Cleaning of Transient Fault Data in Distribution Network Based on Clustering by Fast Search and Find of Density Peaks

Xiaoli Duan, Sanwei Liu, Fuyong Huang, Daoyuan Zhang, Yan Zhao, Jianjia Duan, Zeyu Zeng, Ting Yu, Lipeng Zhong, Bin Dai

***Abstract*—A cleaning model based on clustering by fast search and find of density peaks (CFSSFDP) is proposed to address the problem of many data types and low data quality in the distribution network fault database. First, auto-extraction model of fault data features based on 1D convolutional auto-encoder is established to obtain fault features of massive fault data. Next, the fault data features are clustered by CFSSFDP, the incorrect and invalid fault features are isolated, and the incorrect and invalid fault data in the database are cleaned. Finally, the effectiveness and accuracy of the model for cleaning of fault data in distribution network is verified through the experimental analysis of the cleaning of Simulink simulation fault data.**

***Index Terms*—Data cleaning; Distribution network faults; CFSFDP; Feature extraction**

I. Introduction

The current distribution network is developing at a high speed toward a smart grid with scale, complexity, and intelligence. Higher voltage levels, greater transmission capacity, closer multi-regional interconnections, and rapid development of auxiliary monitoring and distribution secondary systems are other improvement directions. However, the probability of faults and the amount of fault data in such a large and complex smart grid have increased dramatically [1, 2]. Some real faults are recorded in the massive transient fault data. Other errors and invalid fault records caused by various factors also exist, such as interference from external conditions and ageing of the equipment itself. As a result, the fault data received by the main distribution station have a large base and a low qualification rate. They also contain a large number of errors and invalid data. These conditions affect the accurate determination of actual faults on the line by operation and maintenance personnel. The analysis of specific fault events is also affected [3, 4]. Inaccurate or severely deviated fault event analysis results in incorrect fault characterization, which leads to deterioration and proliferation of real faults. Serious cases of deterioration and proliferation can cause damage to distribution equipment in the distribution system. This condition results in large and prolonged power outages and serious economic losses. Therefore, studying the cleaning model of fault data in distribution network is important to maintain the security of the power network.

The initial construction of the power system transmission and distribution equipment system is completed based on the great development of the power industry. The sharing technology of big data cloud platform also provides conditions for fault information fusion and fault data diagnosis of multi-data source. Meanwhile, the power industry is also faced with the technical difficulties of rapid analysis and processing of massive operational data [5]. Currently, three main cleaning methods are used to clean the condition data of distribution network equipment: the first one is to clean the missing and disturbing data by establishing a mapping relationship between fault data and characteristics. This method is simple and straightforward, but it will damage the continuity of the condition data [6, 7]. The second method is to replace the data columns with normal and valid data. This method is completed by detecting and identifying errors and abnormal data in the columns. The method can effectively protect the continuity of the data columns and reduce the corruption rate of the time series data. However, data replacement without identifying the data properties and patterns will lead to the loss of the original data series characteristics and affect the final data cleaning [8, 9]. The third is the use of artificial intelligence unsupervised self-learning techniques for collective feature differentiation and cleaning of large-scale data [2, 10]. This method makes up for the shortcomings of the second method. It also facilitates the cleaning of datasets with large data size and difficulties in manual data feature identification. Currently, the method has good prospects for application.

Manuscript received Nov. 3, 2022; revised Mar. 1, 2023.

Xiaoli Duan is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 272139237@qq.com)

Sanwei Liu is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 604208086@qq.com)

Fuyong Huang is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 2429534301@qq.com)

Daoyuan Zhang is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 1305981795@qq.com)

Jianjia Duan is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 6854759280@qq.com)

Zeyu Zeng is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(e-mail: 3657481289@qq.com)

Ting Yu is a senior engineer of Electric Power Research Institute of State Grid Hunan Electric Power Co., Ltd, Changsha 410036, China(4582265476@qq.com)

Yan Zhao is a graduate student of Chongqing University of Posts and Telecommunications, Chongqing 400065 , China (e-mail: 8616198@qq.com).

Lipeng Zhong is a senior engineer of Hunan University, Changsha 410082, China (e-mail: 6248962546@qq.com).

Bin Dai is a senior engineer of State Grid Yueyang Power Supply Company, Yueyang 414021, China (e-mail: 754796325@qq.com).

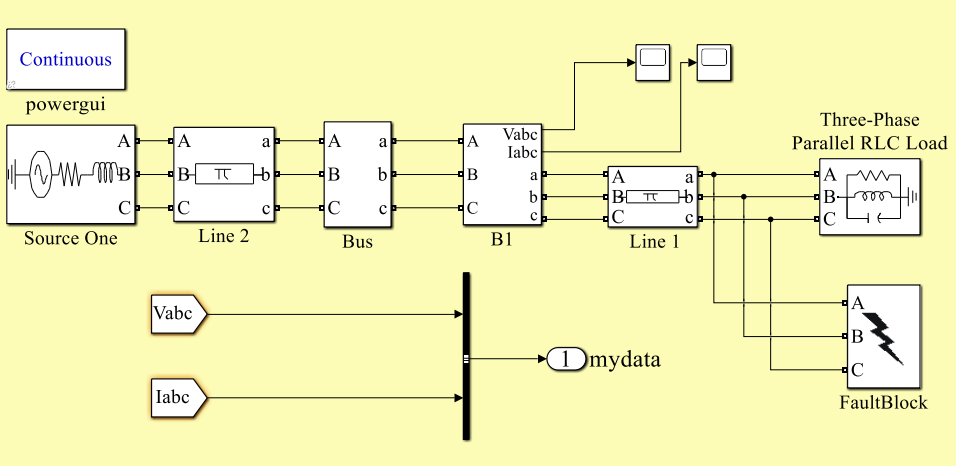


Fig. 1. Simulation model of distribution network

The main reasons are threefold. First, the high sampling frequency of electrical transient fault data in the distribution network, the large data dimension, the long data time series,and the random acquisition time. Second, the multiple different states (steady, transient, and new steady) on each long time series. Third, the data types that include various real faults (single-phase grounding, multi-phase short circuit, and overvoltage), as well as different forms of errors, high-frequency interference fault data, obscure single data characteristics (the fault waveform only appears as individual anomalies), and other problems [3, 11, 12]. Therefore, studying cleaning methods of transient fault data and establishing a clean, high-quality fault database are important for the fusion and analysis of multiple data sources in the distribution network. They are also vital for subsequent fault identification, fault event analysis, and incident rescue.

This study presents a cleaning model of fault data in distribution network based on clustering by fast search and find of density peaks (CFSFDP) algorithm. First, local features of transient fault data are auto-extracted using convolutional auto-encoder (CAE) and principal component analysis (PCA). Then, the transient fault data are cleaned by CFSFDP algorithm. This method effectively eliminates the error and disturbance data in the distribution network dataset. It also effectively improves the cleanliness of the transient fault database through the hierarchical cleaning of the error and disturbance fault data in the distribution network. Meanwhile, the comparative experimental analysis of unsupervised learning multi-cluster cleaning model is launched to demonstrate the effectiveness and accuracy of the self-extraction and cleaning model of transient fault data features.

II. Acquisition of Transient Fault Data in Distribution Network

As shown in Fig.1, the transient fault data of the simulation model in this study are based on a transmission line of 220 kV and 50 Hz with a length of 200 km to simulate a three-phase power system network. The entire network is modeled in the MATLAB environment. The transmission line of the designed network is connected to a three-phase voltage source with positive and zero-sequence resistances of 0.01273 and 0.3864 Ω/km, respectively. Line1 and Line2 are the grid lines. The line voltage and current signals are obtained from the three-phase voltage source side. B1 and Bus modules are the line electrical quantity data in the distribution network. The B1 and Bus modules are the display and acquisition modules for the voltage and current signals collected to provide the data required for detecting and cleaning transmission line faults. The fault block is the fault setting module, and the three-phase loads are RLC-type loads.

As shown in Table I, the design parameters of the network are considered to achieve the various types of faults. Thus, the common seven kinds of fault data are obtained, namely, AG, BG, CG (AG, BG and CG refer to single-phase grounding of phase A, phase B and phase C respectively), AB, ABG and ACG (AB refers to two-phase short circuit of phase A and phase B, ABG and ACG refer to two-phase grounding of phase A and phase B, phase A and phase C respectively) and interference, error fault data. Interference and error fault data for the simulation of fault indicators and line operation process by external interference or their own equipment problems are generated by a class of high- and low-frequency invalid fault data. Fig 2 shows part of the fault data.

TABLE I SIMULATION NETWORK PARAMETERS

Parameter type	Parameter name	Parameter value
Fault	Fault type	AG, BG, CG, AB, ABG, ACG, E-Fault
	Fault distance (km)	1–200
	Fault resistance (Ω)	0.1, 1, 5, 10, 50
	Fault phase angle (ϑ)	45, 135, 270
Line	Positive sequence and zero sequence resistors (Ω/km)	0.01273 and 0.3864
	Positive sequence and zero sequence inductors (H/km)	0.9337e-3 and 4.1264e-3
	Positive sequence and zero sequence capacitors (F/km)	12.74e-9 and 7.751e-9
Voltage source	Inter-phase voltage	220 kV
	Base voltage	220 kV
	Baseline power	60 MVA

III. CFSFDP Cleaning Model Framework

A. CAE feature extraction network

Auto-encoding is an unsupervised learning process that reconstructs input data by encoding and decoding. Moreover,

it extracts low-dimensional feature from the data. 1-D CAE uses a convolutional layer to replace the traditional fully connected layer. Fig 3 shows that the 1-D CAE consists of a convolutional layer, a pooling layer, an unpooling layer, and a deconvolutional layer [13]. In the process of feature extraction, the local perception and weight sharing characteristics of 1-D CAE can obtain more accurate data features and realize more efficient computing [14].

Convolutional layer: The convolution kernel is used to perform convolution calculation on the input signal of the previous layer and output the feature map of this layer by activation function. Thereafter, the feature learning of the convolution layer is completed. The convolution calculation expression is

$$x_{out(i)}^{(l)} = f\left(\sum_{i=1}^{N} x_{out(i)}^{(l)} \odot k_i^{(l)} + b_i^{(l)}\right) \tag{1}$$

where $x_{out(i)}^{(l)}$ represents the i-th feature map in layer l, N represents the number of feature map, $k_i^{(l)}$ represents the i-th convolution kernel in layer l, $b_i^{(l)}$ represents the bias of layer l, $\odot$ represents convolution operation, and f is the activation function.

The activation function is expressed as

$$f(x) = \max(0, x) \tag{2}$$

The output size of the convolutional layer is expressed as

$$Width = \frac{W + 2P - F}{S} + 1 \tag{3}$$

where $Width$ is the convolution output size of this layer, W is the input signal size of the upper layer, P is the filling in the convolution process, F is the size of the convolution kernel, and S is the convolution step length.

Pooling layer: The pooling layer has no parameters to be learned. The number of channels does not change as well. The pooling layer is used to reduce the resolution of convolutional feature maps, compress the dimension of data and parameters, and improve the fault tolerance of the model. It is expressed as

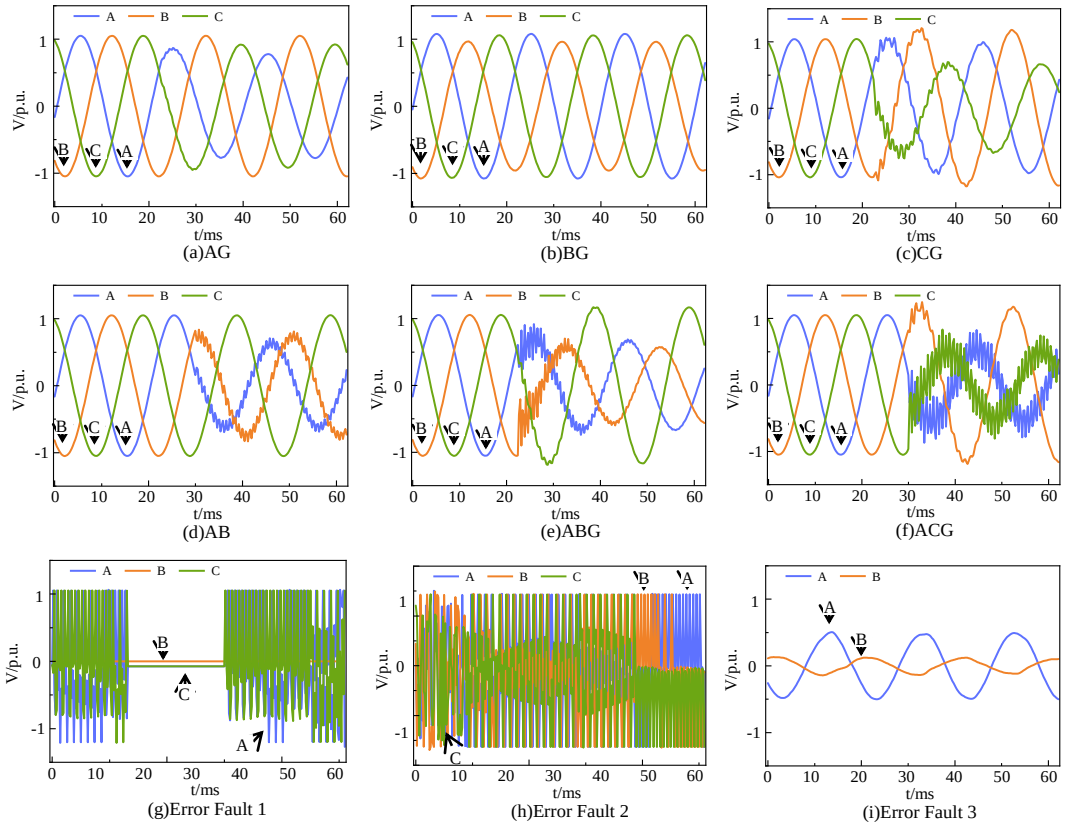


Fig. 2. Partial failure data

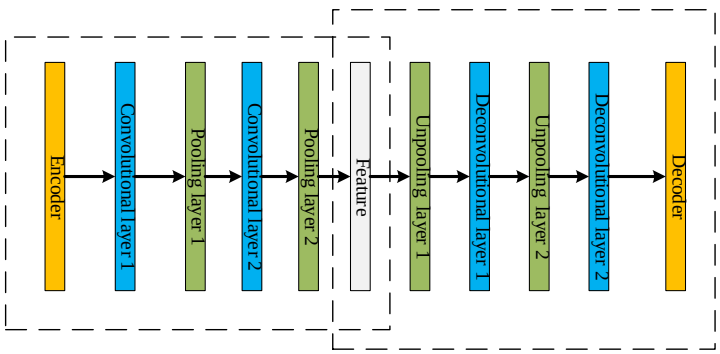


Fig. 3. 1-D CAE structure

$$a_i^{(l)} = \max(x_j^{(l-1)}) \tag{4}$$

where $a_i^{(l)}$ is the *i*-th pooling value of layer *l*, and $x_j^{(l-1)}$ is the element in the pooling window.

The output size of the pooling layer is expressed as

$$Width = \frac{W - F}{S} + 1 \tag{5}$$

where *Width* is the output size of the pooling layer, *W* is the size of the upper input signal, *F* is the size of the pooling window, and *S* is the pooling step length.

Deconvolution layer: The data are upsampled using the transposed convolution matrix. The output size of the deconvolution layer is expressed as

$$Width = S(W-1) - 2P + F \tag{6}$$

where *Width* is the output size, *W* is the size of the upper input signal, *P* is the filling in the convolution process, *F* is the size of the convolution kernel, and *S* is the convolution step length.

Unpooling layer: Unpooling can restore the main information of the data by supplementing bits. This layer has the opposite effect to the pooling layer [15].

B. *Clustering by fast search and find of density peaks*

The CFSFDP algorithm is a data density-based clustering algorithm. It can quickly determine the number of class clusters and class cluster centers through decision diagrams. The algorithm can also rapidly detect the peak class cluster density points, that is, data class cluster centers, for any data shape [16]. The CFSFDP algorithm can highlight anomalous data points and discrete points in a dataset and isolate them effectively. It is suitable for clustering analysis of larger datasets and only requires one traversal for different data to achieve different classes of data clusters. The algorithm is also simpler and more efficient than other iterative clustering algorithms [17].

CFSFDP has some drawbacks in certain cases. For example, different calculation methods are used to calculate the density of different discrete and continuous data in the database when calculating the sample density of the data. Moreover, the truncation distance of the dataset may be different if the sample size of the database is small, which affects the overall clustering effect. In addition, when clustering non-cluster-center data, the non-cluster-center data points that are less dense are clustered in a cluster that is denser and closest to them. The accuracy of the clustering algorithm is reduced [18]. However, this study considers the shortcomings of this particular case and uses a transient fault dataset with a certain amount of data. This way avoids the effect of smaller data samples on the difference in truncation distance. Multiple experiments are conducted on the same dataset with random disruption to determine the accuracy of the clustering algorithm for the chain effect of continuous sample data when cluster occurs. Fig. 4 shows the flowchart of the CFSFDP algorithm.

The CFSFDP algorithm is based on two assumptions. One is that the local density of the cluster centers is greater than the local density of the non-center members of the class cluster. The other is that the centroids with equally high local densities are distant from each other. CFSFDP calculates two important parameters for each piece of data, namely, the local density ρ_i and the distance δ_i, to quickly search for class

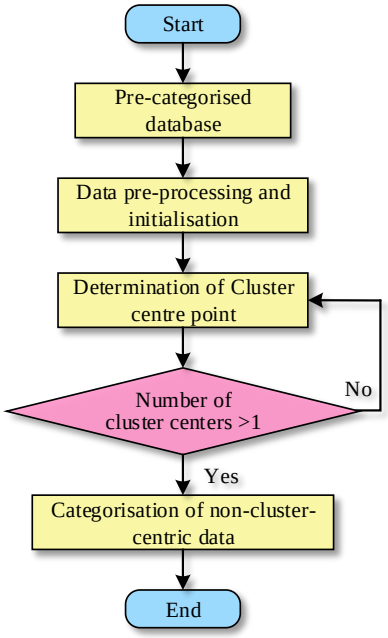


Fig. 4. Flowchart of CFSFDP algorithm

clusters and class cluster centers that meet the above assumptions. For example, data x_i in a certain dataset $Q=\{x_1, x_2, x_3, \ldots, x_n\}$ with key parameters ρ_i, δ_i depending on d_{ij}.

$$d_{ij} = \left\| x^{(i)} - x^j \right\|_2, i \neq j, \tag{7}$$

where $\|\bullet\|_2$ is the Euclidean norm. *i*, *j* = 1, 2, ..., *m*, which is $i \neq j$.

The local density ρ_i is given by

$$\rho_i = \sum_{j=1}^{m} e^{-(\frac{d_{ij}}{d_c})^2} \tag{8}$$

where d_c is the cut-off distance greater than zero, which is often set artificially before clustering is conducted. As clustering databases vary, the cut-off distance is often set indirectly by setting the average percentage of neighbors *P*. d_c is set as

$$d_c = (D_{N1})_k, k = \frac{N \times p}{100}, \tag{9}$$

where $D_{N,1}$ is the d_{ij} of all data in the database in ascending order. *N* is the total amount of data in the database. The *k*-value can be obtained by setting the average percentage of neighborhood *P* to determine the truncation distance. The value is generally set at 1%–2% of all data points in the database.

The data point distance δ_i is calculated as

$$\delta_i = \begin{cases} \max\limits_{j:\rho_j<\rho_i} \left(d_{ij}\right) \\ \min\limits_{j:\rho_j>\rho_i} \left(d_{ij}\right) \end{cases} \tag{10}$$

The display results for ρ_i and δ_i can be derived by calculating the local density ρ_i and the distance δ_i for each data. A visual display of the class cluster core of the database is performed by decision tree construction and the product of the local density and the distance from the data points. Fig 5 shows the local density ρ_i and distance δ_i of the fault sample data after the CFSFDP algorithm. Fig 5 (a) shows the relationship between local density ρ_i and distance δ_i. The purple data points in this figure are the class cluster cores in the clustering process. They have a high local density ρ_i and a large distance δ_i. Fig 5 (b) shows the method of

determining the number of class cluster cores in decreasing order $\gamma_i=\rho_i\times\delta_i$. The yellow shaded area in this figure is the corresponding class cluster centroid highlighted by the red data points in Fig 5 (a).

After the cluster cores for each class of clusters are determined, the feature points need to be assigned to different class clusters based on their own ρ_i and δ_i. Fig 6 shows that the feature points around the cluster cores are either classified as neighboring class clusters or designated as overlapping points. Those classified as neighboring class clusters are true kernel elements, and others are designated as overlapping points because the truncation distance between multiple class clusters overlap and do not belong to any class cluster.

C. CFSFDP cleaning model

A transient fault data cleaning model based on the CFSFDP clustering algorithm is built to eliminate errors and disturbances in the transient database of the distribution network. Clustering algorithms such as decision trees, support vector machines, and K-means can be used at this stage to directly determine the number of clustering centers in the feature set and complete the clustering of fault data. However, some peripheral discrete fault data features in the transient dataset that are less dense locally and farther from the cluster centers are also grouped into relatively adjacent clusters. Obviously, such small local densities and distant from the cluster centers are erroneous, which disturbs transient fault data in the distribution network. If such data are not isolated prior to formal clustering, then discrete fault data will be mixed into the real fault data or high-frequency error data clusters. This condition reduces the effectiveness of the transient fault data cleaning. Therefore, completing the pre-cleaning of discrete data points in the periphery of the transient fault dataset prior to full clustering is necessary to achieve the best cleaning effect on transient fault data. The CFSFDP cleaning model is shown in Fig 7. The model has the following three main components.

(1) Local features of transient fault data in distribution network are obtained by CAE and PCA techniques.

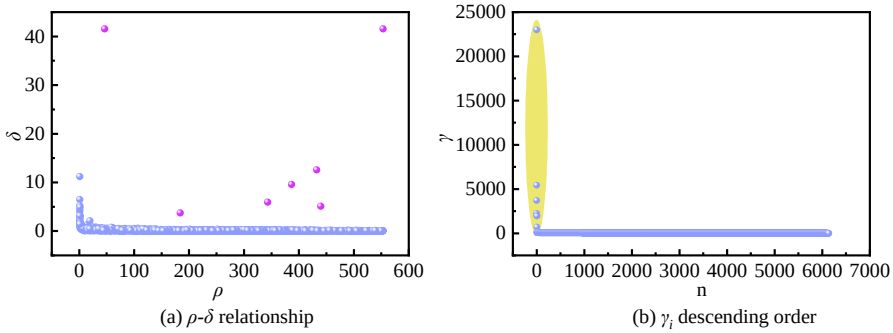
(2) Pre-cleaning of the transient fault data is conducted by CFSFDP algorithm with discrete points.

(3) The CFSFDP algorithm is used again to achieve the complete cleaning of transient fault data.

Fig 8 shows the CFSFDP clustering cleaning process. In the local feature auto-extraction phase of the transient fault data, the transient fault dataset in the 1D distribution network is first extracted locally by convolutional auto-coding network and PCA. In this stage, different types of transient fault data are entered into the feature extraction network to train the weights and bias parameters of the convolutional auto-coding feature auto-extraction. The principal components in PCA are also efficiently acquired in this way.

In the pre-cleaning stage of transient fault data, the CFSFDP algorithm is used to cluster the fault data features obtained in the first part. Then, it locates the discrete points of the fault data by building a ρ–δ relationship decision tree and ρi in descending order. Finally, the algorithm isolates the fault data corresponding to the discrete points for error fault cleaning in the transient fault dataset.

In the complete cleaning stage of the transient fault data, the CFSFDP clustering based on the pre-cleaning in the second stage is performed again on the features of the transient fault dataset by isolating the discrete fault data at the periphery. Then, each class cluster is identified, and the transient fault data type represented by the centroid of each class cluster is detected. The fault type of the centroid of the cluster represents the overall fault data type of the cluster. The high-frequency error fault class cluster data are located by the error or invalid fault type expressed by the cluster centroids. Finally, the fault data corresponding to the error fault clusters are cleaned to achieve a complete cleaning of the transient fault database.



(a) ρ-δ relationship (b) γ_i descending order

Fig. 5 CFSFDP diagram for transient fault data in distribution network

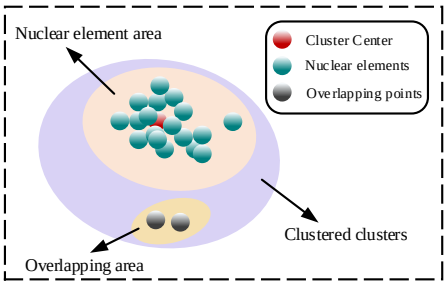


Fig. 6 Definition of elements in clusters

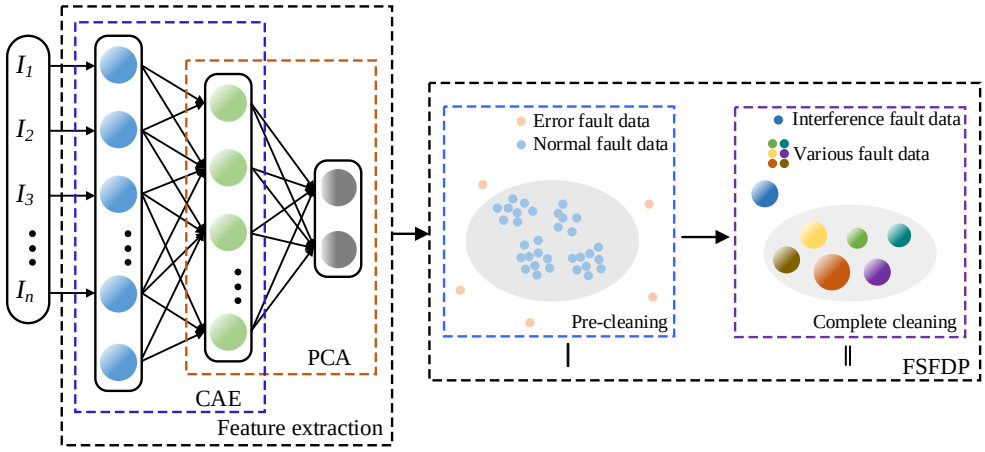


Fig. 7 CFSFDP fault data cleaning model

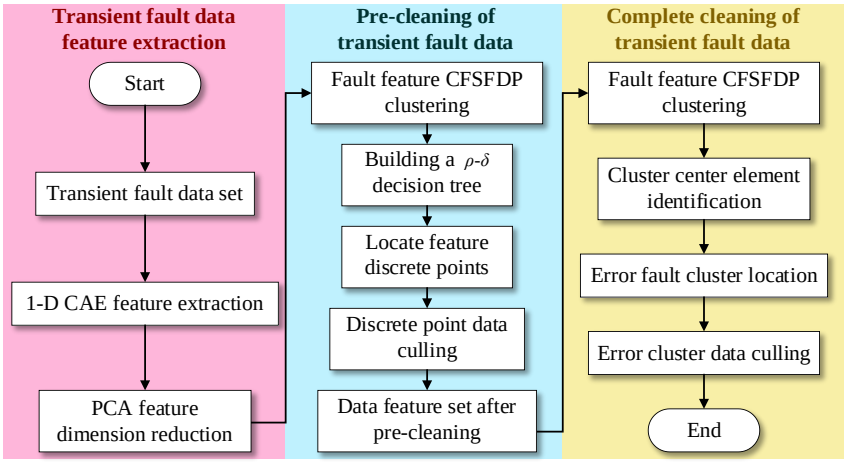


Fig. 8 Flowchart of CFSFDP fault data cleaning

IV. CFSFDP CLEANING IMPLEMENTATION

A. *Self -extraction of the transient fault data*

1-D CAE is used to study the feature extraction of fault data samples. The TensorFlow and Python are used, and the computer configuration is Intel Core i710700CPU@2.9GHz, The memory is 8GB. 1-D CAE framework is built in the TensorFlow, the convolutional coding layer and decoding layer are obtained. The training data samples are imported, the convolution model is compiled, the network structure parameters are optimized. The structure of the 1-D CAE consists of an input layer, an output layer, two convolution pooling layers and deconvolution pooling layers. The two-layer network are enough to obtain the features of the sample data. Due to the low noise of the data, the convolution core size in the network is small. The network has good feature extraction ability. As discussed in our previous work [19], different loss functions are used to show the reconstruction degree of the network in the training process. When the training times reach the fifth time, the binary cross entropy (BCE) loss function value converges to about 0.63. The mean square error (MSE), mean absolute error (MAE) and mean square logarithmic error (MSLE) loss function value converges to approximately 0. In addition, The MSE loss function converges faster than MSLE and MAE. The convergence value of MSE is smaller than those of MSLE and MAE. Thus, the MSE is used as the model objective function. The stable and approximate zero loss function value shows that the error between the output and input of the model reconstruction is very small. The reconstruction effect of 1-D CAE is good.

B. *Fault pre-cleaning for distribution networks based on CFSFDP*

The data are visualized, as shown in Fig. 9, after the 2D feature vector of transient fault data from the distribution network is obtained through the 1D convolutional self-encoder local feature self-extraction technique. The transient fault data corresponding to the 2D feature vector are cleaned by clustering. According to the CFSFDP algorithm, the average neighborhood percentage P is generally set to 1%–2% of all data points in the database. The average neighborhood percentage P=2 is set to meet the truncation distance value.

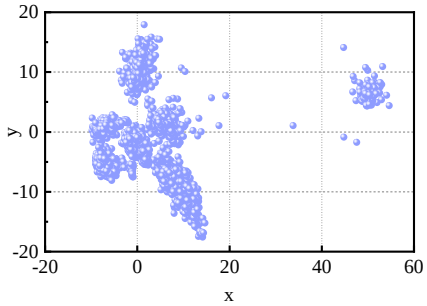
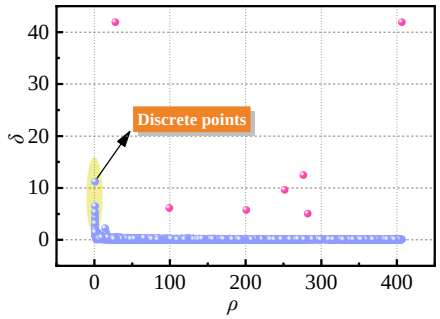
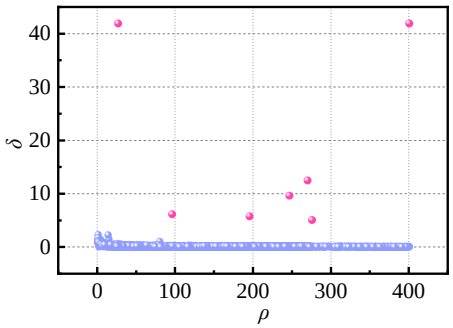


Fig. 9 Visualization of transient fault data features (P=2, x and y denote the first and second features extracted, respectively)

Based on the set mean neighborhood percentages, the truncation distance of the sample dataset is calculated using Equation (7), the local density ρ_i and distance δ_i corresponding to each feature are obtained using Equations (8) and (9), and the decision tree of the ρ–δ relationship is constructed as in Fig 10 (a). The yellow areas with small local densities and large distances represent the discrete points in the fault sample dataset. These discrete points, which are relatively small in number and far from the center of the cluster, are likely to be interference data in the fault data collected by the distribution network. They also need to be isolated as much as possible. The discrete points in the sample dataset can be obtained by ranking the local density of the fault data feature points. Fig 11 (a) shows the local density ranking of the fault data feature points. As shown in the figure, the last section of the sample data points has the lowest local density and is all close to 1. Therefore, these data are discrete points that are far from other classes of clusters. Fig 11(b) shows a zoomed-in view of the local density ranking of the sample data points in this section. The red area formed by the local density of the fault data feature points (ρ<1.2) is the discrete area. The transient fault data corresponding to the feature points in this area are judged as error and interference data, and such data are isolated. The blue area with local density ρ>1.2 is the normal class element area, which represents the real fault data. The local density threshold here is not fixed and can be adjusted in conjunction with the ρ–δ relationship decision tree and the distribution in Fig 11(b). The main consideration for the selection of the ρ threshold is the rapid decrease in local density to a stable value close to 1.



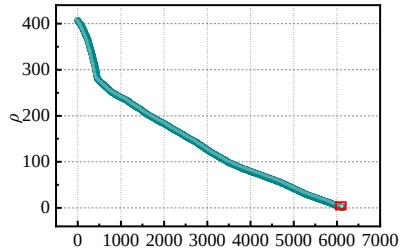
(a) Before pre-cleaning



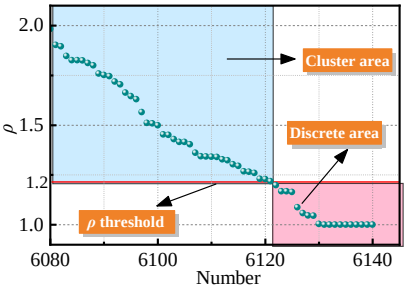
(b) After pre-cleaning

Fig. 10 Building a decision tree (P=2)

After the discrete point data are located and the isolation is completed, the ρ–δ relationship decision tree is again constructed as in Fig 10 (b). Comparison indicates that the discrete points corresponding to the yellow areas in the original figure are effectively isolated. Table II shows the data types corresponding to the isolated discrete points.



(a) ρ_i sorted



(b) ρ_i sorted partial zoom

Fig. 11 ρ_i in descending order

The calculation shows that the pre-cleaning cleaning precision (CP) of the distribution network transient fault data is 84.2%, the correct cleaning accuracy (CA) is 11.4%, and the mistake cleaning rate (MCR) is 0.05%.

$$CA = \frac{\text{Amount of error fault data cleaned}}{\text{Total error fault data}}$$

$$CP = \frac{\text{Amount of error fault data cleaned}}{\text{Total amount of data cleaned}}$$

$$MCR = \frac{\text{Total real fault data for miscleaning}}{\text{Total real fault data}}$$

TABLE II PRE-CLEANING RESULTS

Total pre-cleaned fault data	Error fault data	Real failure data
19	16	3

C. Fault complete cleaning for distribution networks based on CFSFDP

The transient fault data in distribution network are pre-cleaned to isolate some of the errors and interference fault data in the dataset. However, the cleaning cannot be completed for the clusters formed by some high-frequency error fault data. Thus, complete cleaning is needed for the error fault clusters.

After the pre-cleaning is completed, the seven feature points with high local density and large distance in Fig 10 (b) are used as the class cluster centers. The assignment of the remaining feature points is completed as well. Fig 12 shows the clustering results of each class of clusters for different fault types. As observed, the seven clusters represented by the seven fault types are presented as seven regions in the diagram. The data that are clearly assigned to a region are the core elements of the clusters. They have a high local density and are in the central part of the clusters. Region 8 represents the overlap between clusters, that is, the overlap points between clusters. The overlap points are related to the truncation distance set in the clustering process. They have a small local density and do not belong to any of the clusters. Table III, which presents the elements that each cluster has,

shows intersection points between clusters 4 and 5. The overlap points occur between clusters 4 and 5 of the two real fault data clusters. Thus, they do not affect the number of kernel elements in cluster 7 of the high-frequency error clusters. Accordingly, the overlap points present here do not influence the fault data cleaning effect.

TABLE III ANALYSIS OF CLUSTERING ELEMENT RESULTS

Clustered clusters	Total number of elements	Nuclear elements	Overlapping points
1	997	997	0
2	999	999	0
3	1028	1028	0
4	1001	997	4
5	973	971	2
6	1002	1002	0
7	121	121	0

The fault data represented by central elements of the clusters are identified. The type of fault represented by each cluster can be obtained. According to Fig. 12, the distance between cluster 7 and the other clusters are significantly higher than those between any two clusters except cluster 7. The data represented by central elements in cluster 7 are fault. According to the similarity of clustering, the core elements in cluster 7 are the same fault type. Therefore, the fault data represented by this cluster should be eliminated from the dataset. Then, the complete cleaning of the transient fault data is finished.

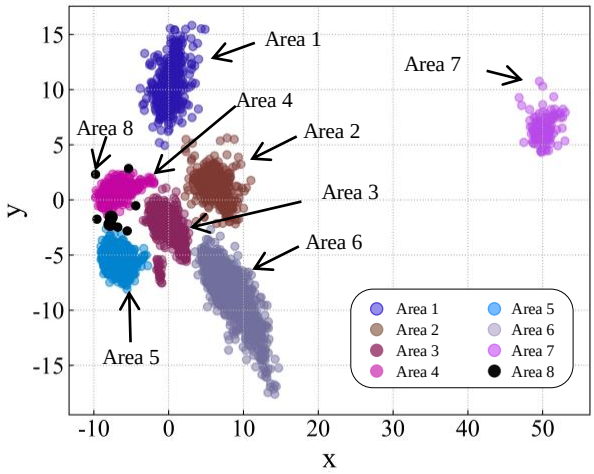


Fig. 12 Complete clustering cleaning results (P=2, x and y denote the first and second features extracted, respectively)

The actual data types of the cleaned cluster elements are shown in Table IV, which shows that all the core elements of the clusters are fault data. As shown in Fig 13, the total CP is 97.9%, the CA is 97.9%, and the MCR is 0.05% for the transient fault data through two incremental cleaning processes, namely, pre-cleaning and complete cleaning.

The results of direct cleaning of transient fault data by CFSFDP are shown in Fig 14. These results are obtained by identifying the error fault cluster 7 and cleaning the transient fault data corresponding to the database. Table V shows the actual data types corresponding to the elements of the direct cleaning clusters. As shown in the table, the CP for direct cleaning of transient fault data is 100%, the CA is 92.1%, and the MCR is 0. The direct cleaning has a good CP, but the CA for all error fault data in the database is low, which results in a certain amount of error data in the transient fault database. The fault database still contains a certain amount of erroneous fault data. Therefore, comparison of the progressive cleaning mode of pre-cleaning and complete cleaning with the direct cleaning mode shows that the two-level progressive cleaning can effectively improve the cleaning quality and cleanliness of the transient fault database.

TABLE IV COMPLETE CLEANING RESULTS

Cleaning data volume	Error fault data	Real fault data
121	121	0

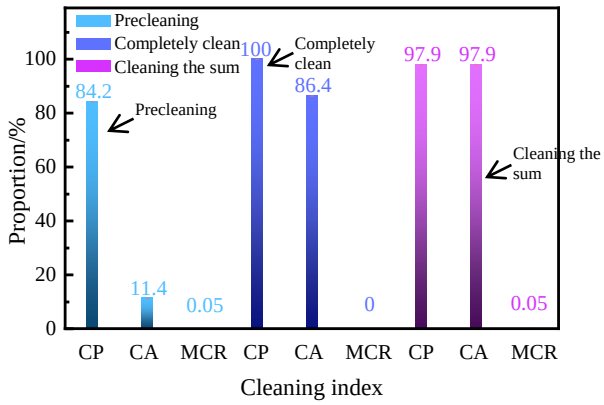


Fig. 13 Results of CFSFDP fault data cleaning

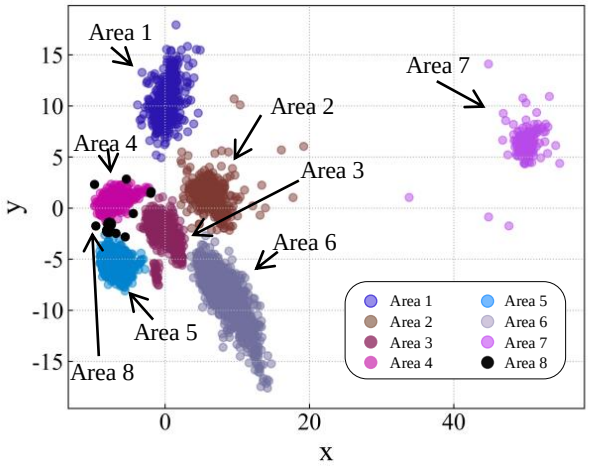


Fig. 14 Results of direct cleaning of CFSFDP fault data

TABLE V DIRECT CLEANING RESULTS

Cleaning data volume	Error fault data	Real fault data
129	129	0

V. EXPERIMENTAL ANALYSIS OF DISTRIBUTION NETWORK FAULT DATA CLEANING

The CFSFDP clustering and cleaning results are compared with those of K-means clustering, density-based spatial clustering of applications with noise (DBSCAN), and other common clustering algorithms for experimental comparison to verify the superiority of the CFSFDP clustering algorithm in clustering and cleaning of transient fault data in distribution networks.

The K-means clustering algorithm is a common unsupervised iterative clustering algorithm that measures the similarity between two or more datasets by the Euclidean distance between the data or the cosine similarity metric.

Notably, the smaller distance between the data means greater similarity and more likely for them to be regarded as the same class of feature data [20]. The aim is to group the entire dataset into *n* clusters, with the center of mass of each cluster being calculated based on the mean of all the data in this cluster. The algorithm first selects *n* random data points from the dataset as the center of mass according to the predetermined number of clusters. Then, it calculates the distance from each number in the dataset to each center of mass, and the data points are assigned to the nearest cluster of the center of mass according to the distance to the center of mass. Next, it recalculates the location of the center of mass of the cluster according to the data of the cluster, and the previous step is repeated several times to continuously update the center of mass of the cluster. Finally, the cluster core is designated as the final cluster core when the position of the cluster cores no longer changes or the distance of each update is less than a set threshold [21].

Four clustering states are set manually for the number of clusters *n*=2, 3, 5, 7 to evaluate the clustering performance of K-means algorithm on transient fault data. Fig 15 shows the visualization in the 2D features of transient fault data under the four clustering states, where *x* and *y* are the first and second features extracted from the fault data, respectively. Regardless of the number of clusters set in K-means, the discrete points in the data cannot be pre-cleaned and are grouped into clusters that are closer together. For the clusters of incorrect and invalid fault data in the top right of the figure, the same amount of fault data is clustered in each clustering state. The isolated cleaning is also calculated to produce a CP of 100% for K-means, 92.1% for CA, and 0% for MCR [22].

The DBSCAN algorithm is also a density-based spatial clustering algorithm, and it is commonly used in cluster analysis and image recognition. It can perform multi-class cluster clustering on arbitrarily shaped datasets and can classify datasets into core points, boundary points, and outliers [23]. The algorithm also does not require a manual setting of the number of clusters. However, it needs to set the neighborhood radius ε and the minimum number of data points in the neighborhood *P*. The DBSCAN algorithm usually starts by detecting the number of data points in a defined neighborhood for each data point. If the number of points is greater than *P*, then a cluster is created with the centroid as the core; otherwise, it is marked as an outlier. Next, it iteratively calculates the core points that can form the clusters and combines the closer clusters. The data clustering is completed until no new data points are added to either class of clusters [24].

The minimum number of data points *P* in the clustering neighborhood is adjusted and the 2D visualization of the fault features corresponding to the minimum number of data points *P* in the clustering neighborhood is shown in Fig 16 for 2, 10, 30, and 60 for verifying the clustering performance of the DBSCAN algorithm on transient fault data. Here, x and y are the first and second features extracted from the fault data, respectively. As observed, the number of clusters increases for smaller and larger *P* settings. Clustering is also ineffective despite the isolated pre-cleaning of the anomalies followed by clustering cluster cleaning. Specifically, the clustering is optimal when *P* = 10, with a CP of 97.8%, a CA of 93.8%, and an MCR of 0.15%. The CA of the DBSCAN clustering algorithm for transient fault data is close to that of the CFSFDP algorithm. However, the accuracy rate of fault data cleaning has decreased, and the overall effect is slightly inferior to that of the CFSFDP algorithm.

As shown in Table VII, compared the proposed CFSFDP with K-means and DBSCAN, the K-means clustering algorithm is less adjustable and has a lower CA. The K-means is unsuitable for cleaning the transient fault data of distribution network. DBSCAN can achieve effective cleaning of fault data, but its CP is lower and is not the best choice. Therefore, the proposed CFSFDP considers both CA and CP of transient fault data. Accordingly, the CFSFDP can effectively eliminate disturbing and erroneous data in massive transient fault data.

TABLE VII COMPARISON RESULTS OF DIFFERENT CLUSTERING METHODS

Clustering methods	CP	CA	MCR
K-means	100%	92.1%	0%
DBSCAN	93.8%	97.8%	0.15%
CFSFDP	97.9%	97.9%	0.05%

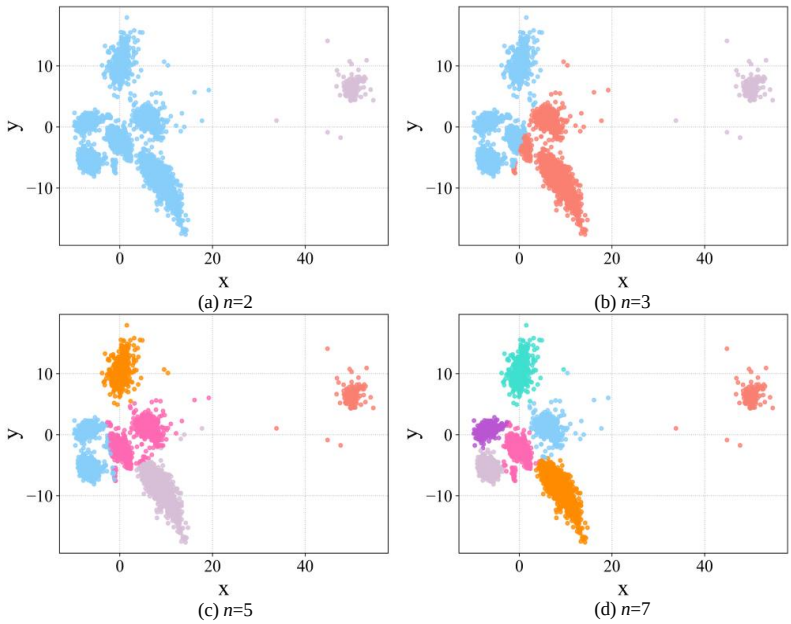


Fig. 15 K-means clustering results

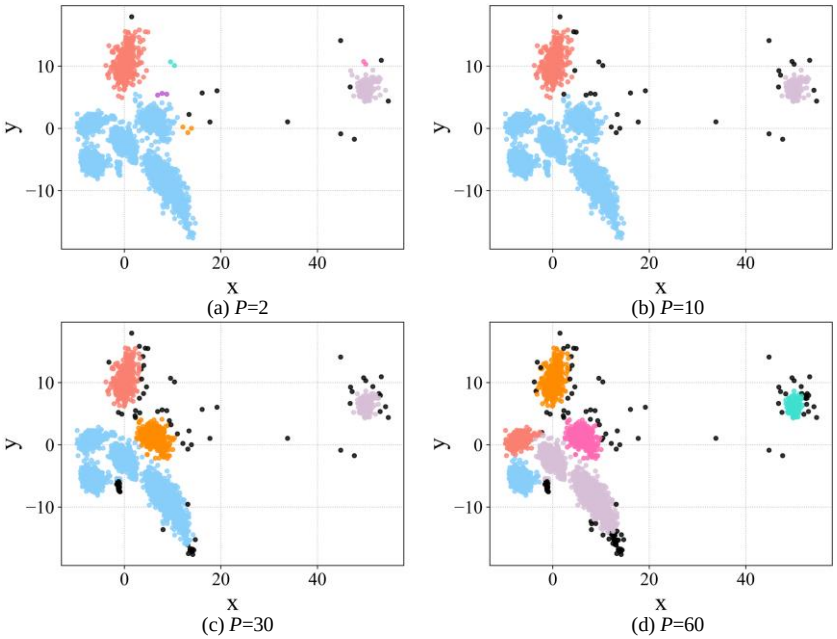


Fig. 16 DBSCAN clustering results

VI. Conclusion

In this study, a transient fault data cleaning model based on density peak fast search clustering is proposed to clean the error and disturbance data in the massive transient fault data and establish a clean and effective fault database. This work lays a solid data foundation for the subsequent work of fault identification, fusion, location, and event analysis of distribution network. It is also particularly important for the future development of power system toward smart grid. The local features of transient fault data are obtained by convolutional self-coding network and PCA techniques. CFSFDP clustering cleaning experiments under transient fault data are conducted as well. The progressive cleaning strategy of CFSFDP for pre-cleaning and complete cleaning of fault data is proposed, and the CP and CA reach 97.9%, and the MCR is 0.05%. Comparison experiments of K-means, DBSCAN, and CFSFDP clustering algorithms are performed based on the same fault data to further validate the usability and superiority of CFSFDP clustering algorithm in the field of transient fault data of distribution network.

References

[1] Jiejie Dai, Hui Song, Yi Yang, et al. A Stack-based Noise Reduction Self-encoder Method for State Data Cleaning of Transmission and Substation Equipment[J]. Power System Automation, 2017,41(12):224-230.

[2] Yingjie Yan, Gehu Sheng, YuFeng Chen, et al. A Time Series Analysis-based Method for Cleaning Transmission and Substation Equipment Condition Big Data[J]. Power System Automation, 2015,39(07):138-144.

[3] Qinzhu Chen, Han Zhang, Jian Yin, et al. Intelligent Self-cleaning Method for Massive Measured Overvoltage Data[J]. High Voltage Electronics, 2019,55(12):227-233.

[4] Shuang Hao, Guoliang Li, Jianhua Feng, et al. A Review of Structured Data Cleaning Techniques[J]. Journal of Tsinghua University (Natural Sciences Edition), 2018,58(12):1037-1050.

[5] Yixuan He, Yi Luo, Guangyu Tu. Framework Design of Ems Data Flow Management System[J]. Power System Automation, 2006(24):33-38.

[6] Zengli Wu, Yongli Zhu, and Jinsha Yuan. A Comprehensive Fault Diagnosis Method for Transformers Based on Bayesian Network Classifier[J]. Journal of Electrical Engineering Technology, 2005(04):45-51.

[7] Xiaowei Yang, Guangquan Zhang, Jie Lu, et al. A Kernel Fuzzy c-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems with Outliers or Noises[J]. IEEE Transactions on Fuzzy Systems, 2011,19(01):105-115.

[8] Feng Meng., Canlin Gao, and Li Bing. Fuzzy Possibilistic Support Vector Machines for Class Imbalance Learning[J]. Journal of Convergence Information Technology, 2013,8(03):692-701.

[9] Jiyu Chen, Wenyuan Li, Adriel Lau, et al. Automated Load Curve Data Cleansing in Power Systems[J]. IEEE Transactions on Smart Grid, 2010,1(02):213-221.

[10] Jinxiao Wei, Buxiang Zhou, and Bing Zhang. Integrated Ddata Cleaning and Unsupervised Learning Techniques for Condition Assessment of Power Equipment[J]. Hydropower Energy Science, 2016,34(09):210-214.

[11] Jixian Zhang, Multi-source Remote Sensing Data Fusion: Status and Trends[J]. International Journal of Image and Data Fusion, 2010, 1(01):5-24.

[12] Yunpeng Hu, Huanxin Chen, Guannan Li, et al. A Statistical Training Data Cleaning Strategy for the PCA-based Chiller Sensor Fault Detection, Diagnosis and Data Reconstruction Method[J]. Energy and Buildings, 2016,112(01):270-278.

[13] Helene Canot, Philippe Durand, and Emmanuel Frenod, Prediction the Strain of Traction-aged Polymer Systems from Artificial Neural Networks with Regularization[J]. IAENG International Journal of Computer Science,2022，49(04):1228-1241.

[14] Li Chen, Zhaoxin Ding, Mu Su, et al. Fault Diagnosis for Distillation Process Based on CNN－DAE[J]. Chinese Journal of Chemical Engineering, 2019,27(03):598-604.

[15] Yuanbin Wang, Yuanyuan Li, and Huaying Wu, Fire Detection Method Based on Improved Convolutional Neural Network with Random Inactivation[J]. IAENG International Journal of Computer Science, 2022，49(04):1297-1304.

[16] Rodriguez Alex., Laio Alessandro. Clustering by Fast Search and Find of Density Peaks[J]. Science, 2014,344(6191):1492-1496.

[17] Junfen Chen, Ming Zhang, Jiacheng Zhao. Fast Search Clustering Algorithm for Density Peaks of Complex High-dimensional Data[J]. Computer Science, 2020,47(03):79-86.

[18] Chuntao Chen. Research and Application of Fast Search and Density Peak Discovery Algorithm[D]: East China Normal University, 2019.

[19] Mi Zou, Yan Zhao, Yan D, et al. Double Convolutional Neural Network for Fault Identification of Power Distribution Network[J]. Electric Power Systems Research, 2022, 210(12): 108085.

[20] Jiawei Liu, Qi Li., Weirong Chen, et al. A Discrete Hidden Markov Model Fault Diagnosis Strategy Based On K-means Clustering Dedicated to PEM Fuel Cell Systems of Tramways[J]. International Journal of Hydrogen Energy, 2018,43(27):12428-12441.

[21] Yingnian Liao, Mengjun Li, Jiqiang Zhang, et al. Multiple Target Location Based on K-means Clustering and Particle Swarm Optimization[J]. Electronic Design Engineering, 2018.
[22] Irlandia Ginanjar, Septie Wulandary, and Toni Toharudin, Empirical Best Linear Unbiased Prediction Method with K-Medoids Cluster for Estimate Per Capita Expenditure of Sub-District Level[J]. IAENG International Journal of Applied Mathematics, 2022, 52(03): 610-616.
[23] Xinyu Tian, Qinghe Zheng, and Nan Jiang, An Abnormal Behavior Detection Method Leveraging Multi-modal Data Fusion and Deep Mining[J]. IAENG International Journal of Applied Mathematics, 2021, 51(01): 92-99.
[24] Unver Mustafa, Erginel Niha, Clustering Applications of IFDBSCAN Algorithm with Comparative Analysis[J]. Journal of Intelligent and Fuzzy Systems, 2020,39(05):6099-6108.