IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2019, accepted July 3, 2019, date of publication July 8, 2019, date of current version July 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927308

Differential Privacy-Preserving Density Peaks Clustering Based on Shared Near Neighbors Similarity

LIPING SUN, SHUTING BAO, SHANG CI, XIAOYAO ZHENG, LIANGMIN GUO, AND YONGLONG LUO
School of Computer and Information, Anhui Normal University, Wuhu 241002, China
Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu 241002, China

Corresponding author: Yonglong Luo (ylluo@ustc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602009 and Grant 61672039, and in part by the Natural Science Foundation of Anhui province under Grant 1808085MF172 and Grant 1908085MF190.

ABSTRACT Density peaks clustering is a novel and efficient density-based clustering algorithm. However, the problem of the sensitive information leakage and the associated security risk with the applications of clustering methods is rarely considered. To address the problem, we proposed differential privacy-preserving density peaks' clustering based on the shared near neighbors similarity method in this paper. First, the Euclidean distance and the shared near neighbors similarity were combined to define the local density of a sample, and the Laplace noise was added to the local density and the shortest distance to protect privacy. Second, the process of cluster center selection was optimized to select the initial cluster centers based on the neighborhood information. Finally, each sample was assigned to the cluster as its nearest neighbor with higher local density. The experimental results on both the UCI and synthetic datasets show that compared with other algorithms, our method more effectively protects the data privacy and improves the quality of the clustering results.

INDEX TERMS Privacy preservation, differential privacy, density peaks clustering algorithm, shared near neighbors similarity.

I. INTRODUCTION

The rapid development of information technology and network technology has brought people from the traditional Internet era into the big data era, the artificial intelligence era, and the IoT era. Facing huge amounts of data, traditional data analysis technology appears to be overstretched [1]. Extracting valuable information and models from vast amounts of data has become vital. As an important technology of data mining, clustering [2] is a kind of unsupervised data analysis method that classifies the original data into clusters based on the similarity between samples such that the similarity of samples in the same cluster is high and the similarity of samples that belong to different clusters is low.

Currently, several companies and organizations have large amounts of sensitive personal information, including patient disease information, online payment transaction records, and personal financial records. Hiding the identifying information (name, ID number, etc.) can protect the user's information security. Clustering is a technology mainly for mining and reuse of data, and the sensitive information may be leaked. Therefore, privacy-preserving clustering becomes an important issue.

Density peaks clustering (DPC) [3] is a density-based clustering algorithm, which can simply and efficiently identify clusters of arbitrary type without iteration. However, it has the following shortcomings: 1) The subjectively selected cutoff distance, which is denoted as d_c, can substantially influence the performance of the DPC algorithm. 2) The cluster centers that are selected by the DPC algorithm may be incorrect when it is applied to a large and complex dataset with large density variations. 3) In the density and distance calculations, private information may be leaked. Recently the proposed and existing privacy-protection models that are based on

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

equivalence classes include K-anonymity [4], l-diversity [5] and t-closeness [6]. They do not provide sufficient security and must be continuously improved according to emerging attack models. Therefore, the differential privacy-preserving model was proposed by Dwork [7]–[10], which overcomes the shortcomings of the traditional privacy- preserving models.

The main contributions of this paper can be summarized as follows:

1) The local density (denoted as ρ) and shortest distance (denoted as δ) with differential privacy are presented. To realize the objective of differential privacy preservation, random noise is added in the calculations of the local density and shortest distance to distort the data. At the same time, the shared near neighbors similarity and Euclidean distance are combined to calculate the local density of the samples, which effectively avoids selecting parameter d_c in the DPC algorithm.

2) Differential privacy-preserving density peaks clustering based on shared near neighbors similarity (DP-DPCSNNS) is proposed. The DP-DPCSNNS algorithm provides ε-differential privacy preservation and improves the accuracy of the clustering results. Shared near neighbors similarity is applied to detect cluster centers with neighborhood information, which improves the accuracy of cluster center selection.

3) Several comparative experiments are performed and six algorithms are compared in terms of ACC, AMI, F-Measure, and ARI metrics on UCI datasets and synthetic datasets. The experimental results verify the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows: Section II reviews the related work. In Section III, we introduce the basic definitions and processes of differential privacy and propose the DP-DPCSNNS schema. In Section IV, experiments are conducted to evaluate the performance of our proposed schema. Finally, we present the conclusions of this work and discuss future work in Section V.

II. RELATED WORK

Clustering analysis is an important field in data mining research and clustering methods have been widely applied in pattern recognition [11]–[13], social networks [14]–[16], and image processing [17]–[19]. Existing clustering algorithms mainly include partitioning, density, hierarchy grids and models. K-means [20] and DBSCAN [21] are classical clustering algorithms that are based on partitioning and density, respectively.

K-means identifies the optimal cluster centers through multiple iterations and allocates the remaining samples according to their distances to each cluster center. However, it cannot detect nonspherical clusters because the sample is always assigned to the nearest cluster center. If the initial cluster centers are selected incorrectly, the algorithm easily falls into a local optimum. DBSCAN measures the closeness of the data distribution by the selection of parameters *eps* (the neighborhood radius) and minpts (the minimum number of samples within the neighborhood radius of the core sample). This algorithm is sensitive to the value of parameter *eps*. A smaller value of *eps* may lead to overclustering, whereas a larger value of *eps* may cause small clusters to be merged.

The DPC algorithm relies on density and distance to detect clusters, and clusters various types of datasets using neither an iterative process nor additional parameters. Moreover, it is not necessary to map the data to the vector space and the performance is not affected by the dimension of the data space. Wu and Wilamowski [22] applied a grid to the DPC algorithm to reduce the number of distance calculations between samples and improve its performance. The density of the grid nodes instead of the samples is calculated. The main advantages of the DPC algorithm are that it detects noise data and identifies clusters of arbitrary shapes. Du *et al.* [23] used the concept of k-nearest neighbors to calculate the local density and reduced the dimension of the data by principal component analysis (PCA). Xie *et al.* [24] proposed a k-nearest neighbors density peaks clustering (KNN-DPC) algorithm. The algorithm defines the local density of the samples to be independent of the cutoff distance d_c and uses two new assignment strategies of samples that are based on k-nearest neighbors. Based on the KNN-DPC algorithm, Xie *et al.* [25] proposed a fuzzy weighted k-nearest neighbors density peaks clustering (FKNN-DPC) algorithm. The weights are defined to assign the remaining samples, which depend on the Euclidean distances between the samples. Mehmood *et al.* [26] proposed a fuzzy DPC algorithm. First, the expected cluster center set is filtered based on the Euclidean distance. Then, the lower density samples are excluded from this set. Next, the near clusters are merged according to the Euclidean distance between the local cluster centers. Xu *et al.* [27] applied a hierarchical clustering algorithm to design a new clustering algorithm. The linear fitting approach is used to determine the potential cluster centers based on the γ value. Some potential cluster centers will be ‘‘degraded’’ as noncenter samples in the hierarchical clustering process. Finally, the leading tree is used to allocate the remaining samples.

The DPC algorithm is often used in the Internet of things, social networks and other fields. Zhang *et al.* [28] combined the DPC algorithm with the k-medoids algorithm to design a dynamic clustering algorithm for sensor technology and wireless communication in the industrial Internet of things. Jia *et al.* [29] used the DPC algorithm to select a hyperspectral band during image processing. The ranking score of each band is computed from the local density and an exponential-based learning rule is employed to adjust the cutoff distance d_c. Wang *et al.* [30] applied the DPC algorithm to identify initial social circles with overlap in social networks. Then, cluster centers with smaller local density values were integrated. However, the DPC algorithm has the following drawbacks:

1) The subjectively selected cutoff distance d_c can substantially influence the performance of the DPC algorithm. The clustering accuracy depends on the density estimate of

the dataset. If there is a large difference between two clusters in terms of density, the local densities of the centers of the two clusters should also differ substantially. However if d_c is very small, the local densities of the two cluster centers will be similar, thereby resulting in unsatisfactory clustering.

2) The DPC algorithm may select incorrect cluster centers if it is applied to a large and complex dataset with large density variations. This is because the density and distance metrics that are used by the DPC algorithm are relatively simple and the process does not take into account the impact of the neighborhood environment of the samples. If the density of a cluster is low and its distance from a cluster with a higher density is small, the cluster center of the low-density cluster is likely to be mistakenly selected because compared to the ρ values of the centers of low-density clusters, border samples of the high-density cluster have higher values of ρ, while their δ values are similar. The erroneous selection of cluster centers will directly lead to incorrect clustering results.

3) Privacy security problem. The purpose of data mining is to analyze the available data and obtain valuable knowledge. In the process of data analysis, it may lead to the outflow of sensitive personal data. And there is the risk of personal privacy disclosure. The selection of cluster centers is easy to divulge personal privacy information during clustering. And the publishing of the local density ρ and shortest distance δ may result in the disclosure of privacy in DPC algorithm.

Recently, differential privacy-preserving clustering has become a research hotspot. Blum *et al.* [31] proposed the differential privacy K-means algorithm on the SuLQ platform. However, the sensitivity of its query function is high and the algorithm does not specify how to set the privacy budget, which reduces the availability of the clustering results. Yu *et al.* [32] proposed an outlier-elimination K-means approach that is based on differential privacy. The outliers are eliminated in the data preprocessing step according to the density of the samples and Laplace noise is added in the selection of cluster centers. Yang *et al.* [33] proposed a differential privacy-preserving K-means clustering (IDP-Kmeans) algorithm. The dataset is divided into several subsets and the center of each subset is calculated after data distortion, which is used as the initial cluster center and improves the clustering performance. Gao and Zhang [34] combined cuckoo search and the particle swarm optimization algorithm to improve the selection process of K-means cluster centers. The algorithm realizes differential privacy protection and applies parallelism on the Apache Spark engine. Ni *et al.* [35] proposed a differential privacy-preserving multicore DBSCAN clustering (DP-MCDBSCAN) algorithm, which redefines the selection process of the initial cluster centers and adds Laplace noise in the calculation of the distance to realize differential privacy preservation.

The DP-DPCSNNS algorithm is proposed in this paper, which overcomes the shortcomings of the DPC algorithm under the framework of the differential privacy-preserving model. It realizes a satisfactory balance between clustering performance and data availability.

III. DP-DPCSNNS ALGORITHM

A. DIFFERENTIAL PRIVACY-PRESERVING MODEL

The differential privacy-preserving model defines a rigorous attack model, which is derived from a hypothesis. On the one hand, the traditional privacy protection technology limits the background knowledge of an attacker. On the other hand, it assumes a special attack mode. A split attack hypothesis is the greatest background knowledge an attacker can grasp. An attacker can have all the data record information except the privacy record of the target being attacked. Differential privacy can still guarantee the privacy of target data under this maximum attack assumption. For other possible attacks, the attacker's background knowledge must be less than the maximum attack assumption.

Consider a dataset D that contains an arbitrary attack object A. Assume the attacker has the highest possible level of background knowledge, namely, the attacker already has information on all records except A. Then, the attacker performs arbitrary query operations f on D and the result is denoted as $f(D)$. Dataset D' is obtained by removing A from D. The attacker performs the same query operation f on D' and the result is denoted as $f\left(D'\right)$. If $f\left(D'\right)$ is approximately equal to $f\,(D)$, then regardless of whether A belongs to D, the attacker cannot obtain additional information about A from similar query results. The differential privacy-preserving model guarantees that the results of query operations on two neighboring datasets (two datasets that differ in at most one row) are similar, which protects the privacy of each sample in the dataset.

Definition 1 (Differential Privacy): Assume R is a random function, S represents all outputs of R, and ε represents the disclosure risk of an event E. For all datasets D and R that differ in at most one row, R provides ε-differential privacy if R satisfies the following inequality:

$$\Pr[R(D) \in S] \leq \exp(\varepsilon) \times \Pr[R(D') \in S] \tag{1}$$

where ε is the privacy budget, which represents the level of privacy protection. The smaller the value of ε, the higher the degree of privacy protection. Usually, ε is less than 1; it is often set to 0.01 or 0.1 or, in some cases, is set to ln2 or ln3. [36].

Differential privacy is achieved by adding noise to the query results, the amount of noise depends on the sensitivity of the query function. The sensitivity is the maximum change in the value of a function that is caused by adding or deleting a single row and is defined as follows:

Definition 2 (Sensitivity): Assume query function $f : D \to R^d$, where D is the input dataset and the output is a d-dimensional real vector. For any datasets D and D' that differ in at most one row, the sensitivity of the function f is defined as follows:

$$\Delta f = \max_{D,D'} \left\| f(D) - f(D') \right\|_1 \tag{2}$$

Definition 3 (Laplace Mechanism): Given dataset D, assume query function $f : D \to R^d$ and its sensitivity Δf.

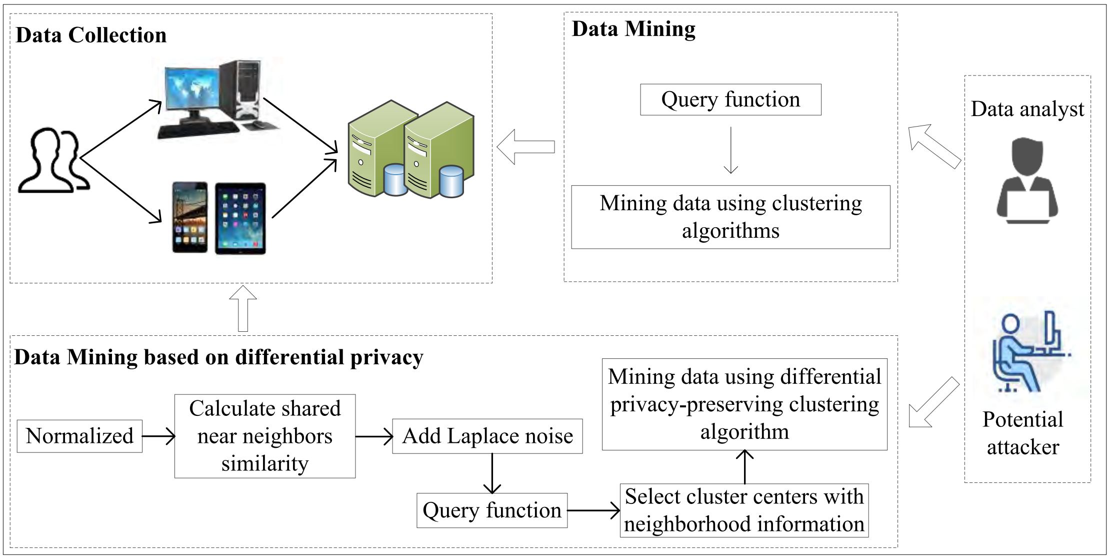

Data Collection
Data Mining
Query function
Mining data using clustering algorithms
Data analyst
Data Mining based on differential privacy
Normalized
Calculate shared near neighbors similarity
Add Laplace noise
Mining data using differential privacy-preserving clustering algorithm
Query function
Select cluster centers with neighborhood information
Potential attacker

FIGURE 1. The system architecture of DP-DPCSNNS schema.

TABLE 1. Notations and descriptions.

Notations	Descriptions
$D=[D_1,D_2,\ldots,D_N]^{\mathrm{T}}$, $D_i=[D_{i1},D_{i2},\ldots,D_{iM}]$	The dataset D with D_i as its i-th sample, where N is the total number of samples and M is the total number of attributes
$X=[x_1,x_2,\ldots,x_N]^{\mathrm{T}}$, $x_i=[x_{i1},x_{i2},\ldots,x_{iM}]$	The dataset X with x_i as its i-th sample, where N is the total number of samples and M is the total number of attributes
$\rho=(\rho_1,\rho_2,\ldots,\rho_N)$	The local density of N samples in X
$\delta=(\delta_1,\delta_2,\ldots,\delta_N)$	The shortest distance of N samples in X
$\gamma=(\gamma_1,\gamma_2,\ldots,\gamma_N)$	The elementwise product of ρ and δ of N samples in X
$C=\{C_1,C_2,\ldots,C_L\}$	The L clusters of X
$U=(u_1,u_2,\ldots,u_N)$	The cluster labels of X
$V=(v_1,v_2,\ldots,v_N)$	The cluster labels of X that are generated by the algorithm

Suppose the random algorithm $R(D) = F(D) + Y$ provides $\mathcal{E}$ differential privacy protection, where $Y \sim Lap(b)$ is random noise with scale parameter $b = \Delta f/\varepsilon$. The probability density function of the Laplace mechanism is:

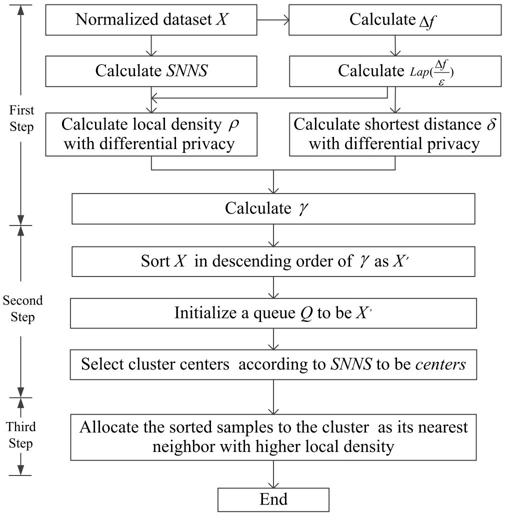
$$P(x) = \frac{1}{2\Delta f/\varepsilon}\exp\left(-\frac{|x|}{\Delta f/\varepsilon}\right) \tag{3}$$

In the era of big data, malicious attackers can use a variety of methods to obtain background knowledge of data privacy. When any record of the data set is changed, an attacker with the highest levels of background knowledge cannot infer relevant details about the stored information in DP-DPSNNS. The system architecture of DP-DPCSNNS is shown in Fig. 1.

B. CORRECTNESS OF THE ALGORITHM

The notations and descriptions of the matrices and vectors that are used in the DP-DPSNNS algorithm are listed in Table 1. The algorithm is based on the assumption that the ideal cluster centers are surrounded by neighbors of lower local density and are relatively far away from another sample with higher local density. The algorithm that is proposed in this paper consists of three main steps: First, Laplace noise is added when measuring the local density ρ and the shortest distance δ of the samples. The noise satisfies the Laplace mechanism according to the definition of differential privacy. Second, the samples that have relatively large ρ and large δ are selected as cluster centers. Third, each remaining sample is assigned to the cluster of its nearest neighbor with higher local density. Fig. 2 demonstrates the main processes of DP-DPCSNNS.

Shared near neighbors similarity is a nonparametric clustering method that is based on k-nearest neighbors, which is suitable for dealing with the practical problems of big data. Typically, a sample and its neighbors are more likely to be assigned to the same cluster. Therefore, the distribution of data can be more correctly reflected by the shared near neighbors similarity between samples.



Normalized dataset X
Calculate Δf
Calculate $SNNS$
Calculate $Lap(\frac{\Delta f}{\varepsilon})$
First Step
Calculate local density ρ with differential privacy
Calculate shortest distance δ with differential privacy
Calculate γ
Sort X in descending order of γ as X'
Second Step
Initialize a queue Q to be X'
Select cluster centers according to $SNNS$ to be *centers*
Third Step
Allocate the sorted samples to the cluster as its nearest neighbor with higher local density
End

FIGURE 2. **The main processes of DP-DPCSNNS.**

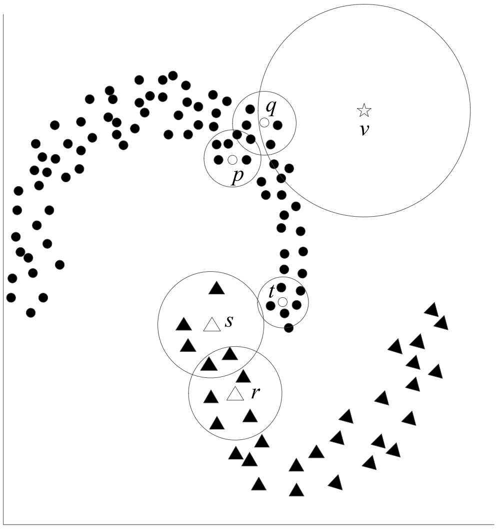


FIGURE 3. **Diagram for the shared near neighbors similarity.**

Definition 4 (Shared Near Neighbors Similarity): Suppose x_i and x_j are random samples in dataset X, $KNN\ (x_i)$ is the set of k-nearest neighbors of sample x_j, and $KNN\ (x_j)$ is the set of k-nearest neighbors of sample x_j. The shared near neighbors similarity of x_i and x_j is calculated as follows:

$$SNNS\left(x_i, x_j\right) = |KNN(x_i) \cap KNN(x_j)| \tag{4}$$

According to the DPC algorithm, the cluster centers and cluster sizes are determined by the local density ρ and the shortest distance δ. The leakage of the local density and the shortest distance information may cause the threat of privacy disclosure. Differential privacy not only protects the usability of clustering results, but also makes it impossible for an attacker to infer the certain privacy information by his background knowledge. This protects the privacy of each individual in the data set. Therefore, differential privacy is used to prevent the leakage of privacy when calculating ρ and δ. To prevent the leakage of privacy and to avoid reducing the availability of the data, the values of ρ and δ are disturbed. The definitions are as follows.

Definition 5 (Local Density With Differential Privacy): The local density of sample x_i is defined as follows:

$$\rho_i = \sum_{j \in KNN(x_i)} \exp\left(-\frac{d(x_i, x_j)}{SNNS(x_i, x_j) + 1}\right) + Lap\left(\frac{\Delta f}{\varepsilon}\right) \tag{5}$$

If $\sum_{j \in KNN(x_i)} SNNS\left(x_i, x_j\right)$ is constant and the distances between x_i and the k-nearest neighbors of x_i are small, that is, if $\sum_{j \in KNN(x_i)} d\left(x_i, x_j\right)$ is small, the local density of the sample x_i is large; if $\sum_{j \in RNN(x_i)} d\left(x_i, x_j\right)$ is constant and the shared near neighbors similarity of x_i and the k-nearest neighbors of x_i are large, that is, if $\sum_{j \in KNN(x_i)} SNNS\left(x_i, x_j\right)$ is large, the local density of sample x_i is large.

To further illustrate the relationship between the shared near neighbors similarity and the Euclidean distance, we present the following example: Fig. 3 shows a distribution of two-dimensional data in which p, q, r, s, t, v are samples of the data and the number of nearest neighbors is $k = 6$. As shown in this figure, there are two clusters: p, q and t are in the upper cluster (dots); s and r are in the cluster below (triangle); and v is an outlier sample. The samples satisfy $SNNS(p, q) = SNNS(q, v) = 2$; however, because $dist(q, v) > dist(p, q)$, the probability that p and q are assigned to the same cluster is large and the probability that q and v are assigned to the same cluster is small. Thus, sample v, which has smaller local density ρ_v and larger shortest distance δ_v, is an outlier sample. In addition, $d(s, t) = d(s, r)$; hence, if only the Euclidean distance is used to measure the similarity of the samples, s and r, which do not belong to the same cluster, will likely be assigned to the same cluster. However, since $SNNS(s, r) = 2$ and $SNNS(s, t) = 0$, it is more likely that s and r will be assigned to the same cluster. Thus, the combination of the shared near neighbors similarity and the Euclidean distance can more accurately measure the similarity of samples so that the information about the cluster structure can be obtained from the local density.

Definition 6 (Shortest Distance With Differential Privacy): The shortest distance of the sample x_i is defined as follows:

$$\delta_i = \begin{cases} \min\limits_j(d(x_i, x_j)) + Lap(\frac{\Delta f}{\varepsilon}), & \rho_j > \rho_i \\ \max\limits_j(d(x_i, x_j)) + Lap(\frac{\Delta f}{\varepsilon}), & \text{otherwise} \end{cases} \tag{6}$$

δ_i is the shortest distance between x_i and other higher local density samples. If x_i is already the highest density sample, δ_i is equal to the maximum distance to the other samples.

The DPC algorithm often uses the decision graph to select cluster centers. This process requires artificial intervention. For a big dataset or a dataset with many cluster centers, selecting the cluster centers by using the decision graph is complicated. If the DPC algorithm uses (7) to select the cluster centers, the neighborhood distribution of the sample is not considered; thus, the error rate may be high.

$$\gamma_i = \rho_i \times \delta_i \tag{7}$$

The erroneous selection of cluster centers will directly lead to the erroneous allocation of the sample; therefore, a new cluster center selection strategy is presented in this paper. In the process of cluster center selection, the cluster centers set *centers* and queue Q must be initialized and the value of γ is calculated according to (7). The samples that are not allocated and have the largest γ values are added to the queue Q in turn until all samples have been visited. The head h of Q is removed and added to *centers*. This is repeated until the condition $length(centers) \leq L$ is satisfied. If sample h satisfies $SNNS(h, p) = 0(\forall p \in centers)$, then h is added to *centers*. Additional details on cluster center selection are presented in Algorithm 2.

Algorithm 1 implements the primary steps of DP-DPCSNNS. Line 4 of algorithm 1 calculates the local density and shortest distance with differential privacy according to (5) and (6). Line 7 of algorithm 1 corresponds to the process of remaining sample allocation. Set *centers* and the corresponding cluster labels are obtained from algorithm 2. Then, the samples are sorted in descending order of local density ρ. The sorted samples are sequentially removed and if a sample is not in *centers*, it is not allocated to any cluster. In this case, the sample is allocated to the same cluster as its nearest neighbor of higher local density.

Algorithm 1 DP-DPCSNNS

```
Input: $D$(a dataset of $N$ samples in $M$ dimensions), $L$(the
        number of clusters), $\varepsilon$(privacy budget parameter),
        $\Delta f$(sensitivity), $k$(the number of nearest neighbors)
Output: $C = \{C_1, C_2, \ldots, C_L\}$
1:  normalize $D$;
2:  use $\Delta f$ and $\varepsilon$ to calculate $Lap(\frac{\Delta f}{\varepsilon})$ by (3);
3:  calculate $SNNS$ by (4);
4:  calculate $\rho$ by (5);
5:  calculate $\delta$ by (6);
6:  use $\rho$ and $\delta$ to calculate $\gamma$ by (7);
7:  $centers = \mathrm{SCC}(D, \rho, \delta, \gamma, L, SNNS)$;
8:  allocate the sorted sample to the cluster
    $C_j \in C(j = 1, 2, \ldots, L)$ as its nearest neighbor with
    higher local density;
9: return $C$.
```

C. TIME COMPLEXITY ANALYSIS

The time complexity analysis of DP-DPSNNS is as follows, where N is the size of dataset. Our method consists of three steps:

Step 1, the calculation of the local density ρ and the shortest distance δ. To calculate ρ, $O\left(N^2\right)$ is required for calculating the distance, $O\left(kN^2\right)$ for calculating the shared near neighbors similarity $O(N)$ for calculating the noise, and $O(kN)$ for querying the k-nearest neighbor information to calculate ρ. The cost of δ is the same as in traditional DPC: $O\left(N^2\right)$. Therefore, the total time complexity of this step is $O\left(N^2\right) + O\left(kN^2\right) + O(N) + O(kN) \sim O\left(N^2\right)$.

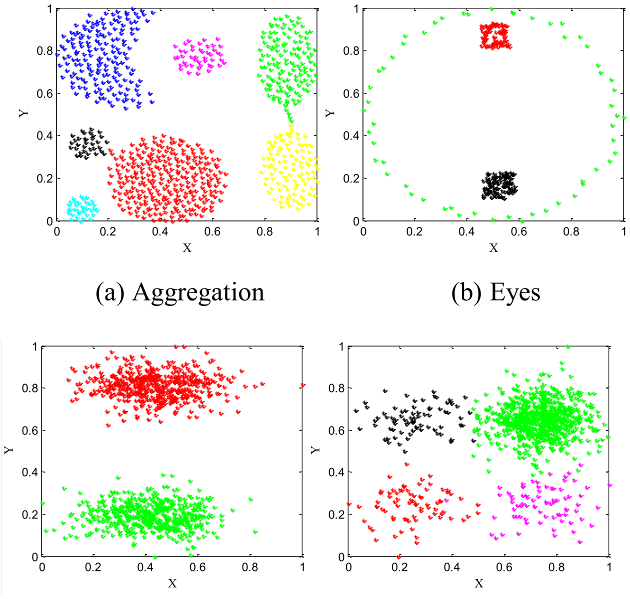
Step 2, the selection of the cluster centers. We require $O(N \log N)$ to sort the samples, $O(N)$ to calculate γ, and $O(N)$ to loop to identify the cluster centers in the worst case. Therefore, the total time complexity of this step is $O(N \log N) + O(N) + O(N) \sim O(N \log N)$.

Step 3, the allocation of the remaining samples. The cost of this step is the same as in the traditional DPC algorithm: $O(N)$.

The overall time complexity of DP-DPCSNNS is $O\left(N^2\right)$. It is the same as that of the traditional DPC algorithm.

Algorithm 2 SCC

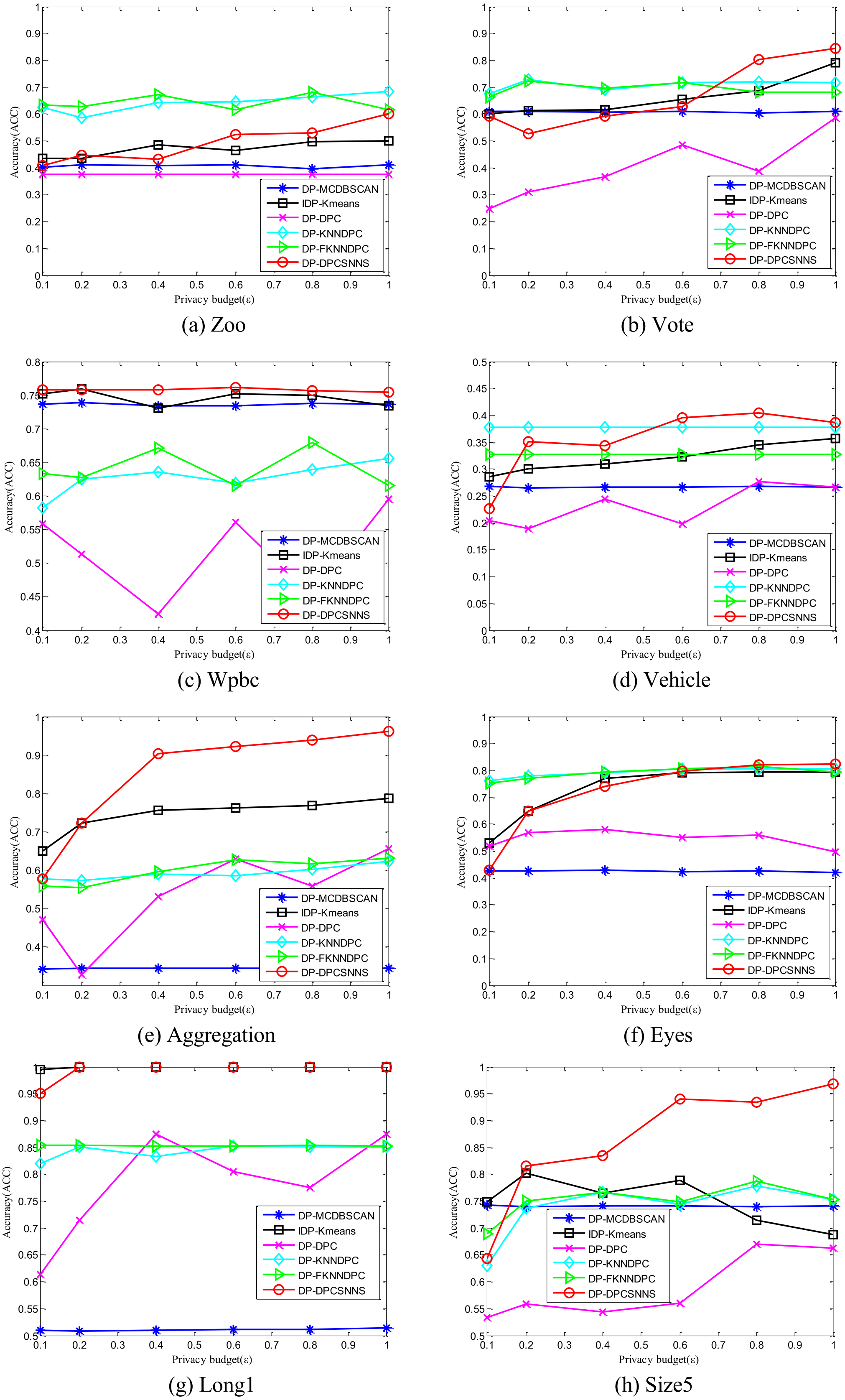
```
Input: $X$(a dataset of $N$samples in $M$ dimensions), $\rho$(local
        density with differential privacy), $\delta$(shortest distance
        with differential privacy), $\gamma$(the elementwise product of
        $\rho$ and $\delta$), $L$(the number of clusters), $SNNS$ (the
        shared near neighbors similarity between samples)
Output:$centers$
1: sort $X$ in descending order of $\gamma$ as
   $X' = \{x'_1, x'_2, \ldots, x'_N\}$ and initialize a queue $Q$ to be $X'$;
2: initialize the set $centers$;
3: remove the head $h$ of $Q$ and add $h$ to $centers$;
4: select the head $h$ of $Q$; if $h$ does not belong to any cluster
    and satisfies $SNNS(h, p) = 0(\forall p \in centers)$ then add $h$ to
   $centers$;
5: remove the head $h$ of $Q$;
6: if $length(centers) \leq L$, then go to 4; else, go to 7;
7: return $centers$.
```



(a) Aggregation (b) Eyes

(c) Long1 (d) Size5

FIGURE 4. Four synthetic datasets.



(a) Zoo

(b) Vote

(c) Wpbc

(d) Vehicle

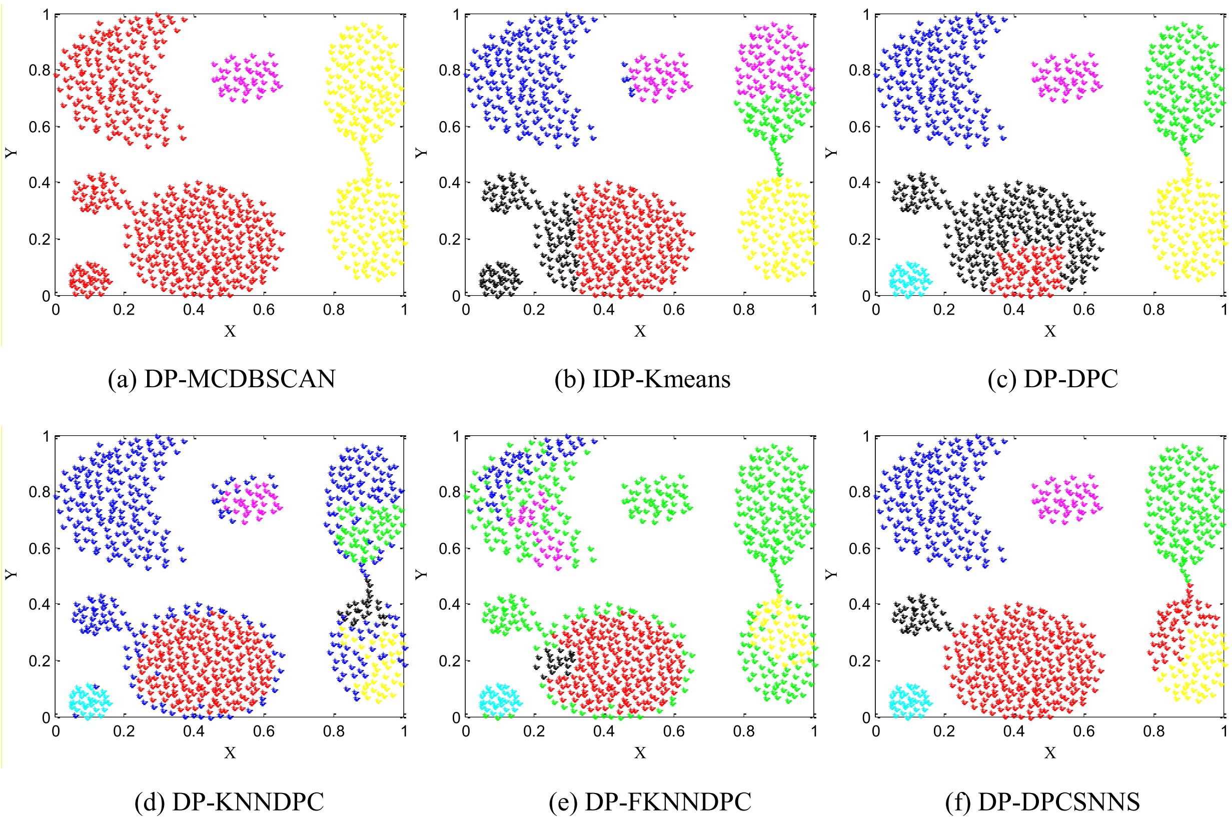
(e) Aggregation

(f) Eyes

(g) Long1

(h) Size5

FIGURE 5. ACC comparison among six clustering algorithms on eight datasets.



(a) DP-MCDBSCAN (b) IDP-Kmeans (c) DP-DPC

(d) DP-KNNDPC (e) DP-FKNNDPC (f) DP-DPCSNNS

FIGURE 6. The clustering results of aggregation by six clustering algorithms.

D. PRIVACY ANALYSIS

Assume S and S' are the query results of the algorithm on D and D' respectively; *Part* represents the clustering result; C is the clustering result without noise; $R\,(D)$ is the clustering result of D with noise; and $R\left(D'\right)$ is the clustering result of D without noise. According to the above equations, the following is obtained:

$$\begin{aligned}\frac{\Pr[S\in Part]}{\Pr[S'\in Part]} &= \frac{\exp(-\frac{\varepsilon|C-R(D)|}{\Delta f})}{\exp(-\frac{\varepsilon|C-R(D')|}{\Delta f})}\\ &\le \exp(\frac{\varepsilon|R(D)-R(D')|}{\Delta f})\\ &\le \exp(\frac{\varepsilon\parallel R(D)-R(D')\parallel_1}{\Delta f})\\ &\le \exp(\varepsilon)\end{aligned}$$

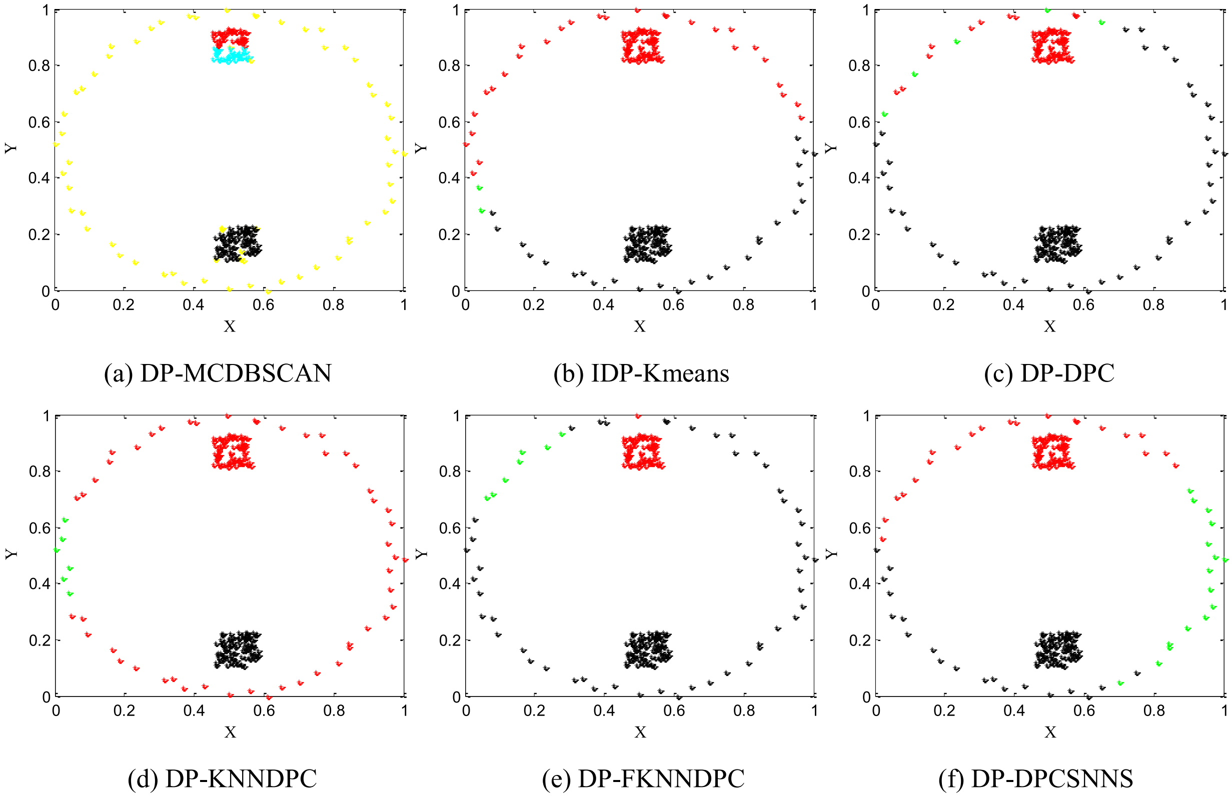
IV. EXPERIMENT

This paper uses real-world datasets and synthetic datasets to evaluate the performance of the DP-DPCSNNS algorithm. These datasets, which are listed in Table 2, are classic datasets in clustering analysis. The four real-world datasets come from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). The four synthetic datasets are Aggregation [37], Eyes [38], Long1 [39] and Size5 [39], which are shown in Fig. 4. The experiment for DP-DPCSNNS is programmed in MATLAB 2013b. The hardware configuration is a computer with the Windows 7 operating system, 4 G physical memory and a 1.9 GHz CPU.

TABLE 2. Datasets.

Dataset	No records N	No attributes M	No clusters L	Type
vote	435	16	2	UCI
zoo	101	16	7	UCI
wpbc	198	33	2	UCI
vehicle	846	18	4	UCI
aggregation	788	2	7	Synthetic
eyes	238	2	3	Synthetic
long1	1000	2	2	Synthetic
size5	1000	2	4	Synthetic

The performance of DP-DPCSNNS is compared with those of five new clustering algorithms that are based on differential privacy including DP-MCDBSCAN [35], IDP-Kmeans [33], DP-DPC (differential preserving DPC), DP-KNNDPC (differential preserving KNNDPC) and DP-FKNNDPC (differential preserving FKNNDPC). Laplace noise is added during the calculation of the distance between samples in DP-MCDBSCAN. Laplace noise is added during the calculation of the cluster centers in IDP-Kmeans.



(a) DP-MCDBSCAN (b) IDP-Kmeans (c) DP-DPC

(d) DP-KNNDPC (e) DP-FKNNDPC (f) DP-DPCSNNS

FIGURE 7. The clustering results of Eyes by six clustering algorithms.

DP-DPC is generated by adding noise to the calculation of the local density ρ and the shortest distance δ in the DPC algorithm. Differential privacy preservation is achieved by adding noise, which affects the clustering results. Therefore, DP-KNNDPC and DP-FKNNDPC are rewritten under the framework of differential privacy preservation by KNN-DPC and FKNN-DPC, respectively.

The six clustering algorithms that are being compared require various parameters to be set: For the MP-MCDBSCAN algorithm, the author of the algorithm provides $minpts = 1/25^{\times N}$ and $\Delta f = M$. To find the optimal value of *eps*, the range of *eps* is obtained from the *dist* graph [21]. Then, the experiment is randomly run 50 times in the range of *eps* and the optimal *eps* value is selected. For the IDP-Kmeans algorithm, the author of the algorithm provides $\Delta f = M + 1$ and the maximum number of iterations is set to 1000. For the DP-DPC algorithm, as a rule of thumb, the author of the DPC algorithm provides percent $= 0.2$ so that the average number of neighbors is between 1% and 2% of the total and Δf is calculated according to (2). For the DP-KNNDPC, DP-FKNNDPC and DP-DPCSNNS algorithms, $k = 6$ and Δf is calculated according to (2). For all the above experiments, to ensure the same level of privacy, we set $\varepsilon = \ln 2$ [36]. Since the addition of noise in the differential privacy framework is random, the above experiments are repeated 20 times and the final results are average values. In this paper, we use (8) to preprocess the dataset so that the effect of eliminating missing values and differences in dimensional ranges is eliminated.

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{8}$$

where x'_{ij} is the normalized result of x_{ij}, which is the element of i-th row and j-th column in dataset X; $\max(x_j)$ is the maximum value in the j-th column attribute in dataset X; and $\min(x_j)$ is the minimum value in the j-th column attribute in dataset X.

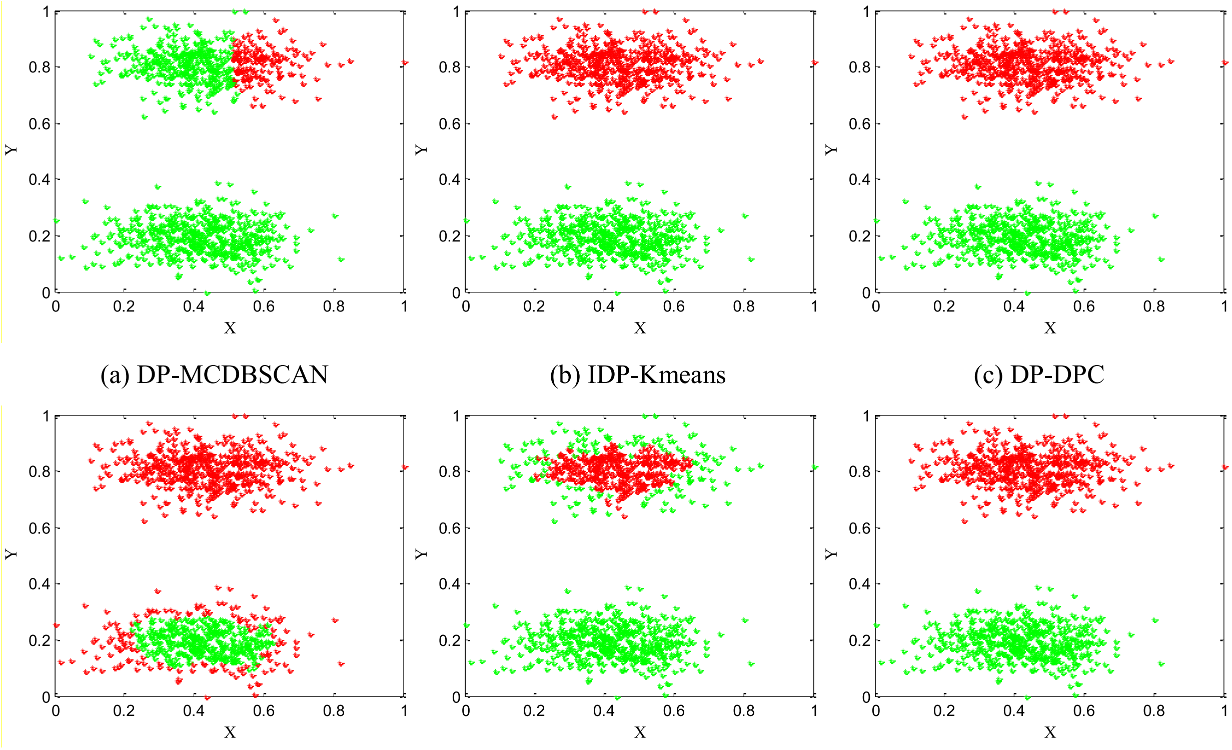
A. EVALUATION METRICS

Accuracy(ACC) [40], adjusted mutual information(AMI) [41], F-Measure [42], adjusted Rand index (ARI) [43] and T-test are used to evaluate the performance of the above clustering algorithms. These five metrics are classical metrics for evaluating the performance of a clustering algorithm. The ranges of ACC and F-Measure are [0,1]. The ranges of AMI and ARI are [−1,1]. The bold values correspond to the better clustering results.

1) ACC METRIC

ACC calculates the proportion of correctly classified samples relative to the total sample. The formula is as follows:

$$ACC = \frac{1}{N}\sum_{i=1}^{N}\omega(u_i, v_i),\ \omega(u_i, v_i) = \begin{cases} 1, u_i = v_i \\ 0, u_i \neq v_i \end{cases} \tag{9}$$



(a) DP-MCDBSCAN (b) IDP-Kmeans (c) DP-DPC

(d) DP-KNNDPC (e) DP-FKNNDPC (f) DP-DPCSNNS

FIGURE 8. The clustering results of Long1 by six clustering algorithms.

where u_i is the standard label of sample x_i and v_i is the class label of sample x_i that is obtained by the clustering algorithm.

2) AMI METRIC

AMI measures the relevance of two event sets in terms of mutual information (MI). $I(U, V)$ is the mutual information between event U and event V. $E\{I(U, V)\}$ is the expected mutual information between U and V. The calculation formula is as follows:

$$\mathrm{AMI}(U, V) = \frac{MI(U, V) - E\{MI(U, V)\}}{\max\{H(U), H(V)\} - E\{MI(U, V)\}} \tag{10}$$

3) F-MEASURE METRIC

F-Measure combines the metrics, namely, precision and recall, and its advantage lies in its overall performance in distinguishing clustering results. The information between two clustering results, namely, $U = \{U_1, U_2, \ldots, U_R\}$ and $V = \{V_1, V_2, \ldots, V_C\}$, can be summarized in the form of a contingency table; the contingency table is shown as Table 3.

In Table 3, n_{ij} represents the number of samples that are in both cluster U_i and cluster V_j, N_{11} represents the number of samples that are in the same cluster in both U and V, N_{00} represents the number of samples that are in different clusters in U and V, N_{01} represents the number of samples that are in the same cluster in U but in different clusters in V, and N_{01} represents the number of samples that are in different clusters in U but in the same cluster in V.

TABLE 3. Contingency table for comparing partitions *U* and *V*.

U\V	V_1	V_2	$\cdots$	V_C	Sum
U_1	n_{11}	n_{12}	$\cdots$	n_{1C}	a_1
U_2	n_{21}	n_{22}	$\cdots$	n_{2C}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_R	n_{R1}	n_{R2}	$\cdots$	n_{RC}	a_R
Sum	b_1	b_2	$\cdots$	b_C	$\sum_{ij} n_{ij} = N$

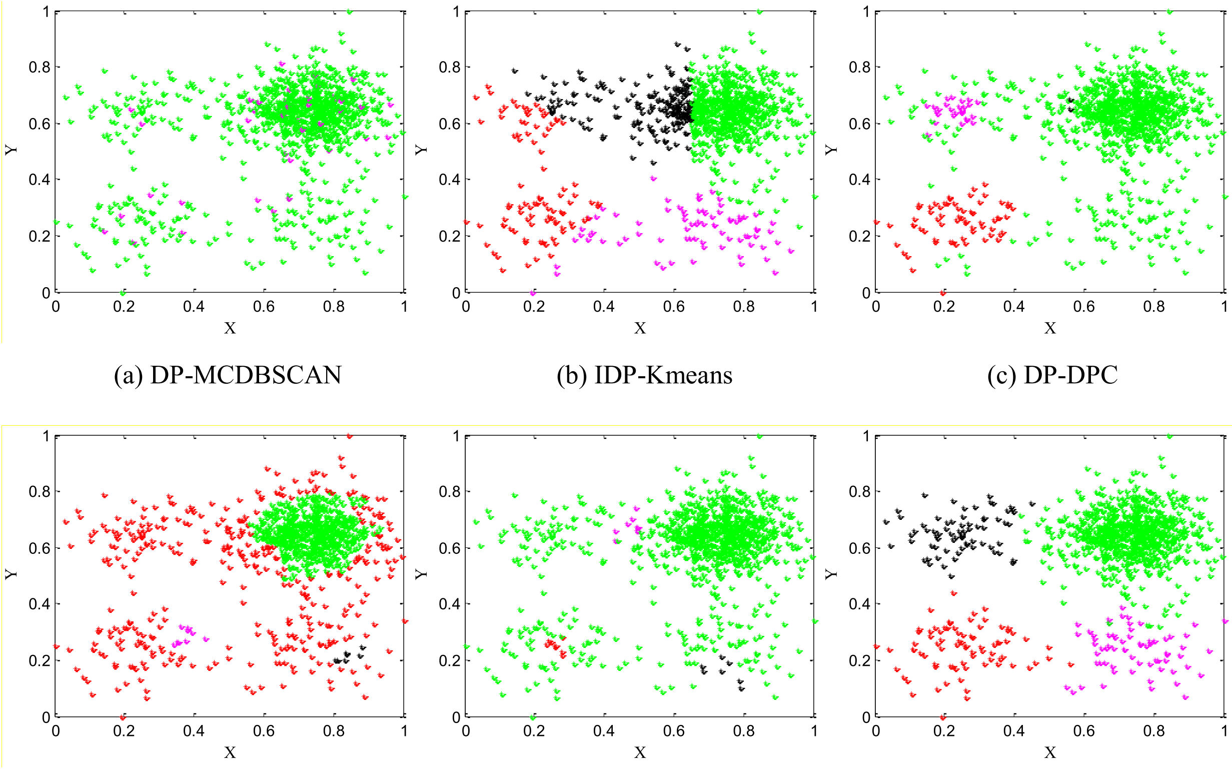
Precision and recall are expressed in (11) and (12), respectively. Recall and precision are a pair of contradictory measures: When the precision is high, the recall tends to be low, and when the recall is high, the precision is often low. F-Measure is expressed in (13).

$$precision = \frac{N_{11}}{N_{11} + N_{10}} \tag{11}$$

$$recall = \frac{N_{11}}{N_{11} + N_{01}} \tag{12}$$

In this experiment, F-Measure is used if $\beta = 1$:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{13}$$



(a) DP-MCDBSCAN (b) IDP-Kmeans (c) DP-DPC

(d) DP-KNNDPC (e) DP-FKNNDPC (f) DP-DPCSNNS

FIGURE 9. The clustering results of Size5 by six clustering algorithms.

4) ARI METRIC

The Rand Index (RI) is expressed as follows:

$$\mathrm{RI} = \frac{N_{11} + N_{00}}{N_{11} + N_{00} + N_{10} + N_{01}} \tag{14}$$

ARI measure the degree of similarity between class label U and class label V of the algorithm. $E\{RI(U, V)\}$ is the expected Rand index between U and V. ARI is expressed as follows:

$$\mathrm{ARI} = \frac{\mathrm{RI}(U, V) - E\{RI(U, V)\}}{\max\{RI(U, V)\} - E\{RI(U, V)\}} \tag{15}$$

5) STATISTICAL TEST

The T-test is a type of significance test that is used to detect whether the difference between two distributions of data is significant. First, we assume that there is no significant difference between the two distributions of data, which is called the null hypothesis. If the probability$p<0.05$, the null hypothesis is rejected, namely, the two distributions of data differ significantly. In contrast, if the probability $p>0.05$, the null hypothesis is accepted, namely, the two distributions of data do not differ significantly. In the experiment, the T-test is used to detect the difference between the labels that are obtained by DP-DPCSNNS and those that are obtained using the other comparison algorithms.

B. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, the DP-DPCSNNS, DP-DPC, DP-KNNDPC, DP-FKNNDPC, and IDP-Kmeans algorithms can automatically identify clusters under specified parameter values. The experiment used four UCI datasets and four synthetic datasets to evaluate the performances of the algorithms in identifying clusters. Table 4 lists the clustering results in terms of the ACC, AMI, F-Measure, and ARI metrics of the four above algorithms on UCI datasets and synthetic datasets. The values in bold are superior experimental results.

Zoo is a commonly used dataset in clustering analysis. It contains 101 samples and 7 clusters, which are linear and inseparable. Zoo is an animal dataset consisting of 16 attributes, 15 of which are Boolean attribute values in (0, 1) and another one attribute is the number of animal legs in (0, 2, 4, 6, 8). Since 15 attributes in Zoo are Boolean attributes, more samples in the dataset have the same attribute value. Therefore, the Euclidean distance between the samples is zero, which makes d_c is zero in the DP-DPC algorithm, resulting in poor clustering performance. DP-MCDBSCAN and IDP-Kmeans treat Euclidean distance as the only measure of similarity, so clustering performance is not great. However, DP-DPCSNNS algorithm avoids the setting of the parameter d_c, and combines the Euclidean distance with the shared near neighbors similarity to calculate the local density. Therefore, compared with the clustering results of other algorithms, DP-DPCSNNS has great

TABLE 4. Comparison of ACC, AMI, F-Measure and ARI metrics for 6 algorithms on datasets.

Algorithm	Zoo					Vote				
	ACC	AMI	F-Measure	ARI	P	ACC	AMI	F-Measure	ARI	P
DP-MCDBSCAN	0.413	0.011	0.375	0.028	$<10^{-5}$	0.611	0.001	0.667	0.008	$<10^{-5}$
IDP-Kmeans	0.517	0.228	0.492	0.215	0.020	0.663	0.086	0.674	0.097	0.037
DP-DPC	0.376	0.223	0.352	0.076	0.047	0.746	0.248	0.737	0.284	0.241
DP-KNNDPC	0.611	0.577	0.659	0.172	0.243	0.699	0.216	0.659	0.172	0.010
DP-FKNNDPC	0.646	0.602	0.610	0.497	0.242	0.740	0.258	0.686	0.254	$<10^{-5}$
DP-DPCSNNS	**0.663**	**0.711**	**0.681**	**0.593**	-	**0.844**	**0.426**	**0.742**	**0.471**	-
Algorithm	Wpbc					Vehicle				
	ACC	AMI	F-Measure	ARI	P	ACC	AMI	F-Measure	ARI	P
DP-MCDBSCAN	0.737	-0.001	0.745	0.010	0.087	0.268	0.001	**0.388**	0.000	0.050
IDP-Kmeans	0.722	-0.001	0.694	-0.074	0.098	0.332	0.063	0.388	0.048	$<10^{-5}$
DP-DPC	0.751	0.002	0.765	-0.008	0.551	0.358	0.100	0.382	0.064	$<10^{-5}$
DP-KNNDPC	0.613	**0.012**	0.623	0.020	0.092	0.378	0.113	0.358	0.057	$<10^{-5}$
DP-FKNNDPC	0.608	0.012	0.615	**0.022**	0.152	0.327	0.078	0.367	0.028	$<10^{-5}$
DP-DPCSNNS	**0.756**	0.000	**0.770**	-0.003	-	**0.401**	**0.145**	0.377	**0.113**	-
Algorithm	Aggregation					Eyes				
	ACC	AMI	F-Measure	ARI	P	ACC	AMI	F-Measure	ARI	P
DP-MCDBSCAN	0.350	0.003	0.352	0.005	$<10^{-5}$	0.421	0.005	0.492	0.004	$<10^{-5}$
IDP-Kmeans	0.780	0.766	0.737	0.674	0.036	0.778	0.518	0.756	0.594	$<10^{-5}$
DP-DPC	0.868	0.905	0.872	0.837	0.380	0.787	0.542	0.765	0.604	$<10^{-5}$
DP-KNNDPC	0.602	0.475	0.482	0.287	0.003	0.796	**0.630**	**0.793**	**0.652**	$<10^{-5}$
DP-FKNNDPC	0.596	0.502	0.489	0.307	$<10^{-5}$	0.794	0.622	0.791	0.649	$<10^{-5}$
DP-DPCSNNS	**0.896**	**0.919**	**0.911**	**0.887**	-	**0.820**	0.567	0.781	0.639	-
Algorithm	Long1					Size5				
	ACC	AMI	F-Measure	ARI	P	ACC	AMI	F-Measure	ARI	P
DP-MCDBSCAN	0.526	0.009	0.641	0.003	$<10^{-5}$	0.743	0.001	0.734	0.011	$<10^{-5}$
IDP-Kmeans	1.000	1.000	1.000	1.000	1.000	0.747	0.572	0.772	0.581	0.002
DP-DPC	1.000	1.000	1.000	1.000	1.000	0.866	0.575	0.859	0.598	0.451
DP-KNNDPC	0.853	0.499	0.759	0.497	$<10^{-5}$	0.768	0.122	0.760	0.158	$<10^{-5}$
DP-FKNNDPC	0.852	0.497	0.758	0.495	$<10^{-5}$	0.747	0.117	0.743	0.132	$<10^{-5}$
DP-DPCSNNS	**1.000**	**1.000**	**1.000**	**1.000**	-	**0.940**	**0.711**	**0.932**	**0.800**	-

advantages. For the dataset Zoo, the ACC metric of DP-DPCSNNS is about 76.3% higher than DP-DPC algorithm, and about 60.5% higher than DP-MCDBSCAN. The ARI metric of DP-DPCSNNS is 87.1% higher than that of DP-DPC. The AMI metric of DP-DPCSNNS is 67.9% higher than that of IDP-Kmeans. The clustering results of DP-DPCSNNS and three comparison algorithms have significant difference according to T-test.

The Vote dataset contains 435 samples and two clusters and each sample is represented by 16 attributes. The three comparison algorithms assign samples that belong to the second cluster to the first cluster. The ACC, AMI, F-Measure and ARI values of DP-DPCSNNS are optimal and the clustering results are closest to the standard classification results. For the dataset Vote, the AMI of DP-DPCSNNS is improved by 71.8% compared with the DP-DPC algorithm. The F-Measure of DP-DPCSNNS is 11.2% higher than that of DP-MCDBSCAN. The ARI of DP-DPCSNNS is 93.3% higher than that of DP-FKNNDPC. DP-DPCSNNS and the four comparison algorithms differ significantly.

Aggregation is a commonly used synthetic dataset for clustering, in which 788 samples are divided into seven clusters. Due to the addition of noise, the overall clustering performance of the algorithm is slightly reduced. One of seven clusters has 272 samples, the density of which is high. Therefore, IDP-Kmeans, DP-DPC, DP-KNNDPC and DP-FKNNDPC select multiple density peaks in the same cluster. For the Aggregation dataset, the F-Measure value of DP-DPCSNNS is 89% higher than that of the DP-KNNDPC algorithm. The AMI value of DP-DPCSNNS is 83% higher than that of DP-KNNDPC. The clustering results of DP-DPCSNNS and the four comparison algorithms differ significantly according to the T-test.

The Size5 dataset contains a total of 1000 samples in four clusters. The first cluster contains 769 samples; the density of this cluster is high and the distribution of samples is dense.

However, the other three clusters each have 77 samples, the densities of these clusters are low and the distributions of samples are sparse. DP-DPCSNNS can correctly identify four cluster centers and distribute the remaining samples. However, the DP-DPC algorithm allocates samples that belong to the sparse second and third clusters to the dense first cluster, which is due to the incorrect estimation of the similarity between samples. Since the distribution of the first cluster is dense, the IDP-Kmeans algorithm identifies two cluster centers in the first cluster and the samples of the first cluster are incorrectly allocated to two clusters. For the Size5 dataset, the ACC value of DP-DPCSNNS is 25.8% higher than that of the IDP-Kmeans algorithm. The ARI value of DP-DPCSNNS is 33.8% higher than that of DP-DPC. The clustering results of DP-DPCSNNS and the four comparison algorithms differ significantly according to the T-test.

From the p-values that are listed in Table 4, the clustering results of DP-DPCSNNS and the comparison algorithms differ significantly on most datasets. For the Wpbc dataset, DP-DPCSNNS and the comparison algorithms do not differ significantly; this is because the total numbers of clusters and records of dataset Wpbc are small. For the Long1 dataset, the p-values of DP-DPCSNNS between DP-DPC and IDP-Kmeans are 1; this is because these three algorithms all obtain the optimal results.

From the above detailed analysis of the performance comparison of DP-DPCSNNS with other clustering algorithms on UCI and synthetic datasets, the experimental results show that DP-DPCSNNS performs well in clustering. Comparing DP-DPCSNNS with DP-DPC, the clustering results are optimized though the local density calculation method and the cluster center selection mechanism that are utilized in DP-DPCSNNS. On the UCI and synthetic datasets, DP-DPCSNNS has substantial advantages over the other algorithms under the framework of the ε-differential-privacy-preserving model.

Fig. 5 shows a comparison of the ACC values from the six algorithms running on Zoo, Vote, Wpbc, Vehicle, Aggregation, Eyes, Long1 and Size5. It can be seen from the figure, the general trend is an increase of ACC as the privacy budget ε increases. However, the noise is random. There may be fluctuations in the ACC of some algorithms. The DP-DPCSNNS algorithm has substantial advantages on Wpbc, Vehicle and the four synthetic datasets. On the Zoo and Vote datasets, the ACC of DP-DPCSNNS is low if ε is low; however, as ε increases, the advantage of the DP-DPCSNNS algorithm increases. Figs. 6-9 show the performances of the six clustering algorithms on the synthetic datasets.

V. CONCLUSION

Differential privacy-preserving density peaks clustering based on shared near neighbors similarity, which is denoted as DP-DPCSNNS, was proposed in this paper. The DP-DPCSNNS algorithm provides ε-differential privacy preservation and improves the clustering performance. The shared near neighbors similarity and the Euclidean distance are combined to calculate the local density of samples, which avoids the setting of the parameter d_c. Laplace noise is added in the calculations of the local density and the shortest distance under the framework of ε-differential privacy, which overcomes the problem of privacy leakage that was encountered with the original DPC algorithm. From the detailed analysis of the performance of DP-DPCSNNS compared to other clustering algorithms on UCI and synthetic datasets, the experimental results show that DP-DPCSNNS can accurately select the cluster centers and ensure better clustering performance under the framework of the differential privacy-preserving model. In future work, the next step will be to use privacy budget allocation strategies to improve the clustering performance.

REFERENCES

[1] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec. 1996.

[2] A. K. Jain and R. C. Dubes, "Algorithms for clustering data," *Technometrics*, vol. 32, no. 2, pp. 227–229, 1988.

[3] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[4] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[5] A. Machanavajjhala, M. Venkitasubramaniam, D. Kifer, and J. Gehrke, "L-diversity: Privacy beyond K-anonymity," in *Proc. IEEE 22nd Int. Conf. Data Eng.*, Atlanta, GA, USA, Apr. 2006, pp. 1–12.

[6] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond K-anonymity and L-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Istanbul, Turkey, Apr. 2007, pp. 106–115.

[7] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloq. Automata, Lang., Program.* Venice, Italy: Springer, 2006, pp. 1–12.

[8] C. Dwork, F. Mcsherry, A. Smith, and K. Nissim, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. Berlin, Germany: Springer, 2006, pp. 637–648.

[9] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[10] J. Ren, Z. Yao, J. Xiong, Y. Zhang, and A. Ye, "A secure data deduplication scheme based on differential privacy," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, Shenzhen, China, Dec. 2017, pp. 1241–1246.

[11] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[12] D. Liu, S.-F. Cheng, and Y. Yang, "Density peaks clustering approach for discovering demand hot spots in city-scale taxi fleet dataset," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Las Palmas, Spain, Sep. 2015, pp. 1–6.

[13] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[14] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Netw.*, vol. 31, no. 2, pp. 155–163, May 2009.

[15] K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 1275–1276.

[16] M. S. Handcock, J. M. Tantrum, and A. E. Raftery, "Model-based clustering for social networks," *J. Roy. Stat. Soc. A, Statist. Soc.*, vol. 170, no. 2, pp. 301–354, 2010.

[17] C. Yang, S. Liu, L. Bruzzone, R. Guan, and P. Du, "A feature-metric-based affinity propagation technique for feature selection in hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1152–1156, Sep. 2013.

[18] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.

[19] J. Yu, R. Hong, M. Wang, and J. You, "Image clustering based on sparse patch alignment framework," *Pattern Recognit.*, vol. 47, no. 11, pp. 3512–3519, 2014.

[20] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.* Berkeley, CA, USA: Univ. California Press, 1967, pp. 281–297.

[21] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Sacramento, CA, USA, AAAI Press, 1996, pp. 226–231.

[22] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1620–1628, Aug. 2017.

[23] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowl.-Based Syst.*, vol. 99, pp. 135–145, May 2016.

[24] J. Xie, H. Gao, and W. Xie, "K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset," *Scientia Sinica, Informationis*, vol. 46, no. 2, pp. 258–280, 2016.

[25] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Inf. Sci.*, vol. 354, pp. 19–40, Aug. 2016.

[26] R. Mehmood, R. Bie, H. Dawood, and H. Ahmad, "Fuzzy clustering by fast search and find of density peaks," in *Proc. IEEE Int. Conf. Identificat., Inf., Knowl. Internet Things*, Beijing, China, Oct. 2016, pp. 785–793.

[27] J. Xu, G. Wang, and W. Deng, "DenPEHC: Density peak based efficient hierarchical clustering," *Inf. Sci.*, vol. 373, no. 12, pp. 200–218, 2016.

[28] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao, and P. Li, "An incremental CFS algorithm for clustering large data in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1193–1201, Jun. 2017.

[29] S. Jia, G. Tang, J. Zhu, and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 88–102, Jan. 2016.

[30] M. Wang, W. Zuo, and Y. Wang, "An improved density peaks-based clustering method for social circle discovery in social networks," *Neurocomputing*, vol. 179, pp. 219–227, Feb. 2016.

[31] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst.*, Baltimore MD, USA, 2005, pp. 128–138.

[32] Q. Yu, Y. Luo, C. Chen, and X. Ding, "Outlier-eliminated K-means clustering algorithm based on differential privacy preservation," *Appl. Intell.*, vol. 45, no. 4, pp. 1179–1191, 2016.

[33] L. Yang, H. Zhifeng, and W. Wen, "Research on differential provacy preserving K-means clustering," *Comput. Sci.*, vol. 40, no. 3, pp. 287–290, 2013.

[34] Z.-Q. Gao and L.-J. Zhang, "DPHKMS: An efficient hybrid clustering preserving differential privacy in spark," in *Proc. Int. Conf. Emerg. Internetworking, Data Web Technol.*, 2017, pp. 367–377.

[35] L. Ni, C. Li, X. Wang, H. Jiang, and J. Yu, "DP-MCDBSCAN: Differential privacy preserving multi-core DBSCAN clustering for network user data," *IEEE Access*, vol. 6, pp. 21053–21063, 2018.

[36] C. Dwork, "Differential privacy in new settings," in *Proc. ACM-SIAM Symp. Discrete Algorithms*. Austin, TX, USA: SIAM, 2010, pp. 174–183.

[37] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, pp. 1–30, 2007.

[38] X. Tao, R. Wang, R. Chang, C. Li, R. Liu, and J. Zou, "Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies," *Knowl.-Based Syst.*, vol. 170, pp. 26–42, Apr. 2019.

[39] A. M. Gong, L. Jiao, L. Bo, W. Ling, and X. Zhang, "Image texture classification using a manifold-distance-based evolutionary clustering method," *Opt. Eng.*, vol. 47, no. 7, pp. 685–694, 2008.

[40] G. Carpaneto and P. Toth, "Algorithm 548: Solution of the assignment problem," *ACM Trans. Math. Softw.*, vol. 6, no. 1, pp. 104–111, 1980.

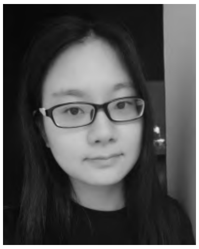
[41] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

[42] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater*, vol. 1, no. 5, pp. 1–5, 2007.

[43] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.

LIPING SUN received the M.S. degree from the School of Computer Science, Chongqing University, in 2008, and the Ph.D. degree from the School of Geography and Tourism, Anhui Normal University, in 2015, where she has been a Professor with the School of Computer and Information, since 2017.

Her research interests include data mining and information security.

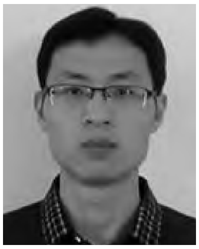


SHUTING BAO received the B.E. degree in computer science and technology from the Wangjiang College, Anhui Normal University, in 2016, where she is currently pursuing the M.S. degree with the School of Computer and Information.

Her research interests include data mining and information security.

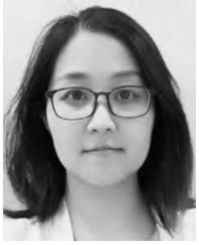
SHANG CI received the B.E. degree from the School of Computer and Information, Anqing Normal University, in 2017. He is currently pursuing the M.S. degree with the School of Computer and Information, Anhui Normal University.

His research interests include cloud computing security and data mining.



XIAOYAO ZHENG received the M.S. degree from the School of Computer Science, Hefei University of Technology, in 2005, and the Ph.D. degree from the School of Geography and Tourism, Anhui Normal University, in 2018, where he has been an Associate Professor with the School of Computer and Information, since 2016.

His research interests include information security and social networks.



LIANGMIN GUO received the Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China, in 2011.

Since 2013, she has been an Associate Professor with the School of Computer and Information, Anhui Normal University. Her research interests include cloud computing, information security, and recommender systems.

YONGLONG LUO received the Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China, in 2005.

Since 2007, he has been a Professor with the School of Computer and Information, Anhui Normal University. He is currently the Director of the Anhui Provincial Key Laboratory of Network and Information Security. His research interests include information security and spatial data processing.