model *MO*$_P$ for training, so as to output the final indicator contribution and complete the dimensionality reduction.

D. *Steps for the algorithm of wind turbine feature selection*

Based on the Blending XGBoost constructed above, the steps of wind turbine feature extraction are as follows:

(1) Input the sample *x* of the wind turbine and the loss function *L*.

(2) Utilize the greedy algorithm to build a tree and learn a new function to fit the residual of the last prediction.

(3) Iteratively train *L* to minimize the error.

(4) Define the regular term, calculate the complexity, and divide the tree into two parts: structure *s* and weight *m*.

The objective function $Y(t)$ is transformed into $\tilde{Y}(t)$:

$$\tilde{Y}(t)=\sum\nolimits_{j=1}^{W}\left[G_j m_j+\frac{1}{n}(H_j+\theta)m_j^{\,2}\right]+\mu W \qquad (20)$$

(5) After node iteration, the optimal value of the loss difference before and after node segmentation is obtained, which can be used to calculate the average gain of features to denote the feature importances. Then choose relatively more important features to achieve dimensionality reduction:

$$G=\frac{1}{2}\left[\frac{G_L^{\,2}}{H_L+\theta}+\frac{G_R^{\,2}}{H_R+\theta}-\frac{(G_L+G_R)^2}{H_L+H_R+\theta}\right]-\mu \qquad (21)$$

where $\frac{G_L^{\,2}}{H_L+\theta}$ represents the score of the left sub-tree; $\frac{G_R^{\,2}}{H_R+\theta}$ represents the score of the right sub-tree; $\frac{(G_L+G_R)^2}{H_L+H_R+\theta}$ represents the score obtained without splitting; μ represents the complexity of the new node.

(6) Repeat the above steps until enough trees are generated to make the predicted value closest to the real value, and the algorithm ends.

This paper applies Blending to the above algorithm to reduce the dimension of clustering indexes. The larger the gain of each node splitting means the larger difference between the model before and after splitting, that is, the more the gradient of the objective function decreases and the closer to the optimal solution of the objective function. Therefore, when splitting each feature, the *G* value is recorded, and the total *G* value is divided by the number of times the feature is used to split the node to obtain a quantitative score of the contribution degree of the feature. According to the order of contribution from low to high, the features are deleted one by one and re-clustered. After traversal, the index combination corresponding to the clustering result with the highest contour value is obtained.

III. Improved DBSCAN Clustering Based on DTW

This section introduces the similarity calculation method based on DTW and Euclidean distance as well as DBSCAN. Compared with the traditional distance and density, DTW eliminates the influence of different dynamic response time of the wind turbine (non-aligned time series), and it improves the model significantly; Compared with traditional clustering methods such as k-means, DBSCAN does not need to predetermine the number of groups, and it is not sensitive to noise. Besides, it can identify non-spherical data sets (the clustering results of other methods are all spheres). Facing massive data, its training speed is faster and it can eliminate the influence of irregular shape and noise of the data set on the clustering results, and accurately identify the offline fans.

A. *Error Analysis of Calculating Similarities between fans Based on Traditional Euclidean Distance*

Define two sequences *P* and *Q* are with lengths *m* and *n* respectively. When *m* = *n*, Euclidean distance can represent the distance between these two points, as shown in(22).

$$D(E)=\sqrt{\sum_{i=1}^{m}\left(p_i-q_i\right)^2} \qquad (22)$$

Due to the mismatching of different wind turbines in the recovery characteristics after faults, and the traditional Euclidean distance adopted as the similarity bases on the dynamic response of different wind turbines at the same time point, the clustering error is large, and the equivalent units can not accurately simulate the detailed dynamic behaviors of the wind farm. To verify the above theory, this paper leverages Matlab / Simulink simulation platform to build a wind farm composed of 30 DFIGs with a rated power of 1.5MW.

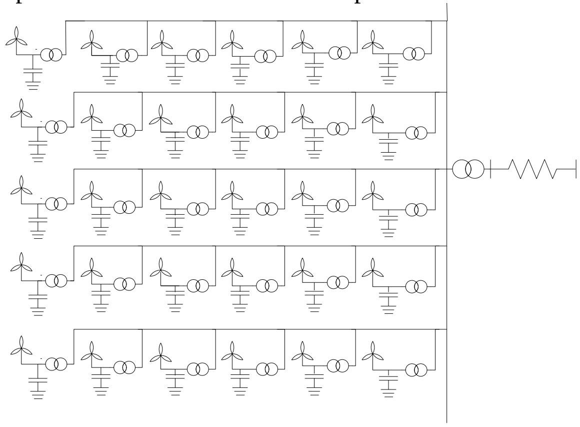


Fig.3 Wind farm simulation figure

The terminal voltage of DFIG is 575v, which is locally increased to 25kV by the unit wiring mode of one machine and one transformer. Every six transformers in the field are connected by overhead lines and transmitted to the 25kV / 220kV substation and the external power grid. Taking the active power of fan 1 and fan 2 as an example, the three-phase short circuit fault occurs at the parallel point of the wind farm at the 12s, and the fault is cleared at the 12.1s.

As shown in Figure 4, when the fault occurs, the active power of the wind turbine will drop. After a short-term transient (tens of milliseconds), the steady-state value during the fault is quickly reached. After the fault is cleared, the active power of the fan undergoes downward and upward overshoots for a certain period, and the active power returns to the normal working state at a certain slope after a short-term transient state (tens of milliseconds). Therefore, for system-level analysis, especially in electro-mechanical transient analysis, the dominant dynamics of wind turbines concerned mainly include: the transient steady state during the duration of the fault and the power recovery process after the fault is cleared.

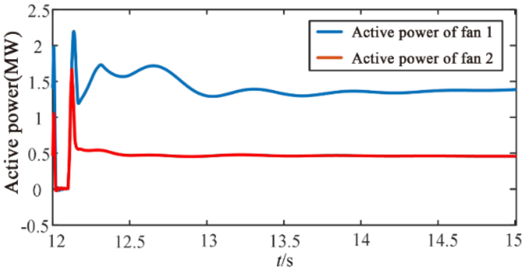


Fig.4 Simulation diagram of active power transient response

The difference in the active power leading dynamic behavior of wind turbines operating at different wind speeds after the fault is cleared is as follows: the initial recovery power of wind turbines operating at different wind speeds is different; although the wind turbines working at different wind speeds all return to the normal working state according to a certain speed, the wind turbines with low wind speed can quickly reach the steady-state value, while the wind turbines with high wind speed need a long time under the condition of the same set speed.

At 12.5s, fan 2 has reached a steady-state, while fan 1 is still recovering. The similarity of the transient response of thees fans is smaller and the similarity of the steady-state response of thees fans are too larger if they are calculated based on the traditional Euclidean distance. It requires dividing the periods corresponding to transient and steady-state responses manually.

B. *DTW: Calculating the Similarity of fans*

Due to the influence of wind direction, topography and wake flow, the wind speed of each power unit in the same wind farm is different. Even for the same type of wind turbine, its dynamic response time is also different. Due to the structure of the internal collector network of the wind farm, the actual faults received by different wind turbines during the fault period are different. For example, when the wind farm outlet fails, some wind turbines may experience low voltage ride through, while the other part will not be affected. In these complex situations, the operating data of different wind turbines are not aligned in the time series. Even if the Euclidean distance is exploited in the DBSCAN clustering algorithm, the similarity measured by the distance between these two time series cannot be effectively obtained. Therefore, this paper applies DTW to calculate the similarity of time series data.

DTW is an important method to measure the similarity between two sequences of different length[18]. The core of the algorithm is that it can break the restriction of inconsistent sequence length. By extending and shortening the time series with constraints and pruning, DTW finds the optimal path to calculate the similarity between the two time series. Th sum of Euclidean distances between all aligned points is called warp path distance (WPD), and the minimum WPD measures the similarity between the two time series, as (23) shows.

$$\left\{\begin{array}{c} D(0,0)=0 \\ D(i,j)=d(p_i,q_j)+\min\left\{\begin{array}{c} D(i-1,j-1) \\ D(i-1,j) \\ D(i,j-1) \end{array}\right\} \end{array}\right\} \quad (23)$$

Suppose the sequence-regulated path is R， k denotes the length of the path, then the regular path is $R=\left(r_1,r_2,\cdots r_k\right)$, and the distance function of R is $Y(k)=\min\sum_{i=1}^{k} r_i$. The defined regular path must satisfy the following constraints: (1) Boundary: the starting and ending point of the two sequences P and Q must correspond, i.e. $R_1=(1,1)$, $R_k=(m,n)$. (2) Monotonicity: The points on the sequence P and Q must be monotonic so that the two sequences will not intersect. (3) Continuity: The points in the sequence can only be matched with adjacent points, and cannot be matched by strides, namely $0 \le i-i' \le 1$.

The time-series features of wind turbines selected in this paper have reached 13 dimensions, including electrical and mechanical characteristics. The electromagnetic torque feature is taken as an example to compare DTW distance and traditional Euclidean distance, as shown in Figure 5. In the process of fault recovery, DTW obtains the alignment path between the wind turbine characteristics through the transformation of the time axis.

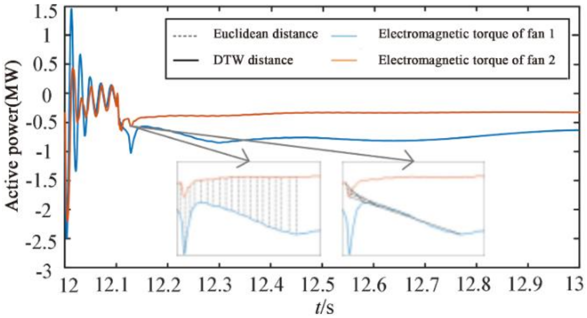


Fig.5 Similarity between wind farm 1 and wind farm 2

C. *Error Analysis of K-means*

Sources of errors:

(1) Noise: for each iteration, k-means must calculate the mean value of all points in the cluster, so if there is an outlier point extremely far from the other points, it will greatly affect the location of the centroid, which in turn affects the division of clusters. If there is a fan whose wind speed is significantly different from other fans in the wind field, the abnormal fan may be wrongly classified, which will affect the subsequent equivalence problem. For example, the clustering data set is x = [1, 1.1, 1.2, 1.2, 1.45, 1.38, 1.68, 1.72, 1.75, 3, 3.3, 3.15, 3.2, 4, 3.9, 4.1, 4.15], y = [1, 1.1, 0.9, 1.25, 1.35, 1.55, 1.5, 1.65, 1.68, 3, 3.1, 3.05, 2.9, 4, 4.1, 3.9, 3.95], and the noise point is (2, 2.5).The clustering results before adding noise points are shown in Figure 6. One color represents one category. When the noise points are added, the result of clustering changes. The algorithm classifies the noise points into cluster 2 instead

of identifying the noise points as expected, and the two points in the box that should belong to cluster 2 are wrongly classified into cluster 1. If the samples are distributed more scattered, this phenomenon of centroid shift will be more obvious. For example, off-grid wind turbines are equivalent to noise points, and k-means clustering cannot identify off-grid wind turbines, and it even affects the centroid of clusters and causes shifting clusters, which seriously affects the accuracy. Figure 7 shows the k-means clustering after adding noise.

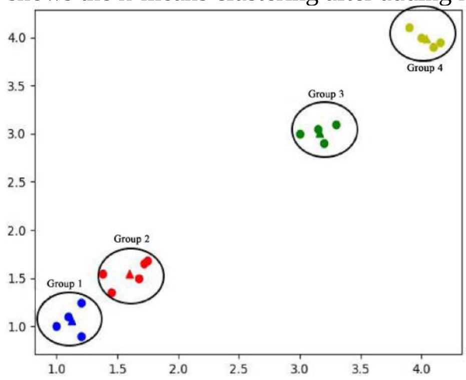


Fig.6 K-means clustering before adding noise

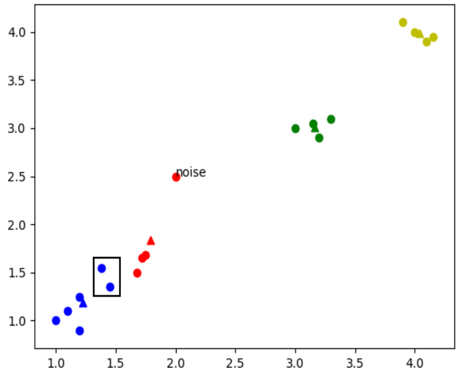


Fig.7 K-means clustering after adding noise

(2) Convex data set: k-means is based on Euclidean distance, which essentially divides the sample space into *k* Voronoi regions, so the obtained clusters are all convex, making its application range limited, and it is difficult to find some non-convex properties in the data set. However, in the wind farm case of this study, the data set is not strictly convex.

D. DBSCAN Division of Clusters

The similarity between multi-dimensional time series obtained by DTW needs to be substituted into DBSCAN for clustering. The algorithm divides regions with sufficient density into clusters, and finds clusters of arbitrary shapes in a noisy spatial database. It defines clusters as the largest set of connected points with a density.

Different from distance-based clustering algorithms, there are two hyper-parameters that require manual intervention, namely *Epsilon* and *MinPts*. *Epsilon* represents the clustering radius and *MinPts* represents the minimum number of samples in a category. *Epsilon* and *MinPts* can divide all samples into three categories: (1) Core point: select a point *M* in the sample, and set *N* to be the number of samples with an attainable density. When $N \geqslant MinPts$, point *M* is the core point; (2) Border point: the point that is not the core point but falls in the neighborhood of the core point; (3) Noise point: if point *Q* and any core point do not meet the density reachability, then *Q* is a noise point, as shown in Figure 8. After setting the two parameters, a point in the sample can be arbitrarily selected as the core point, and all the samples satisfying the condition of density can be found as a category to ensure that all the points in the category are in the epsilon neighborhood. Let the sample set be *E* and select a point *m* randomly, as (24) defines. The algorithm flow is shown in Figure 9.

$$Epsilon(m,M)=\left\{m\in E \mid d(m,M)\le Epsilon\right\} \quad (24)$$

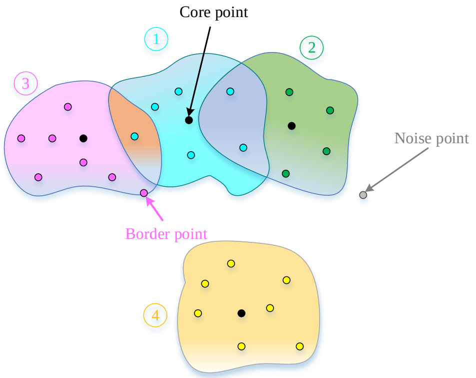


Fig.8 Clustering schematic

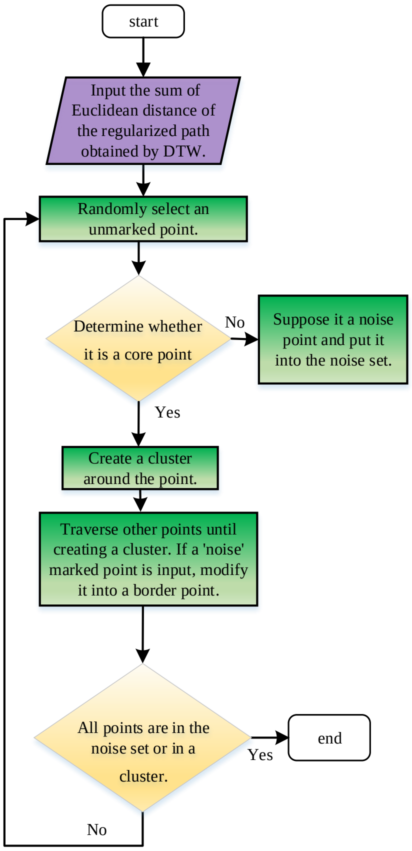

start
Input the sum of Euclidean distance of the regularized path obtained by DTW.
Randomly select an unmarked point.
Determine whether it is a core point
No
Suppose it a noise point and put it into the noise set.
Yes
Create a cluster around the point.
Traverse other points until creating a cluster. If a 'noise' marked point is input, modify it into a border point.
All points are in the noise set or in a cluster.
Yes
end
No

Fig.9 DBSCAN clustering flowchart

Comparing to the above k-means experiment, clustering the same data set with DBSCAN achieves awesome robustness. Adding noise does not affect the clustering result, and the algorithm will automatically identify the noise points, as shown in Figure 10 and Figure 11.

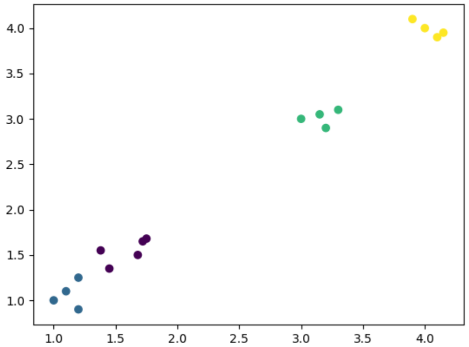


Fig.10 DBSCAN clustering before adding noise

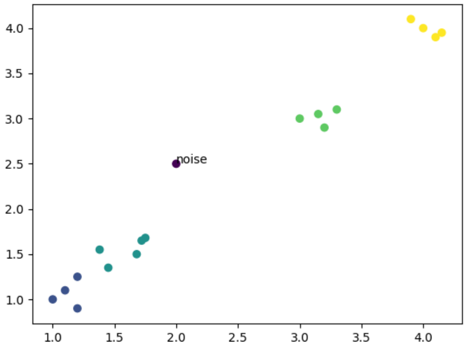


Fig.11 DBSCAN clustering after adding noise

For example, grid faults usually cause voltage drops, causing that the wind turbines are disconnected to prevent transient over-current from damaging their internal electronic components. When the off-grid wind turbines reach a certain scale, it will seriously affect the stability of the system. To solve the problem of large-area disconnection of wind turbines under grid faults, the units with similar low voltage ride through (LVRT) capacity can be equivalent to a group. Through the coordinated control of the same group of units, it is not only convenient for the power dispatching department to unify the management of large-scale wind farms, but also can realize the optimal configuration of reactive power compensation devices. The LVRT bearing capacity of off-grid wind turbines is poor, and DBSCAN clustering can dynamically identify off-grid wind turbines, which improves the accuracy and reliability of clustering results.

IV. The Equivalent Value of Wind Turbine Clusters

In this paper, each wind turbine group is equivalent according to the capacity weighting method and the principle of constant loss. The equivalent capacity of each cluster is the sum of the capacity of all wind turbines in the cluster.

$$S_G=\sum_{i=1}^{m_f}S_i \quad (25)$$

where S_G is the capacity of the equivalent generator; S_i is the capacity of generator i in the cluster; m_f is the number of fans in the cluster. Suppose the equation of motion of the rotor of the i-th generator is:

$$T_{Ji}\frac{d\omega_i}{dt}=P_{mi}-P_{ei}-D_i(\omega_i-1) \quad (26)$$

where parameters are standard unit values (based on the rated capacity of the generator), which can be transformed into (27).

$$\left(T_{Ji}\frac{d\omega_i}{dt}\right)\frac{S_i}{S_G}=\left[P_{mi}-P_{ei}-D_i(\omega_i-1)\right]\frac{S_i}{S_G} \quad (27)$$

By adding the equations of generators in the cluster, the following results can be obtained:

$$\sum_{i=1}^{m_f}\left(\frac{S_i}{S_G}T_{Ji}\right)\frac{d\omega}{dt}=\sum_{i=1}^{m_f}\left\{\frac{S_i}{S_G}\left[P_{mi}-P_{ei}-D_i(\omega_i-1)\right]\right\} \quad (28)$$

If the rotor motion equation of the equivalent machine is:

$$T_{JG}\frac{d\omega}{dt}=P_{mG}-P_{eG}-D_G(\omega-1) \quad (29)$$

where parameters are the unit value (based on the equivalent capacity). The following equations can be got:

$$T_{JG}=\sum_{i=1}^{m_f}\frac{S_i}{S_G}T_{Ji}=\frac{\sum_{i=1}^{m_f}S_iT_{Ji}}{\sum_{i=1}^{m_f}S_i} \quad P_{mG}=\sum_{i=1}^{m_f}\frac{S_i}{S_G}P_{mi}=\frac{\sum_{i=1}^{m_f}S_iP_{mi}}{\sum_{i=1}^{m_f}S_i}$$

$$D_G=\sum_{i=1}^{m_f}\frac{S_i}{S_G}D_i=\frac{\sum_{i=1}^{m_f}S_iD_i}{\sum_{i=1}^{m_f}S_i} \quad P_{eG}=\sum_{i=1}^{m_f}\frac{S_i}{S_G}P_{ei}=\frac{\sum_{i=1}^{m_f}S_iP_{ei}}{\sum_{i=1}^{m_f}S_i} \quad (30)$$

where: T_{JG} is the moment of inertia; D_G is the damping coefficient; P_{mG} is the mechanical power; P_{eG} is the electromagnetic power.

The equivalent formula of line impedance is as follows:

$$Z_{\mathrm{eq}}=\frac{\sum_{i=1}^{m_f}\left(\sum_{k=1}^{i}\left(Z_k\sum_{j=k}^{n_f}P_j\right)P_i\right)}{\left(\sum_{i=1}^{m_f}P_i\right)^2} \quad (31)$$

where: the number of wind turbines in the trunk-line fan branch of the wind farm n_f; Z_k is the impedance of the k-th branch in the trunk-type branch.

For line equivalent ground admittance:

$$Y_{\mathrm{eq}}=\sum_{i=1}^{m_f}Y_i \quad (32)$$

The equivalent capacity and impedance of the transformer directly connected to the fans in the group are as follows:

$$\begin{cases}S_{Teq}=\mathrm{m_f}S_T\\ Z_{Teq}=Z_T/\mathrm{m_f}\end{cases} \quad (33)$$

where: S_{Teq} is the equivalent capacity of the transformer; Z_{Teq} is the equivalent impedance of the transformer.

In the engineering application of safety analysis, the degree of influence of the equivalent error on the safety of the power grid is not only the magnitude of the error relative to its true value. Therefore, the relative e_r is used to represent the equivalent error, and the specific expression is as follows:

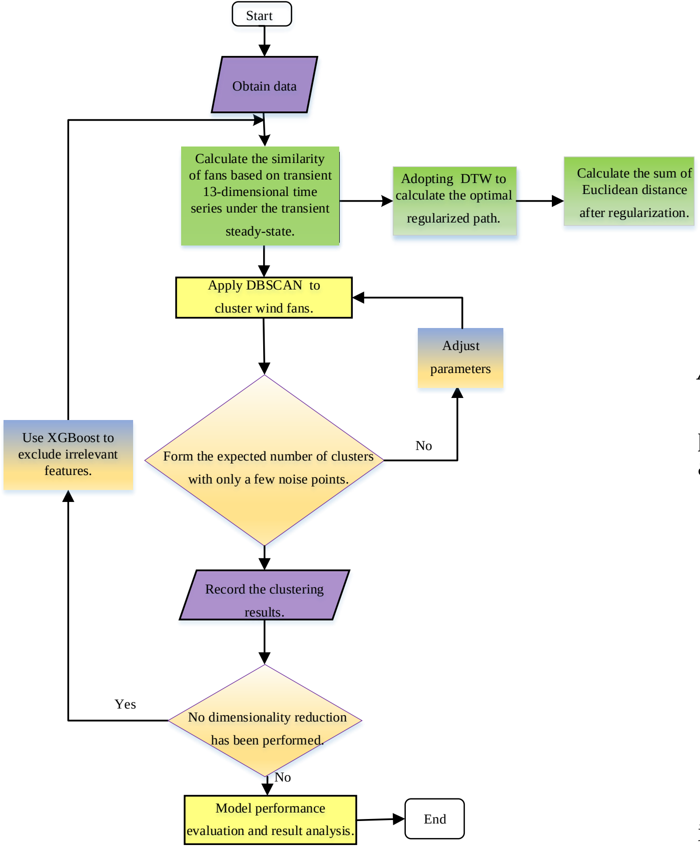
$$e_{\mathrm{r}} = \left| \frac{x - x^{eq}}{x_b} \right| \times 100\% \qquad (34)$$

where: x and x^{eq} represent the true value and estimated value.

V. Dimensionality Reduction and Index Selection Based on XGBoost-Blending & Fan Clustering Based on DBSCAN-DTW

A. Data Acquisition

To verify the equivalent effect after clustering, this paper establishes a DFIG wind farm model through Matlab / Simulink. It sets a three-phase short-circuit fault at the outlet of the wind farm during a certain period, and obtains the 13-dimensional characteristic time series data of the transient steady state of each wind turbine. Among them, the data feature is 13-dimensional of each wind turbine, including 4 mechanical indexes: rotor angular speed *wr*, pitch angle *Pitch*, electromagnetic torque *Tem*, mechanical torque *Tm*; and 9 electrical indexes: stator voltage *Vs*, active power *P*, and reactive power *Q*, rotor voltage d-axis component *Vrd*, rotor voltage q-axis component *Vrq*, stator current d-axis component *Isd*, stator current q-axis component *Isq*, rotor current d-axis and q-axis component *Ird*, *Irq*.



Start
Obtain data
Calculate the similarity of fans based on transient 13-dimensional time series under the transient steady-state.
Adopting DTW to calculate the optimal regularized path.
Calculate the sum of Euclidean distance after regularization.
Apply DBSCAN to cluster wind fans.
Adjust parameters
Use XGBoost to exclude irrelevant features.
Form the expected number of clusters with only a few noise points.
No
Record the clustering results.
Yes
No dimensionality reduction has been performed.
No
Model performance evaluation and result analysis.
End

Fig.12 Algorithm flowchart

B. Algorithm Flow

The algorithm is implemented by python, and the flowchart is shown in Figure 12. DTW is employed to calculate the optimal integration path between fans, and the similarity is calculated according to the Euclidean distance after integration, which is used as the clustering basis of DBSCAN. Then continuously adjust the parameters to obtain the optimal clusters. At this time, the calculation dimension is extremely high, and there may be a strong correlation and redundancy between features. Therefore, Blending and XGBoost are applied to select clustering indicators to achieve feature dimensionality reduction. Next, re-process DBSCAN-DTW clustering, and output clustering results.

C. Model Validation

The model is tested under various wind speed scenarios, including gusts, gradual winds and integrated winds, as shown in Figure 13. The scheme of model validation is as follows: the wind farm is equivalent according to the clustering results and the equivalent model is built by Matlab / Simulink; the dynamic response (active power, reactive power, voltage) error of the equivalent model and the actual model is calculated, and compared that with the traditional k-means clustering error.

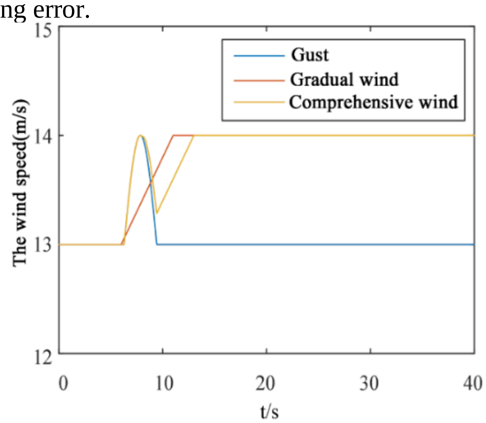


Fig.13 Different wind speed disturbance scenarios

VI. Analysis of Cases

A. Introduction to calculation examples

This paper utilizes MATLAB / Simulink simulation platform to construct a wind farm composed of 16 DFIGs with a rated power of 1.5MW, as shown in Figure 14.

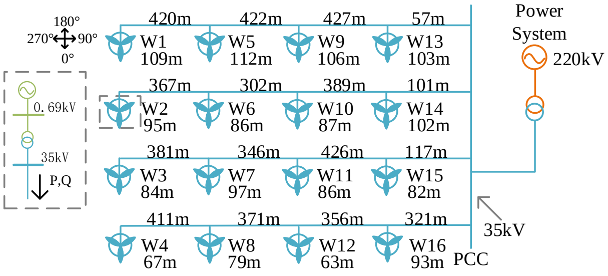


Fig.14 Schematic diagram of a wind farm

The terminal voltage of DFIG is 690V, which is locally increased to 35kV by the unit connection mode of one machine and one transformer, and then transmitted to 35kV /

220kV substation and external power grid through overhead lines. The parameters of DFIG and transformers are in [19]. Suppose the wind direction is 0° and the wind speed before reaching this wind field is 13m/s. Calculate the wind speed at different locations based on the wind speed wake effect.

Run the simulation model. When all DFIGs enter the steady-state, perform the following experiments. Standardize and normalize the 13-dimensional time-series data obtained in each experiment.

Experiment 1: A three-phase short-circuit fault occurs at the outlet of the wind farm in 12s, and the fault lasts for 100ms. Sample from 11s to 17s, with a 50us step size.

Experiment 2:

(1) Gust disturbance: based on experiment 1, program the s_function module in Simulink and add a sine half-wave disturbance wind speed component to each basic wind speed to simulate the disturbance of gusts. By changing the occurrence of gusts, set the simulation duration to 40s, and set the gust disturbance duration to π s. Three sets of data are measured for data analysis, as shown in Table 1.

Tab.1 Gust disturbance experiment disturbance component parameter

Experiment group number	Amplitude (m/s)	Moment of occurrence (s)
1	1	3π
2	1	2π
3	1	π

(2) Progressive wind disturbance. Truncate the basic constant wind speed, that is, set the time of occurrence of progressive wind. The progressive process is simulated with a proportional function. After a certain period of time, it grows to another constant wind speed, completing the gradual wind simulation. Via setting the time of occurrence of progressive wind, set the simulation duration to 40s, and the duration of progressive wind disturbance to 5s. Two data sets are measured for analysis, as shown in Table 2.

Tab.2 Disturbance component parameter of progressive wind disturbance experiment

Experiment group number	Amplitude (m/s)	Moment of occurrence (s)
1	1	6
2	1	12

(3) Comprehensive wind disturbance. Set gust to occur at 2π s and lasts for π seconds, with an amplitude of 1m / s; set gradual wind occurs at 8s and lasts for 5s.

Experiment 3: by adjusting the wind speed, multiple groups of data are collected from the above four experiments. Implement Blending on the contribution of multiple groups of XGBoost indexes. Utilize the fused clustering results of multi-components to build an equivalent model of the wind farm.

B. Software and Hardware Platform Configuration

The proposed algorithm is realized by python, and the Linux server is for model training. The server configuration is in the appendix. In terms of software configuration, this example uses the machine learning library Sklearn, Vim integrated development editor, Anaconda environment management software and other software suitable for python.

C. Experiment 1

1) Selection and Dimensionality Reduction of Clustering Indexes

This experiment applies XGBoost-Blending and random forest (RF) to reduce the dimension of 13-dimensional DFIG time-series features collected before, and the order of index contribution is shown in Table 3 in the appendix. The training time of XGBoost-Blending is 3.53s, and the training time of RF is 52.85s. The training time of XGBoost-Blending is much shorter than that of RF, which verifies the significant advantages of XGBoost-Blending's fast training speed. After using XGBoost-Blending to reduce the dimensionality, the contour value of the scheme with the highest contour value is 0.954, and the corresponding index combination is: *Tem*, *Vrq*, and *Ird*; After RF reduces the dimension, the highest contour value is 0.897, and the corresponding indexes are: *Tem*, *Ird*, *Irq*. The contour value of the former is obviously higher than that of the latter. Since the contour value of outliers [19] is set to be 0 in this paper, the contour value index is actually an optimization target with fewer outliers and high clustering similarity of DFIG comprehensively, indicating that XGBoost-Blending can select the clustering indexes more accurately.

2) Division of Wind Turbine Clusters

The Euclidean distance of the regular paths obtained by DTW represents the similarity for clustering. Set the maximum search radius as the maximum similarity between the turbines, and traverse all high-dimensional space points to filter the solution space. Table 3 shows the results of cluster division by k-means, DBSCAN-DTW, DBSCAN-DTW after dimension reduction of RF, and DBSCAN-DTW after dimension reduction of XGBoost-Blending. From Table 3:

(1) The clustering time of DBSCAN-DTW is much shorter than that of k-means, and the contour value of DBSCAN-DTW is also significantly higher than that of k-means, indicating that the clustering speed and accuracy of DBSCAN-DTW are much higher than that of k-means.

(2) Both XGBoost-Blending and RF reduce the index dimension from 13 to 3, which improves the calculation speed, filters out noise and redundant data, and significantly improves the contour value. Xgboost-Blending not only reduces the dimension faster than RF, but also obtains higher contour value with corresponding clustering index combination, so the model is more accurate and faster simplified.

Finally, leverage Xgboost-Blending to opt clustering indexes and DBSCAN-DTW to cluster, as shown in Figure 18.

Tab.3 Cluster Division Results

Grouping method	Grouping result	Contour value	operation hours
k-means	{1,5,9,13};{2,6,7,10,14};{3,11,15,16};{4,8,12}	0.689	0.01444s
DBSCAN-DTW	{1,5,9};{2,6,7,10,13,14};{3,8,11,15,16};{4,12}	0.854	0.00121s
DBSCAN-DTW after random forest dimensionality reduction	{1,5,9};{2,6,7,10,11,13,14};{3,4,8,12,15,16}	0.897	0.00024s
DBSCAN-DTW after XGBoost dimensionality reduction	{1,5,9};{2,6,7,10,11,13,14};{3,8,15,16};{4,12}	0.954	0.00021s

3) *Equivalent Effect Test after Clustering*

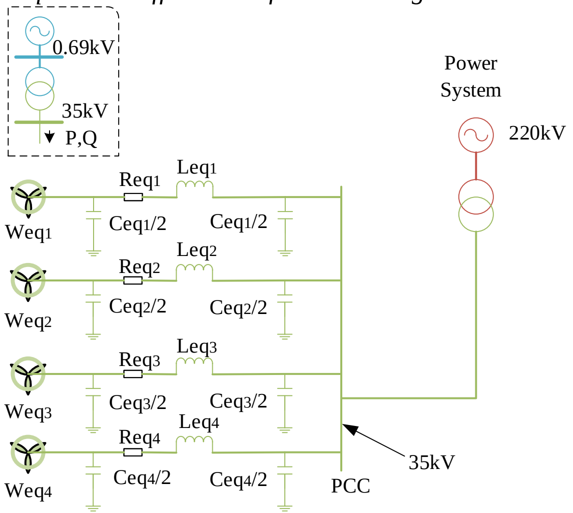


Fig.15 Schematic diagram of the four-machine equivalent wind farm

The equivalent parameters of DFIG are calculated by the capacity weighting method, as shown in Figure 15. The equivalent parameters of the internal collection network of the wind farm are obtained according to the principle of constant loss[20]. The corresponding Matlab / Simulink simulation model is constructed and compared with the export dynamic response of the original model.

To verify the DBSCAN-DTW clustering of XGBoost-Blending dimension reduction, this paper chooses benchmarks including k-means, DBSCAN-DTW and DBSCAN-DTW clustering of RF dimension reduction. The response curves of voltage, active power and reactive power at the parallel node are shown in Figure 16, 17 and 18.

This paper adopts three evaluation indexes, namely, the relative deviation of voltage, the relative deviation of active power and the relative deviation of reactive power at the junction point [21]. As shown in Table 4, compared with the equivalent model obtained by other methods, the proposed DBSCAN-DTW clustering of XGBoost-Blending dimension reduction can reduce the voltage deviation, active power deviation and reactive power deviation of the equivalent model, and significantly improve its accuracy.

Tab.4 Dynamic equivalent relative deviation comparison

Grouping method	Voltage deviation	Active deviation	Reactive deviation
k-means	0.2648%	0.2783%	15.6434%
DBSCAN-DTW	0.0836%	0.2774%	14.7549%
DBSCAN-DTW after random forest dimensionality reduction	0.1254%	0.2311%	11.5044%
DBSCAN-DTW after XGBoost dimensionality reduction	0.0696%	0.1223%	7.3521%

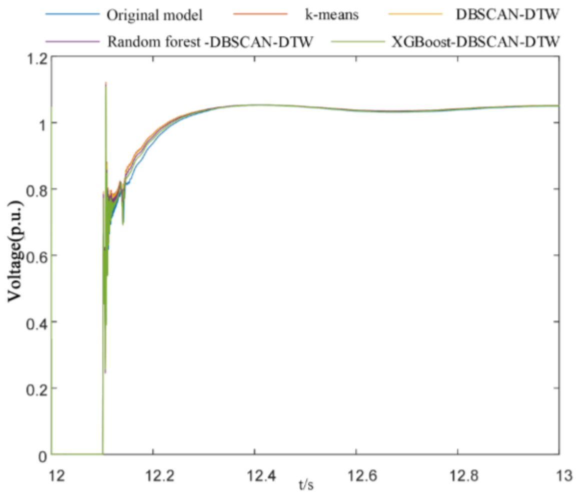


Fig.16 Voltage response curve at the grid connection point

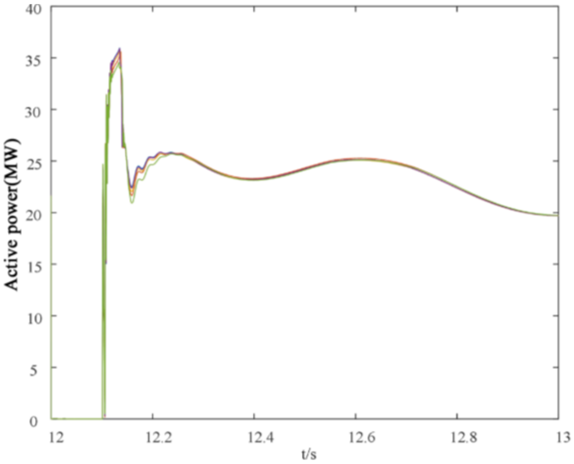


Fig.17 Active power response curve at the grid connection point

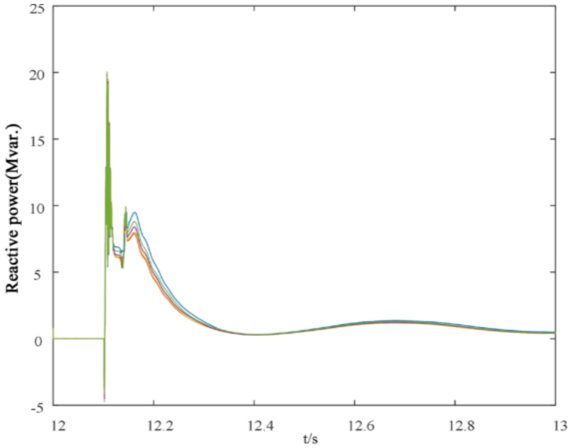


Fig.18 Reactive power response curve at the grid connection point

4) *Model Evaluation of Experiment 1*

(1) When a three-phase short-circuit fault occurs at the grid connection point, considering the different dynamic response time of the same model of DFIG under different wind speeds, and the difference is enlarged considering different electrical distances, different fault locations or different fault types. However, the premise of traditional Euclidean distance and cosine distance in the clustering algorithm is time series correspondence, so DFIG can not be effectively clustered. By adopting DTW to calculate the regularized path, it solves the above problem by reforming time series, increasing the model accuracy.

(2) Due to the randomness of the distribution of DFIG in the multi-dimensional feature space, similar DFIGs are usually irregular clusters, while traditional distance-based clustering algorithms can only form spherical clusters, which may cause abnormal points in the clustering process. Abnormal points can cause centroids shifting, thereby reducing the contour value. By applying DBSCAN to implement density-based clustering, the above problem is solved and the model is enhanced.

(3) Due to the multi-dimensional characteristics of wind farms, such as stable operation, fault occurrence and fault recovery, the traditional clustering methods will reduce clustering accuracy DBSCAN has better anti-noise ability so as to improve the model accuracy.

(4) To fully consider the characteristics of DFIG, the electrical and mechanical indicators are both selected, which are in high dimensionality and probably contain a strong correlation between the indicators and noise, which will reduce the clustering speed and accuracy. XGBoost-Blending can effectively select clustering features, improve the clustering speed, filter out the noise and redundant data so that the contour value has been significantly improved; XGBoost-Blending is not only faster than RF in dimensionality reduction, but also the contour value of the corresponding index combination is higher.

D. *Experiment 2*

Utilize the three sets of experimental data under progressive wind in experiment 2 to repeat the steps in experiment 1. The index contribution obtained by XGBoost-Blending is in Table 4 in the appendix. The clustering result gained via DBSCAN-DTW is shown in Table 5.

Under the disturbance of gust 1 and comprehensive wind, the wind farm is equivalent to a 5-machine model after clustering, as shown in Figure 19. It is also equivalent to a 4-machine model under gust 2 and gust 3. Under the condition of a three-phase short circuit fault at the wind farm outlet, the dynamic response of the wind farm outlet under different algorithms is obtained in different wind speed scenarios, as shown in Table 5 in the appendix. The results verify that the proposed algorithm has better performance under different wind speed conditions with satisfactory value accuracy and strong robustness.

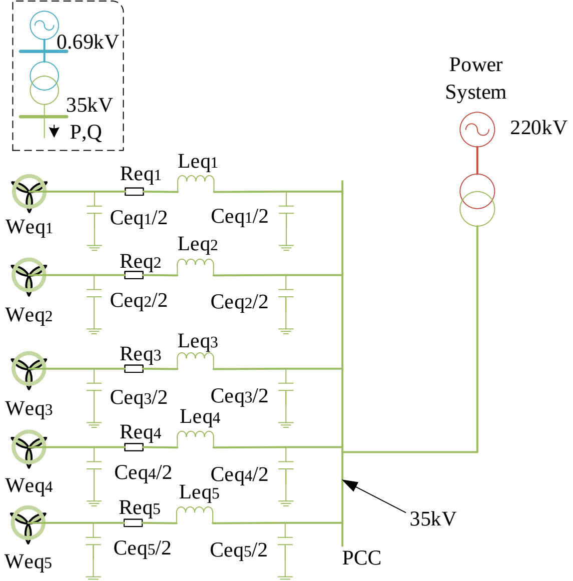


Fig.19 Schematic diagram of the five-machine equivalent wind farm

E. *Experiment 3*

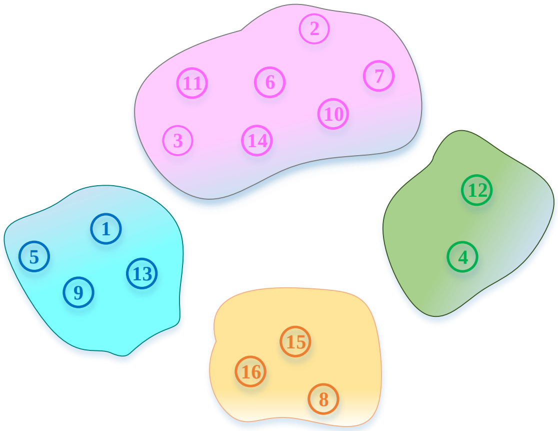


Fig.20 Schematic diagram of results obtained by the clustering ensemble

By modifying the parameters, plenty of data under wind speed scenarios are collected, and the clustering results under different conditions are fused, which are shown in Figure 20. Compared with the equivalent model gained by other methods, the DBSCAN-DTW clustering of XGBoost-Blending dimensionality reduction can make the voltage deviation, active power deviation, and reactive power deviation of the equivalent model all achieve the minimum under various wind speed scenarios. Its accuracy is significantly improved with stronger robustness, as shown in Table 5.

Tab.5 Equivalent deviation of the fusion model under different wind speed disturbance scenarios

Experiment group number	Voltage deviation	Active deviation	Reactive deviation
Gust1	0.26%	0.21%	7.34%
Gust2	0.17%	0.24%	10.86%
Gust3	0.35%	0.37%	20.82%
Progressive wind 1	0.22%	0.28%	18.28%
Progressive wind 2	0.30%	0.37%	16.13%
Comprehensive wind	0.39%	0.48%	20.57%

VII. CONCLUSION

This paper proposes a method to reduce the dimension of clustering indexed by Blending with XGBoost, cluster via DBSCAN optimized by DTW, and fuse the clustering results. Through dividing machines inside the wind farm, the following conclusions are drawn:

(1) DTW takes into account that the dynamic response time of the same type of DFIGs in the wind farm is different under different wind speeds, and effectively solves the problem of missing data in the actual data of a wind farm.

(2) DBSCAN avoids similar DFIG outliers based on density, and its anti-noise ability has also been significantly enhanced.

(3) The multi-machine equivalent clustering method based on clustering ensemble can satisfy the demands of establishing a fixed multi-machine model with wide applicability in multi-working conditions.

VIII. REFERENCES

[1] Wang Peng, Zhang Zenyuan, Huang Qi, et al. Improved wind farm aggregated modeling method for large-scale power system stability studies[J]. IEEE Transactions on Power Systems, 2018, 33(6): 6332-6342.

[2]XIA Yu. Research on Equivalent Modeling of Large Wind Farm [D]. Hefei University of Technology,2019.

[3] Huang Wei, Zhang Xiaozhen. Equivalent Modeling of Large Scale Wind Farm Based on Feature Analysis [J].Power System Technology, 2013, 37(08): 2271-2277.

[4] Han Ji, Miao Shihong, Li Lixing, Yang Weichen, Li Yaowang.Division of Aircraft Groups in Wind Field and Comprehensive Optimization of Equivalent Wind Field Parameters Based on Multi-view Transfer Learning [J].Proceedings of the CSEE,2020,40(15):4866-4881.

[5] Mi Zengqiang, Su Xunwen, Yang Qisen, et al.Multi-machine characterization method of wind farm dynamic equivalent model [J].Transactions of China Electrotechnical Society, 2010,25 (5) : 162-169.

[6] Cao Na, Yu Qun. Grouping Method of Wind Turbines in Grid-connected Wind Farm under Wind Speed Fluctuation [J].Automation of Electric Power Systems, 2012, 36(02): 42-46.

[7]ZHANG Xing, LI Longyuan, HU Xiaobo, et al.Dynamic Equivalence of Wind Farm Based on Wind Turbine Output Time Series Data Clustering [J].Power System Technology, 2015,39 (10) : 2787-2793.

[8] Mi Zengqiang, Su Xunwen, Yu Yang, Wang Yi, Wu Tao. Dynamic Equivalent Model of Double-fed Wind Farm [J].Automation of Electric Power Systems, 2010, 34(17): 72-77.

[9] Zou J, Peng C, Xu H, et al. A Fuzzy Clustering Algorithm-Based Dynamic Equivalent Modeling Method for Wind Farm With DFIG[J]. IEEE Transactions on Energy Conversion, 2015, 30(4): 1-9.

[10] Chen Shuyong, Wang Cong, Shen Hong, Gao Ningchao, Zhu Lin, Lan Hua.Dynamic Equivalence of Wind Farm Based on Clustering Algorithm. Proceedings of the CSEE, 2012, 32(04): 11-19+24.

[11] YANG Mao, DONG Juncheng. Study on Wind Power Fluctuation Characteristics Based on Mixed Distribution Model [J].Proceedings of the CSEE, 2016, 36(S1): 69-78.

[12] Zou Jianxiao, Peng Chao, Xu Hongbin, et al． A fuzzy clustering algorithm-based dynamic equivalent modeling method for wind farm with DFIG[J]． IEEE Transactions on Energy Conversion， 2015， 30(4) ：1329-1337.

[13] WANG Hui, LIU Da, WANG Jilong. Prediction of Ultra-short-term Wind Speed Based on Spectral Clustering and Optimized Extreme Learning Machine [J].Power System Technology, 2015, 39(05): 1307-1314.

[14] LIN Li, PAN Wancan, ZHANG Lingyun, et al.Wind farm cluster division based on immune outlier data and sensitive initial center K-means algorithm [J].Proceedings of the CSEE, 2016,36 (20) : 5461-5468.

[15] Chao Pupu, Li Weixing, Jin Xiaoming, Qi Jinling, Chang Xuefei. Practical Equivalent Method for Doubly-fed Wind Farm Based on Active Power Response [J].Proceedings of the CSEE, 2018, 38(06): 1639-1646+1900.

[16]LIU Bo, QIN Chuan, JU Ping, et al. Short-term Bus Load Prediction Based on XGBoost and Stacking Model Fusion [J].Electric Power Automation Equipment, 2020, 40(03): 147-153.

[17]LI Yonggang, WANG Yue, LIU Fengrui, et al. Combined Model of Short-term Wind Speed Prediction Based on Stacking Fusion [J].Power System Technology, 2020, 44(08): 28752882.

[18]Weng Hanli, Guo Yida, Li Haowei, Wan Yi, Li Zhenxing, Huang Jingguang. Countermeasures against misoperation of converter transformer zero sequence differential protection under inrush current conditions[J/OL]. Automation of Electric Power Systems, 1-14[2020-09-14].

[19]Peter J.Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis[J]. Journal of Computational and Applied Mathematics, 1987, 20: 53-65.

[20]Li Weixing, Chao Pupu, Liang Xiaodong, et al. A practical equivalent method for DFIG wind farms[J]. IEEE Transactions on Sustainable Energy, 2018, 9(2): 610-620.

[21] Xu Yuqin, Wang Na. Research on dynamic equivalent model of doubly-fed wind farm based on cluster analysis[J]. Journal of North China Electric Power University (Natural Science Edition), 2013(03): 1-5.

[22] Zhang Bo, Zhu Jun, Su Hang. Towards the third generation of artificial intelligence[J]. Science in China: Information Science, 2020, 50(09): 1281-1302.

Appendix

Tab.1 Initial wind speed

Fan number	Wind speed (m/s)	Fan number	Wind speed (m/s)	Fan number	Wind speed (m/s)
1	12.32	11	9.49	21	8.72
2	8.22	12	8.07	22	7.66
3	6.95	13	6.99	23	10.10
4	10.05	14	10.58	24	7.07
5	9.59	15	9.39	25	8.59
6	8.21	16	10.95	26	9.21
7	6.75	17	9.76	27	7.25
8	10.51	18	8.59	28	10.51
9	9.45	19	7.49	29	8.46
10	10.65	20	11.35	30	10.65

Tab.2 Initial wind speed

Fan number	Wind speed (m/s)	Fan number	Wind speed (m/s)
1	13.0000	9	13.0000
2	12.3149	10	12.2824
3	11.8144	11	11.9069
4	10.7757	12	11.1397
5	13.0000	13	13.0000
6	12.2889	14	12.3851
7	12.0159	15	11.7793
8	11.4231	16	11.7355

Tab.3 Server hardware configuration

Device	model
processor	2.3 GHz Eight core Intel Core i9
RAM	32 GB 2667 MHz DDR4
Graphics card	Tesla 418.87.00 P100-PCIE-12GB
operating system	ubuntu18.04

Tab.4 Index contribution ranking

Indicator name	*XGBoost* Calculate contribution	Indicator name	Random forest Calculate contribution
Tem	0.445088	*Tem*	0.215671
Vrq	0.272319	*Ird*	0.174328
Ird	0.208157	*Irq*	0.132072
P	0.028390	*Q*	0.108823
Vrd	0.024280	*Isd*	0.100213
Tm	0.010050	*Isq*	0.087055
Isq	0.003549	*Vrq*	0.083371
Q	0.002683	*Vrd*	0.043248
Pitch	0.002115	*Pitch*	0.020629
Irq	0.001941	*Tm*	0.017948
wr	0.000605	*P*	0.008925

Isd	0.000587	*wr*	0.006242
Vs	0.000238	*Vs*	0.001475

Tab.5 Index contribution ranking

Gust 1	Gust 2	Gust 3	Progressive wind 1	Progressive wind 2	Comprehensive wind
Ird	Tem	Tem	Tem	Tem	Isq
Tem	Vrq	Vrq	Pitch	Pitch	Tem
Vrq	Ird	Ird	Ird	Ird	Ird
Isq	Vs	Vs	Isq	Vrq	Pitch
P	Tm	Tm	Q	P	Irq
Q	P	Isq	Isd	Q	P
Pitch	wr	P	Vrq	Tm	Isd
Vrd	Pitch	wr	Vrd	Vs	Vrd
wr	Isq	Pitch	Tm	Irq	Q
Isd	Isd	Irq	Irq	Isq	Vs
Tm	Irq	Isd	Vs	Vrd	Tm
Vs	Q	Vrd	wr	Isd	wr
Irq	Vrd	Q	P	wr	Vrd

Repeat the steps of Experiment 1, and the clustering results obtained by DBSCAN-DTW are shown in Table 10.

Tab.6 Cluster division results under different wind speed disturbance scenarios

Experiment group number	Grouping results	Contour value	operation hours
Gust 1	{1, 5};{2,6,7,10,14};{3, 8, 11, 12, 15, 16};{9, 13};{4}	0.921	0.00020s
Gust 2	{1,5};{2, 3, 6, 7, 9, 10, 11, 13, 14};{4, 8, 12};{15, 16}	0.950	0.00024s
Gust 3	{1,5};{2, 3, 6, 7, 9, 10, 11, 13, 14};{4, 8, 12};{15, 16}	0.951	0.00020s
Progressive wind 1	{1, 5, 9, 13};{2,6};{3, 7, 8, 11, 15, 16};{4, 12};{10, 14}	0.986	0.00013s
Progressive wind 2	{1, 5, 9, 13};{2, 3, 6, 7, 10, 11, 14, 15, 16};{8, 12};{4}	0.923	0.00019s
Comprehensive wind	{1, 2, 5, 9, 13};{3, 7, 8, 10, 11, 15, 16};{4, 12};{6};{14}	0.951	0.00020s

Tab.7 Dynamic equivalent relative deviation comparison

Experiment group number	Grouping results	Voltage deviation	Active deviation	Reactive deviation
Gust1	k-means	0.24%	0.45%	11.38%
	DBSCAN-DTW	0.32%	0.37%	15.04%
	Random Forest-DBSCAN-DTW	0.15%	0.23%	7.26%
	XGBoost—DBSCAN-DTW	0.09%	0.12%	7.20%
Gust2	k-means	0.37%	0.32%	17.23%
	DBSCAN-DTW	0.28%	0.24%	13.23%
	Random Forest-DBSCAN-DTW	0.21%	0.22%	12.47%
	XGBoost—DBSCAN-DTW	0.06%	0.14%	9.98%
Gust3	k-means	0.30%	0.57%	21.33%
	DBSCAN-DTW	0.23%	0.32%	21.12%
	Random Forest-DBSCAN-DTW	0.15%	0.23%	19.54%
	XGBoost—DBSCAN-DTW	0.08%	0.19%	15.18%
Progressive wind 1	k-means	0.24%	0.26%	17.22%
	DBSCAN-DTW	0.18%	0.27%	13.15%
	Random Forest-DBSCAN-DTW	0.12%	0.20%	11.22%
	XGBoost—	0.10%	0.16%	10.24%

	DBSCAN-DTW			
Progressive wind 2	k-means	0.34%	0.46%	21.78%
	DBSCAN-DTW	0.28%	0.33%	17.99%
	Random Forest-DBSCAN-DTW	0.19%	0.27%	16.48%
	XGBoost—DBSCAN-DTW	0.14%	0.21%	15.37%
Comprehensive wind	k-means	0.38%	0.31%	27.48%
	DBSCAN-DTW	0.29%	0.35%	19.27%
	Random Forest-DBSCAN-DTW	0.26%	0.28%	14.47%
	XGBoost—DBSCAN-DTW	0.23%	0.25%	11.26%