

Retraction

Retracted: Power Operation Violation Identification Method Based on Point Cloud Data Preprocessing and Deep Learning under the Architecture of IoT

Journal of Electrical and Computer Engineering

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

Copyright © 2023 Journal of Electrical and Computer Engineering. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] S. Yang, W. Fu, L. Zhang, and Z. Wang, "Power Operation Violation Identification Method Based on Point Cloud Data Preprocessing and Deep Learning under the Architecture of IoT," *Journal of Electrical and Computer Engineering*, vol. 2023, Article ID 6859102, 8 pages, 2023.

Research Article

Power Operation Violation Identification Method Based on Point Cloud Data Preprocessing and Deep Learning under the Architecture of IoT

Shibo Yang , Weiping Fu, Lishuo Zhang, and Zhaolei Wang 

State Grid Hebei Extra High Voltage Company, Shijiazhuang, Hebei 050070, China

Correspondence should be addressed to Shibo Yang; 634506327@qq.com

Received 24 June 2022; Revised 17 August 2022; Accepted 24 August 2022; Published 20 February 2023

Academic Editor: Xuefeng Shao

Copyright © 2023 Shibo Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problems of low recognition accuracy and large memory occupation when using point cloud information for power operation violation, A power operation violation recognition method based on point cloud data preprocessing and deep learning under the architecture of Internet of things (IoT) is proposed. First, voxel filtering and statistical filtering methods are used to properly simplify the power operation point cloud data on the premise of ensuring the quality of reverse modeling, and the moving least square method is used to smooth the point cloud to obtain a complete and closed three-dimensional model; second, the process of power operation violation behavior recognition is divided into two stages. In the first stage, PointRCNN extracts the semantic features of each point, separates the front scenic spots, and extracts the preselection box. In the second stage, the candidate box is refined by integrating the semantic features and classification confidence of the first stage to obtain a more accurate bounding box. Finally, the experiments show that the average accuracy of the proposed method is the highest, with an average accuracy of 0.919 in the simple difficulty scenario, 0.897 in the medium difficulty scenario, and 0.839 in the difficult difficulty scenario, which are higher than those of the compared methods. Therefore, the proposed method can effectively improve the accuracy of power operation violation identification.

1. Introduction

There are many live equipment on the power operation site, and the operators must strictly abide by the operation safety specifications [1]. However, in the process of power operation, it is difficult to control the working range and movement track of operators. Some workers do not wear safety helmets to carry out dangerous operations in order to save trouble. They often enter or cross the live area in violation of regulations in order to bypass the short way, which is easy to lead to safety production accidents [2–5]. Therefore, it is very important to track the trajectory of operators, safety control, and monitoring the operation process.

In the process of traditional power production, the hidden dangers of safe operation are found, eliminated, and avoided mainly through standardized operation training for operators,

self-supervision of operators, and manual supervision and observation through a traditional video monitoring system, so as to ensure the safety of operation. The target detection task in three-dimensional scene needs to be completed by a large amount of data. These data can be divided into two-dimensional image, 2.5D RGB-D image, and three-dimensional point cloud data according to the dimension. Although these two types of data are easy to obtain and have high processing efficiency, they have great limitations [6–8]. 3D point cloud data can just make up for these deficiencies. The essence of 3D target detection problem is the division of points. The point cloud data is a collection composed of multiple disordered points, which itself contains rich spatial information. At the same time, it can make up for the defect that the camera is not suitable for outdoor large-scale scenes due to field of view. Therefore, the point cloud data is of great significance for understanding 3D scenes [9–14].

In recent years, in the field of computer vision, target detection and tracking technology have made great progress, and a series of target detection algorithms and target tracking algorithms with superior performance have emerged, which make their application possible [15–17]. The application of target detection and tracking method to power workplace to realize production intelligence has become a key research topic, which has an important application value in the field of power production safety [18–20].

The traditional method based on the feature extraction of power equipment has poor adaptability due to the complexity of operation and migration. In the face of complex situations such as rapid illumination change, similar environmental background color to the target and slow movement of the target, the traditional target detection methods usually perform poorly, are greatly affected by the interference of environmental noise, and have the problem of too high time complexity [19, 20]. Therefore, limited by the complexity of the actual environment of power production and workplace, the traditional target detection methods cannot meet the needs of power production and operation. Reference [21] takes the point cloud and image data as the input, projects the point cloud data to the two-dimensional plane with a specific perspective, and then uses the region based representation to deeply fuse the features of the two data, and proposes a target detection method based on multiview three-dimensional network (MV3D). In reference [22], the frame is generated in the 2D target detector by a serial method, and then projected onto the 3D point cloud to further optimize the frame. This kind of method improves the detection efficiency, realizes the dimensional (2D-3D) positioning, and shortens the search time of point cloud. However, the whole process relies heavily on 2D detection effect, and there is no way to solve the occlusion problem in the original data. Reference [23] obtains a two-dimensional point map by fine designing the bounding box coding and projecting the point cloud to the front view, and uses the full convolution network to intensively predict the three-dimensional frame. Reference [24] encodes the point cloud as a descriptive volume representation and connects it to RPN to generate detection. A depth network combining feature extraction and boundary box prediction is proposed. Reference [25] proposes a target detection method based on PV EncoNet. The model removes a large number of invalid point clouds through the filtering algorithm and then adds texture information through point cloud coloring to enhance features. Reference [26] introduces deformable convolution into the point cloud target detection network to enhance the adaptability of the network to detection targets in different directions and shapes. Aiming at the imprecision of target position and the uncertainty of foreground target depth distribution, reference [27] proposes a target detection method based on improved GUPNET.

However, the above methods often have the problems of low recognition accuracy and large memory occupation. In order to overcome the above problems, a method for identifying violations of power operation based on point cloud data preprocessing and in-depth learning under the

Internet of things architecture is proposed. The innovations are as follows:

- (1) The voxel filtering and statistical filtering methods are used to simplify the point cloud, and the moving least square method is used to smooth the point cloud, which reduces the memory occupation and running time of the method.
- (2) In the first stage, PointRCNN extracts the semantic features of each point, separates the front scenic spots, and extracts the preselected box. In the second stage, the candidate box is refined by integrating the semantic features and classification confidence of the first stage, which effectively improves the accuracy of the method in identifying violations of power operation.

2. Point Cloud Data Preprocessing

2.1. Point Cloud Data. The point cloud data, in this study, are collected from the deployed Internet of things devices. Maptek i-SITE 8200 3D laser scanner is used to acquire point cloud data and build a data set of power workplace. We use the annotation tool to annotate the target boxes of the operator target and the head target in the image, and generate an annotation file, including the position, size, and category of the target box. The point cloud uses a large number of coordinates (x, y, z) to describe the three-dimensional shape of the object, and the matrix of $n \times 3$ size stores this information, where n is the total number of points. The coordinate value range in the point cloud data is $(-\infty, +\infty)$. Compared with the two-dimensional image with the value range of $(0, 255)$, the point cloud data has a wider range and a greater amount of information.

The unstructured of point cloud brings great challenges to data processing. Figures 1(a) and 1(b) in Figure 1, respectively, show a structured two-dimensional image and a nonstructured point cloud. Each pixel of the image in Figure 1(a) is limited in the grid, and there is a fixed distance and adjacency relationship between pixels. The point cloud in Figure 1(b) has no fixed structure, and there is no fixed distance and adjacency relationship between points. In two-dimensional image processing, the convolution kernel of the convolution neural network has a fixed structure. This convolution kernel can only deal with data with regular structure, so it cannot deal with nonstructural data such as the point cloud.

There are great differences between the point cloud and the image. All its information are reflected in the coordinates of the points, and the order between the points will not provide any valuable information for the data. In other words, arbitrarily changing the order of points in the point cloud data will not change its shape.

2.2. Pretreatment Method. When precise equipment such as high-resolution camera is used to collect point cloud data, the obtained point cloud is denser. In order to accelerate the progress of reconstruction, voxel filtering or resampling is

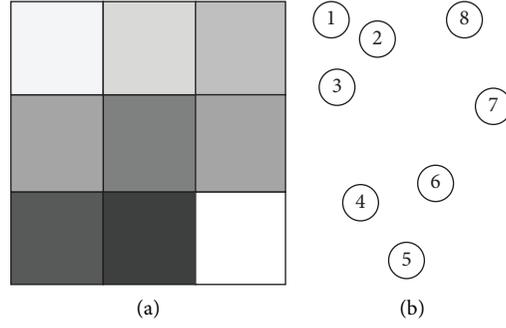


FIGURE 1: Unstructured and disordered point cloud.

usually used to properly simplify the point cloud on the premise of ensuring the quality of reverse modeling.

Voxel filtering obtains a new subset by retaining only the barycenters of all points in the 3D voxel grid. After voxel filtering, the density of point cloud decreases and the amount of data decreases greatly, but its shape, characteristics, and spatial structure information can remain basically unchanged. The main steps are as follows:

- (1) Find the coordinate range and calculate the maximum and minimum values on the three-dimensional coordinate axis
- (2) Calculate the side length of the minimum bounding box according to the maximum and minimum values in (1): $l_x = x_{\max} - x_{\min}$, $l_y = y_{\max} - y_{\min}$ and $l_z = z_{\max} - z_{\min}$
- (3) Determines the side length of the voxel grid. Set the side length cell of voxel grid and divide the minimum bounding box into $M * N * L$ voxel grid:

$$\begin{aligned} M &= \lfloor \frac{l_x}{\text{cell}} \rfloor, \\ N &= \lfloor \frac{l_y}{\text{cell}} \rfloor, \\ L &= \lfloor \frac{l_z}{\text{cell}} \rfloor, \end{aligned} \quad (1)$$

where $\lfloor \rfloor$ indicates rounding down. (x, y, z) represents the three-dimensional shape of an object.

- (4) Voxel grids are numbered. Determine the number of voxel grid to which each point belongs (a, b, c) , which is calculated as follows:

$$\begin{aligned} a &= \lfloor \frac{(x_i - x_{\min})}{\text{cell}} \rfloor, \\ b &= \lfloor \frac{(y_i - y_{\min})}{\text{cell}} \rfloor, \\ c &= \lfloor \frac{(z_i - z_{\min})}{\text{cell}} \rfloor. \end{aligned} \quad (2)$$

- (5) Point cloud simplification. Calculate the center of gravity of each voxel grid:

$$C = \frac{1}{k} \sum_{i=1}^k p_i, \quad (3)$$

where C is the center of gravity of the voxel grid, p_i is the point in the voxel grid, and k is the number of points.

The principle of statistical filtering is to traverse all points, statistically analyze all points in the neighborhood of each point, and calculate the average Euclidean distance from each point in the neighborhood. If the average distance from a point to all points in the near neighborhood exceeds the threshold, it will be regarded as an outlier and eliminated from the data set.

Therefore, in order to improve the quality of the reconstructed model as much as possible and get a complete, restored, and closed three-dimensional model as much as possible, the point cloud must be smoothed in advance before the point cloud surface is reconstructed.

The least square method is used to find z' the best approximation b for the equations $Az = b$:

$$|b - Az'| \leq |b - Az|. \quad (4)$$

The weighted least square method introduces the weight function to enhance the important influencing factors and weaken the secondary influencing factors through weighting, that is,

$$J = \sum_{i=1}^n \theta(s) (f(x_i) - y_i)^2, \quad (5)$$

where s is the Euclidean distance from the independent variable x to the nearby sample independent variable.

Moving least squares smoothing is to calculate the weighted least squares method for each sample, and then calculate the function value $f_X(x_i)$ of the independent variable x_i of the sample, and the obtained $(x, f_X(x_i))$ is the smoothing result.

After statistical filtering, some outliers such as noise generated in the scanning process will be eliminated to improve the quality of point cloud and avoid wrong calculation in the subsequent modeling process. After moving least squares (MLS) point cloud smoothing, some local objects with sparse point cloud density will increase the

number of some point clouds due to interpolation, so as to strengthen the detailed features in the modeling process and improve the reconstruction quality. The specific process of point cloud data simplification is shown in Figure 2.

3. Recognition Method of Violation Behavior in Power Operation Based on PointRCNN

3.1. PointNet. PointNet is a deep neural network that can directly process disordered point cloud data, and its formal representation is shown in equation.

$$T(\{q_1, q_n\}) \approx g[h(q_1), h(q_n)], \quad (6)$$

where $\{q_1, q_n\}$ represents the input unordered point cloud $q_i \in R^d$, q_i represents the i_{th} point in the input point cloud, R represents the real number, d represents the dimension, N represents the number of input points, and h is a continuous function of nonlinear transformation.

PointNet uses multilayer perceptron to learn nonlinear transformation function. Aiming at the displacement invariance of point cloud data, by designing symmetric function g , we can approximate a function that can solve the disorder of point cloud f . The PointNet network structure is shown in Figure 3. The input end is N three-dimensional points, using transformation matrix F_{T-net} predicts the rotation matrix to realize the alignment of data and features, rotates the point cloud to an optimal angle, and enters the classification network. The multilayer perceptron is used to extract the abstract features of the point cloud. The pooling layer adopts a way of downsampling, uses the symmetric function Max pooling to retain the important information in the high-dimensional abstract features, and maps the point cloud to a one-dimensional descriptor. The introduction of symmetry function solves the disorder problem of the point cloud data, and the use of transformation matrix can deal well with the rotation invariance of point cloud data. In PointNet classification network, the features of independent point cloud are extracted by convolution based on each individual point.

3.2. PointRCNN Phase I. For an outdoor scenic spot cloud, the number of background points is much larger than that of the previous scenic spots. Therefore, in order to solve the problem of imbalance between positive and negative samples, focal loss is selected as the classification loss in the prescenic spot prediction task. The form of the preselected box is $(x, y, z, w, l, h, \theta)$, where (x, y, z) is the center coordinate of the preselected box, (w, l, h) is the size of the preselected box, and θ is the orientation of the preselected box in the top view.

In order to limit the scope of generating preselected boxes, PointRCNN uses a regression loss based on small blocks. In order to predict the central coordinates of a target, PointRCNN divides the surroundings under the top view of each front scenic spot into small blocks. Then, set the one-dimensional search distance S and determine which block different targets are in according to S and fixed parameters. A small block classification cross entropy loss is also used

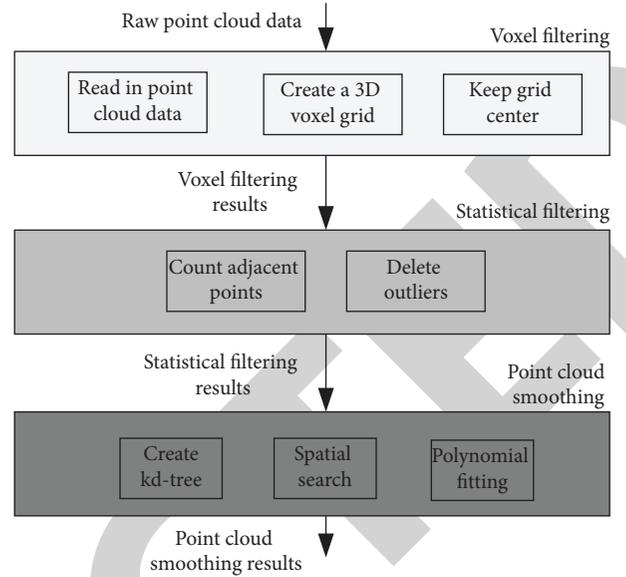


FIGURE 2: Flow chart of point cloud data preprocessing.

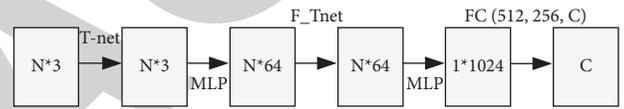


FIGURE 3: Pointnet network structure.

here, which can improve the accuracy and robustness of location.

The regression loss of (x, y) is composed of two parts, one is the classification loss function of which small block the target center is located, and the other is the regression loss of (x, y) relative to this small block. For the regression of z value, because the fluctuation of the target in the z direction is small, the regression loss function is directly used. The regression target can be obtained from equation.

$$\begin{aligned} \text{bin}_x^{(p)} &= \lfloor \frac{x^p - x^{(p)} + S}{\delta} \rfloor, \\ \text{bin}_y^{(p)} &= \lfloor \frac{y^p - y^{(p)} + S}{\delta} \rfloor, \\ \text{res}_u^{(p)} &= \frac{1}{C} \left(\text{up} - u^{(p)} + S - \left(\text{bin}_u^{(p)} \delta + \frac{\delta}{2} \right) \right), \\ \text{res}_y^{(p)} &= y^p - y^{(p)}, \end{aligned} \quad (7)$$

where $(x, y^{(p)}, z^{(p)})$ is the coordinate of the front scenic spot of interest, $(x^{(p)}, y^{(p)}, z^{(p)})$ is the central coordinate of the target corresponding to the point, $(\text{bin}_x^{(p)}, \text{bin}_y^{(p)})$ is the coordinate of the small block to which the label is mapped, $(\text{res}_x^{(p)}, \text{res}_y^{(p)})$ is the distance of the point relative to the corresponding small block, and C is the parameter for normalization.

For parameters (w, l, h) , the smooth L1 loss function is directly used for regression, and the expression of smooth L1

function is shown in (8). For θ , first divides 2π into n parts, and then predicts which part $\text{bin}_\theta^{(p)}$ it belongs to and the angle value $\text{res}_\theta^{(p)}$.

$$\text{Smooth}_{L1}(x) = \begin{cases} |x|, & -0.5|x| \geq 1, \\ 0.5x^2, & \text{otherwise.} \end{cases} \quad (8)$$

Let N_{pos} be the number of front scenic spots, $\text{bin}_u^{(p)}$ and $\text{res}_u^{(p)}$ be the predicted small blocks of the point and the corresponding deviation value, F_{reg} be the cross-entropy loss, and F_{cls} be the smooth L1 loss. The overall loss function of the first stage can be expressed as follows:

$$\begin{aligned} L_{\text{bin}}^{(p)} &= \sum_{u \in \{x, z, \theta\}} \left(F_{\text{cls}}(\text{bin}_u^{(p)}, \text{bin}_u^{(p)}) + F_{\text{reg}}(\text{res}_u^{(p)}, \text{res}_u^{(p)}) \right), \\ L_{\text{res}}^{(p)} &= \sum_{v \in \{y, h, w, l\}} F_{\text{reg}}(\text{res}_v^{(p)}, \text{bin}_v^{(p)}), \\ L_{\text{reg}} &= \frac{1}{N_{\text{pos}}} \sum_{p \in \text{pos}} (L_{\text{bin}}^{(p)} + L_{\text{res}}^{(p)}). \end{aligned} \quad (9)$$

3.3. PointRCNN Phase II. After obtaining the preselected boxes in the first step, these preselected boxes need to be further refined to obtain more accurate bounding boxes. For any preselection box $b_i = (x_i, y_i, z_i, w_i, l_i, h_i, \theta_i)$, PointRCNN will be slightly expanded to obtain nearby semantic information, that is $b_i^e = (x_i, y_i, z_i, w_i + \eta, l_i + \eta, h_i + \eta, \theta_i)$.

For any point $p = (x^{(p)}, y^{(p)}, z^{(p)})$, see if it is in b_i^e . If true, keep the point. In order to further refine the preselector box, the points located b_i^e in the corresponding target points are converted to a standard coordinate system.

Although this coordinate system can obtain local features well, it will inevitably lose depth information. In order to retain this information, define a feature $d^{(p)} = \sqrt{(x^{(p)})^2 + (y^{(p)})^2 + (z^{(p)})^2}$. Finally, the $[r^{(p)}, d^{(p)} + m^{(p)}]$ connection is used as the final local feature. A multilayer perceptron is used to extract the feature, and a feature with the same dimension as the first stage $f^{(p)}$ is obtained. Then, this feature is $f^{(p)}$ connected to get a new feature. This feature is input into a network that extracts point cloud features to obtain the final refined 3D box and the confidence of the corresponding box. Only when the IOU between the dimension box and the box is greater than a threshold, the box will participate in the final back propagation.

In terms of loss function, the loss function similar to that in the first stage is adopted here, but when selecting the search distance s , a smaller search distance is selected. Moreover, because all preselected boxes and labels will be converted to the standard coordinate system; that is, the preselected box $b_i = (x_i, y_i, z_i, w_i, l_i, h_i, \theta_i)$ and label box $b_i^l = (x_i^l, y_i^l, z_i^l, w_i^l, l_i^l, h_i^l, \theta_i^l)$ will be converted to equation.

$$\begin{aligned} \tilde{b}_i &= (0, 0, 0, w_i, l_i, h_i, 0), \\ \tilde{b}_i^l &= (x_i^l - x_i, y_i^l - y_i, z_i^l - z_i, w_i^l, l_i^l, h_i^l, \theta_i^l - \theta_i). \end{aligned} \quad (10)$$

The loss function of the second stage is as follows:

$$L_{re} = \frac{1}{\|B\|} \left(\sum_{i \in B} F_{\text{cls}}(\text{prob}_i, \text{label}_i) + \sum_{i \in B_{\text{pos}}} (\tilde{L}_{\text{bin}}^{(i)} + \tilde{L}_{\text{res}}^{(i)}) \right). \quad (11)$$

Among them, B is the preselection box obtained in the first stage, B_{pos} is the number of positive samples predicted, prob_i is the confidence of prediction, and label_i is the corresponding annotation. Finally, after ranking the confidence of the prediction frame results, the maximum value is suppressed and the overlapping three-dimensional frames are filtered to get the final results.

3.4. Identification Process of Violation Behavior in Power Operation Based on PointRCNN. The identification process of power operation violations based on PointRCNN is shown in Figure 4. First, the input power operation point cloud data is simplified by voxel filtering and statistical filtering methods, and the point cloud is smoothed by the moving least square method, which reduces the memory occupation and running time of the method. Second, in the first stage of PointRCNN, the semantic feature of each point is extracted through a network, and then this feature is used to separate the front scenic spots and extract the preselection box. Finally, by fusing the semantic features and classification confidence of the first stage, PointRCNN further refines the candidate boxes proposed in the first stage and then uses the nonmaximum suppression method to screen the candidate boxes to get the final result.

4. Experiment and Analysis

4.1. Experimental Environment and Data Set. The whole experiment was completed under Ubuntu 6.04 operating system based on Linux kernel, and various functions were realized in the experiment mainly using python. The hardware resources used mainly include Intel Xeon Series CPU and two NVIDIA Tesla series GPUs, as well as 64 g memory and 300 g external memory. The experiments of the proposed method are implemented under the OpenPCDet framework. The main software and hardware environments and corresponding versions involved in this experiment are shown in Table 1:

The maptek i-SITE 8200 3D laser scanner is used to obtain the point cloud data, and 8000 interested frame images are selected from the data set of power workplace. Use the annotation tool (Lable_tool) to annotate the target frame of the operator target and head target in the image, generate the annotation file including the position, size, and category of the target frame, and finally complete the

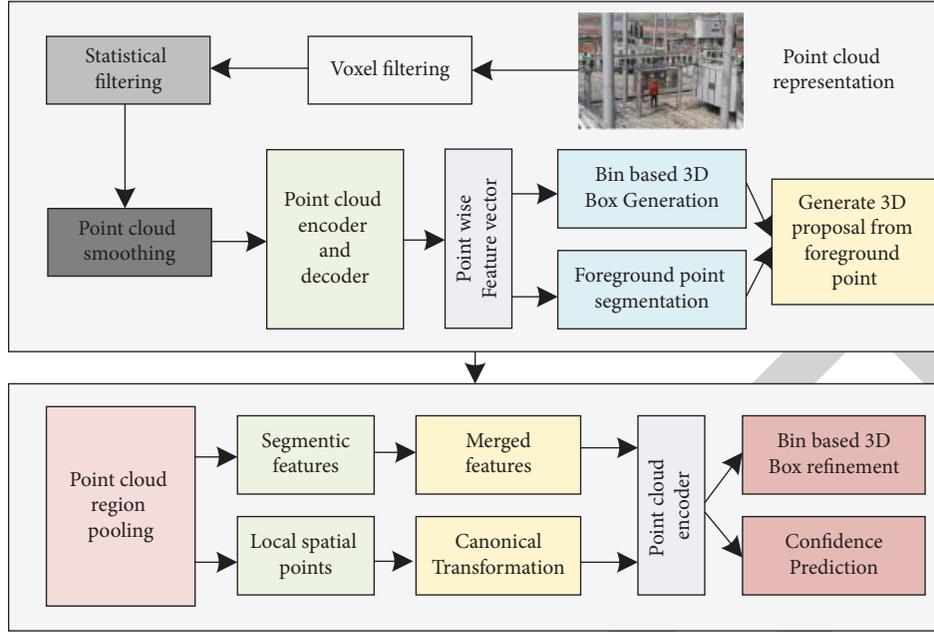


FIGURE 4: Identification process of power operation violations based on PointRCNN.

TABLE 1: Experimental environment information.

Experimental environment	Model of version
Ubuntu	16.04.1
Python	3.6
CPU	Intel xeon gold 5218 CPU@2.3 GHz
GPU	Nidia tesla T4x2
CUDA	10.0.130
OpenPCDet	2.0
Py Torch	1.3
Spconv	1.2

construction of the training data set. The data set, in this study, is divided into three levels according to the degree of occlusion and truncation: easy, medium, and hard. Among them, simple scenes have no occlusion, and the truncation rate is 15%; The medium scene contains partial occlusion, and the truncation rate is 30%; Difficult scenes contain a lot of occlusion, and the truncation rate is 50%.

4.2. Training Curve. The same training set is used to train the proposed method in simple and medium difficult scenarios. Figure 5 shows the change of the average accuracy of the data with the training step. It can be seen from the figure that under the same training process, the average accuracy of this method gradually decreases with the increase of scene difficulty under the same epoch. After 300 rounds of training, each curve has been stable, the network has converged, and the detection performance has been in the best state. Through the analysis of the training curve, the curve of the network hardly fluctuates after 300 rounds of training. Therefore, it can be known that the network can achieve the optimal effect after 300 rounds of training. At this time, the

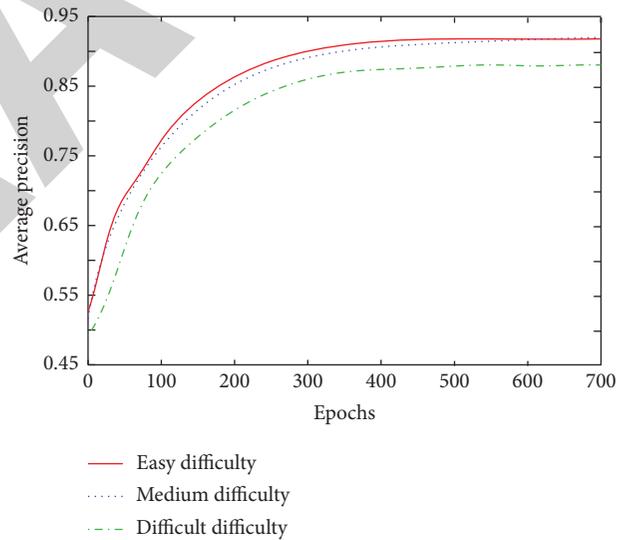


FIGURE 5: Variation of average precision in different difficult scenarios.

average accuracy in simple difficulty scenario is 0.913, that in medium difficulty scenario is 0.887, and that in difficult difficulty scenario is 0.859.

4.3. Performance Comparison with Other Methods. In order to prove the performance of the proposed method, it is compared with the methods in reference [26] and reference [27] under the same experimental conditions. The comparison results are shown in Table 2. Reference [26] introduces deformable convolution into the point cloud target detection network to enhance the adaptability of the network to detection targets in different directions and shapes.

TABLE 2: Comparison of average precision of different methods in easy, moderate, and hard scenarios.

Test method	Easy difficulty	Moderate difficulty	Hard difficulty
References. [26]	0.896	0.879	0.783
References. [27]	0.900	0.856	0.802
Proposed method	0.919	0.897	0.839

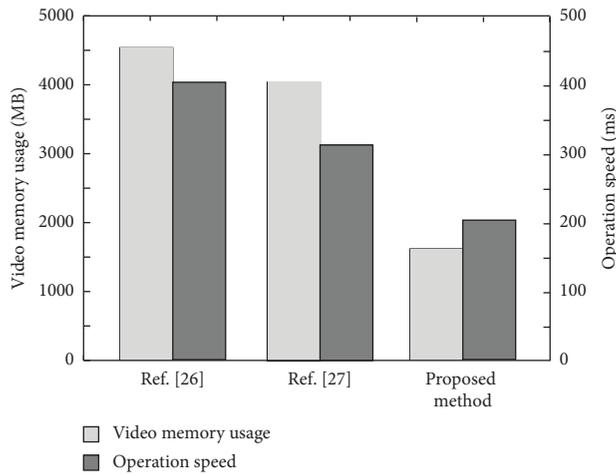


FIGURE 6: Video memory usage and running time of different methods.

Reference [27] proposed a target detection method based on improved GUPNET. The experimental results show that the average accuracy of the proposed method is the highest, with an average accuracy of 0.919 in simple difficult scenes, 0.897 in medium difficult scenes, and 0.839 in difficult scenes; reference [27] has an average accuracy of 0.900 in simple difficult scenes, 0.856 in medium difficult scenes, and 0.802 in difficult scenes; the average accuracy of reference [26] is the lowest, with the average accuracy of only 0.896 in simple difficult scenes, 0.879 in medium difficult scenes, and 0.783 in difficult scenes. This is because the proposed method divides the process of power operation violation recognition into two stages. In the first stage, PointRCNN extracts the semantic feature of each point and then uses this feature to separate the front scenic spots and extract the preselection box. In the second stage, the candidate frames are refined by integrating the semantic features and classification confidence of the first stage, and then the candidate frames are screened by nonmaximum suppression. This improves the recognition efficiency of the proposed method, while the comparison method cannot effectively mine the data features in the face of a large number of point cloud data. Therefore, the recognition performance is poor.

In order to analyze the impact of each classifier on the speed and memory consumption of the method, under the experimental conditions of moderately difficult scenes, it is compared with the methods in reference [26] and reference [27], and the comparison results are shown in Figure 6. The memory occupation of the method in reference [26] is

4504 MB, and the running time is 399.7 ms. The memory occupation of the proposed method is only 1687 mb, and the running time is 200.3 ms. This is because in the point cloud data preprocessing stage, the point cloud data is filtered and smoothed, which reduces the density of the point cloud and greatly reduces the amount of data, but its shape characteristics and spatial structure information can remain basically unchanged. The processing of point cloud data by the comparison method is not perfect, takes up a lot of computer memory, and the calculation time is long.

5. Conclusion

In order to solve the problems of low accuracy and large memory occupation of existing methods using point cloud information for power operation violation identification under the Internet of things architecture, a power operation violation identification method based on point cloud data preprocessing and deep learning under the Internet of things architecture is proposed. The voxel filtering and statistical filtering methods are used, and the moving least square method is used to smooth the point cloud to obtain a complete and closed three-dimensional model. The process of identifying violations in power operation is divided into two stages to improve the recognition accuracy of the proposed method. Finally, the experiments show that the proposed method effectively improves the recognition accuracy. In addition, the representation forms of point cloud data are mainly voxel, point, and graph structure. In the next step, the point cloud processing method integrating multiple representation schemes will be studied. In the future, the accuracy can be further improved by fusing image information.

Data Availability

The data that support that findings of the study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are thankful to Science and Technology project funding from State Grid Corporation of China (Project number: kj2022-037).

References

- [1] X. Zhang, B. Qian, and W. Zhou, "An intelligent method of violation identification for power operation," in *Proceedings of the 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, pp. 891–893, IEEE, Chongqing, China, November 2021.
- [2] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.
- [3] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: future vision, architecture, challenges and services," in

- Proceedings of the 2014 IEEE world forum on Internet of Things (WF-IoT)*, pp. 287–292, IEEE, Seoul, South Korea, March 2014.
- [4] J. S. Kumar and D. R. Patel, “A survey on internet of things: security and privacy issues,” *International Journal of Computer Application*, vol. 90, no. 11, pp. 10–21, 2014.
 - [5] L. Atzori, A. Iera, and G. Morabito, “The internet of things: a survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
 - [6] G. Ruan, Y. Zhong, and J. Jiang, “A 3D space violation detection method of substations based on the deep neural network,” in *Proceedings of the 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 16–19, IEEE, Chongqing, China, July 2021.
 - [7] H. Woo, E. Kang, S. Wang, and K. H. Lee, “A new segmentation method for point cloud data,” *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167–178, 2002.
 - [8] M. Y. Yang and W. Förstner, “Plane detection in point cloud data,” in *Proceedings of the 2nd International Conference on Machine Control Guidance*, vol. 8, no. 1, pp. 95–104, Bonn, Germany, March 2010.
 - [9] X. Liu, M. Yan, and J. Bohg, “MeteorNet: deep learning on dynamic 3D point cloud sequences,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9246–9255, Seoul, South Korea, October 2019.
 - [10] X. Yao, J. Guo, J. Hu, and Q. Cao, “Using deep learning in semantic classification for point cloud data,” *IEEE Access*, vol. 7, no. 12, pp. 37121–37130, 2019.
 - [11] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, “A review of deep learning-based semantic segmentation for point cloud,” *IEEE Access*, vol. 7, no. 4, pp. 179118–179133, 2019.
 - [12] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, “Review: deep learning on 3D point clouds,” *Remote Sensing*, vol. 12, no. 11, pp. 1729–1738, 2020.
 - [13] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, “Deep learning for 3D point clouds: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.
 - [14] C. R. Qi, H. Su, and K. Mo, “PointNet: deep learning on point sets for 3D classification and segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, Honolulu, HI, USA, July 2017.
 - [15] X. Hu and Y. Yuan, “Deep-learning-based classification for DTM extraction from ALS point cloud,” *Remote Sensing*, vol. 8, no. 9, pp. 730–738, 2016.
 - [16] S. Ravanbakhsh, J. Schneider, and B. Poczos, “Deep learning with sets and point clouds,” *arXiv*, vol. 9, no. 3, pp. 210–215, 2016.
 - [17] B. Li, “3D fully convolutional network for vehicle detection in point cloud,” in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1513–1518, IEEE, Vancouver, BC, Canada, September 2017.
 - [18] A. Kashefi, D. Rempe, and L. J. Guibas, “A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries,” *Physics of Fluids*, vol. 33, no. 2, pp. 104–112, 2021.
 - [19] Y. Cui, R. Chen, and W. Chu, “Deep learning for image and point cloud fusion in autonomous driving: a review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 9, pp. 2000–2009, 2021.
 - [20] J. Wang, C. Xu, L. Dai, J. Zhang, and R. Zhong, “An unequal deep learning approach for 3-D point cloud segmentation,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7913–7922, 2021.
 - [21] X. Chen, H. Ma, and J. Wan, “Multi-view 3d object detection network for autonomous driving,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, Las Vegas, NV, USA, July 2017.
 - [22] C. R. Qi, W. Liu, and C. Wu, “Frustum pointnets for 3d object detection from rgb-d data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, Salt Lake City, UT, USA, June 2018.
 - [23] B. Li, T. Zhang, and T. Xia, “Vehicle detection from 3d lidar using fully convolutional network,” *Robotics: Science and Systems 2016*, vol. 5, no. 15, pp. 1098–1104, 2016.
 - [24] Y. Zhou and O. Tuzel, “Voxelnet: end-to-end learning for point cloud based 3d object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, Salt Lake City, UT, USA, June 2018.
 - [25] Z. Ouyang, X. Dong, and J. Cui, “PV-EncoNet: fast object detection based on colored point cloud,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 16, pp. 387–397, 2021.
 - [26] W. Zhang, X. Fu, and W. Li, “Learning deformable network for 3D object detection on point clouds,” *Mobile Information Systems*, vol. 2021, no. 11, Article ID 3163470, 9 pages, 2021.
 - [27] H. Hu, M. Zhu, M. Li, and K. L. Chan, “Deep learning-based monocular 3D object detection with refinement of depth information,” *Sensors*, vol. 22, no. 7, pp. 2576–2586, 2022.