

Research Article

A Novel DBSCAN Clustering Algorithm via Edge Computing-Based Deep Neural Network Model for Targeted Poverty Alleviation Big Data

Hui Liu ^{1,2}, Yang Liu ³, Zhenquan Qin ¹, Ran Zhang ¹, Zheng Zhang ⁴,
and Liao Mu ¹

¹School of Software Technology, Dalian University of Technology, Dalian 116024, China

²Faculty of Business and Management, Universiti Teknologi MARA Sarawak Branch Jalan Meranek, 94300 Kota Samarahan, Sarawak, Malaysia

³International School, Shenyang Jianzhu University, China

⁴International School of Information Science and Engineering, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Yang Liu; liuyang@sjzu.edu.cn and Zhenquan Qin; qzq@dlut.edu.cn

Received 14 January 2021; Accepted 15 June 2021; Published 28 June 2021

Academic Editor: Jun Cai

Copyright © 2021 Hui Liu 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.

Big data technology has been developed rapidly in recent years. The performance improvement mechanism of targeted poverty alleviation is studied through the big data technology to further promote the comprehensive application of big data technology in poverty alleviation and development. Using the data mining knowledge to accurately identify the poor population under the framework of big data, compared with the traditional identification method, it is obviously more accurate and persuasive, which is also helpful to find out the real causes of poverty and assist the poor residents in the future. In the current targeted poverty alleviation work, the identification of poor households and the matching of assistance measures are mainly through the visiting of village cadres and the establishment of documents. Traditional methods are time-consuming, laborious, and difficult to manage. It always omits lots of useful family information. Therefore, new technologies need to be introduced to realize intelligent identification of poverty-stricken households and reduce labor costs. In this paper, we introduce a novel DBSCAN clustering algorithm via the edge computing-based deep neural network model for targeted poverty alleviation. First, we deploy an edge computing-based deep neural network model. Then, in this constructed model, we execute data mining for the poverty-stricken family. In this paper, the DBSCAN clustering algorithm is used to excavate the poverty features of the poor households and complete the intelligent identification of the poor households. In view of the current situation of high-dimensional and large-volume poverty alleviation data, the algorithm uses the relative density difference of grid to divide the data space into regions with different densities and adopts the DBSCAN algorithm to cluster the above result, which improves the accuracy of DBSCAN. This avoids the need for DBSCAN to traverse all data when searching for density connections. Finally, the proposed method is utilized for analyzing and mining the poverty alleviation data. The average accuracy is more than 96%. The average F -measure, NMI, and PRE values exceed 90%. The results show that it provides decision support for precise matching and intelligent pairing of village cadres in poverty alleviation work.

1. Introduction

In recent years, deep learning has achieved great success in some fields; especially, the deep neural network (DNN) method has achieved good results on various tasks, such as autonomous driving, intelligent speech, and image recognition [1–4]. DNN

is mainly composed of multiple convolutional layers and fully connected layers. All the layers process the input data, transmit it to the next layer, and output the results at the final layer.

The high-precision DNN model has many network layers and requires more computing resources. Currently, for intelligent application tasks, DNN is usually deployed on cloud

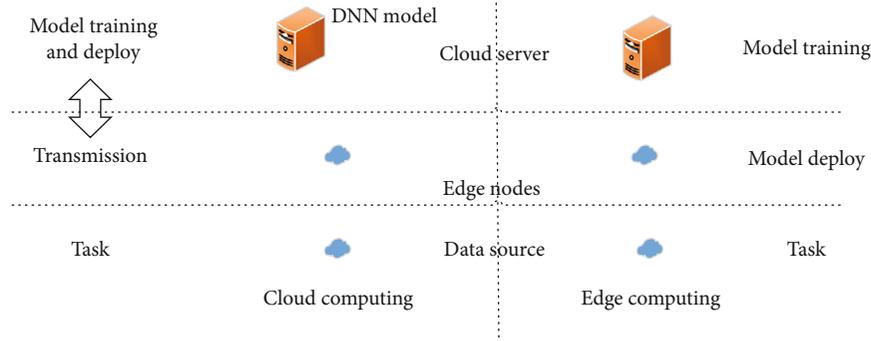


FIGURE 1: Comparison of cloud computing and edge computing under the deep learning model.

servers with sufficient computing resources. In this way, the data source needs to transfer the task data to the model in the cloud. The cloud computing model has the problems with high latency, low privacy, and high communication cost; so, it cannot meet the task requirements well. Some researchers attempt to deploy the cloud model to mobile devices, but due to the limited resources of mobile devices, only simple machine learning methods can be run, which results in low recognition accuracy [5–7].

As a new computing model, edge computing pushes computing power to the edge, which has attracted extensive attention from researchers [8–10]. In the edge computing scenario, the DNN model is deployed on the edge computing nodes around the device [11]. Edge computing nodes are much closer to data sources than cloud services; so, the low latency feature can be easily implemented [12]. However, the processing capacity of current edge computing devices is limited, and a single edge computing node may not be able to complete the inference task of complex network model well; so, multiple edge computing nodes are required to jointly deploy the DNN model. The main challenge of deploying DNN is how to select the appropriate computing nodes to deploy the model. Taking the segmentation of the neural network model, the computing requirements of the model and the network conditions of the edge computing nodes were taken into account; so, it optimizes the delay when multiple computing nodes jointly run the neural network model. Figure 1 shows the deep learning model under cloud computing and edge computing, respectively. In the cloud computing scenario, the data sent by the user device is uploaded to the cloud server through the network. Under the edge computing scenario, the cloud server only conducts the model training process of delay tolerance and then utilizes the trained model on the edge node. The data source transmits the task data to the edge node, and the DNN model returns the operation result for the data source.

Targeted poverty alleviation is a way to accurately identify, assist, and manage poverty alleviation targets through scientific and effective procedures in accordance with the conditions of farmers living in different poverty areas. Targeted poverty alleviation based on big data is to build a poverty portrait of poor households through the mining and analysis of poverty alleviation data and to carry out all-round identification and evaluation for

poor people [13–16], so as to find out the poor population, figure out the causes of poverty, and send poverty alleviation policies. Poverty portrait generally includes poverty index, poverty characteristics, and matching assistance measures of poor households. However, the data mining method is rarely adopted for targeted poverty alleviation. Data mining analyzes data from the vast amounts of data contained in the hidden rule. The data mining includes prior knowledge-based classification, clustering without prior knowledge, association rule mining based on association rules, and intelligent algorithm based on machine learning.

Cluster analysis is an important data mining technology, which plays a very important role in data mining [17–19]. Cluster analysis technology is widely used in other research fields, such as machine learning, artificial intelligence, image processing, and cloud computing. Clustering is the process of dividing a data set into different clusters according to the similarity between data objects. The data objects belonging to the same cluster are as similar as possible. Data objects belonging to different clusters are as different as possible. So far, many different clustering algorithms have been proposed, including partition-based clustering algorithm, hierarchical clustering algorithm, density-based clustering algorithm, grid-based clustering algorithm, and fuzzy-based clustering algorithm.

The partition-based clustering algorithm uses the iterative control strategy to optimize an objective function and constantly changes the data objects in the clustering center by iterative relocation, thus improving the partitioning results in each time. The K -means algorithm is a classical partition-based clustering algorithm. Since the K -means algorithm randomly selects the clustering center, the selection of the initial clustering center will have a great impact on the clustering results, and the algorithm cannot deal with nonspherical clusters. The hierarchical clustering algorithm can handle nonspherical clusters. The Chameleon algorithm is a kind of hierarchical clustering algorithm that mixes “top down” and “bottom up” strategies. The Chameleon algorithm first constructs the k -neighborhood graph on the original data set then uses an efficient graph partition algorithm to partition the k -neighborhood graph and get the initial class cluster; finally, it merges the subclusters. However, the algorithm is sensitive to noise points and has high time complexity. The density-based clustering algorithm [20] is measured by density correlation between data points; according

to the setting threshold, the density of the density exceeds the threshold of the adjacent areas connected data cluster. Compared with partitioning-based clustering, density-based clustering can find the clustering with arbitrary shapes, and the unique outlier processing strategy can process the abnormal data effectively. The DBSCAN algorithm can discover clusters with arbitrary shape, and it is not sensitive to noise. But the time complexity is very high, and it is more sensitive to the neighborhood parameter, since different parameters can lead to different clustering results. In order to solve the problem of parameter sensitivity, Bryant et al. [21] proposed a density estimation method using the reverse nearest neighbor as the data object and used a k -neighborhood graph similar to DBSCAN for clustering.

The grid-based clustering algorithm divides the data object space into finite units according to different dimensions, and all processing takes the unit as the object [22]. In this method, the clustering operation of data sets is transformed into the blocks processing in data space, thus improving the efficiency of the algorithm. The CLIQUE algorithm [23] combines the characteristics of grid-based and density-based clustering algorithms and makes use of the priori properties of frequent patterns and association rule mining to obtain the monotony of dense elements in terms of dimension. Then, the clustering is performed by identifying dense elements. The FCM algorithm is a classic fuzzy-based clustering algorithm, which optimizes an objective function iteratively and allocates data objects according to membership matrix. Because the number of class clusters is specified in advance, if the parameter is not selected properly, it is easy to fall into local optimum.

Rodriguez et al. [24] proposed the density peak clustering (DPC) algorithm in 2014, which did not need to specify the number of class clusters in advance, and it only required fewer parameters to find nonspherical clusters and was insensitive to noise. However, the DPC algorithm also has many shortcomings including the following aspects: (1) the artificial setting of the truncation distance has a certain randomness, which has a great influence on the clustering results, and (2) when calculating the local density of data objects, it does not consider the structure difference within the data sets. When the data density difference between the clusters is large, the ideal clustering result cannot be obtained; (3) it cannot handle high dimensional data and large scale data well. In view of the shortcomings of the DPC algorithm, many algorithms have been proposed to solve the above problems. Chen et al. [25] proposed an algorithm to redefine the measurement method of truncation distance and local density by combining the concept of k -nearest neighbor, which could generate the truncation distance adaptively for any data set and make the calculation result of local density more consistent with the real distribution of data. At the same time, distance ratio contest was introduced in the decision graph to replace the original distance parameter so that the center of class cluster was more obvious in the decision graph. Guo et al. [26] adopted the idea of k -nearest neighbor to calculate the local density of data objects. Although the influence of truncation distance parameter on clustering results was largely solved, the choice

of parameter k needed further study. Jin et al. [27] introduced natural neighbors to calculate the local density of the data object, thus solving the problem of parameter k selection. Li et al. [28] proposed a prominent peak clustering (PPC) algorithm based on significant density peak. The main idea of the algorithm was to divide data objects into multiple clusters and then merged the clusters with no obvious density peak to obtain accurate clustering results. Chen et al. [29] defined the local density of data objects through the law of universal gravitation and established a two-step strategy based on the first cosmic velocity to allocate the remaining data objects, so as to make the allocation of the remaining data objects more accurate. Yu et al. [30] improved the DPC algorithm by introducing weighted local density sequence and two-step allocation strategy and then used the nearest neighbor dynamic table to improve the clustering efficiency of the algorithm. Du et al. [31] introduced the k -nearest neighbor and PCA method into the DPC algorithm, which made it handle high-dimensional data well. Xu et al. [32] used the similarity index MS to allocate data points of the data set and then redefined noise points in the boundary region by the d_c -nearest neighbor method. This algorithm was suitable for data sets with high dimensional data and complex data structure, but it could not automatically determine the clustering center. Xu et al. [33] proposed a method by using grid to process large-scale data. For the above methods, they still have low efficiency when processing high dimension data.

The DBSCAN clustering algorithm uses fixed Eps and minPts (two input parameters), and the effect of processing multidensity data is not ideal. The time complexity of the algorithm is $O(N^2)$. To solve the above problems, a DBSCAN multidensity clustering algorithm based on region division is proposed. Our main contributions are as follows:

- (1) First, we deploy an edge computing-based deep neural network model
- (2) Then, in this constructed model, we execute data mining for poverty-stricken family
- (3) In this paper, the DBSCAN clustering algorithm is used to excavate the poverty features of the poor households and complete the intelligent identification of the poor households
- (4) In view of the current situation of high-dimensional and large-volume poverty alleviation data, the algorithm uses the relative density difference of grid to divide the data space into regions with different densities and adopts the DBSCAN algorithm to cluster the above result, which improves the accuracy of DBSCAN
- (5) Experiments show that the algorithm can cluster multidensity data effectively and has strong adaptability to various data and better efficiency

This paper is organized as follows. In section 2, the DNN model based on edge computing for this paper is introduced.

Then, we give the detailed proposed DBSCAN algorithm in Section 3. Section 4 displays the experiment and analysis. There is a conclusion in Section 5.

2. Model Deployment Based on Edge Computing

To deploy the DNN model in an edge computing scenario, the training process of the model needs to be completed on the cloud server firstly. The structure of DNN is an ordered sequence between layers. Each layer receives the output data from the previous layer and processes it, then transmits it to the next layer. So, the DNN can be constructed into a DNN model with multiple branches, each branch is composed of a specific neural network layer, and the multiple branches constitute a complete DNN model. The edge computing nodes are composed of a variety of devices with different computing performances, which are numerous, independent, and scattered around the users. A single edge computing node can only run a simple model with low accuracy [34, 35]; so, the branching DNN model is distributed to multiple edge computing nodes. Because each branch has a different network layer structure, and this will lead to different requirements on the computing resources of the deployment nodes for each branch; that is, the deployment of the same model to different nodes will lead to different running delays. Moreover, considering the different network conditions between different edge computing nodes, the distributed deployment of the branch neural network needs to comprehensively consider the computing capacity of nodes, branch model structure, and data transmission between nodes. Therefore, it is necessary to select the best edge computing node deployment for a given DNN branching model structure.

In the edge computing scenario, given edge computing node set $E = \{e_1, e_2, \dots, e_1, \dots, e_m\}$ and DNN model set with n branches $D = \{d_1, d_2, \dots, d_z, \dots, d_n\}$, where d_z stands for the n th branch, and the n model branches require the running order, f_{ij} represent the running order of branch model i and j .

3. Proposed DBSCAN Algorithm

The DBSCAN algorithm is a classical clustering algorithm based on density. This algorithm calculates the Eps neighborhood of each data object and obtains clustering results by clustering the densified data objects into a class cluster. The DBSCAN algorithm can automatically determine the number of class clusters, find any shape of class clusters, and is not sensitive to noise data. Given a d -dimensional data set $D(i = 1, 2, \dots, d)$, DBSCAN is defined as follows:

Definition 1. The Eps neighborhood of the data object p . The Eps neighborhood $N_{Eps}(p)$ of the data object $\forall p \in D$ is defined as the set of points contained within the region of a d -dimensional hypersphere with p as the center of sphere and Eps as the radius, i.e., $N_{Eps}(p) = \{q \in D | \text{dist}(p, q) \leq \text{Eps}\}$,

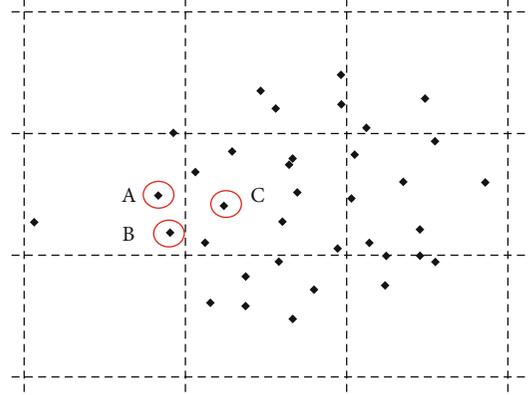


FIGURE 2: Boundary processing.

where D is the data set in the d -dimensional space, and $\text{dist}(p, q)$ represents the distance between points p and q in D .

Definition 2. Core data objects. Given the parameters Eps and minPts, for the data object p , if the object number of p in the Eps neighborhood meets $|N_{Eps}(p)| \geq \text{minPts}$, then p is called the core object.

Definition 3. Directly density reachable. Given Eps and minPts, for the data object $p, q \in D$, if p satisfies the two conditions: $p \in N_{Eps}(q)$ and $|N_{Eps}(q)| \geq \text{minPts}$, then p is called directly density reachable about Eps and minPts. In addition, the directly density reachable does not satisfy symmetry.

Definition 4. Density reachable. Given Eps and minPts, for the data object $p, q \in D$, if there is object sequence $p_1, p_2, \dots, p_n \in D, p_1 = q, p_n = p, p_{i+1}$ is directly density reachable, then p is called density reachable about Eps and minPts. In addition, the density reachable does not satisfy symmetry.

Definition 5. Density connection. Given Eps and minPts, for the data object $p, q \in D$, if there is a data o that makes p and q are density reachable, so p and q are called density connection about Eps and minPts. Therefore, density connection satisfies symmetry.

When given Eps and minPts, the simple flow of the DBSCAN algorithm is as follows. It selects any unpartitioned data object and determines whether it is a core data object. If so, it finds all data objects with density reachable and labels these data objects as a class. Otherwise, the noise data will be judged. If it is a noise point, it will be labeled; if it is not a noise, the object will not be processed. This is repeated until all data objects have been partitioned.

Given a d -dimensional data set $D(i = 1, 2, \dots, d)$, the number of data is N , and any dimension attribute A_i in D is bounded. Let the value of the i th dimension be in the interval $Rg_i = [l_i, h_i]$, and then $S = Rg_1 \cdot Rg_2 \cdot \dots \cdot Rg_d$ is the d -dimensional data space. Each dimension of data space is divided into an interval with equal length and mutually disjoint to form a grid unit. These grids are left closed and right

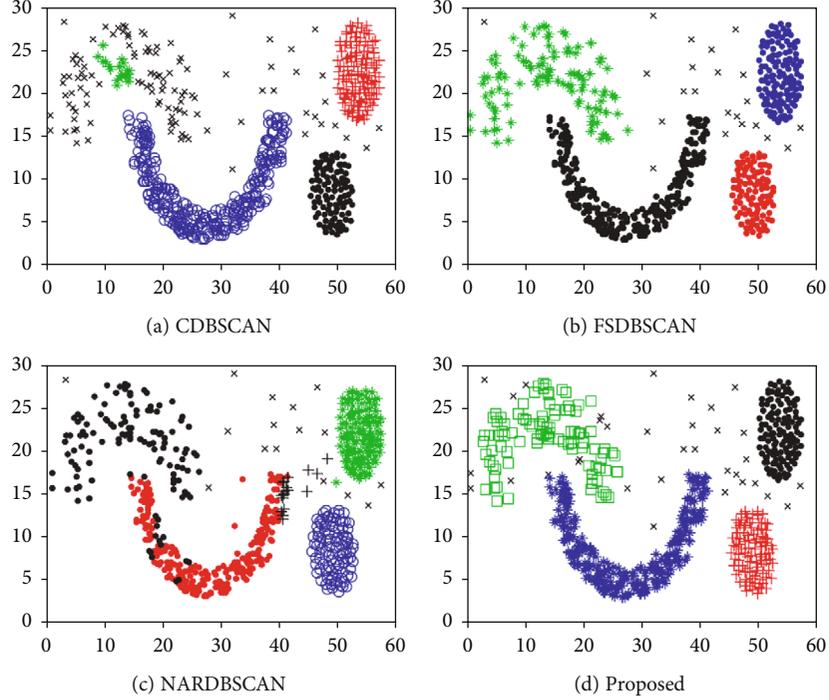


FIGURE 3: DS1 cluster result.

open in every dimension. In this way, the data space is divided into $\prod \text{num}_i$ super rectangular grid cells with equal volume (num_i is the number of intervals on the i th dimension in the data space). Set the grid edge length as

$$\text{length} = \alpha \cdot \sqrt[d]{\prod_{i=1}^d \frac{(h_i - l_i)}{N}}, \quad (1)$$

where α is the grid control factor, which is used to control the size of the grid. $\alpha = 1.5$ is used in all experiments in this paper. According to the grid side length, the number of intervals on each dimension can be calculated as

$$\text{num}_i = \lceil (h_i - l_i) / \text{length} \rceil. \quad (2)$$

It maps each object in the data set to the corresponding grid. For each data object $X(x_1, x_2, \dots, x_d)$, the subscript of the corresponding grid on each dimension is

$$\text{ind}_i = \lceil (x_i - l_i) / \text{length} \rceil. \quad (3)$$

For each object X in the data set, it is mapped to the corresponding grid g according to equation (3), and the number of objects in grid g is $\text{den}(g)$.

Adjacent grid is defined as if the grid g_1 and g_2 are adjacent, then $|\text{ind}_i(g_1) - \text{ind}_i(g_2)| \leq 1$ ($i = 1, 2, \dots, d$), and there are at most $3^d - 1$ adjacent grids.

The relative density difference of the grid is used; that is, two grid cells g_1 and g_2 have the density of $\text{den}(g_1)$ and $\text{den}(g_2)$, respectively, and the relative density difference of

TABLE 1: Comparison results with different methods.

Method	F -measure	NMI	PRE
CDBSCAN	0.8346	0.8246	0.8535
FSDBSCAN	0.9035	0.8935	0.9179
NARDBSCAN	0.9177	0.9085	0.9215
Proposed	0.9219	0.9126	0.9377

g_2 relative to g_1 is defined as

$$\text{rgdd}(g_1, g_2) = \frac{|\text{den}(g_1) - \text{den}(g_2)|}{\text{den}(g_1)}. \quad (4)$$

Formula (4) is used as a condition for grid merging. Firstly, the grid with the highest density is selected as the initial cell grid g_0 , and the relative density difference $\text{rgdd}(g_0, g)$ between adjacent grid g and g_0 is calculated according to equation (4) in turn. If $\text{rgdd}(g_0, g) < \varepsilon$ (ε is a given parameter), then g and g_0 are merged. The initial unit grid g_0 becomes the merged big grid area, the initial unit grid density $\text{den}(g_0) = \text{den}(g_0) + \text{den}(g) / G_{\text{num}}$, where G_{num} represents the merged grid number; that is, the initial grid cell density is dynamic. It continues to extend outward combined grid with this method, until all the boundary grid does not meet the $\text{rgdd}(g_0, g) < \varepsilon$, because some points in the boundary grid may be boundary points for this region, as shown in Figure 2. Data points A and B are the boundary points of cluster C, which cannot be regarded as noise loss. Therefore, the boundary grid is also merged into the region (the

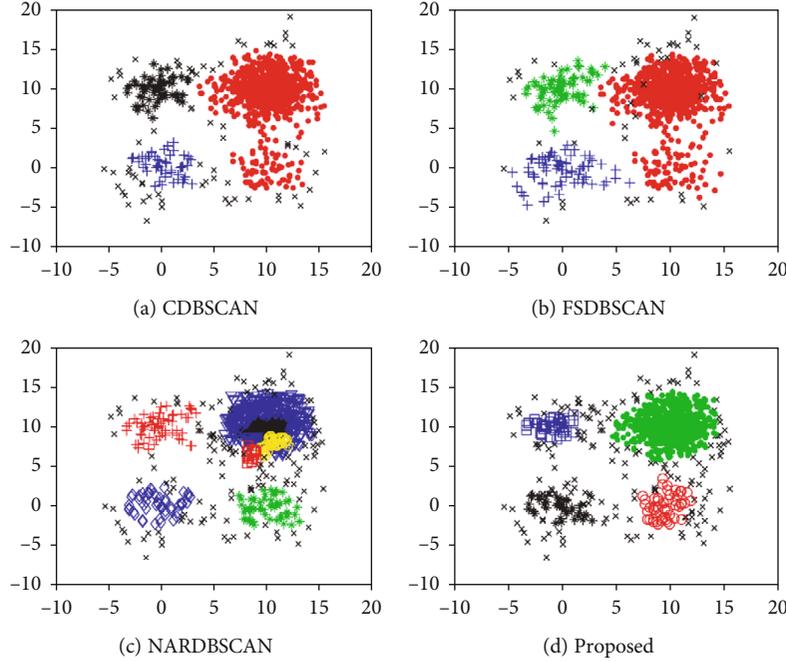


FIGURE 4: DS2 cluster result.

boundary grid is no longer extended outward), and the merged grid forms a region, whose data set is denoted as D_1 .

Then, it takes the densest of the remaining unprocessed grids as the initial unit grid and repeats the above steps until the remaining data points can no longer be clustered. So, the data set D is divided into D_1, D_2, \dots, D_K and the initial noise point.

The above region division method has roughly divided the data set D into K data regions and noise points with different densities. Then, DBSCAN clustering is carried out for each region with different densities. According to the grid partitioning method, the grid with the largest density of the remaining grid is taken as the initial grid unit every time; so, the density from the first region to the k th region is basically decreasing.

According to the grid-based method, the grid with the largest density is used as the initial grid cell in each division. So, the density basically decreases from the first region to the K th region. When conducting cluster, this paper inputs Eps and $minPts$ for the first region D_1 to perform DBSCAN clustering. Eps is automatically obtained from D_2 to D_K by the following formula:

$$Eps_i = \frac{Eps \times num_1}{G_1} / \frac{num_i}{G_i}, i = 2, 3, \dots, K, \quad (5)$$

where num_i represents the data number in D_i . G_i represents the merged grid number in the i th region.

In this paper, the data space is first roughly divided into different data regions through grid partitioning. Then, Eps is automatically obtained according to different densities for

TABLE 2: Comparison results with different methods.

Method	F -measure	NMI	PRE
CDBSCAN	0.8752	0.8165	0.9069
FSDBSCAN	0.9169	0.8747	0.9344
NARDBSCAN	0.9158	0.8732	0.9317
Proposed	0.9428	0.9336	0.9572

each region, and DBSCAN clustering is performed. Region partitioning reduces the unnecessary query operation of density connection in the DBSCAN algorithm and improves the efficiency. Eps of the DBSCAN algorithm is automatically obtained according to the different densities of their respective regions, which makes it more adaptable to data, especially when processing multidensity data, and the effect is better.

The specific description of the proposed algorithm is as follows.

Input: dataset D , Eps , $minPts$, and ϵ .

According to equations (1)~(3), the grid is partitioned, and the data is boxed. The data set D is mapped to the partitioned grid, and the density of each grid is counted.

According to equation (4) and grid density difference parameter ϵ , the grids with similar densities and adjacent distances are merged. In this way, the data space is divided into different regions, and data blocks D_1, \dots, D_K , and preliminary noise points are generated.

DBSCAN clustering is performed for data regions with different densities according to parameter Eps , $minPts$, and equation (5).

Output: clustering results.

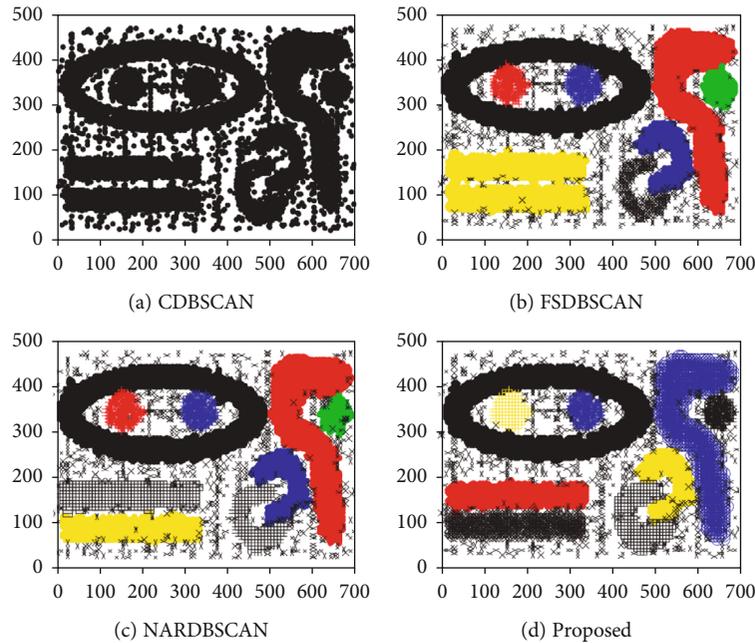


FIGURE 5: DS3 cluster result.

4. Experiments and Analysis

All algorithms in this paper are realized and processed by MATLAB. The experimental environment is as follows: CPU-Intel i7, Memory-4 GB, and Windows 10. We compare the proposed algorithm in this paper with the CDBSCAN [36], FSDBSCAN [37], and NARDBSCAN [38]. In this paper, three indexes including precision (PRE), normalized mutual information (NMI), and F -measure, which are widely used in clustering algorithms, are adopted as the performance measurement criteria for the clustering algorithms, where the value of NMI, F -measure, and PRE is $[0,1]$. The higher value denotes the better clustering result. The experiment data sets are from UCI machine learning repository (<http://archive.ics.uci.edu/ml/datasets.php>).

4.1. Experiment 1. Data set DS1 has four classes: noise points and multiple densities. The clustering results are shown in Figure 3.

As can be seen from Figure 3(a), data loss in low-density regions is serious when noise points are not absorbed in high-density regions. Conversely, the low-density region does not lose data, the high-density region will absorb the noise point, or the two U-shaped data on the left will be merged into a class. The clustering results in Figure 3(b) are ideal, which can identify classes with noise in the multidensity data, but the algorithm efficiency needs to be improved. In Figure 3(c), the processing effect of irregular graph boundary points is poor. In Figure 3(d), the clustering algorithm in this paper adopts the grid division method to perform DBSCAN clustering for different densities, which not only identifies four classes but also almost does not absorb noise points. In DS1 data set, the average values with different methods are shown in Table 1.

Table 1 shows the clustering results of the four algorithms on the dataset DS1. It can be seen from the comparison of F -measure, NMI, and ACC that the clustering of the proposed algorithm in this paper is higher than other algorithms on dataset DS1.

4.2. Experiment 2. This experiment uses a data set DS2 with class boundary interference. The results are shown in Figure 4.

As can be seen from Figure 4(a), CDBSCAN identifies three classes, and the class with low density and close to high density is absorbed into the high density region and absorbs more noise points. In Figure 4(b), the data center in the upper right corner has a high density, and the density gradually decreases along the edge, which is suitable for the edge data. As a result, the data in the lower right corner, which is close to the upper right corner and has a similar density, is merged into a class. In Figure 4(c), the NARDBSCAN algorithm divides the high-density region into multiple classes, and it lacks data processing from the density center to the edge. The algorithm in this paper adopts grid division and DBSCAN clustering algorithm with different parameters for different density blocks, which can well identify clusters with different density and absorb fewer noise points. Table 2 is the comparison result on data set DS2.

4.3. Experiment 3. In this experiment, data sets DS3 with large data volume, many clusters, and rich shapes are used for comparison. The data contains nine clusters. The experiment result is shown in Figure 5.

In Figure 5(b), the cluster processing of the FSDBSCAN method is ideal, but inevitably, the two classes that are close to each other are merged into one class. In Figure 5(c), low-density edges are treated as noise points, and relatively isolated points in the class are treated as noise points. In

TABLE 3: Comparison results with different methods on DS3.

Method	F-measure	NMI	PRE
CDBSCAN	0.7526	0.7359	0.7955
FSDBSCAN	0.8932	0.8132	0.8928
NARDBSCAN	0.9425	0.8872	0.9603
Proposed	0.9458	0.8893	0.9612

Figure 5(d), with the proposed method, the high-density region does not absorb too many noise points, and the low-density region is not divided or treated as noise; so, the multi-density data is well processed. Table 3 shows the detailed index values.

The above experiments mainly show the adaptability of the proposed algorithm to multidensity and arbitrary shape data clustering. The following is a quantitative analysis. The original data DS1, DS2, and DS3 are three-dimensional data, and the third dimension is the class label. If the class label in the clustering result is the same as the class label of the original data, the clustering is correct. The correct number of samples in clustering is denoted as R , and the total number of data as T [39]. The formula for calculating accuracy is

$$\text{Accuracy} = R/T \times 100\%. \quad (6)$$

The data in the following table is the average value after 20 times. DS1, DS2, and DS3 experimental results are shown in Tables 4–6.

4.4. Experiment 4. Identification of Poverty Features. This section adopts the proposed DBSCAN algorithm to process the real poverty alleviation data. The data source of poverty alleviation comes from a prefecture-level city, including 11,423,500 rural population, 196,700 rural households, and 68,000 poverty-stricken households. The original data includes the data of archived card, the data of visit, the data of agricultural cloud project, the data of education, and health and sanitation departments. According to the designed poverty alleviation data indicator system, the data of poverty alleviation can be filled with ETL tool. Finally, the proposed DBSCAN algorithm in this paper is applied to analyze the poverty alleviation index data to intelligently identify the poverty characteristics of the poor households and carry out the corresponding visual display and analysis to provide decision support for the regional assistance work.

In this paper, the proposed DBSCAN clustering algorithm based on local sensitive hash is applied to complete the mining and analysis of the preprocessed poverty alleviation index data matrix to identify the poverty characteristics of the poor households in different regions. Feature classification is carried out by clustering algorithm, and 8 data clusters are finally identified by consensus with each cluster corresponding to a feature category. For each data cluster, it takes the central data object as the cluster. The index whose value exceeds the threshold value of 0.7 is the impoverishment index of the data object. The poverty characteristics identi-

TABLE 4: DS1 result.

Method	Time/ms	Accuracy/%
CDBSCAN	262.9	82.63
FSDBSCAN	218.5	96.3
NARDBSCAN	114.5	95.28
Proposed	45.9	99.24

TABLE 5: DS2 result.

Method	Time/ms	Accuracy/%
CDBSCAN	208.4	85.84
FSDBSCAN	125.8	93.64
NARDBSCAN	63.1	94.82
Proposed	38.7	96.87

TABLE 6: DS3 result.

Method	Time/ms	Accuracy/%
CDBSCAN	669.4	88.95
FSDBSCAN	263.7	94.62
NARDBSCAN	90.7	96.27
Proposed	55.5	99.36

TABLE 7: Poverty feature recognition accuracy.

Method	Accuracy	Time
CDBSCAN	91.0%	One week
FSDBSCAN	92.3%	4 days
NARDBSCAN	94.5%	2.5 days
Proposed	96.5%	1358s

fied in this paper are compared and corresponded to the causes of poverty in the registered cards of poor households. On the one hand, it refines and complements the general causes of poverty in the archival registration cards to facilitate the matching of assistance measures and assistance cadres in the later period. On the other hand, the identification results of this paper and the causes of poverty can be used to verify each other.

Table 7 shows the comparison accuracy between the poverty characteristics identified in this paper and the causes of poverty caused by registered card. We give the overall comparison results. Because this article identifies poverty characteristics compared with cross tent card of poverty causes more refined, index, and rich content, the farmers that have the same poverty characteristics may correspond to multiple poverty reasons. When calculating the accuracy, for the poverty characteristics of a certain household, some or all of the poverty characteristics correspond to the causes of poverty caused by registered card, and we believe that the recognition result is consistent with the result of registered card.

5. Conclusions

Targeted poverty alleviation aims to build a well-off society in an all-round way. However, the current poverty alleviation work mainly relies on the establishment of documents, visiting by village cadres, and so on, which costs a lot of manpower and material resources, and lacks scientific and effective means with supervision management. This paper mainly studies the intelligent identification method of poverty characteristics for poor households, designs an intelligent identification scheme of poverty characteristics, and further provides data reference and guidance for accurate matching of assistance measures and pairing of assistance cadres. However, if the dimension of the datasets is very higher, the cluster effect is not perfect with the current experimental environment. In the future, we will adopt newly deep learning methods to improve the cluster effectiveness. The improved and advanced clustering technologies will be applied in targeted poverty alleviation of the poverty counties in China.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] Q. Zhang, C. Bai, Z. Chen et al., "Deep learning models for diagnosing spleen and stomach diseases in smart Chinese medicine with cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, p. 1, 2021.
- [2] P. Li, Z. Chen, L. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.
- [3] L. Zhao, Z. Chen, Y. Yang, V. C. M. Leung, and Z. Jane Wang, "Incomplete multi-view clustering via deep semantic mapping," *Neurocomputing*, vol. 275, pp. 1053–1062, 2018.
- [4] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.
- [5] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi et al., "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, no. 1, pp. 65–69, 2019.
- [6] R. Travadi and S. Narayanan, "Total variability layer in deep neural network embeddings for speaker verification," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 893–897, 2019.
- [7] P. Li, Z. Chen, L. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An incremental deep convolutional computation model for feature learning on industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1341–1349, 2019.
- [8] Z. Ning, X. Kong, F. Xia, X. Wang, and W. Hou, "Green and sustainable cloud of things: enabling collaborative edge computing," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 72–78, 2019.
- [9] A. A. Laghari, H. He, A. Khan, N. Kumar, and R. Kharel, "Quality of experience framework for cloud computing (QoC)," *IEEE Access*, vol. 6, pp. 64876–64890, 2018.
- [10] L. Liu, G. Han, Z. Xu, L. Shu, B. Peng, and M. Martinez-Garcia, "Predictive boundary tracking based on motion behavior learning for continuous objects in industrial wireless sensor networks," *IEEE Transactions on Mobile Computing*, 2021.
- [11] A. Laghari, H. He, S. Karim, H. A. Shah, and N. K. Karn, "Quality of experience assessment of video quality in social clouds," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 8313942, 10 pages, 2017.
- [12] G. Han, H. Wang, H. Guan, and M. Guizani, "A mobile charging algorithm based on multicharger cooperation in internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 684–694, 2021.
- [13] Y. Zhou, Y. Guo, Y. Liu, Y. Li, and W. Wu, "Targeted poverty alleviation and land policy innovation: some practice and policy implications from China," *Land Use Policy*, vol. 74, pp. 53–65, 2018.
- [14] Q. Zhang and T. Zhang, "Advances and practices for targeted poverty alleviation in China," *China Economic Transition*, vol. 1, no. 1, pp. 134–141, 2018.
- [15] W. Luo, Y. Tang, and X. Zhang, "Analysis on factors influencing farmers' participation behavior in rural tourism targeted poverty alleviation: based on embedding social structure theory," *Journal of Hunan Agricultural University*, vol. 20, no. 5, pp. 24–30, 2019.
- [16] C. Chen and J. Pan, "The effect of the health poverty alleviation project on financial risk protection for rural residents: evidence from Chishui City, China," *International Journal for Equity in Health*, vol. 18, no. 1, p. 79, 2019.
- [17] L. Teng, H. Li, and S. Yin, "Modified pyramid dual tree direction filter-based image denoising via curvature scale and nonlocal mean multigrade remnant filter," *International Journal of Communication Systems*, vol. 31, no. 16, article e3486, 2018.
- [18] M. Nagano, A. Komesu, and H. Miyata, "An evolutionary clustering search for the total tardiness blocking flow shop problem," *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1843–1857, 2019.
- [19] M. Parmar, D. Wang, X. Zhang et al., "REDPC: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.
- [20] X. Cui, J. Wang, F. Wu et al., "Extracting main center pattern from road networks using density-based clustering with fuzzy neighborhood," *International Journal of Geo Information*, vol. 8, no. 5, p. 238, 2019.
- [21] A. Bryant and K. Cios, "RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1109–1121, 2018.
- [22] L. Wang, S. Ding, Y. Wang, and L. Ding, "A robust spectral clustering algorithm based on grid-partition and decision-graph," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 5, pp. 1243–1254, 2021.
- [23] M. Hassani, Y. Kim, and T. Seidl, "Subspace MOA: subspace stream clustering evaluation using the MOA framework," in *International Conference on Database Systems for Advanced Applications*, Wuhan, China, 2013.

- [24] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
- [25] J. Chen and P. Yu, "A domain adaptive density clustering algorithm for data with varying density distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2310–2321, 2021.
- [26] Z. Guo, T. Huang, Z. Cai, and W. Zhu, "A new local density for density peak clustering," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Melbourne, VIC, Australia, 2018.
- [27] H. Jin and X. Z. Qian, "Optimized density peak clustering algorithm by natural nearest neighbor," *Journal of Frontiers of Computer Science and Technology*, vol. 13, no. 4, pp. 711–720, 2019.
- [28] L. Ni, W. Luo, W. Zhu, and W. Liu, "Clustering by finding prominent peaks in density space," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 727–739, 2019.
- [29] H. Chen, M. Ge, and Y. Xue, "Clustering algorithm of density difference optimized by mixed teaching and learning," *SN Computer Science*, vol. 1, no. 3, 2020.
- [30] D. Yu, G. Liu, M. Guo, X. Liu, and S. Yao, "Density peaks clustering based on weighted local density sequence and nearest neighbor assignment," *IEEE Access*, vol. 7, pp. 34301–34317, 2019.
- [31] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.
- [32] X. Xu, S. Ding, H. Xu, H. Liao, and Y. Xue, "A feasible density peaks clustering algorithm with a merging strategy," *Soft Computing*, vol. 23, no. 13, pp. 5171–5183, 2019.
- [33] X. Xu, S. Ding, and Z. Shi, "An improved density peaks clustering algorithm with fast finding cluster centers," *Knowledge-Based Systems*, vol. 158, pp. 65–74, 2018.
- [34] Z. Liao, G. Han, H. Wang, and L. Liu, "Multistation-based collaborative charging strategy for high-density low-power sensing nodes in industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7575–7588, 2021.
- [35] A. Laghari, A. K. Jumani, and R. A. Laghari, "Review and state of art of fog computing," *Archives of Computational Methods in Engineering*, pp. 1–13, 2021.
- [36] T. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemosometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92–96, 2013.
- [37] D. Abdolzadegan, M. Moattar, and M. Ghoshuni, "A robust method for early diagnosis of autism spectrum disorder from EEG signals based on feature selection and DBSCAN method," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 482–493, 2020.
- [38] Z. Han, M. Cheng, F. Chen, Z. Deng, and Y. Wang, "A spatial load forecasting method based on DBSCAN clustering and NAR neural network," *Journal of Physics Conference Series*, vol. 1449, article 012032, 2020.
- [39] Z. Xu, G. Han, H. Zhu, L. Liu, and M. Guizani, "Adaptive DE algorithm for novel energy control framework based on edge computing in IIoT applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5118–5127, 2021.