This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3468012

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Research on Point Cloud Segmentation Method based on Local and Global Feature Extraction of Electricity Equipment

Ze Zhang, Danyin Wen, Weijun Shi, Yan Wu, Jiajia Zhou and Kaiming Bian

State Grid Taizhou Power Supply Company, Taizhou Jiangsu 225300, China

Corresponding author: Ze Zhang (e-mail: 786571551@qq.com).

This research was supported by State Grid of Jiangsu Electric Power Co.(China), grant number J2023060."

ABSTRACT Intelligent inspection has become an important trend in the development of substation operation and maintenance technology, and fast and accurate point cloud segmentation of the large amount of point cloud data collected in the process of intelligent inspection is an important basis for effective fault localization and risk early warning of electricity equipment. However, the existing point cloud segmentation methods have problems such as messy spatial information, low classification accuracy and unstable model. Therefore, a new point cloud component segmentation method is proposed by combining the local and global information characteristics of the point cloud and applied to the diagnosis of electricity equipment. Firstly, apply dynamic-k neighbor retrieval for each point to facilitate subsequent local feature extraction. Then, the input point cloud data is processed by graph convolution to capture the local characteristics of the data; then, the global information is extracted by cyclic use of a set abstraction layer, and the two information features are fused and the model is trained to obtain a more powerful and efficient point cloud component segmentation model. Finally, experimental validation was carried out using point cloud data of lightning arrester devices in electricity equipment. The experimental results show that the accuracy and average mIoU of the part segmentation results of this method reaches more than 90%. With 2%-4% improvement and 4-10 times stability improvement compared to traditional point cloud processing methods, this method has better stability and accuracy in point cloud part segmentation tasks. This method not only provides an effective method for the segmentation of 3D point cloud data, but also provides strong technical support for the subsequent fault diagnosis and detection work.

INDEX TERMS Electricity Equipment, Intelligent Inspection, Point Cloud Part Segmentation, Deep Learning Model, Feature Fusion

I. INTRODUCTION

Point cloud data is a data representation that describes the surface of an object in three-dimensional space, which is widely used in computer vision, geographic information science, remote sensing technology and other fields. Point cloud segmentation is one of the key techniques for processing point cloud data, which aims to divide point cloud data into multiple parts with different attributes or semantic information [1], [2]. In recent years, point cloud part segmentation has played a crucial role in various fields, such as automated driving, machine vision, and equipment overhauling, bringing significant advantages to tasks such as fault diagnosis and defect identification [3]. For point cloud data in complex environments, how to accurately extract individual components of equipment from large-scale unorganized point cloud data is an important foundation for subsequent fault diagnosis and defect identification of electricity equipment [4], [5].

Since the stable operation of electricity equipment is crucial for people's life and safety assurance, it is vital for the detection of electricity equipment, and the use of more stable and effective point cloud segmentation methods can provide better support for the subsequent tasks. Currently, few research addressed the detailed analysis and practical application of point cloud component segmentation in the context of power equipment. Although some substations have initiated the utilization of point cloud data, the process of component segmentation remains predominantly manual.

Given the substantial quantity of equipment within a station, the manual segmentation of point cloud components is both time-consuming and labor-intensive. Consequently, the exploration of automated methods for the segmentation of point cloud components in power equipment holds significant value and is imperative for advancing the field.

In early research, segmentation methods based on point cloud region growth were most widely used [6]. In recent years, many deep learning models specialized in processing point cloud data have appeared at home and abroad [7], [8], [9], which can be specifically classified into deep learning models based on RGB-D images, voxels, and other representation elements[10].

In terms of deep learning models, 3D ShapeNets [11] proposed by Princeton University and VoxNet [12] proposed by Daniel Maturana transform point cloud data into voxel lattices and then process them using 3D convolutional networks, which lose part of the information when extracting spatial features. In addition, the model proposed by Hugues Thomas [13] defines a learnable geometric kernel, however, its high computational complexity makes it difficult to meet the real-time requirements.

Using relationships between points directly and retrieving features is a simple and intuitive approach. The model PointNet [14] proposed by Stanford University is a point cloud based method which uses a shared Multilayer Perceptron (MLP) to mine the point-by-point cloud features and extracts the global features directly from the original 3D coordinates, but it is weak in recognizing the local structures. After this Stanford University proposed PointNet++ [15], which is based on PointNet and employs a hierarchical neural network structure to mine local structure information, which enhances the inadequacy of PointNet to some extent [16]. PointWeb [17], proposed by the University of Hong Kong, China, inserts an adaptive feature adjustment module layer after the clustering layer in order to embed the interaction information between point clouds into each point cloud. These strategies enhance the representation of learned point cloud features. However, these MLP-based feature extraction methods are limited in that the method treats all points as equally important and misses capturing the key information of the object, i.e., the extraction of localized features does not do the trick.

The dynamic graph convolutional neural network proposed by MIT [18] employs dynamic edge convolution operation to effectively capture the local features of point cloud data, however, the graph structure is connected in such a way that it does not pay attention to the role of each point in the overall point cloud, and thus it is relatively weak in dealing with global features [19], [20]. Beijing Institute of Technology introduced the CSA module based on PointNet++ [21], which adds an attention module to each layer of the feature and spatial dimensions to improve the accuracy of local feature extraction, however, due to the loss of efficiency in feature extraction at each layer of the model, the complexity of feature propagation is increased, and due to the use of large-scale downsampling in this model, it cannot adequately capture the local features when dealing with complex point clouds.

To summarize, existing methods have problems such as spatial information clutter, low classification accuracy, and high model complexity when performing point cloud segmentation of electricity equipment. Moreover, the evaluation of point cloud part segmentation is not only limited to the accuracy of the segmentation, but also needs to take into account the model size as well as the detection speed. Most of the current feature extraction combination methods are too complex, greatly delaying the model training time, increasing the model complexity and poor comprehensibility.

Based on this, this paper proposes a new method for segmenting point cloud data, Dynamic Graph Convolution Pointnet (DGCPointnet). The method organically combines the edge convolution module (Edge Conv) in graph convolutional networks with the use of MLP global feature extraction to fuse and classify at the feature level in order to fully utilize the respective advantages of both. The edge convolution operation with dynamic-k neighborhood points retrieval is used to fully capture the local features of the point cloud to make up for the shortcomings of the MLP method for local feature ex-traction, and the MLP hierarchical learning architecture is used to optimize the low efficiency of the dynamic graph convolutional neural network for global feature learning. Through the use of laser equipment and visible light equipment for data acquisition and feature matching, the self-built point cloud data of lightning arrester equipment in the substation is formed, and the proposed method is applied to the in-stance segmentation of this dataset, and the results show that the method of this paper has a very good performance in the accuracy and stability of the point cloud component segmentation task.

The point cloud segmentation strategy DGCPointnet proposed in this study optimizes the model and improves the point cloud segmentation. This method proposes a dynamic k-neighborhood point retrieval based approach with simultaneous tandem local and global feature extraction and finally detection in power equipment dataset. This method not only provides new ideas for subsequent point cloud segmentation feature processing, but also pioneers the combination of point cloud part segmentation with power equipment, which provides support for subsequent research.

II. MATERIALS AND METHODS

Modeling encompasses a number of processes, as shown in Fig. 1. This model not only designs the dynamic k-neighborhood retrieval method for local feature extraction, but

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3468012

also optimizes the global feature strategy, which improves the detection segmentation accuracy while optimizing the model size as well as the detection speed.

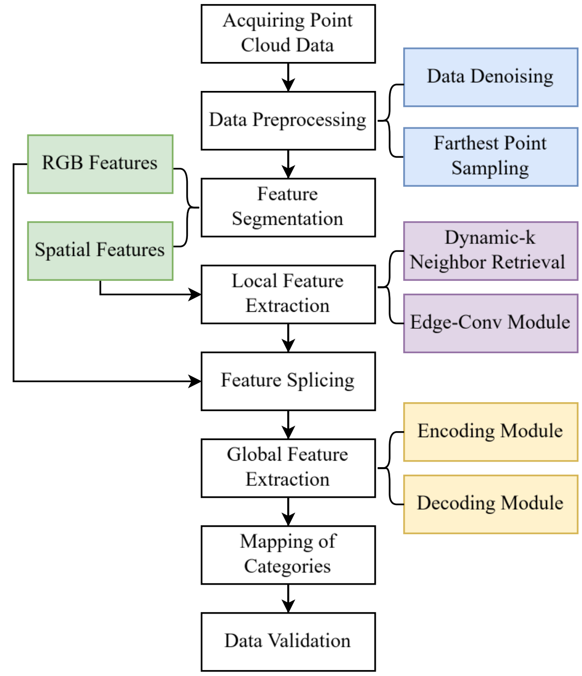


FIGURE 1. **Model design flowchart.**

Firstly, the point cloud data are acquired. After data processing such as denoising and downsampling, the point cloud features are segmented. Local feature extraction based on "dynamic k-neighborhood retrieval" and "Edge-Conv module" is applied to different dimensional data. After that, the features are spliced and global feature extraction is performed. Finally, the segmentation results are returned and verified.

The DGCPointnet takes into account both the recognition of local features and the extraction of global features of the point cloud, and Fig. 2 shows the flowchart of the model.

As shown in Fig. 2, the input point cloud data is first obtained; the point cloud spatial coordinate data as well as the data of other dimensional features of the point cloud (e.g., color information, etc.) are separated; the local features in the point cloud coordinate data are extracted by using edge Conv; the extracted features are spliced with the feature dimensional data; the spliced data are subjected to the global feature extraction; and the classification information of each point is obtained.

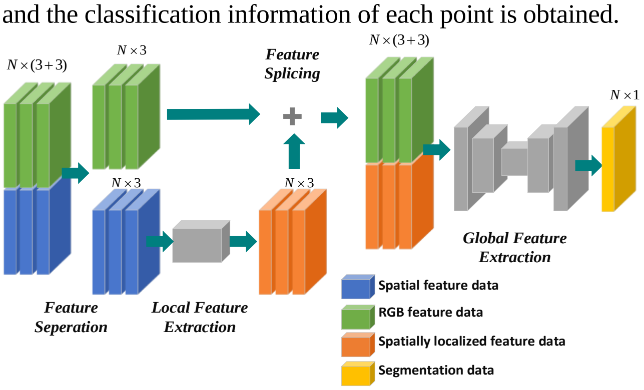


FIGURE 2. **Schematic flow of the DGCPointnet model.**

In this case, the dataset is set to be $X=\{X_1, X_2,..., X_{total}\}$, where each X_i object instance is represented by a point cloud. Use $P=\{p_1, p_2,..., p_N\}=X_i$ to represent the point cloud of an instance, the data format of the point cloud is $(N, D+F+L)$.

N denotes that each instance contains N points, and the input data format for deep learning is fixed, so all point cloud data needs to undergo pre-data processing to downsample it to the same number of point clouds;

D is the spatial coordinate data of each point, since coordinate data in xyz three dimensions are needed to determine the position in 3D space, so here $D=3$;

F is the feature data of each point, which can characterize the additional features of each point, such as the color information of RGB, or temperature data, depth data and so on;

L is the label data of each point, which means that each point belongs to a category, or belongs to a certain part.

A. EXTRACT LOCAL FEATURES

Edge Conv is a graph convolution structure for point clouds, which dynamically builds a graph structure on each layer of the network, characterizes each point as a center point with its edge features with respect to each neighboring point, and then aggregates these features to obtain a new representation of the point. Multiple edge Conv modules can be stacked head-to-tail to obtain a multilayered, semantically richer representation. Because Edge Conv mainly extracts the local information of the point, and the color information for different points in the object has no obvious local connection, so set the input feature dimension as $D=3$, that is, the three-dimensional spatial coordinates, the subsequent extraction of spatial local information after the data and the RGB data splicing, for the local feature extraction model schematic shown in Fig. 3.

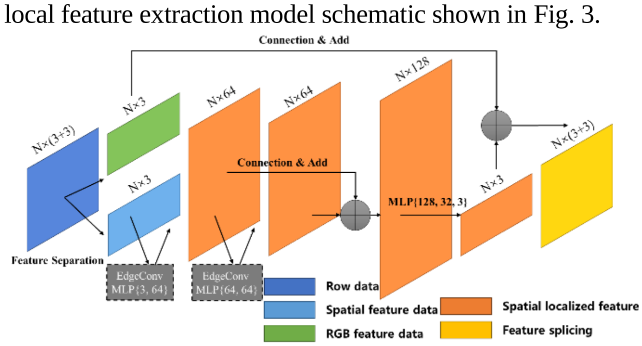


FIGURE 3. **Flowchart of local feature extraction.**

1) DYNAMIC-K NEIGHBORHOOD RETRIEVAL

Our research extracts local features through graph convolution, and the key is to find the appropriate neighborhood interval for each point to extract local features. Generally, k-nearest neighbors or spherical neighbors were commonly used to search for neighborhood points. We have proposed a Dynamic-K Neighborhood Point Retrieval method based on edge feature extraction, which can dynamically update the size of the retrieval neighborhood and find the trend of large changes in the normal vector for edge local feature extraction. The specific process is shown in Fig. 4.

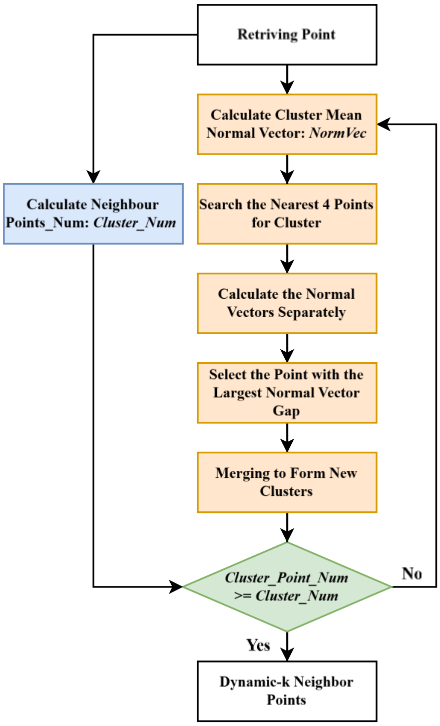


FIGURE 4. Flowchart of dynamic-k neighborhood retrieval.

Firstly, calculate the volume V of the bounding box and the number of point clouds $Point_{num}$ for the point cloud data, and obtain the local density of the point cloud data as shown in (1).

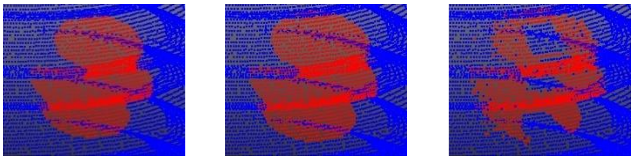
$$density = \frac{Point_{num}}{V} \quad (1)$$

Afterwards, for the point P_i to be searched, find the closest point P_j and calculate its interval bounding box volume V'. Calculate the number of iteration points based on local features, as shown in (2).

$$Cluster_{num} = density \times V' \quad (2)$$

Retrieve the 4 nearest points to P_i and select the point with the largest difference with the normal vector of point P_i as the clustering point. View the clustered cluster as a new whole and retrieve the 4 nearest points and select the point with the largest difference from the average normal vector of the cluster as the clustering point. This keeps looping the above iterations to select $Cluster_{num}$ points as clustered clusters.

For insulator data with obvious changes in local features, some of its point cloud data is taken as an example, as shown in Fig. 5.



(a) K-Nearest (b) Spherical Neighbor (c) Dynamic-k Neighbor

FIGURE 5. Flowchart of local feature extraction.

The left image in Fig. 5 shows k-neighborhood retrieval, the middle image shows spherical neighborhood retrieval, and the right image shows Dynamic-k Neighborhood retrieval for edge feature extraction. It can be seen that due to the addition of normal vector constraints in this method, points that are close to the normal vector of the points to be retrieved are filtered out, which can better characterize the local features of complex point clouds with local structures. For example, in the insulator point cloud data shown in the Fig. 5, this method can better retrieve the neighboring edge points of the points to be retrieved, facilitating subsequent local feature extraction.

2) EDGE-CONV LOCAL FEATURE EXTRACTION

According to Fig. 3, the local feature extraction part first performs point cloud data input; separates spatial coordinate dimension data and feature dimension data; for coordinate dimension data, uses the Edge Conv module to perform feature extraction, and the extracted result rises from 3 dimensions to 64 dimensions; performs another Edge Conv of this result, and splices together the results of the two feature extractions mentioned above; and then performs data downscaling to 3 dimensions by multiple MLPs to downscale the data to 3 dimensions and stitch the feature dimension data together to return a 6-dimensional point cloud data.

The Edge Conv module is shown in Fig. 6, the input of the module is $N \times F$, which N represents the number of input point clouds, which F represents the dimension of the input point cloud, and the output of an Edge Conv module is in the form of $N \times a_n$.

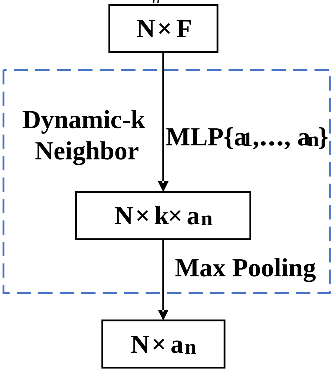


FIGURE 6. Graph Edge Conv schematic flowchart.

In the Edge Conv module, assume that a directed graph is $Graph = (\nu, \varepsilon)$ used to represent the local structure of the point cloud, where the vertices are $\nu = \{1, 2, ..., n\}$, and the edges are $\varepsilon = \nu \times \nu$. First, a k-nearest neighbor graph is created in 3D space, assuming that the closest points $p_{j_1}, ..., p_{j_k}$ to p_i contains many directed edges $(i, j_1), ..., (i, j_k)$, then the edge features can be defined as: $e_{ij} = h_{\Theta}(x_i, x_j)$, where h_{Θ} is a nonlinear function consisting of a learnable parameter h_{Θ}, and finally a symmetric aggregation operation is added to the edge features associated with all the edges, starting from each vertex.

$$p_i^{'} = \underset{j:(i,j)\in\varepsilon}{\square} h_\Theta(p_i, p_j) \tag{3}$$

For h_Θ and $\square$ in equation, different modes can be chosen to realize different effects. For example, fixing the number of neighboring points, a similar convolutional implementation can be followed, as shown in (2).

$$p_{im}^{'} = \sum_{j:(i,j)\in\varepsilon} \theta_m p_j \tag{4}$$

where $\Theta = (\theta_1, \theta_2, \ldots, \theta_M)$, M denotes the number of convolutional kernels and also the number of output channels, where the aggregation operation is $\square = \sum$.

In the light of local and global features, the choice was made to adopt a combination of global information captured centered on p_i, while considering the use of $p_j - p_i$ links between concerns and neighborhoods, as shown in (3).

$$e_{ij} = h_\Theta(p_i, p_j - p_i) \tag{5}$$

After that a certain point characterizing the new features will be obtained fed into the perceptual machine and using the max pooling, which is shown in (4).

$$p_{im}^{'} = \max_{j:(i,j)\in\varepsilon} \theta_m e_{ij} \tag{6}$$

B. GLOBAL FRATURE EXTRACTION AND PART SEGMENTION

After completing the local feature extraction, the extracted local spatial feature information needs to be spliced with the color features, followed by the global feature analysis. The overall structure of global feature recognition is shown in Fig. 7.

It can be seen that the model structure introduces a multi-level structure, which ensures that more detailed features can be extracted from complex point cloud data. The global feature recognition model is divided into two parts: encoding and decoding. For the encoding part, a multi-layer ensemble feature layer is applied to achieve the effect of point cloud feature extraction; for the decoding part, multiple feature propagation layers are applied, and the features of the ensemble feature layer are connected to the propagation layer across the layers, and finally the MLP is used to segment the objects based on the collected point cloud features.

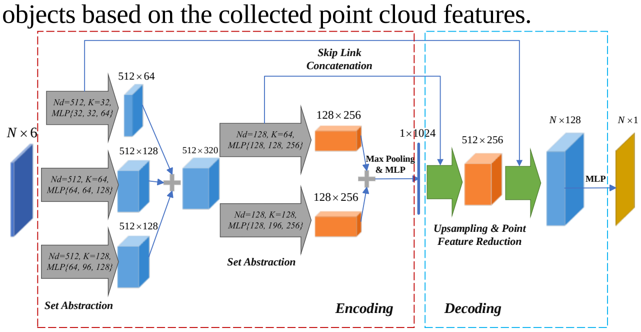


FIGURE 7. Schematic diagram of global feature extraction and part segmentation process.

1) FEATURE EXTRACTION AND DIMENSION REDUCTION

In this paper, the ensemble feature layer is applied for feature extraction and dimensionality reduction, and the structure design is shown in Fig. 8.

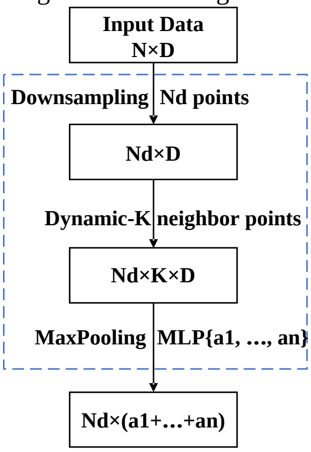


FIGURE 8. Schematic diagram of the principle and flow of the Set Abstraction Layers.

The input to each ensemble feature layer is $N \times D$, N is the number of input points, and D is the feature dimension of the input points. It contains three processing steps.

① Downsampling.

Farthest Point Sampling (FPS) is used to downsample the point cloud. Compared to random downsampling, FPS can cover the entire point set and ensure good uniformity, and compared to voxel downsampling, FPS does not change the coordinates of the original points and the number of downsampled points N_d can be specified.

② Localized retrieval.

The Ball query method is used to generate local regions N_d, i.e., for each point after downsampling, a spherical neighborhood point cloud is retrieved, and each local region contains up to K points.

③ Feature extraction.

Use the point cloud convolutional layer, the convolutional layer contains MLP and pooling operations, this step completes the point cloud data feature extraction.

2) UPSAMPLING AND POINT FEATURE REDUCTION

After the first half of the encoding part, the global features of the point cloud data are obtained, while for point cloud segmentation, the "point features" of each point before downsampling need to be accurate. In order to solve the above problem, the model uses multiple feature propagation layers and passes the features of the ensemble feature layer by connecting them across the layers.

In the feature propagation layer, point features are propagated from N_{l-1} points to N_l points, where N_l and $N_{l-1}(N_{l-1} \le N_l)$ are the size of the point set for the input and output of the ensemble feature layer, respectively, and the size of the point set for the output and input of the corresponding feature propagation layer.

First, an inverse distance weighted average based on k nearest neighbors is used (as in (5), $u = 2, k = 3$ by default). For the encoded data, find a set of points that are neighbors in the original spatial coordinates which contains k points $p_1, p_2,.., p_k$, and then complete the computation of the features for the new point p_x based on these points k by using the following interpolation equation.

$$f^{(j)}(p)=\frac{\sum_{i=1}^{k} w_i(p) f_i^{(j)}}{\sum_{i=1}^{k} w_i(p)} \tag{7}$$

Where,

$$w_i(p)=\frac{1}{d(p,p_i)^u}, j=1,2,...,C \tag{8}$$

C in the above equation denotes the feature layer dimension. Based on this, the upsampling of point features is completed. However, since the encoded data itself is the global features of the point cloud and does not represent the initial spatial and color features, the up-sampled data can only be used to characterize the global information. As shown in Fig. 8, the features obtained through the first layer of interpolation are the post-coding features, but the feature information in the second aggregate feature layer (gray arrows) in Fig. 7 is also needed to restore the point features. The cross-layer join is to splice the representations from the corresponding ensemble feature layer during the encoding process into the corresponding feature propagation layer. It connects the interpolated features at N_{l-1} points with the cross-layer join point features from the ensemble feature layer and merges the features. In this way, the pre-encoding information features can be added to the global features obtained, and the two information can be superimposed to obtain a "point feature" for each point.

Finally, the connected features are passed through a "unit point network", similar to convolution-by-convolution in a convolutional neural network, with a number of shared fully-connected layers and a ReLU activation function layer to update the feature vectors at each point. The process is repeated until these features are propagated to the original set of points. Finally, the MLP is used to perform the segmentation task of the point cloud for each point feature.

III. EXPERIMENTS AND RESULTS ANALYSIS

Overall, this paper uses the Edge Conv module for local feature extraction and the MLP structure for global feature recognition. This design allows the model to take into account both local and global information of the point cloud data, thus improving the accuracy and efficiency of the point cloud component segmentation task.

The method proposed in this paper is applied to the point cloud component segmentation of substation surge arrester equipment. By extracting the point cloud of the arrester equipment instances, it is convenient to carry out the subsequent work of defect diagnosis, fault detection, and high-temperature warning about the arrester. The data collected from the substation point cloud data is shown in Fig. 9. The collected substation lightning arrester data has a total of data 430 point cloud data after the farthest point down sampling down to 8192 points, the dataset can be accessed through IEEEDataPort[22].

FIGURE 9. Scanning point cloud schematic of substation.

When processing the data, the initial segmentation of the region was first performed using a clustering algorithm, after which the lightning arrester was divided and the components were labeled. Next, the obtained data are subjected to data enhancement operations such as scaling, panning and downsampling, and in order to maintain the same number of point clouds for all the data, the farthest point downsampling method is used to standardize the number of point clouds as $N = 2048$. The pre-processed and supervised classified point cloud is shown in Fig. 10.

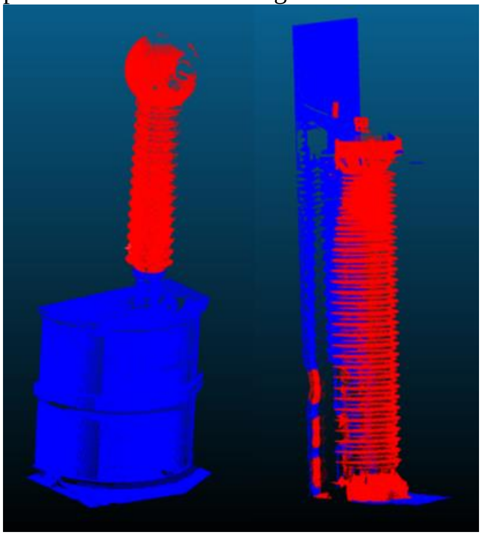


FIGURE 10. Lightning arrester point cloud schematic (left and right are different types of arresters).

The hybrid model described in the previous section is used to perform the segmentation task of point cloud parts, and Pointnet and Pointnet++ are used as model comparisons. In order to avoid the influence of parameters, the above models are set uniformly, the optimizer is set to "Adam", the learning rate is 0.001, the learning rate attenuation is 0.0001, etc., and the number of point clouds searched in the neighborhood of the Edge Conv module is set $k = 40$.

In the testing of data, the evaluation metrics are concerned with the precision and accuracy of the test, which are important indicators for judging the results of the model operation. For the testing of point cloud data, this paper proposes the following test metrics: Test Accuracy, Class Avg mIoU, and Instance Avg mIoU. Test Accuracy is an intuitive evaluation metric that calculates the ratio of all correctly predicted samples to the total number of samples.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

TP (True Positives) is the number of points correctly predicted to be in the positive category. TN (True Negatives) is the number of points correctly predicted to be in the negative category. FP (False Positives) is the number of points incorrectly predicted to be in the positive category. FN (False Negatives) is the number of points incorrectly predicted to be in the negative category.

The Class Avg mIoU is a measure of the degree of overlap between predicted and real regions in the segmentation task for different category point clouds.

$$\text{mIoU} = \frac{1}{N}\sum\nolimits_{c=1}^{N} \frac{TP_c}{TP_c+FP_c+FN_c} \tag{10}$$

N is the total number of categories. The other parameters are shown in equation (7) and refer to the relevant meanings in category c .

The Instance Avg mIoU is a measure of the degree of overlap between the predicted and real regions in the segmentation task for different instance point clouds.

$$\text{IoU} = \frac{TP}{TP+FP+FN} \tag{11}$$

For each instance, the calculation is similar to the Class Avg mIoU, but for a single instance rather than the entire category. In the following practical tests, the difference between the Class Avg mIoU and the Instance Avg mIoU for the same test is not more than 0.1%, so the Test Accuracy and t Class Avg mIoU are uniformly used as the basis for judging the test results of the point cloud data in the subsequent tests.

In the DGCPointnet, the extraction of local features of the point cloud is optimized by using the introduction of Edge Conv, because Edge Conv is mainly used for feature extraction through the neighboring points in the 3D spatial coordinates, so firstly, only the 3D spatial coordinates of the point cloud are used for training and testing, with reference to the point cloud format in the Methods, enter The point cloud format is (2048×(3+0+1)), the number of point clouds $N=2048$, the spatial information dimension $D=3$, the feature dimension $F=0$ and the label dimension $L=1$. At this point, the accuracy of the test data during the training process is obtained as shown in Fig. 11 and Fig. 12.

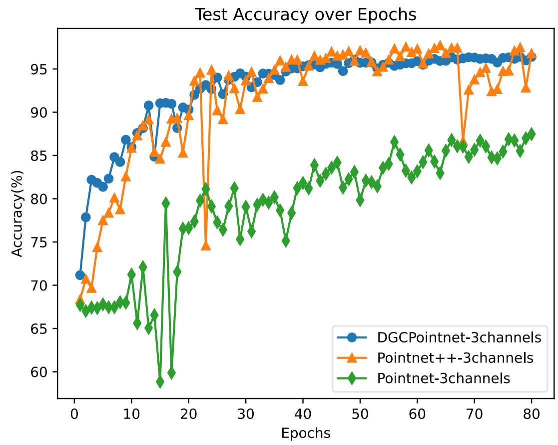


FIGURE 11. Test Accuracy considering spatial features.

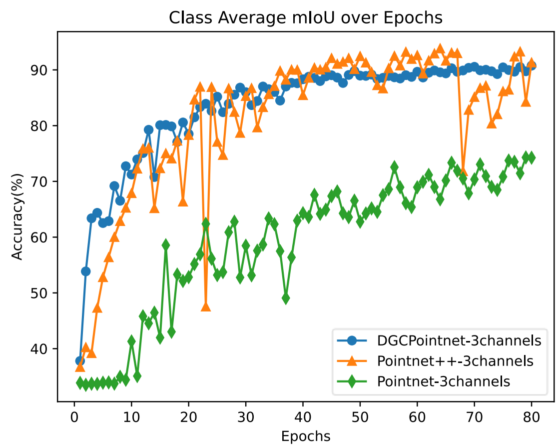


FIGURE 12. Class Average mIoU considering spatial features.

As shown in the above two figures, it can be found that the Pointnet model has a large gap in both the test accuracy and the results of the analogous average intersection and comparison, and its stability is not well guaranteed; while Pointnet++ and the DGCPointnet can improve the test accuracy in a shorter period of time. However, Pointnet++ has obvious fluctuations in the training, it can be seen that the magnitude of fluctuations is very large, even after stabilization, with the increase in the training epoch, its up and down range is much larger than the DGCPointnet, while the DGCPointnet not only has to achieve an accuracy of 90% and above, the stability of the model is also much better than the other two models. As can be seen in Fig. 11 and 12, the overall accuracy of the category-averaged intersection and merger ratios is lower compared to the test accuracy. The results of each category test considered in the category intersection and merger ratio are more representative of the test accuracy of the model details, as some of the misdetections due to small offsets have a larger penalty value for mIoU.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3468012

After this, add the existence of its own RGB three-channel color data, at this time feature dimensions $F = 3$, the results obtained as shown in Fig. 13 and Fig. 14.

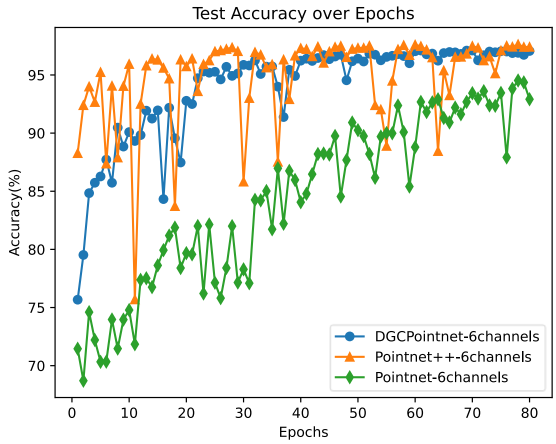


FIGURE 13. **Test Accuracy considering spatial and color features.**

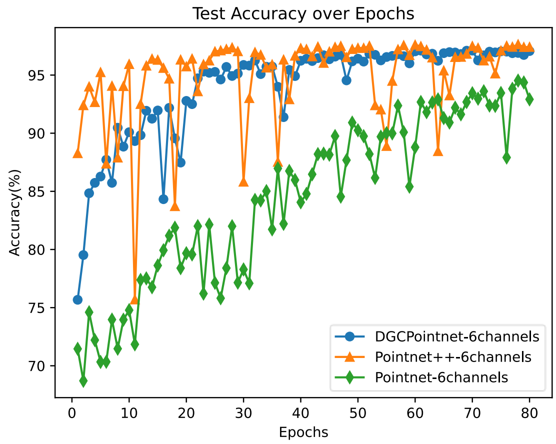


FIGURE 14. **Class Average mIoU considering spatial and color features.**

It can be seen that after adding the three dimensional features, the overall accuracy of the three models is further improved, but as analyzed above, the accuracy of the Pointnet model is still not high; the Pointnet++ model has a higher accuracy at the be-ginning of the training compared to the spatial coordinates only, but the stability is poorer and the amplitude of fluctuation is larger; although the accuracy of the DGCPointnet is slightly lower at the beginning of the training, the accuracy gradually increases and quickly stabilizes at 90% and above as the training progresses. Although the accuracy of the DGCPointnet in this paper is a little lower at the beginning of training, with the training, the accuracy of the DGCPointnet gradually increases and quickly stabilizes at 90% and above.

After training, this paper uses different models to detect the data in the test set, and each round of detection is carried out 10 times and solves the mean value, maximum value, minimum value, standard deviation and other parameters for detailed comparison, and the comparison results are shown in Table I, Table II, Fig. 15 and 16.

TABLE I
TABLE OF TEST ACCURACY RESULTS

Feature Dimension	Algorithm	Test Accuracy	Standard Deviation
Consider only the spatial feature dimension	Pointnet	86.13±1.46	0.9780
	Pointnet++	94.86±2.64	1.8516
	DGCNN	95.32±0.92	0.8462
	DGCPointnet	96.19±0.44	0.1982
Consider the spatial and the color feature dimension	Pointnet	92.83±4.92	1.8809
	Pointnet++	96.98±1.85	0.7850
	DGCNN	95.84±1.13	0.6783
	DGCPointnet	96.86±0.59	0.2437

TABLE II
TABLE OF CLASS AVERAGE MIOU RESULTS

Feature Dimension	Algorithm	Test Accuracy	Standard Deviation
Consider only the spatial feature dimension	Pointnet	71.96±3.48	2.1301
	Pointnet++	87.01±6.64	4.2700
	DGCNN	88.37±2.13	1.9727
	DGCPointnet	90.01±0.76	0.4523
Consider the spatial and the color feature dimension	Pointnet	82.82±8.83	3.5480
	Pointnet++	90.50±5.48	2.3415
	DGCNN	89.79±1.87	1.2857
	DGCPointnet	91.47±1.40	0.6084

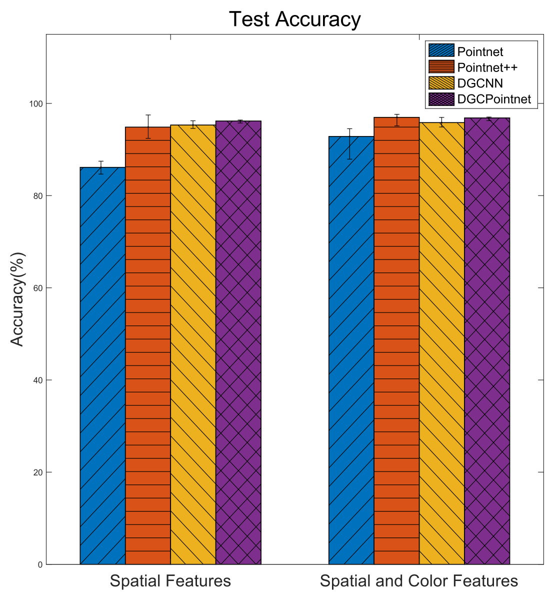


FIGURE 15. **Test Accuracy schematic.**

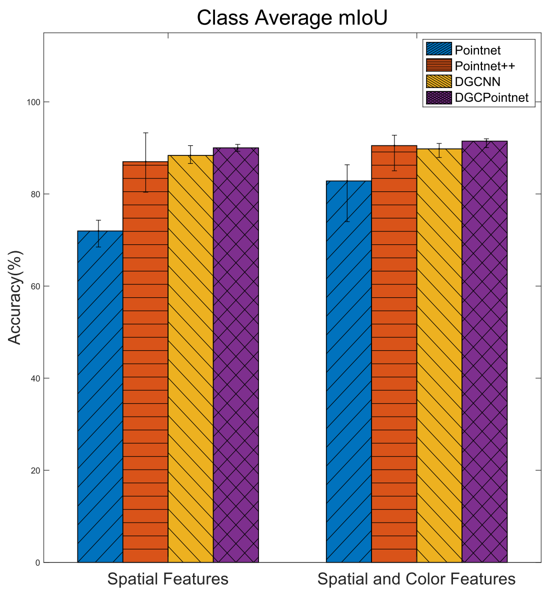


FIGURE 16. Class Average mIoU schematic.

Combined with the analysis in Table I and Table II as well as Fig. 15 and 16, it can be seen that: under the above two channel models as well as the two test rubrics with a total of four scoring metrics, Pointnet does not perform well in terms of both accuracy and stability; whereas, Pointnet++ ranks the top in terms of the maximum accuracy in all the tests; DGCNN performs well with only spatial features, second only to DGCPointnet; however, in general, the DGCPointnet leads the remaining three models in terms of both the average test accuracy as well as stability ahead of the remaining two models. In Fig. 15 and Fig. 16, it can be seen that the upward and downward fluctuations of the accuracy of the DGCPointnet are very small, and the standard deviation of the DGCPointnet is much lower than that of the remaining three models as can be seen from the tabular data.

In summary, the DGCPointnet shows excellent performance in the task of segmenting the point cloud components of substation lightning arrester equipment. The above results prove that the DGCPointnet achieves superior performance in the lightning arrester point cloud component segmentation task, which will have a broader application prospect in the future defect diagnosis and high temperature warning tasks.

IV. CONCLUSION

In this paper, a point cloud component segmentation method that fuses local feature extraction with global feature extraction is proposed, and tested and validated for the point cloud data of lightning arrester equipment in electricity equipment. The method first separates the point cloud data with multidimensional features into spatial coordinate dimension data and other feature dimension data (e.g., color data), and applies the local feature extraction model constructed by the Edge Conv map convolution structure to the point cloud data in the spatial coordinate dimension; then, the extracted result is spliced with the data in the other feature dimensions, and the spliced data are fed into the global feature extraction model constructed by the multilayer ensemble feature layer and feature propagation layer; finally, after the fusion of multiple ensemble feature layers and feature propagation layers, a global feature extraction model is constructed. feature extraction model; finally, the classification results are output to the label dimension of each point cloud after multiple multilayer perceptrons. By comparing and analyzing with the traditional algorithms, the experimental results prove that its performance is significantly better than the Pointnet and Pointnet++ models. The specific conclusions obtained in this paper are as follows:

1) A dynamic-kneighborhood point retrieval is proposed in the model to find different neighborhood retrieval regions for different points by two features of the point cloud, local density and normal vector trend. The proposed method can better characterize the local features you of different points and provide solution ideas for subsequent tasks.

2) Fast training speed. By monitoring the training process, it can be seen that due to the extraction and fusion of local and global feature information, the model in this paper can quickly reach a high level of testing accuracy. When only spatial coordinate data is considered, the training speed is better than the remaining two models; after adding color dimension data, the model in this paper significantly outperforms Point-net in terms of accuracy, and significantly outperforms Pointnet++ in terms of stability.

3) Excellent performance in test accuracy and stability. In the final test, several experiments are repeated and it can be seen that the DGCPointnet always maintains high accuracy and stability. In contrast, although Pointnet++ is ahead in terms of maximum accuracy, its average accuracy and stability are lower than that of the paper's model. This result fully demonstrates the power of the DGCPointnet in dealing with complex and changing real-world environments.

4) Point cloud feature fusion analysis. Compared with the fusion at the decision layer, the DGCPointnet directly fuses at the data layer and the feature layer, which can take the differences of features between different data and the fusion effect into account more, and significantly improve the accuracy and stability of part classification.

The research in this paper provides a new approach to solve point cloud component segmentation. The method provides a dynamic k-neighborhood retrieval based on local density and normal vector trends, proposes a point cloud component segmentation by concatenating local and global features in series, and also combines the point cloud segmentation with power equipment to provide effective support for subsequent tasks. The model can learn more detailed features of point cloud data, and for subsequent more complex point cloud data, this method proposes the idea of combining local and global feature extraction can be used not only in part segmentation, but also in point cloud generation, noise removal and other point-cloud processing fields. In the future, the local feature extraction method and the feature fusion method can be optimized, and the features can be fused using, for example, the attention mechanism to better ensure the feature extraction effect.

In conclusion, this method not only provides a better point cloud component segmentation method, but also lays the foundation for the subsequent maintenance of power equipment, while the innovation of point cloud segmentation method can provide ideas for the subsequent research.

REFERENCES

[1] E. Grilli, F. Menna, and F. Remondino, "A REVIEW OF POINT CLOUDS SEGMENTATION AND CLASSIFICATION ALGORITHMS," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLII-2-W3, pp. 339–344, Feb. 2017, doi: 10.5194/isprs-archives-XLII-2-W3-339-2017.

[2] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, F. Poiesi, and E. Ricci, "Compositional Semantic Mix for Domain Adaptation in Point Cloud Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14234–14247, Dec. 2023, doi: 10.1109/TPAMI.2023.3310261.

[3] L. Rauch and T. Braml, "Semantic Point Cloud Segmentation with Deep-Learning-Based Approaches for the Construction Industry: A Survey," *Appl. Sci.*, vol. 13, no. 16, p. 9146, Aug. 2023, doi: 10.3390/app13169146.

[4] A. Kuriyal, V. Kumar, and B. Lohani, "pCTFusion: Point Convolution-Transformer Fusion with Semantic Aware Loss for Outdoor LiDAR Point Cloud Segmentation," *SN Comput. Sci.*, vol. 5, no. 3, p. 272, Feb. 2024, doi: 10.1007/s42979-024-02627-5.

[5] LU Jian, JIA Xurui, ZHOU Jian, LIU Wei, ZHANG Kaibing, and PANG Feifei, "A review of deep learning based on 3D point cloud segmentation," Control and Decision Making, vol. 38, no. 3, pp. 595–611, 2023, doi: 10.13195/j.kzyjc.2021.1648.

[6] F. Poux, C. Mattes, Z. Selman, and L. Kobbelt, "Automatic region-growing system for the segmentation of large point clouds," *Autom. Constr.*, vol. 138, p. 104250, Jun. 2022, doi: 10.1016/j.autcon.2022.104250.

[7] J. Zhang, X. Wang, Y. Li, and Y. Liu, "CHBS-Net: 3D Point Cloud Segmentation Network with Key Feature Guidance for Circular Hole Boundaries," *Machines*, vol. 11, no. 11, p. 982, Oct. 2023, doi: 10.3390/machines11110982.

[8] X. Yang, Y. Wen, S. Jiao, R. Zhao, X. Han, and L. He, "Point Cloud Segmentation Network Based on Attention Mechanism and Dual Graph Convolution," *Electronics*, vol. 12, no. 24, p. 4991, Dec. 2023, doi: 10.3390/electronics12244991.

[9] TIAN Yujie, GUAN Youqing, and GONG Rui, "A Robust Deep Neural Network for Multi-Feature Point Cloud Classification and Segmentation," Computer Engineering, vol. 47, no. 11, pp. 234–240, 2021, doi: 10.19678/j.issn.1000-3428.0060004.

[10] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, and A. Mian, "Deep Learning Based 3D Segmentation: A Survey." arXiv, Jul. 26, 2023. doi: 10.48550/arXiv.2103.05423.

[11] Z. Wu *et al.*, "3D ShapeNets: A Deep Representation for Volumetric Shapes," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920. Accessed: Sep. 14, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Wu_3D_ShapeNets_A_2015_CVPR_paper.html

[12] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 922–928. doi: 10.1109/IROS.2015.7353481.

[13] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6411–6420. Accessed: Sep. 14, 2023. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Thomas_KPConv_Flexible_and_Deformable_Convolution_for_Point_Clouds_ICCV_2019_paper.html

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660. Accessed: Sep. 14, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Qi_PointNet_Deep_Learning_CVPR_2017_paper.html

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Sep. 14, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html

[16] LIANG Zhenhua and WANG Feng, "Attention weighted feature aggregation PointNet network for part segmentation," Computer Application Research, vol. 40, no. 5, pp. 1571-1576+1582, 2023, doi: 10.19734/j.issn.1001-3695.2022.08.0423.

[17] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5565–5573. Accessed: Sep. 14, 2023. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Zhao_PointWeb_Enhancing_Local_Neighborhood_Features_for_Point_Cloud_Processing_CVPR_2019_paper.html

[18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *ACM Trans. Graph.*, vol. 38, no. 5, p. 146:1-146:12, Oct. 2019, doi: 10.1145/3326362.

[19] T. He, C. Shen, and A. V. D. Hengel, "Dynamic Convolution for 3D Point Cloud Instance Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, 2022, doi: 10.1109/TPAMI.2022.3216926.

[20] SHI Yi, WEI Dong, SONG Qiang, HE Lian, and WANG Jingshuang, "Point cloud data classification and segmentation network based on dynamic graph convolution and discrete Hartley transform different pooling," Computer Applications, vol. 42, no. S1, pp. 292－297, 2022.

[21] QU Yanlin, WANG Yue, ZHANG Qian, and HAN Shaokun, "Point Cloud Analysis Method Based on Spatial Feature Attention Mechanism," Advances in Lasers and Optoelectronics, vol. 60, no. 24, pp. 240–250, 2023.

[22] Haoyu Song, "Lightning arrester point cloud segmentation dataset." IEEE Dataport, 2024. [Online]. Available: 10.21227/2gr7-vz15

ZE ZHANG was born in 1978 in Suqian City, Jiangsu Province, he received his Bachelor of Engineering degree in 2000 from Hehai University, majoring in power system and automation.

He has been engaged in production and operation management for a long time as the deputy chief engineer of Taizhou Power Supply Branch of State Grid Jiangsu Electric Power Co. As a project leader, he has undertaken two provincial science and technology projects and participated in five science and technology projects, and is familiar with the R&D implementation and management process of science and technology projects.

Since 2021, he has won 15 various scientific and technological awards, issued one national standard as the second completion unit, and led the issuance of two enterprise standards.

DANYIN WEN was born in 1989 in Jingjiang City, Jiangsu Province. She received her M.E. degree in Power System and Automation from Hehai University in 2013.

From August 2013, she has been working in the Substation Operation and Maintenance Center of Taizhou Power Supply Branch of State Grid Jiangsu Electric Power Company Limited, where she has participated in writing a safety skills practical training textbook. Her research interests include substation operation and maintenance management, infrared thermal imaging technology, intelligent inspection construction and application, and automatic acceptance of anti-error lockout.

Dr. WEN's awards and honors include Jiangsu Provincial Skilled Worker, State Grid Corporation Skilled Worker, and Taizhou May Day Labor Medal.

WEIJUN SHI was born in 1988 in Xinghua City, Jiangsu Province, China, he received his M.E. degree from Hehai University, specializing in power system and automation, in 2013..

From August 2013, he has been engaged in substation operation and maintenance, authorized 4 patents, and participated in writing a textbook for practical training of safety skills. His research interests include substation O&M management, substation 3D modeling and intelligent inspection construction and application.