International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 7700-7711

https://doi.org/10.70517/ijhsa464663

Deep learning modeling and optimization of three-dimensional point cloud size characteristics of cables

Fengfei Su[1,*], Zhen Xu[2], Yiqing Shi[3], Gang Chen[1], Lei Lei[3,4] and Qingyun Cheng[3,4]
[1] State Grid Weinan Power Supply Company, Weinan, Shaanxi, 714000, China
[2] State Grid Shaanxi Electric Power Co., Ltd., Xi'an, Shaanxi, 710000, China
[3] State Grid (Xi'an) Environmental Technology Center Co., Ltd., Xi'an, Shaanxi, 710000, China
[4] Electric Power Research Institute of State Grid Shaanxi Electric Power Co., Ltd., Xi'an, Shaanxi, 710000, China
Corresponding authors: (e-mail: fengfei1115@126.com).

Abstract Within the theoretical framework of 3D point cloud data, this paper proposes the use of laser radar sensors to collect 3D point cloud data of cable size features. Due to the presence of redundant interference data in the data, an adaptive filtering algorithm is used to preprocess the data. To better extract cable dimension features, a cable dimension feature extraction model based on the ADGCNN network is designed. Through feature enhancement and fusion, a deep learning training model for cable dimension features is established. To address the issue of suboptimal model training performance, the Adadelta optimization algorithm is applied to optimize the model, and its optimization effects are verified and analyzed. The accuracy rate before model optimization was 0.894. After applying the Adadelta optimization algorithm, the model's accuracy rate improved to 0.975, confirming the effectiveness of the Adadelta optimization algorithm in model optimization.

Index Terms adaptive filtering algorithm, ADGCNN network, Adadelta optimization algorithm, cable dimension features, 3D point cloud data

I. Introduction

As urbanization accelerates, the importance of urban power systems has become increasingly evident. With the growth in electricity demand, the safe, stable, and reliable operation of power systems has become particularly critical [1]. However, the construction process of cable laying and the quality management of cable accessories installation involve practical challenges and complex issues, such as unstable installation quality due to manual installation, narrow environments, and insufficient lighting, all of which increase the difficulty of quality control [2]-[5]. Therefore, researching how to improve the installation quality of cable accessories, reduce fault frequencies, minimize power outages, and enhance engineering construction quality and management standards has become an urgent issue to address during cable laying processes [6], [7].

In recent years, the rapid development of precise measurement and analysis technologies such as 3D laser scanning, point cloud processing, and 3D modeling has provided new avenues for addressing these challenges [8], [9]. For example, 3D laser scanning technology can accurately and quickly measure the dimensions and positions of cables and cable accessories. Point cloud processing technology processes scan data to obtain the shapes and installation conditions of cables and cable accessories. 3D modeling technology uses scan data to create models, visually demonstrating the installation status of cables [10]-[13]. The application of these technologies can enhance the quality management efficiency of cable installations, reduce the operational risks of power systems, and drive technological innovation and upgrading in the power industry [14], [15].

Among these, a point cloud is a dispersed set of points that must undergo a series of processing steps to obtain the cable point cloud information required in this paper [16]. A 3D point cloud is a digital representation form composed of various information such as spatial coordinates and color textures [17]. Therefore, during calculations related to cable information, point cloud data can accurately reflect geometric parameters and other information about the cable, playing a crucial role. Literature [18] utilizes point clouds for three-dimensional structural visualization analysis of overhead communication cables between utility poles, combining machine learning methods to improve cable modeling accuracy, thereby assisting in the automation of infrastructure monitoring tasks. Literature [19] indicates that manually grinding cable joints results in surface shrinkage, uneven grinding, and irregular shapes. Therefore, a method for measuring the outer diameter parameters of cable joints based on three-dimensional point cloud processing is proposed, which not only accurately measures the outer diameter parameters but also preserves the original characteristics of the cable. Literature [20] designed a point cloud-based automated surveying method for power line corridors. By extracting cross-sections of the original point cloud in the newly

constructed reference system, it enables automatic detection of power cables and surrounding obstacles, providing crucial support for automated power management. Literature [21] addresses the welding process of cable multi-tube crossing structures in a common duct configuration, proposing a robot trajectory optimization model that combines weld seam 3D point cloud data with the NSGA-II optimization method. This model can accurately fit the cable model to achieve high-quality and efficient welding processes. Literature [22] introduces a cable joint point cloud remapping and image segmentation fusion method for cable joint defect measurement. By remapping preprocessed point cloud data onto a saliency image for explicit defect detection, it achieves good defect monitoring results. Reference [23] investigates a measurement method for stress cones in high-voltage cable joints based on 3D point clouds. By constructing a feature curve model that fits the radius of the circle to reflect the structural characteristics of the cable joint, and calculating the intersection points with the fitted straight line, the method enables automatic measurement of the cable joint point cloud. Although most studies have recognized that automated technologies such as machine learning can achieve high quality and efficiency when processing point cloud data, the data extraction and training methods are not yet fully developed and have room for further optimization.

This paper first uses a lidar sensor to collect 3D point cloud data of cable dimensional features. During the collection process, noise inevitably affects the data, resulting in interference information in the 3D point cloud data. Therefore, an adaptive filtering algorithm is used to filter, denoise, and register the data. After completing the data preprocessing, the ADGCNN network is employed to extract cable size features from the 3D point cloud data. The extracted features are then enhanced and fused to establish a deep learning training model for cable size features. Finally, to address the issue of suboptimal model performance, the Adadelta optimization algorithm is applied to optimize the model, and the effectiveness of the Adadelta optimization algorithm in model optimization is explored through loss values and accuracy rates.

II. Three-dimensional point cloud data technology

II. A. Theoretical Framework of 3D Point Cloud Data

Three-dimensional point cloud data, which consists of a large set of three-dimensional coordinate points, can accurately and realistically reflect the three-dimensional shape and surface details of an object, providing robust data support for practical applications across various fields [24]. Its objective is to reconstruct the geometric surface model of an object using a discrete set of three-dimensional points. It has found widespread application in fields such as medical image analysis, autonomous vehicle environmental perception, industrial inspection, and cultural heritage digitization.

Three-dimensional point cloud data, as an important data type in modern engineering, refers to a collection of points obtained through three-dimensional scanning equipment, which contain their coordinate information in three-dimensional space. These data points are typically distributed in a disordered manner in three-dimensional space, forming a cloud-like structure, hence the name “point cloud.” Mathematically, 3D point cloud data can be represented as a set of points P, where each point p_i has three-dimensional coordinates (x_i, y_i, z_i) as shown in Equation (1):

$$P = \{p_i = (x_i, y_i, z_i) \mid i = 1, 2, \cdots, N\} \tag{1}$$

In this context, N represents the total number of points. The acquisition of 3D point cloud data is a critical step in fields such as 3D reconstruction, machine vision, and remote sensing monitoring. In the early days, contact-based measurement methods were widely used, but with technological advancements, non-contact methods such as structured light scanning and laser scanning have gradually become the mainstream. Structured light scanning is fast and highly accurate, making it suitable for medium-sized objects, but it is sensitive to environmental conditions; laser scanning, on the other hand, measures three-dimensional coordinates by calculating the time of flight or phase difference of laser beams. Although it is more expensive, it offers greater adaptability.

II. B. Generation and Preprocessing of Point Cloud Data

II. B. 1) Generation of point cloud data

LiDAR is a sensor that measures distance by emitting laser pulses and measuring the time it takes for the reflected pulses to return. A LiDAR system typically consists of a laser transmitter, receiver, and processing unit, enabling the efficient generation of high-precision 3D point cloud data. During data collection, point cloud data inevitably becomes affected by noise, resulting in errors and outliers in the data. Noise sources include sensor errors, environmental interference, and data processing errors. Noise impacts the quality of point cloud data and requires filtering and denoising through preprocessing techniques.

II. B. 2) Point cloud data preprocessing

(1) Local noise level estimation

In 3D point cloud data, noise typically manifests as random point shifts. To accurately estimate the noise level in a local region before filtering, the local variance of the point cloud data is used to characterize the noise magnitude. Let the local region of the point cloud data be P_i, which contains N_i points $P_i = (x_i, y_i, z_i)$. The local noise level σ_i can be estimated using equation (2). That is:

$$\sigma_i^2 = \frac{1}{N_i}\sum_{j=1}^{N_i} \| P_j - \mu_i \|^2 \tag{2}$$

Among them, μ_i is the centroid of the local region, which is defined as shown in Equation (3). It is:

$$\mu_i = \frac{1}{N_i}\sum_{j=1}^{N_i} P_i \tag{3}$$

By calculating the noise level σ_i of each local region, we can provide a basis for subsequent filter parameter adjustment.

(2) Adaptive adjustment of filtering parameters

The core of adaptive filtering lies in dynamically adjusting the size and weight distribution of the filter kernel based on the local noise level. Let the standard deviation of the filter kernel be σ_r, and its adaptive adjustment is shown in Equation (4). That is:

$$\sigma_f = f(\sigma_i) \tag{4}$$

Among them, $f(\sigma_i)$ is a monotonically increasing function used to adjust the size of the filter kernel according to the local noise level σ_i. Therefore, by using a linear function or exponential function, it can be expressed as shown in Equation (5). That is:

$$\sigma_j = \alpha \cdot \sigma_i + \beta \tag{5}$$

In this case, α and β are adjustment coefficients determined based on the specific application scenario.

(3) Edge detection and retention

Edge points in a 3D point cloud typically manifest as points with relatively large local gradients. To retain edge details during the filtering process, a gradient-based edge detection method is employed. Let the local gradient of point P_i be ∇P_i, whose calculation is shown in Equation (6). It is:

$$\nabla P_j = \sum_{\lambda \in N_j} \frac{P_j - P_\lambda}{\| P_j - P_\lambda \|} \tag{6}$$

Among them, N_i represents the neighborhood point set of point P_i. When $\| \nabla P_i \|$ exceeds a certain threshold τ, the point is judged to be an edge point, and a higher weight ω_i is assigned to it during the filtering process, as shown in Equation (7). That is:

$$\omega_i = 1 + \gamma \max(0, \| \nabla P_i \| - \tau) \tag{7}$$

where γ is the adjustment coefficient.

(4) Filtering process

The final filtering operation is achieved through weighted averaging. Given the filtering result P' of point P_i, the calculation is shown in Equation (8). It is:

$$P_j' = \frac{\sum_{k \in N_j} \omega_k \cdot P_k \cdot \exp\left(-\frac{\| P_j - P_k \|^2}{2\sigma_j^2}\right)}{\sum_{k \in N_j} \omega_k \cdot \exp\left(-\frac{\| P_j - P_k \|^2}{2\sigma_j^2}\right)} \tag{8}$$

Among them, ω_i is the weight of the k th point, and σ_i is the standard deviation of the adaptive adjusted filter kernel.

II. C.Three-dimensional point cloud data filtering

Point cloud filtering is a preprocessing technique for point cloud data that improves data quality by removing noise, smoothing data, and retaining important features. As shown below:

(1) Neighborhood search. Find the set of points within the neighborhood of each point.

(2) Calculate the mean and standard deviation. Calculate the mean and standard deviation of the distances between each point and its neighboring points.

(3) Set the threshold. Set the distance threshold based on the mean and standard deviation.

(4) Remove outliers. Remove points whose distances exceed the threshold.

II. D.Point Cloud Registration

Point cloud registration is the process of aligning multiple point cloud data sets collected from different perspectives or at different times into the same coordinate system. The goal of point cloud registration is to find a rigid body transformation (including rotation and translation) such that one point cloud (source point cloud) is aligned as closely as possible with another point cloud (target point cloud) after the transformation. The rigid body transformation model is shown in Equation (9). It is:

$$T = [R \mid t] \tag{9}$$

In this context, R is the rotation matrix, and t is the translation vector. The registration process determines the optimal rigid transformation by optimizing a distance metric (such as Euclidean distance).

III. Deep Learning Model Training, Construction, and Optimization

III. A. Building Deep Learning Training Models

III. A. 1) Extraction of cable size features from point cloud data

Three-dimensional point cloud data containing cable size features exhibit sparsity and irregularity. Existing methods for extracting cable size features do not account for the varying importance of different feature channels. To address this, we propose an ADGCNN network combined with an attention mechanism, introducing a channel attention module into the EdgeConv structure to assign different weights based on the importance of feature channels, thereby enhancing the network's expressive capability. Additionally, we employ a method combining max pooling and average pooling to address the disordered nature of point clouds, preventing information loss that would occur if only max pooling were used.

(1) Network Learning Process

The ADGCNN network structure is shown in Figure 1. The three-dimensional point cloud data containing cable dimensions is sequentially input into 32-dimensional, 32-dimensional, 64-dimensional, and 128-dimensional EdgeConv modules, respectively, to obtain 32-dimensional, 32-dimensional, 64-dimensional, and 128-dimensional cable dimension features. These hierarchical cable dimension features are concatenated into a 256-dimensional vector and fed into a multi-layer perceptron for training, resulting in a 512-dimensional vector. Subsequently, max pooling and average pooling operations are performed, and the concatenated result yields a 1×1024-dimensional global feature. Finally, three fully connected layers with 256, 128, and C units are used as classifiers, where C represents the number of categories in the dataset. Batch normalization and Leaky ReLU activation functions are applied in all fully connected layers.

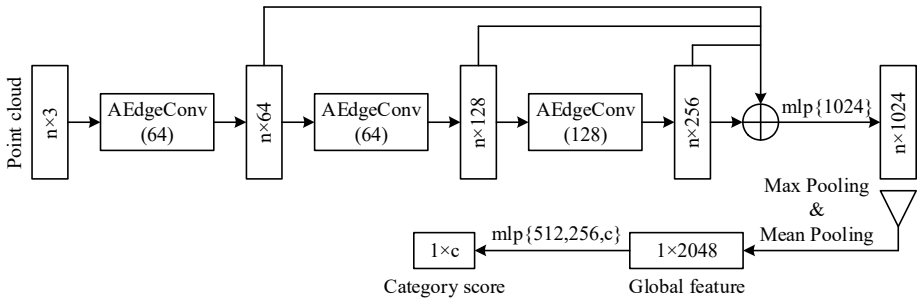


Figure 1: ADGCNN network structure

(2) AEdgeConv Module

Existing methods for processing point cloud data often ignore differences in channel importance when extracting cable size features. To address this issue, the AEdgeConv module was designed based on the EdgeConv structure in DGCNN network 3, as shown in Figure 2. This module uses a channel attention mechanism, as shown in Figure 3, to calculate the weighted vector of feature channels, enhance feature vectors, increase the weight of useful information, and thereby improve the network's learning of useful information.

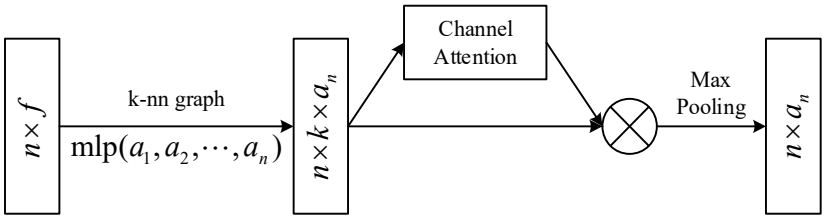


Figure 2: AEdgeConv structure diagram

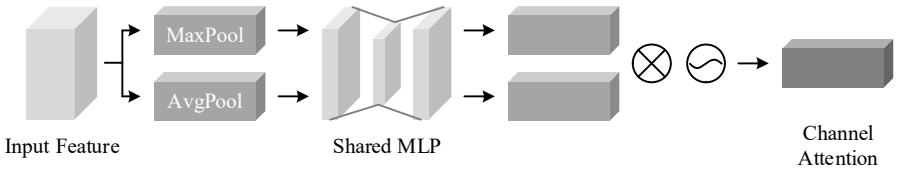


Figure 3: Channel attention mechanism structure

PointNet only considers the global features of point cloud data, as shown in Equation (10). For:

$$h_\theta(X_i, X_j) = h_\theta(X_i) \tag{10}$$

ADGCNN is similar to DGCNN in that it uses graph neural networks to extract local features from point clouds. V represents the edge features of vertex X_i, as shown in Equation (11). It is defined as:

$$V = h_\Theta(X_i, X_j) = h_\Theta(X_i, X_j - X_i) \tag{11}$$

The vector V is input into the channel attention mechanism to emphasize the importance differences of feature channels. Here, α_1, α_2 represent 1×1 convolution operations, and σ represents the Sigmoid function, which is used to obtain weight parameters β between 0 and 1 to differentiate the importance of feature channels. The process is shown in Equation (12) as follows:

$$\beta = \sigma(\alpha_1(\alpha_2(MAX(V))) + \alpha_1(\alpha_2(MEAN(V)))) \tag{12}$$

Multiply the weight β by the marginal feature information V to obtain the feature-enhanced vector V', where $A(')$ is the attention map, i.e., the weight β multiplied by the feature vector V. This process is shown in Equation (13). It is:

$$V' = A(V) \tag{13}$$

Finally, the complete edge features of X_i undergo a pooling operation. The AEdgeConv module uses max pooling to aggregate the edge features, which is represented by Equation (14).

$$X_i' = \max_{j:(i,j)\in E}(V') \tag{14}$$

(3) Symmetric functions

Symmetric functions are commonly used to address the issue of point cloud disorder. Most models use max pooling to summarize point cloud feature information, but using max pooling alone can lead to feature loss to a certain extent. Average pooling can reduce estimation errors caused by differences in neighborhood size, so different pooling methods can obtain features from different perspectives.

III. A. 2) Feature Enhancement

To extract rich and effective cable size features from 3D point cloud data, a feature enhancement module was designed, incorporating depth-separable convolutions, skip connections, and pixel addition operations. The feature

enhancement, as shown in Figure 4, includes five 3×3 depth-separable convolutions, three skip connections, and pixel addition operations. Each depth-separable convolution in the feature enhancement module includes a BN normalization operation and a Gelu activation function. After passing through the feature enhancement module, both shallow and deep feature maps obtain more rich cable dimension features. Depthwise separable convolutions effectively reduce the number of parameters in convolution operations while maintaining performance comparable to conventional convolutions. The skip-connection operation uses the output of one layer in the feature enhancement module as the input for the next layer, effectively addressing the issue of degradation, facilitating gradient propagation, and accelerating the training process of the cable dimension feature deep learning model, thereby enabling the model to capture more detailed information about cable dimensions. The BN normalization operation has the advantages of avoiding gradient vanishing and gradient explosion, accelerating model convergence, and improving generalization. The Gelu activation function has a smooth and continuous derivative, making it easier to propagate gradients during the training of the slender defect segmentation network. The output values of the Gelu activation function have a large range, which helps accelerate the convergence speed of the model.

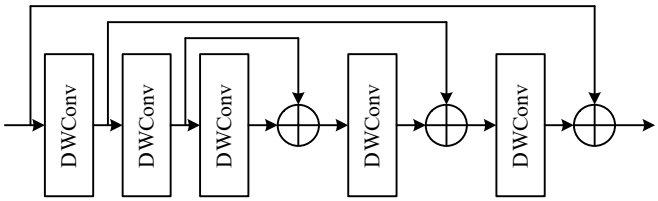


Figure 4: Feature enhancement

III. A. 3) Feature Fusion

Three-dimensional point cloud data of cable size features have high resolution and contain more positional information, but less semantic information. Deep features have richer semantic information but poorer perception of cable size feature details. The size of the receptive field affects the accuracy of training deep models for cable size features. The larger the receptive field of a deep learning model, the more cable size features it typically obtains. Dilated convolutions can effectively increase the receptive field. To improve the accuracy of training deep models for cable dimension features, a feature fusion method was designed to increase the receptive field while enhancing the features of elongated defect edges. First, the input 3D point cloud data is subjected to a Concatenate operation, which fuses information by overlaying feature channels. Then, three parallel branches are used to enhance the fused features and expand the receptive field. Each parallel branch consists of two 3×3 depth-separable convolutions and one dilated convolution with skip connections and pixel-wise addition operations. The dilation rates of the dilated convolutions in the three parallel branches are 2, 3, and 4, respectively. Each convolution operation in the parallel branches includes a Batch Normalization (BN) standardization operation and a Gelu activation function. Finally, the features from the three parallel branches are further fused using pixelwise addition and 1×1 convolution operations. The pixelwise addition operation adds features with the same number of channels, increasing the information per pixel to complete feature fusion.

III. B. Optimization of Deep Learning Training Models

III. B. 1) Overview of the Optimizer

In deep learning model training, the first step is to clearly define the problem we aim to solve and the appropriate model to use. Next, we organize the training data and test data, then select a training framework such as TensorFlow or Caffe. Through specific rules, the model extracts features from the training data, gradually updating and adjusting the values of variables in each layer, ultimately producing a trained model. This model captures the features of the training data, serving as a good representation or mapping of the entire dataset. In machine learning and deep learning, optimization algorithms include common methods like gradient descent (SGD) and MBGD, as well as alternative optimizers such as Adadelta, Adagrad, and RMSProp. For optimization algorithms, the optimization target is the parameter values θ in the model, and the optimization objective function is the loss function. The independent variable of the loss function L is θ, where the parameters of L are the entire training set. In other words, the objective function is determined by the entire training set. If the training set differs, the loss function will also differ.

III. B. 2) Adadelta Optimizer

Common optimization algorithms for deep learning training models include gradient descent (SGD), MBGD, Adadelta, Adagrad, RMSProp, and other optimizers. Given that the research content of this paper is the cable size features in three-dimensional point cloud data, the Adadelta optimizer is used to optimize the deep learning training model for cable size features mentioned above. The Adadelta algorithm is an improvement over Adagrad. Compared to Adagrad, Adadelta replaces the G in the denominator with the decayed average of the gradient squared, as shown in the following formula (15):

$$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \grave{o}}} \cdot g_t \tag{15}$$

The new form of the denominator is equivalent to the root mean square of the gradient. In mathematical statistical analysis, the sum of the squares of all values is calculated, then the mean is calculated, and then the square root is taken to obtain the root mean square, which is generally abbreviated as RMS, as shown in formula (16):

$$\Delta\theta_t = -\frac{\eta}{RMS[g]_t} \cdot g_t \tag{16}$$

The formula for calculating E is as follows. The value at time t depends on the average of the previous moment and the current gradient, as shown in formula (17) below:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \tag{17}$$

In addition, the learning rate η is replaced by $RMS[\Delta\theta]$ using this adaptive method, so that we do not even need to set the learning rate in advance, as shown in formula (18) below:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} \cdot g_t \tag{18}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \tag{19}$$

IV. Point cloud data preprocessing and model verification analysis

IV. A. Point cloud data preprocessing analysis

IV. A. 1) Acquiring Point Cloud Data

Using a lidar sensor, three-dimensional point cloud data of cable size characteristics was obtained, with a total of 53,135 points collected. Due to limitations in the scanning accuracy of the equipment, operator inexperience, complex environmental conditions, and surface characteristics of the objects, the collected point cloud data contains noise points and outliers, as well as issues such as data unevenness. Repeated scanning of local features also results in large data volumes and data redundancy. These shortcomings can interfere with subsequent research work, affecting cable dimension feature extraction, deep learning model training, and model optimization. Therefore, this paper employs an improved adaptive filtering algorithm to preprocess the acquired point cloud data. The performance of the improved adaptive filtering algorithm will be validated and analyzed in the following sections.

IV. A. 2) Experimental Results

To validate the effectiveness of the noise reduction performance of the algorithm proposed in this paper, we first compared its noise reduction performance with that of existing algorithms for different noise types, including Gaussian noise, random noise, and composite noise (a combination of Gaussian and random noise). The experimental results of the preprocessing of 3D point cloud data are shown in Table 1. The radius filtering algorithm consumes excessive time for noise reduction, with an average runtime of 2.111 seconds, resulting in low efficiency; The median denoising algorithm achieves denoising performance comparable to that of the radius filtering algorithm. It uses the optimal number of neighboring points from statistical filtering as the input parameter for radius filtering, thereby reducing the number of experiments required to find the optimal parameters for radius filtering during the experimental process, which improves efficiency to some extent. However, the single-run execution time of the algorithm is longer than that of the radius filtering algorithm, reaching 2.901 seconds. After processing the noise model using the denoising algorithm proposed in this paper, there are no residual noise points in the subjective visual effect, regardless of whether it is a Gaussian noise model, random noise model, or composite noise model. The algorithm demonstrates strong robustness across different noise types. After noise reduction processing by the algorithm proposed in this paper, the average number of remaining points in the point cloud model is 30,454, with

a corresponding algorithm runtime of 0.308 seconds, comparable to the 0.295-second runtime of the statistical filtering algorithm. When comprehensively comparing noise reduction effectiveness and efficiency, the algorithm proposed in this paper outperforms the other three algorithms.

Table 1: Experimental results of 3D point cloud data preprocessing

Algorithm type	Gaussian noise		Random noise		Compound noise		Means	
	Remaining point	Time /s	Remaining point	Time /s	Remaining point	Time /s	Remaining point	Time /s
Statistical filtering	33442	0.118	35928	0.662	34022	0.105	34464	0.295
Radius filtering	31273	0.753	34871	2.834	33511	2.745	33218	2.111
Median filtering	31233	2.854	34201	2.943	32468	2.905	32634	2.901
The algorithm of this article	30683	0.136	30389	0.715	30291	0.074	30454	0.308

To further validate the stability of the algorithm proposed in this paper, we will compare its denoising performance on point cloud models under different noise intensities. Gaussian noise with signal-to-noise ratios (SNR) of 60 dB, 80 dB, 100 dB, and 120 dB was added to the 3D point cloud data. The deviation analysis data before denoising for the algorithm under different noise intensities is shown in Table 2, and the deviation analysis data after denoising is shown in Table 3. The data shows that after denoising by the algorithm under different noise intensities, the average deviation is 0.0008 mm, with an absolute value far below the pre-denoising value of 0.2959 mm. The average standard deviation is 0.2228 mm, a 42.14% reduction compared to the pre-denoising value of 0.3851 mm. Additionally, the average percentage of point cloud data within tolerance increased from 16.78% before denoising to 29.45%, representing a 75.51% improvement. This significantly enhances the accuracy of the point cloud data, ensuring the validity of subsequent research results.

Table 2: Deviation analysis data before denoising

Noise intensity(dB)	Average deviation(mm)	Standard deviation(mm)	Proportion within the tolerance
60	-0.3342	0.4011	17.24%
80	-0.2746	0.4006	16.32%
100	-0.3126	0.3741	17.11%
120	-0.2621	0.3646	16.44%
Mean	-0.2959	0.3851	16.78%

Table 3: Deviation analysis data after denoising

Noise intensity(dB)	Average deviation(mm)	Standard deviation(mm)	Proportion within the tolerance
60	0.0021	0.2229	29.49%
80	0.0004	0.2231	29.42%
100	0.0004	0.2226	29.44%
120	0.0004	0.2226	29.44%
Mean	0.0008	0.2228	29.45%

IV. B. Model validation analysis

IV. B. 1) Cable size feature extraction analysis

(1) Evaluation indicators

For the analysis of cable size features in 3D point cloud data, this section will use extraction accuracy and average extraction accuracy as the evaluation indicators for this study. The specific formulas are shown below:

$$A_{overall} = \frac{M}{N} \times 100\% \tag{20}$$

$$A_{avg} = \frac{\left(\frac{M_1}{N_1} + \frac{M_2}{N_2} + \cdots + \frac{M_n}{N_n} \right)}{n} \times 100\% \tag{21}$$

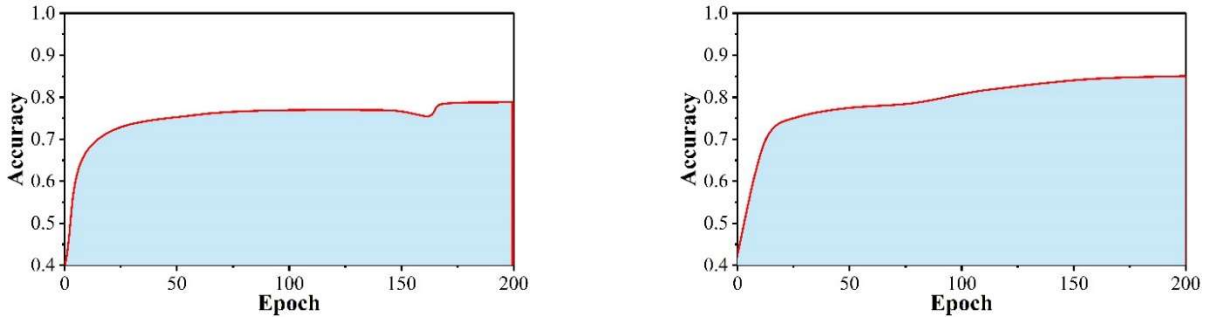
In the equation, $(M_1, M_2, \cdots, M_n)$ is the category data for each cable size feature, $(N_1, N_2, \cdots, N_n)$ is the 3D point cloud data for each cable size feature, and n is the number of categories in the sample.

(2) Experimental Environment

The hardware environment configuration includes an Intel® Core™ i7@2.4GHz CPU, 64.0 GB of memory, an NVIDIA GeForce GTX 4080 Ti GPU, Ubuntu 16.08 operating system, TensorFlow 1.16.0 deep learning framework, OpenGL version 4.8, and Python 3.6.4 programming language.

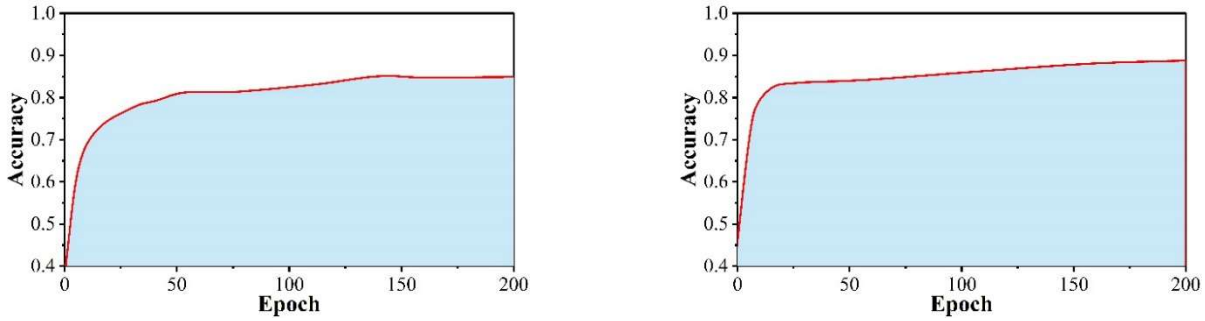
(3) Result Analysis

Under the influence of evaluation metrics, the cable size feature extraction performance of the ADGCNN network was validated and analyzed, with comparison methods including AlexNet, VGGNet-16, and ResNet50. The feature extraction accuracy results are shown in Figures 5–8, where (a) and (b) represent extraction accuracy and average extraction accuracy, respectively. The results show that the ADGCNN network outperforms AlexNet, VGGNet-16, and ResNet50, indicating that the ADGCNN network can effectively extract cable dimension features from 3D point cloud data, thereby validating the effectiveness of the ADGCNN network for cable dimension feature extraction.



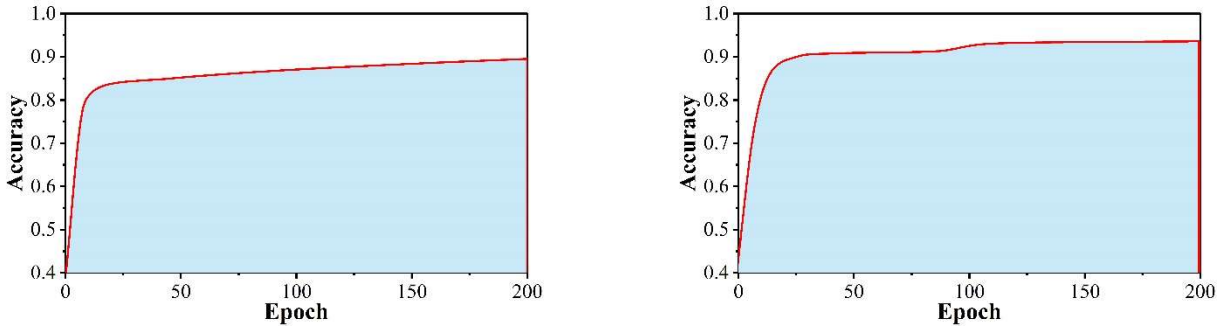
(a)Extraction accuracy (b)Average extraction accuracy

Figure 5: Feature extraction accuracy rate(AlexNet)



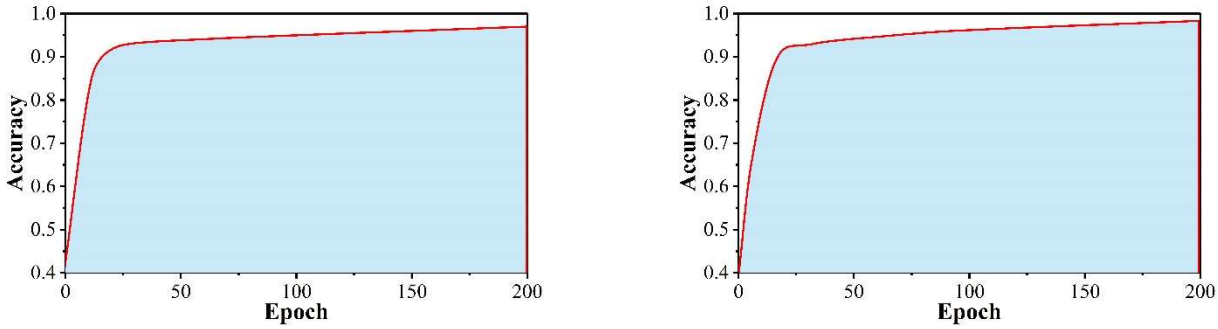
(a)Extraction accuracy (b)Average extraction accuracy

Figure 6: Feature extraction accuracy rate(VGGNet-16)



(a)Extraction accuracy (b)Average extraction accuracy

Figure 7: Feature extraction accuracy rate(ResNet50)



(a)Extraction accuracy (b)Average extraction accuracy

Figure 8: Feature extraction accuracy rate(ADGCNN)

To further validate the cable size feature extraction of the ADGCNN network, the loss curves for cable size feature extraction across different networks are compared below, as shown in Figures 9–12. When compared with AlexNet, VGGNet-16, and ResNet50, the ADGCNN network exhibits smaller fluctuations in loss values during the cable dimension feature extraction process, indicating that the ADGCNN network performs more stably in this task.

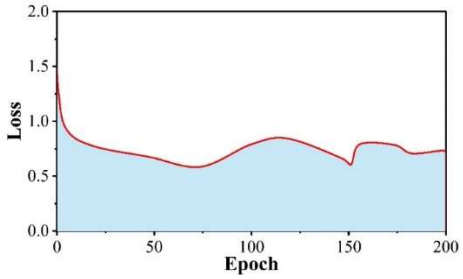


Figure 9: The loss curve of feature extraction(AlexNet)

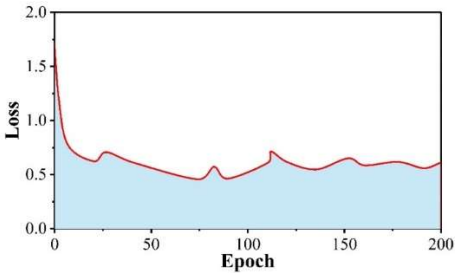


Figure 10: The loss curve of feature extraction(VGGNet-16)

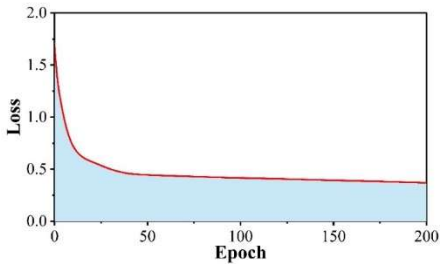


Figure 11: The loss curve of feature extraction(ResNet50)

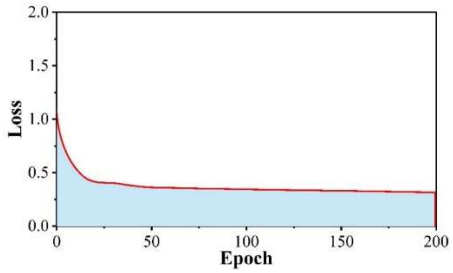
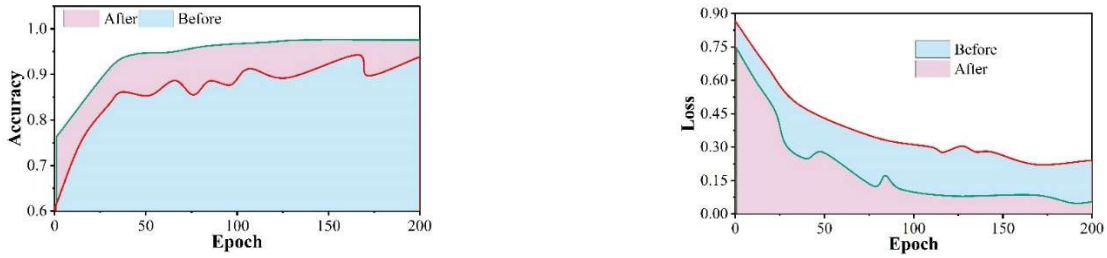


Figure 12: The loss curve of feature extraction(ADGCNN)

IV. B. 2) Analysis of Model Optimization Effects

As analyzed above, the ADGCNN network can effectively extract cable size features from 3D point cloud data, and then through feature enhancement and fusion, ultimately complete the construction of a deep learning training model. To improve the performance of the model training process, this paper uses the Adadelta optimization algorithm to optimize the model and conducts an in-depth analysis of its optimization effects, primarily considering loss values and accuracy rates. During training, the loss value gradually approaches 0 as the number of iterations increases, thereby reflecting the network's performance. Accuracy is measured by evaluating the output results of the trained network using a separate test set. A network model was constructed using transfer learning methods and trained. When the Adadelta optimization algorithm was selected, the training accuracy and loss accuracy are shown in Figure 13, where (a) and (b) represent the loss value and accuracy, respectively. Based on the loss value and accuracy data in the figure, it can be seen that under the Adadelta optimization algorithm, the model accuracy improved from 0.894 to 0.975, and the model training loss value was also optimized, demonstrating the optimization effect of the Adadelta optimization algorithm on the model in this paper.



(a)Accuracy (b)Loss

Figure 13: Training accuracy and loss accuracy

V. Conclusion

This paper utilizes three-dimensional point cloud data technology to obtain three-dimensional point cloud data of cable size features using a lidar sensor. To avoid the influence of interference information on the research results, an adaptive filtering algorithm is employed for data preprocessing. Based on this, the ADGCNN network is used to extract cable size features from the three-dimensional point cloud data. Through feature enhancement and fusion, a deep learning training model for cable size features is constructed. To address the issue of suboptimal model training performance, the Adadelta optimization algorithm is applied to optimize the model. The effectiveness of the algorithm in optimizing the deep learning training model for cable dimension features is validated from two aspects: loss value and accuracy rate. After validation analysis, it was found that under the Adadelta optimization algorithm, the accuracy of the model increased from the initial 0.894 to 0.975, and the corresponding loss value was also optimized. Based on the research results, the effectiveness of the research scheme was fully validated.

Funding

Science and Technology Project of State Grid Shaanxi Electric Power Co., Ltd.: Research on Intelligent Detection Technology of Cable Geometric Parameters Based on 3D Laser Scanning (5226WN240005).

References

[1] Li, Y., Jiang, L., Xie, M., Yu, J., Qian, L., Xu, K., ... & Wang, Y. (2024). Advancements and challenges in power cable laying. Energies, 17(12), 2905.

[2] Chen, X., Chen, Z., Liu, G., Chen, K., Wang, L., Xiang, W., & Zhang, R. (2021). Railway overhead contact system point cloud classification. Sensors, 21(15), 4961.
[3] Heisler, P., Steinmetz, P., Yoo, I. S., & Franke, J. (2017). Automatization of the cable-routing-process within the automated production of wiring systems. Applied Mechanics and Materials, 871, 186-192.
[4] Fang, C., Lu, W., Liu, J., Yang, X., & Zhang, J. (2025). Dynamic Simulation of Underground Cable Laying for Digital Three-Dimensional Transmission Lines. Applied Sciences, 15(2), 979.
[5] Rizzo, G., Romano, P., Imburgia, A., & Ala, G. (2019). Review of the PEA method for space charge measurements on HVDC cables and mini-cables. Energies, 12(18), 3512.
[6] Arastounia, M. (2017). An enhanced algorithm for concurrent recognition of rail tracks and power cables from terrestrial and airborne lidar point clouds. Infrastructures, 2(2), 8.
[7] Sánchez-Rodríguez, A., Soilán, M., Cabaleiro, M., & Arias, P. (2019). Automated inspection of railway tunnels' power line using LiDAR point clouds. Remote Sensing, 11(21), 2567.
[8] Merkle, D., Frey, C., & Reiterer, A. (2021). Fusion of ground penetrating radar and laser scanning for infrastructure mapping. Journal of Applied Geodesy, 15(1), 31-45.
[9] Pastucha, E., Puniach, E., Ścisłowicz, A., Ćwiąkała, P., Niewiem, W., & Wiącek, P. (2020). 3D reconstruction of power lines using UAV images to monitor corridor clearance. Remote Sensing, 12(22), 3698.
[10] Wang, F., Li, B., Xu, Y., Zhang, J., & Zhang, J. (2025). Automated measurement method for cable shapes based on laser scanners and cameras. Journal of Civil Structural Health Monitoring, 15(2), 717-729.
[11] Tu, X., Xu, C., Liu, S., Lin, S., Chen, L., Xie, G., & Li, R. (2020). LiDAR point cloud recognition and visualization with deep learning for overhead contact inspection. Sensors, 20(21), 6387.
[12] Siranec, M., Höger, M., & Otcenasova, A. (2021). Advanced power line diagnostics using point cloud data—Possible applications and limits. Remote Sensing, 13(10), 1880.
[13] Tiseanu, I., Zani, L., Tiseanu, C. S., Craciunescu, T., & Dobrea, C. (2015). Accurate 3D modeling of cable in conduit conductor type superconductors by X-ray microtomography. Fusion Engineering and Design, 98, 1176-1180.
[14] Ortega, S., Trujillo, A., Santana, J. M., Suárez, J. P., & Santana, J. (2019). Characterization and modeling of power line corridor elements from LiDAR point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 152, 24-33.
[15] Schofield, O. B., Lorenzen, K. H., & Ebeid, E. (2020, August). Cloud to cable: A drone framework for autonomous power line inspection. In 2020 23rd Euromicro Conference on Digital System Design (DSD) (pp. 503-509). IEEE.
[16] Zhang, T., Yuan, H., Qi, L., Zhang, J., Zhou, Q., Ji, S., ... & Li, X. (2025, April). Point cloud mamba: Point cloud learning via state space model. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 10, pp. 10121-10130).
[17] Han, X. F., Jin, J. S., Wang, M. J., Jiang, W., Gao, L., & Xiao, L. (2017). A review of algorithms for filtering the 3D point cloud. Signal Processing: Image Communication, 57, 103-112.
[18] Inoue, M., Niigaki, H., Shimizu, T., Honda, N., Oshida, H., & Ebine, T. (2021). Visualization of 3D cable between utility poles obtained from laser scanning point clouds: a case study. SN Applied Sciences, 3, 1-11.
[19] Xie, H., Liu, G., Deng, L., Song, T., & Qin, F. (2024). Measurement of outer diameter parameters of manual grinding cable joints based on 3D point cloud processing. Measurement, 236, 115095.
[20] Nardinocchi, C., Balsi, M., & Esposito, S. (2020). Fully automatic point cloud analysis for powerline corridor mapping. IEEE Transactions on Geoscience and Remote Sensing, 58(12), 8637-8648.
[21] Liu, Y., Tang, Q., Tian, X., & Yang, S. (2023). A novel offline programming approach of robot welding for multi-pipe intersection structures based on NSGA-Ⅱ and measured 3D point-clouds. Robotics and computer-integrated manufacturing, 83, 102549.
[22] Deng, L., Deng, H., Liu, G., Zhao, J., Huang, H., & Li, L. (2024). XLPE cable joint defects measurement method based on point cloud remapping. Measurement, 226, 114139.
[23] Zheng, Z., Gao, Y., He, Z., Wang, C., Xie, H., & Liu, G. (2023). Dimensional measurement of anti-stress cone of HV cable joints based on 3D point cloud. Measurement Science and Technology, 34(11), 115009.
[24] Haoran Ye,Hongjie Guo,Wenhua Li,Shanying Lin,Gen Li & Lei Hong. (2025). An adaptive control method of non-metallic armoured Optical-electrical cable winch system in Oceanographic Research Vessel based on three-dimensional laser scanning point cloud. Ocean Engineering,333,121549-121549.