IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 May 2023, accepted 27 May 2023, date of publication 31 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281652

RESEARCH ARTICLE

A Density Peaking Clustering Algorithm for Differential Privacy Preservation

HUA CHEN, KEHUI MEI, YUAN ZHOU, NAN WANG, MENGDI TANG, AND GUANGXING CAI

School of Science, Hubei University of Technology, Wuhan 430068, China

Corresponding author: Kehui Mei (792768156@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502156, in part by the Teaching and Research Project of Hubei Provincial Department of Education under Grant 282, and in part by the Doctoral Startup Fund of Hubei University of Technology under Grant BSQD13051.

ABSTRACT The privacy protection problem in data mining has received increasingly attention and is a hot topic of current research. To address the problems of large accuracy loss and instability of clustering results of clustering algorithms under differential privacy protection requirements, a density peak clustering algorithm for differential privacy protection (DP-chDPC) is proposed. Firstly, the original DPC algorithm is improved, by using the dichotomy method to automatically determine the truncation distance to avoid the subjectivity of manual selection, and by setting the threshold of local density and center offset distance to automatically obtain the clustering center, which overcomes the uncertainty of the original DPC algorithm to select the clustering center based on the decision graph. Then, noise is added to the local density by using the Laplace mechanism to realize the differential privacy protection of the algorithm during the clustering analysis. Finally, the Chebyshev distance is used to replace the Euclidean distance to calculate the distance matrix, which reduces the interference on the clustering results after the algorithm adds noise, and reduces the loss of clustering accuracy, so that the stability of the algorithm is improved. The experimental results show that the DP-chDPC algorithm can effectively reduce the loss of clustering accuracy after the algorithm adds noise, and the clustering results are more stable.

INDEX TERMS Cluster analysis, differential privacy, Chebyshev distance, dichotomous method, Laplace mechanism.

I. INTRODUCTION

In today's society, various behaviors of people can be saved by data, and massive data can be obtained in a short time [1], and with the gradual rise of speech recognition, deep learning and the rapid development of the Internet, various new data mining algorithms have been proposed [2]. Data mining can build different models to analyze data, find out the intrinsic laws, and obtain valuable information, but when using data mining algorithms to analyze data, it can lead to the leakage of personal privacy, which can cause great losses [3], [4], [5], [6]. Clustering is a common algorithm for data mining, and how to achieve privacy protection in clustering is a hot topic of current research [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Amir Masoud Rahmani.

Differential privacy [8] (DP) was proposed by Dwork in 2006, and it can be proved by rigorous derivation of mathematical knowledge [9]. Its protection mechanism is to distort the data by adding noise to the data to achieve the effect of privacy protection, and the processed data is still available [10]. And in the big data environment, the clustering algorithm oriented to differential privacy protection has high compatibility and can be applied to recommendation systems, face recognition, biomedicine, transportation, etc. to realize people's need for personal privacy protection [11].

DPC algorithm is a relatively efficient and novel clustering algorithm, which has been studied by many scholars. Wang et al. [12], [13], [14] conducted a series of studies on clustering algorithms. Firstly, they proposed a variational density peak clustering algorithm (VDPC), which constructed a unified clustering framework and could

automatically cluster data sets with different density distribution types. Then, a pseudo-label guided density peak clustering algorithm (PLDPC) was proposed. In this algorithm, a pseudo-label generation method based on co-occurrence theory was designed, and mutual information maximization method was used to obtain better clustering results. Finally, an adaptive and improved CMNN algorithm (AVCMNN) was proposed. Firstly, the data points in some small clusters were misidentified as noise points, and a new voting strategy was adopted to redistribute these data points, which improves the clustering results. Then the parameters of the proposed method were optimized by using mutual information maximization to construct the objective function. Finally, better parameter values and clustering results were obtained. Li et al. [15] proposed a density peak clustering algorithm (CFDPC) based on clustering fusion strategy, which solved the problem that data point allocation was prone to joint errors, and selected the clustering center correctly. Ding et al. [16] proposed an improved density peak clustering algorithm (IDPCNNMS) based on the natural neighborhood merging strategy, which could adaptively identify the natural neighbor set of each data, obtain its local density, and effectively eliminate the influence of truncation parameters on the final result. Zou and Wang [17] introduced the idea of connectivity on the basis of the original DPC algorithm, and proposed an improved density peak clustering algorithm (ConDPC), which improved the acquisition of clustering center points and the sample allocation strategy, and improved the clustering accuracy of the algorithm. Yin et al. [18] improved the density peak clustering algorithm. In view of the selection of parameter d_c, the K-nearest neighbor idea was adopted to sort the nearest neighbor distance of each data, and the global bifurcation points were found to divide data of different densities. For the selection of clustering centers, the local density and distance of each data point in each data partition were found out and γ-map was drawn. The average value of γ height difference was calculated, and the maximum discontinuity point was found through two screening. The clustering center and the number of clustering center were determined automatically, thus improving the clustering accuracy of the algorithm. Li et al. [19] proposed a new density peak clustering method based on fuzzy semantic units and introduced relative semantic distance, which made the decision graph of clustering center selection clearer and the clustering results better. Liu and Wang [20] proposed a density-based optimal peak density algorithm (DTDPC), which automatically selected cluster centers according to the weight trends of cluster centers. A density-based two-step allocation strategy was designed to divide core points and boundary points, and the clustering results were relatively stable.

Applying differential privacy protection in clustering analysis can protect personal privacy while obtaining valuable information. Blum et al. [21] first introduced differential privacy into clustering analysis by proposing the SuLQ framework and querying the database using the DPk-means algorithm, but such algorithms were more sensitive to the type of data set and were prone to obtain local optimal solutions. Therefore, Wu and Huang et al. [22] proposed the DP-DBSCAN algorithm based on differential privacy protection, which combined differential privacy with density clustering, and could maintain the effectiveness of clustering while obtaining differential privacy protection, expanding its applicability. Ni et al. [23] proposed a differential privacy-preserving multicore DBSCAN (DP-MCDBSCAN) clustering scheme based on differential privacy protection to address the privacy protection problem in clustering analysis of web user data, which effectively solved the privacy leakage problem in data mining and improved the clustering accuracy of DBSCAN under differential privacy protection. Wang et al. [24] made improvements in OPTICS algorithm and introduced differential privacy DP-OPTICS algorithm, which solved the problem of low data availability, but required a relatively accurate estimation of the user's query probability. Sun et al. [25] combined Euclidean distance and shared nearest neighbor similarity to redefine local density and proposed a differential privacy-preserving algorithm DP-DPCSNNS based on shared nearest neighbor similarity, which overcame the privacy leakage problem of the original DPC algorithm and improved the accuracy of clustering results, but did not consider the stability of clustering results. Chen et al. [26] introduced the reachable centroid definition and proposed the DP-rcCFSFDP algorithm, which could perform effective clustering while protecting data privacy, but also did not consider the stability problem of the algorithm after adding noise. Chen et al. [27] proposed an adaptive clustering center density peak clustering algorithm based on differential privacy for the poor adaptive ability of DPC algorithm on high-dimensional data, the inability to automatically determine the cluster center, and the privacy problems in cluster analysis. This algorithm not only solved the privacy problem in cluster analysis but also greatly improved the clustering accuracy, but still did not consider the stability problem after the algorithm added noise.

Currently, most of the research on the combination of differential privacy with density clustering focuses on improving the algorithm to improve the accuracy of clustering. However, the loss of clustering accuracy caused by adding noise and the stability of clustering results are not taken into account. In this paper, we combine density peak clustering (DPC) algorithm with differential privacy, and propose a density peak clustering algorithm DP-chDPC oriented to differential privacy protection, aiming to protect personal privacy while reducing the loss of clustering accuracy after the algorithm adds noise, improving the stability of clustering results, and this algorithm has a better clustering effect.

The main works of this paper are summarized as follows.

1) To address the instability of clustering results under differential privacy protection, a density peak clustering algorithm DP-chDPC for differential privacy protection is

proposed. Different from the DP-DPC algorithm, this algorithm can reduce the loss of clustering accuracy and improve the stability of the algorithm. Privacy analysis proves that the DP-chDPC clustering algorithm not only clusters the data, but also prevents the data information from being leaked.

2) To address the problem that the DPC algorithm requires manual determination of truncation distance and clustering centroids, we use the dichotomous method to automatically determine the truncation distance to avoid the subjectivity of manual selection. The thresholds of local density and center offset distance is set to automatically obtain the clustering centroids, which overcomes the uncertainty of the DPC algorithm to select the clustering centers based on the decision map.

3) We demonstrate the privacy-preserving performance of the DP-chDPC algorithm and conduct experiments to validate our algorithm. The experimental results show that the algorithm can effectively improve the stability of the algorithm on low-dimensional datasets and high-dimensional datasets, reduce the loss of clustering accuracy after the algorithm adds noise, and have significant advantages in clustering results on low-dimensional datasets.

The rest of this paper is organized as follows. In Section II, we introduce the definition related to differential privacy and the concepts and specific steps related to the DPC algorithm. In Section III, we propose the DP-chDPC algorithm and prove its privacy. In Section IV, we conduct experiments to evaluate the performance of our proposed algorithm. Finally, we present the findings of this work and discuss future research work in Section V.

II. RELATED WORKS

A. RELEVANT DEFINITIONS OF DIFFERENTIAL PRIVACY

Definition 1 (ε-Differential Privacy) [28]: For an algorithm M, $M(D)$ is the set formed by M on a dataset D. Suppose D_1 and D_2 are two datasets with the same attributes and the number of records differs by 1. S is a subset of $M(D)$, and if algorithm M satisfies equation (1), then M is said to provide ε-differential privacy.

$$P_r[M(D_1) \in S] \leq e^{\varepsilon} \times P_r[M(D_2) \in S] \tag{1}$$

where the parameter ε represents the privacy protection budget, which can reflect the privacy protection strength provided by the algorithm M. The smaller the value, the higher the protection strength, and the practical application needs to balance the availability of data and the level of privacy protection.

Definition 2 (Global Sensitivity) [29]: Assuming that the function $f : D \to R^d$, for any input dataset D_1 and its adjacent dataset D_2, the global sensitivity of function f is:

$$GS_f = \max_{D_1, D_2} |f(D_1) - f(D_2)| \tag{2}$$

Definition 3 (Local Sensitivity) [30]: Assuming that the function $f : D \to R^d$, the input is the dataset D_1, and its neighboring dataset is D_2, the local sensitivity of the function f on D_1 is:

$$LS_f = \max_{D_2} |f(D_1) - f(D_2)| \tag{3}$$

B. IMPLEMENTATION MECHANISM OF DIFFERENTIAL PRIVACY PROTECTION

The privacy preserving mechanisms commonly used in differential privacy are Laplace mechanism and exponential mechanism, mainly by adding random noise to the algorithm to make its query results or output results satisfy differential privacy protection, so as to ensure that the privacy information is not leaked, and the size of the added noise depends on the privacy preserving budget and sensitivity. Among them, the Laplace mechanism is applicable to the privacy protection of numerical datasets, while the exponential mechanism is applicable to the privacy protection of non-numerical datasets.

Theorem 1 (Laplace Mechanism) [31]: Given a dataset D, assume that the function $f : D \to R^d$, the sensitivity of the function f is Δf, and if the algorithm M satisfies $M(D) = f(D) + Lap(b)$, then M provides ε-differential privacy protection. where $Lap(b)$ obeys a Laplace distribution with a location parameter of 0 and a scale parameter of $b = \Delta f / \varepsilon$. The magnitude of the provided noise is inversely proportional to ε and proportional to Δf.

Theorem 2 (Exponential Mechanism) [31]: Given a dataset D, assume that the output of the randomized algorithm M is an entity object $r \in R_{Range}(M)$, $q(D, r)$ is the availability function, and Δq is the sensitivity of $q(D, r)$. If r is selected and output from $R_{Range}(M)$ with probability $M \propto \exp(\varepsilon \times q(D, r)/2\Delta q)$, then M provides ε-differential privacy protection.

C. DPC ALGORITHM

The DPC algorithm is a density-based clustering method proposed by Rodriguez and Laio [32] in 2014, which does not require iteration and enables efficient clustering of arbitrarily shaped datasets. The algorithm is based on two important assumptions: 1) the centers of class clusters are surrounded by other data points of lower density and 2) the distance between the centers of class clusters is relatively far [33]. In order to satisfy these two assumptions, it is crucial to calculate the local density ρ_i of sample point i and the center offset distance δ_i between sample point i and other denser points in order to find the class cluster centers accurately.

Definition 4 (Euclidean Distance) [34]: For any n-dimensional vectors α and β, the Euclidean distance is the true distance between any two points in the n-dimensional space, which is calculated as:

$$d(\alpha, \beta) = \sqrt{\sum_{i=1}^{n} (\alpha_i - \beta_i)^2} \tag{4}$$

Definition 5 (Local Density) [35]: Suppose there is a dataset $X_{N \times M} = (x_1, x_2, \cdots, x_N)^T$, $x_i = (x_{i1}, x_{i2}, \cdots, x_{iM})$, N is the number of samples and M is the number of sample

dimensions. Then the local density ρ_i is calculated in two ways: for larger datasets a truncation kernel is usually used, and for smaller datasets a Gaussian kernel is usually used.

The truncation kernel is:

$$\rho_i = \sum_{i \neq j} \chi\left(d_{ij} - d_c\right), \quad \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{5}$$

The Gaussian kernel is:

$$\rho_i = \sum_{i \neq j} \exp\left[-\left(\frac{d_{ij}}{d_c}\right)^2\right] \tag{6}$$

where d_{ij} is the distance between data points after normalization; d_c is the truncation distance, which is the only input parameter, and the truncation percentage is usually set to about 2%.

Definition 6 (Center Offset Distance) [15]: The center offset distance δi is the distance closest to sample point i among all the distances between sample point i and other points of higher density, and is calculated as follows.

$$\delta_i = \begin{cases} \max\limits_{j}(d_{ij}), & \rho_i = \max(\rho) \\ \min\limits_{j:\rho_j > \rho_i}(d_{ij}), & \rho_i < \max(\rho) \end{cases} \tag{7}$$

That is, for the sample with the highest density, the center offset distance is calculated as the distance to its farthest point, and for the remaining data points, the center offset distance is calculated as the distance to the nearest point of any other point with higher density [34].

The DPC algorithm is a relatively new clustering algorithm, and its specific steps are shown in Algorithm 1 [33]. First, the input data set is standardized and the distance matrix is calculated to determine the truncation distance; then, the local density ρ_i and the center offset distance δ_i are calculated for each data point, and the decision map is drawn to select the clustering center points; finally, the non-clustering center data points are categorized and the data division results are output.

Algorithm 1 DPC Algorithm

Input: Dataset $X_{N\times M} = (x_1, x_2, \cdots, x_N)^T$
Output: Category label Y
1: Standardized dataset $X_{N\times M}$.
2: The distance matrix $D = (d_{ij})$ of the sample dataset is calculated according to Equation (4), where $d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2}$.
3: Set the truncation percentage p to calculate the truncation distance d_c.
4: calculate the local density ρ_i and the central offset distance δ_i according to Eqs. (5), (6) and (7).
5: Plotting decision diagrams based on ρ_i and δ_i and selecting cluster centroids.
6: Categorize the non-clustering center data points and output the data division results.

III. DPC ALGORITHM FOR DIFFERENTIAL PRIVACY PROTECTION

At present, the research of DPC algorithm based on differential privacy protection mainly improves the original DPC algorithm and improves the clustering accuracy of the algorithm after adding noise, but does not consider the stability of the clustering results after the algorithm adds noise. In this paper, we propose a differential privacy-preserving algorithm DP-chDPC for the above problems, which can effectively reduce the loss of clustering accuracy and improve the stability of the algorithm after adding noise. The advantages of the DP-chDPC algorithm are clarified through a detailed comparative analysis with the DP-DPC algorithm, and Figure 1 shows the basic flow of each algorithm.

The basic flow of DPC, DP-DPC, and DP-chDPC algorithms is given in Figure 1. The comparative analysis shows that the DP-DPC algorithm only adds noise to the local density using the Laplace mechanism on the basis of the DPC algorithm, thus achieving the privacy protection of the algorithm. And the DP-chDPC algorithm first uses Chebyshev distance instead of Euclidean distance to calculate the distance matrix, which improves the stability of the algorithm. Then the truncation distance is automatically calculated using the dichotomy method, which saves the time of finding the truncation percentage. Then the same Laplace mechanism is used to add noise to the local density to achieve privacy protection. Finally, the centroids of clusters are found automatically by setting the thresholds of local density and center offset distance after adding noise.

A. CHEBYSHEV DISTANCE

Exploring a suitable method to calculate the distance matrix is beneficial to improve the performance of clustering after adding noise. The DP-chDPC algorithm calculates the distance matrix in terms of Chebyshev distance, which can improve the stability of the algorithm after adding noise. For any n-dimensional vectors α and β, the Chebyshev distance between them is specifically defined as:

$$d(\alpha, \beta) = \max_{i=1,2,\cdots,n} |\alpha_i - \beta_i| \tag{8}$$

B. DICHOTOMOUS METHOD TO DETERMINE THE TRUNCATION DISTANCE

The original DPC algorithm is selected manually according to experience, and the truncation percentage is artificially given a value to calculate the truncation distance d_c, so the selection is too subjective and it is difficult to get good results by only relying on one parameter. In contrast, this paper introduces two new parameters to automatically calculate the truncation distance d_c by simply giving a range. In comparison, this new method can reduce the searching time and achieve better clustering results. In this paper, we use the dichotomous method to calculate the truncation distance, and determine

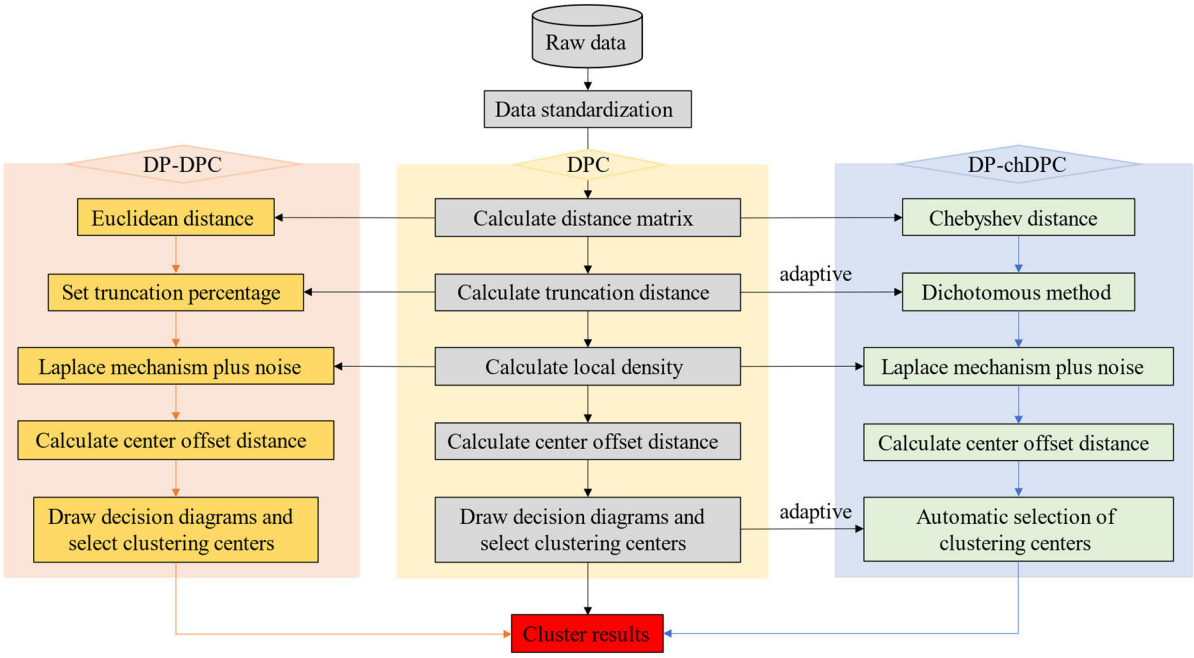


FIGURE 1. Basic flow of different algorithms.

the truncation distance automatically by setting the values of parameters r_1, r_2, and the specific steps are:

Algorithm 2 Dichotomous Method to Determine the Truncation Distance

Input: Distance matrix $D = (d_{ij})$
Output: Truncation distance d_c
1: Initialize the truncation distance d_c as the average of the maximum and minimum values of the distance matrix, and let $d_{\max}$, $d_{\min}$ denote the maximum and minimum values of D respectively, then $d_c = (d_{\max} + d_{\min})/2$.
2: Calculate the number n of points with d_{ij} less than d_c and calculate its percentage $r = n/N$, where N denotes the total number of samples.
3: If $r < r_1$, assign d_c to $d_{\min}$, return to step 1, and continue the loop.
4: If $r > r_2$, assign d_c to $d_{\max}$, return to step 1, and continue the loop with the new d_c instead of the initial value.
5: If $r_1 \le r \le r_2$ or $d_{\max} - d_{\min} < 0.0001$, then exit the loop and output d_c.

C. DP-DPC ALGORITHM

The DP-DPC algorithm can protect the information of the data from being leaked while clustering and analyzing the data, and its specific steps are shown in Algorithm 3. Firstly, the input data set is standardized and the distance matrix is calculated using Euclidean distance. Then, the truncation percentage p is set manually, and the truncation distance is calculated. Next, the local density ρ_i is calculated and noise is added to obtain ρ_i', followed by the center offset distance δ_i'. Finally, a decision diagram is drawn based on ρ_i' and δ_i', cluster centroids are selected, and non-cluster centroid data points are grouped, and the data division results are output.

Algorithm 3 DP-DPC Algorithm

Input: dataset $X_{N\times M} = (x_1, x_2, \cdots, x_N)^T$, privacy-preserving budget ε, Truncation percentage p, number of clustering centers
Output: Category label Y
1: Standardized dataset $X_{N\times M}$.
2: Calculate the distance matrix $D = (d_{ij})$ of the sample data set according to Eq. (4), where $d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2}$.
3: Set the truncation percentage p to calculate the truncation distance d_c.
4: Calculate the local density ρ_i according to equations (5) and (6), and then add Laplace noise to ρ_i by Laplace mechanism, then the local density $\rho_i' = \sum_{i\neq j}\exp(-(d_{ij}/d_c)^2) + Lap(b)$ after adding noise.
5: Calculate the center offset distance δ_i' according to ρ_i' and equation (7).
6: Plotting decision diagrams based on ρ_i' and δ_i' and selecting cluster centroids.
7: Categorize the non-clustering center data points and output the data division results.

D. DP-CHDPC ALGORITHM

The DP-chDPC algorithm is mainly used to solve the problem of loss of clustering accuracy after the algorithm adds noise, so as to improve the stability of the algorithm. The specific

steps are shown in Algorithm 4. Firstly, the input data set is standardized and the Chebyshev distance is used to calculate the distance matrix. Then, according to algorithm 2, the dichotomy method is used to automatically determine the truncation distance. Secondly, the local density ρ_i is calculated, and the noise is added to get ρ_i', and then the center deviation distance δ_i is calculated. Finally, by setting the threshold values of local density and center offset distance, the clustering central points are automatically selected, and the non-clustering central data points are classified, and the data division results are output.

Algorithm 4 DP-chDPC Algorithm

Input: dataset $X_{N\times M} = (x_1, x_2, \cdots, x_N)^T$, privacy-preserving budget ε
Output: Category label Y
1: Standardized dataset $X_{N\times M}$.
2: Calculate the distance matrix $D = (d_{ij})$ of the sample data set according to Eq. (8), where $d_{ij} = \max_{k=1,2,\cdots,n} |x_{ik} - x_{jk}|$.
3: Calculate the truncation distance d_c according to Algorithm 2.
4: Calculate the local density ρ_i according to equations (5) and (6), and then add Laplace noise to ρ_i by Laplace mechanism, then the local density $\rho_i' = \sum_{i\neq j} \exp(-(d_{ij}/d_c)^2) + Lap(b)$ after adding noise.
5: Calculate the center offset distance δ_i' according to ρ_i' and equation (7).
6: Let the thresholds of local density and central offset distance be $\rho = (\rho_{\min} + \rho_{\max})/2$, $\delta = (\delta_{\min} + \delta_{\max})/2$, respectively, and select the points of $\rho_i' > \rho$ and $\delta_i' > \delta$ as cluster centroids. Where $\rho_{\min}$, $\rho_{\max}$ are the minimum and maximum values in ρ_i', $\delta_{\min}$, $\delta_{\max}$ are the minimum and maximum values in δ_i', respectively.
7: Categorize the non-clustering center data points and output the data division results.

E. ANALYSIS OF PRIVACY

The DP-chDPC algorithm achieves differential privacy protection by adding Laplace noise to the local density ρ_i and setting the privacy budget to ε. Assuming that D_1 and D_2 are mutually neighboring datasets, M is a noise-addition algorithm, and ρ_i is ρ_i' after noise addition, the privacy proof procedure of the DP-chDPC algorithm is as follows.

$$\begin{aligned}\frac{\Pr[M(\rho_i) = \bar{\rho}_i]}{\Pr[M(\rho_i') = \bar{\rho}_i]} &= \frac{\exp(-\frac{\varepsilon|\bar{\rho}_i - \rho_i|}{\Delta f})}{\exp(-\frac{\varepsilon|\bar{\rho}_i - \rho_i'|}{\Delta f})} \\ &= \exp(\frac{\varepsilon(|\bar{\rho}_i - \rho_i'| - |\bar{\rho}_i - \rho_i|)}{\Delta f}) \\ &\leq \exp(\frac{\varepsilon|\rho_i - \rho_i'|}{\Delta f}) \\ &\leq \exp(\frac{\varepsilon(\max|\rho_i - \rho_i'|)}{\Delta f}) \\ &= \exp(\varepsilon)\end{aligned}$$

TABLE 1. Experimental datasets.

Dataset	Attributes	Size	Clusters	Sources
Aggregation	2	788	7	*Synthetic*
Compound	2	399	6	*Synthetic*
Flame	2	240	2	*Synthetic*
Jain	2	373	2	*Synthetic*
Pathbased	2	300	3	*Synthetic*
R15	2	600	15	*Synthetic*
Banknote	4	1372	2	*UCI*
Seeds	7	210	3	*UCI*
Iris	4	150	3	*UCI*
Wine	13	178	3	*UCI*
Abalone	8	4177	3	*UCI*
Ecoli	7	336	8	*UCI*

From the above: the clustering algorithm DP-chDPC with noise addition to the local density ρ_i satisfies ε-differential privacy protection.

F. ANALYSIS OF ALGORITHM COMPLEXITY

For a dataset with sample size n, the time complexity of the DPC algorithm is determined by 3 main parts: a) Calculate the distance matrix using Euclidean distance. b) Calculate the local density ρ_i for each sample. c) Calculate the center offset distance δ_i for each sample. The time complexity of each part is $O(n^2)$, so the total time complexity is $O(n^2)$.

The time complexity of the DP-chDPC algorithm proposed in this paper is determined by six main components: a) Calculating the distance matrix using Chebyshev distance with a time complexity of $O(n^2)$. b) Compute the truncated distance d_c using the dichotomous method with a time complexity of $O(\log n)$. c) Calculate the local density ρ_i for each sample with time complexity $O(n^2)$. d) Add Laplace noise to the local density ρ_i to get the local density ρ_i' after adding noise, with time complexity $O(n)$. e) Calculate the center offset distance δ_i' from ρ_i' with time complexity $O(n^2)$. f) Set the thresholds of local density and center offset distance to automatically select the clustering centroids with time complexity $O(n)$.

Therefore, the time complexity of DP-chDPC algorithm is $O(n^2)$, which is of the same order of magnitude as the time complexity of the DPC algorithm.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the performance of the DP-chDPC algorithm, this paper uses the operating system Windows 11, Python 3.9.7 development environment, processor Intel(R) Core (TM) i5-1155G7@2.50GHz2.50GHz, and 8.00 GB of memory. experimental data are used in six classical synthetic datasets and six UCI real datasets to test and evaluate the algorithm, and each experimental dataset is shown in Table 1, which describes the characteristics of each dataset respectively.

A. EVALUATION INDICATORS

The DP-chDPC algorithm can not only perform cluster analysis on data, but also prevent the disclosure of privacy information. It also considers the stability of clustering results

after the algorithm adds noise. Its evaluation index needs to be considered from two perspectives, namely differential privacy protection intensity and clustering accuracy.

(1) Differential privacy protection strength. The DP-chDPC algorithm realizes differential privacy protection based on Laplace noise mechanism, and its protection strength is related to Δf and ε. By $|f(D_1) - f(D_2)| = 1$ is $\Delta f = 1$. Then the smaller ε, add the greater the noise, protect the higher strength; The larger the ε, the less noise is added and the lower the protection strength.

(2) The accuracy of clustering. The experimental datasets used in this paper all contain true labels, so the external metrics of clustering are used for the metrics. Adjusted mutual information [36] (AMI), adjusted Rand coefficient [37] (ARI), and Fowlkes-mallows index [38] (FMI) are chosen as the accuracy evaluation criteria of clustering by DP-chDPC algorithm, and all three metrics take values in the range of [0,1], and the larger the value the better the clustering effect.

Suppose Y_{true} and Y_{pred} are two classes of labels for the sample data set, Y_{true} is the true label and Y_{pred} is the clustering label, and the entropy of these two classes of labels is:

$$H(Y_{true}) = \sum_{i=1}^{|Y_{true}|} P(i)\log(P(i)),$$
$$H(Y_{pred}) = \sum_{j=1}^{|Y_{pred}|} P(j)\log(P(j)) \tag{9}$$

where $P(i) = |Y_{true}(i)|/N$, $P(j) = |Y_{pred}(j)|/N$, respectively, denotes the number of percentages of each label in the dataset. Then the adjusted mutual information AMI is:

$$AMI = \frac{\sum_{i=1}^{|Y_{true}|}\sum_{j=1}^{|Y_{pred}|} P(i,j)\log(\frac{P(i,j)}{P(i)P(j)})}{\sqrt{H(Y_{true})H(Y_{pred})}} \tag{10}$$

Adjusting the Rand coefficient ARI and the Fowlkes-Mallows index FMI for:

$$ARI = \frac{2(A \times D - B \times C)}{(B+D)(A+C)+(C+D)(A+B)} \tag{11}$$

$$FMI = \frac{A}{\sqrt{(A+B)(A+C)}} \tag{12}$$

where A denotes the number of data point pairs that are the same class in both Y_{true} and Y_{pred}. B denotes the number of data point pairs that are the same class in Y_{true} but not the same class in Y_{pred}. C denotes the number of data point pairs that are not the same class in Y_{true} but the same class in Y_{pred}. D denotes the number of data point pairs that are not the same class in both Y_{true} and Y_{pred}.

B. THE SETTING OF EXPERIMENTAL PARAMETERS

In order to test the clustering performance of each algorithm on each dataset more objectively, the parameter settings of each algorithm are tuned in different datasets. chDPC, DP-chDPC algorithm requires setting parameters r_1, r_2 to calculate the truncation distance d_c so that the average number of neighbors of each sample point is about 1%-2% of the total number of sample points in the dataset, but this does not apply to all datasets and it is necessary to adjust the parameters for clustering according to the characteristics of the dataset. For Aggregation, Flame and Wine datasets $r_1 = 0.01$ and $r_2 = 0.04$; for Compound, Jain, Seeds and Ecoli datasets $r_1 = 0.01$ and $r_2 = 0.03$; for Pathbased, R15, Banknote, Iris and Abalone datasets $r_1 = 0.02$ and $r_2 = 0.04$.The DPC, DP-DPC algorithms need to set the parameter truncation percentage p, $p = 2$ on the Jain, Pathbased, Seeds, Banknote, Wine, Abalone, Ecoli datasets. $p = 3$ on Compound, Flame and Iris datasets. $p = 4$ in Aggregation and R15 datasets.

C. ANALYSIS OF EXPERIMENTAL RESULTS OF SYNTHETIC DATASETS

In order to verify that the DP-chDPC algorithm using Chebyshev distance can effectively reduce the loss of clustering accuracy after the algorithm adds noise, six representative synthetic datasets are selected for comparison experiments with Euclidean distance. The other conditions of this experiment are consistent and only the distance is compared. When noise is added, its privacy-preserving budget ε is taken as (0.1, 10) with a step size of 0.5, and the average values of ARI, AMI, and FMI corresponding to all ε are calculated, and then the same experiment is repeated 10 times, and the average values of ARI, AMI, and FMI of 10 times are taken as the final clustering results. Where, the parameter c represents the number of clustering centers of the algorithm. As shown in Table 2, for Aggregation, Compound and Jain datasets, the DP-chDPC algorithm has the same clustering results using Euclidean distance and Chebyshev distance before adding noise, but after adding noise, the clustering results using Chebyshev distance are better. On the Flame dataset, the DP-chDPC algorithm has slightly better clustering results using Euclidean distance than Chebyshev distance before adding noise, but also better clustering results using Chebyshev distance after adding noise. On the Pathbased dataset, the clustering results of Euclidean distance using Euclidean distance are slightly better than Chebyshev distance before adding noise, but after adding noise, Euclidean distance is slightly higher than Chebyshev distance on AMI, and Chebyshev distance is better than ARI and FMI. On the R15 dataset, the clustering results of DP-chDPC algorithm using Euclidean distance are better than those of Chebyshev distance before adding noise, but after adding noise, Euclidean distance is slightly higher than Chebyshev distance in ARI, and Chebyshev distance performs better in AMI and FMI.

The loss of clustering accuracy of DP-chDPC algorithm before and after adding noise is further analyzed. As can be seen from Table 3, on the Aggregation dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.0088, 0.0062, and 0.0052, respectively, which are lower than the results of Euclidean distance. On the Compound dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.0111, 0.0209, and 0.0079, respectively, which are also lower than the results of Euclidean distance. On the Flame

TABLE 2. Comparison results of synthetic datasets before and after adding noise.

Dataset	Distance	Noise	ARI	AMI	FMI	c
Aggregation	Euclidean distance	No	0.8266	0.9021	0.8761	5
		Yes	0.7678	0.8597	0.8407	
	Chebyshev distance	No	0.8266	0.9021	0.8761	6
		Yes	**0.8178**	**0.8959**	**0.8709**	
Compound	Euclidean distance	No	0.7402	0.8041	0.8304	3
		Yes	0.7061	0.7620	0.8116	
	Chebyshev distance	No	0.7402	0.8041	0.8304	4
		Yes	**0.7291**	**0.7832**	**0.8225**	
Flame	Euclidean distance	No	1.0000	1.0000	1.0000	2
		Yes	0.7728	0.8061	0.8714	
	Chebyshev distance	No	0.9832	0.9632	0.9922	2
		Yes	**0.8768**	**0.8656**	**0.9266**	
Jain	Euclidean distance	No	0.7055	0.6439	0.8779	2
		Yes	0.5786	0.5437	0.8206	
	Chebyshev distance	No	0.7055	0.6439	0.8779	2
		Yes	**0.6228**	**0.5769**	**0.8463**	
Pathbased	Euclidean distance	No	0.5031	0.5737	0.6812	3
		Yes	0.4734	**0.5526**	0.6679	
	Chebyshev distance	No	0.5006	0.5640	0.6799	3
		Yes	**0.4821**	0.5524	**0.6714**	
R15	Euclidean distance	No	0.9927	0.9938	0.9932	15
		Yes	**0.9789**	0.9782	0.9723	
	Chebyshev distance	No	0.9785	0.9833	0.9799	15
		Yes	0.9741	**0.9816**	**0.9759**	

TABLE 3. Accuracy decline of synthetic datasets.

Dataset	Distance	ARI	AMI	FMI
Aggregation	Euclidean distance	0.0588	0.0424	0.0354
	Chebyshev distance	**0.0088**	**0.0062**	**0.0052**
Compound	Euclidean distance	0.0341	0.0421	0.0188
	Chebyshev distance	**0.0111**	**0.0209**	**0.0079**
Flame	Euclidean distance	0.2272	0.1939	0.1286
	Chebyshev distance	**0.1064**	**0.0976**	**0.0656**
Jain	Euclidean distance	0.1269	0.1002	0.0573
	Chebyshev distance	**0.0767**	**0.0670**	**0.0316**
Pathbased	Euclidean distance	0.0297	0.0211	0.0133
	Chebyshev distance	**0.0185**	**0.0116**	**0.0085**
R15	Euclidean distance	0.0138	0.0156	0.0209
	Chebyshev distance	**0.0044**	**0.0017**	**0.0040**

dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.1064, 0.0976, and 0.0656, respectively, which are still lower than the results of Euclidean distance. On the Jain, Pathbased and R15 datasets, it remains that the loss of clustering accuracy using Chebyshev distance is smaller. It can be seen that the DP-chDPC algorithm can effectively reduce the loss of clustering accuracy when the algorithm adds noise on the low-dimensional datasets.

The trend of clustering accuracy with privacy protection budget is further analyzed. FMI is taken as the evaluation index. It can be seen from Figure 2 that, on the Aggregation data set, the FMI with Chebyshev distance increases in a certain range first, reaches a critical value at $\varepsilon = 0.6$, and then maintains a steady state. However, the variation trend of FMI using Euclidean distance firstly increases within a certain range and reaches a critical value at $\varepsilon = 1.1$, and then the variation has a small amplitude fluctuation. In the Compound data set, FMI using Chebyshev distance has a stable variation trend, while FMI using Euclidean distance has a large fluctuation in a certain range, reaches a critical value at $\varepsilon = 1.9$, and then has a stable variation trend. On Flame data set, FMI with Chebyshev distance has a tendency to fluctuate and increase within a certain range, reaching a critical value at $\varepsilon = 1.8$, and then remaining stationary. However, the FMI with Euclidean distance first fluctuates and increases within a certain range, and then reaches a critical value at $\varepsilon = 2.4$, and then the variation still has a large fluctuation range. In Jain data set, FMI with Chebyshev distance first fluctuates in a certain range, reaches a critical value at $\varepsilon = 1.9$, and then changes very steadily. However, the FMI with Euclidean distance always fluctuates greatly. In the Pathbased dataset, the trend of FMI using Chebyshev distance and Euclidean distance is consistent, which first increases within a certain range, reaches a critical value at $\varepsilon = 1.3$, and then maintains a stable state. In R15 data set, the FMI with Chebyshev distance keeps steady, while the FMI with Euclidean distance first increases in a certain range and reaches a critical value at $\varepsilon = 0.7$, and then maintains a steady state. It can be seen that the DP-chDPC algorithm is not only effective in reducing the loss of clustering accuracy after the algorithm adds noise on low-dimensional data sets, but also improves the stability of the algorithm.

D. ANALYSIS OF EXPERIMENTAL RESULTS OF UCI DATASETS

It is further verified that the DP-chDPC algorithm using Chebyshev distance on the UCI dataset can effectively reduce the loss of clustering accuracy after the algorithm adds noise, and its privacy-preserving budget ε takes the same value as the synthetic dataset when noise is added, and the same experiment is still repeated 10 times, and the average of ARI, AMI, and FMI of 10 times is taken as the final clustering result. Where, the parameter c represents the number of clustering centers of the algorithm. As can be seen from Table 4, on the Banknote dataset, the clustering results of the DP-chDPC algorithm using Chebyshev distance before adding noise performed better on ARI and AMI, and the clustering results using Euclidean distance performed better on FMI. However, after adding noise, the clustering results using Chebyshev distance were better on both. On Seeds and Wine datasets, the clustering results using Euclidean distance were better on DP-chDPC algorithm before adding noise. However, after adding noise, the clustering results using Euclidean distance performed better on AMI and FMI, and

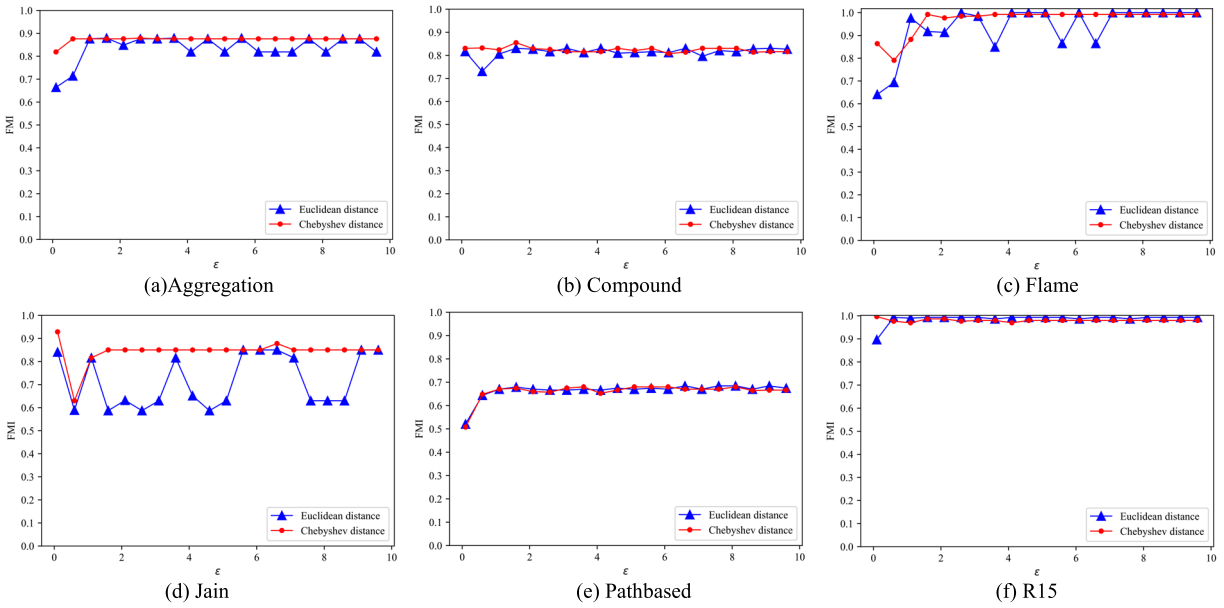


FIGURE 2. Trends of FMI in synthetic datasets.

the clustering results using Chebyshev distance performed better on ARI. On the Iris and Ecoli datasets, the clustering results using Euclidean distance were better on DP-chDPC algorithm before adding noise. However, after adding noise, the clustering results using Euclidean distance performed better on ARI and AMI, and the clustering results using Chebyshev distance performed better on FMI. On the Abalone dataset, the DP-chDPC algorithm has better clustering results using Euclidean distance before adding noise. However, after adding noise, the clustering results using Euclidean distance performed better on ARI, and the clustering results using Chebyshev distance performed better on AMI and FMI.

The loss of clustering accuracy of DP-chDPC algorithm before and after adding noise was further analyzed. As shown in Table 5, on the Banknote dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.0321, 0.0409, and 0.0343, respectively, which are lower than the results of Euclidean distance. On the Seeds dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.0035, 0.0174, and 0.0254, respectively, which are also lower than the results of Euclidean distance. On the Iris dataset, the losses of ARI, AMI, and FMI using Chebyshev distance are 0.0035, 0.0068, and 0.0067, respectively, which are still lower than the results of Euclidean distance. In the Wine, Abalone and Ecoli datasets, Chebyshev distance is still used to reduce the loss of clustering accuracy. It can be seen that the DP-chDPC algorithm is also effective in reducing the loss of clustering accuracy after the algorithm adds noise on high-dimensional data sets.

Further analysis of the trends of clustering accuracy with the privacy protection budget on the UCI datasets, taking FMI as the evaluation index, it can be seen from Figure 3 that on the Banknote dataset, the trends of FMI using Chebyshev distance increases within a certain range first, reaches a critical value at $\varepsilon = 1.8$, and then remains a steady state.

TABLE 4. Comparison results of UCI datasets before and after adding noise.

Dataset	*Distance*	*Noise*	*ARI*	*AMI*	*FMI*	*c*
Banknote	*Euclidean distance*	*No*	*0.7067*	*0.7097*	*0.8422*	*4*
		Yes	*0.6238*	*0.6603*	*0.7894*	
	Chebyshev distance	*No*	*0.7579*	*0.7955*	*0.8303*	*4*
		Yes	***0.7258***	***0.7546***	***0.7960***	
Seeds	*Euclidean distance*	*No*	*0.7997*	*0.7542*	*0.8659*	*3*
		Yes	*0.6941*	***0.6798***	***0.8103***	
	Chebyshev distance	*No*	*0.7407*	*0.6953*	*0.8265*	*3*
		Yes	***0.7372***	*0.6779*	*0.8011*	
Iris	*Euclidean distance*	*No*	*0.5681*	*0.7316*	*0.7715*	*2*
		Yes	***0.5437***	***0.7014***	*0.7463*	
	Chebyshev distance	*No*	*0.5438*	*0.6900*	*0.7580*	*2*
		Yes	*0.5403*	*0.6832*	***0.7513***	
Wine	*Euclidean distance*	*No*	*0.6637*	*0.6982*	*0.7549*	*2*
		Yes	*0.6379*	***0.6647***	***0.7058***	
	Chebyshev distance	*No*	*0.6483*	*0.6539*	*0.6917*	*2*
		Yes	***0.6428***	*0.6407*	*0.6785*	
Abalone	*Euclidean distance*	*No*	*0.8258*	*0.8349*	*0.8875*	*2*
		Yes	***0.8147***	*0.8193*	*0.8673*	
	Chebyshev distance	*No*	*0.8129*	*0.8243*	*0.8748*	*3*
		Yes	*0.8022*	***0.8204***	***0.8674***	
Ecoli	*Euclidean distance*	*No*	*0.3760*	*0.4479*	*0.6494*	*3*
		Yes	***0.3620***	***0.4245***	*0.6269*	
	Chebyshev distance	*No*	*0.3566*	*0.4148*	*0.6354*	*3*
		Yes	*0.3536*	*0.4137*	***0.6339***	

TABLE 5. Accuracy decline of UCI datasets.

Dataset	Distance	ARI	AMI	FMI
Banknote	Euclidean distance	0.0829	0.0494	0.0528
	Chebyshev distance	**0.0321**	**0.0409**	**0.0343**
Seeds	Euclidean distance	0.1056	0.0744	0.0556
	Chebyshev distance	**0.0035**	**0.0174**	**0.0254**
Iris	Euclidean distance	0.0244	0.0302	0.0252
	Chebyshev distance	**0.0035**	**0.0068**	**0.0067**
Wine	Euclidean distance	0.0258	0.0335	0.0491
	Chebyshev distance	**0.0055**	**0.0132**	**0.0132**
Abalone	Euclidean distance	0.0111	0.0156	0.0202
	Chebyshev distance	**0.0107**	**0.0039**	**0.0074**
Ecoli	Euclidean distance	0.0140	0.0234	0.0225
	Chebyshev distance	**0.0030**	**0.0011**	**0.0015**

However, the variation trend of FMI using Euclidean distance has been fluctuating, and the fluctuation range is large. In the Seeds data set, FMI using Chebyshev distance first increases in a certain range, reaches a critical value at $\varepsilon = 1.6$, and then changes steadily. However, the variation trend of FMI using Euclidean distance is that it first increases within a certain range, reaches the critical value at $\varepsilon = 2.1$, and then still has a small amplitude fluctuation. In Iris data set, FMI with Chebyshev distance increases in a certain range at first, reaches a critical value at $\varepsilon = 2.1$, and then changes steadily. However, the variation trend of FMI using Euclidean distance first fluctuates within a certain range and also reaches the critical value at $\varepsilon = 2.1$, and then changes very steadily. On the Wine dataset, FMI with Chebyshev distance increases in a certain range, reaches a critical value at $\varepsilon = 1.6$, and then maintains a steady state. FMI using Euclidean distance first fluctuates within a certain range, reaches a critical value at $\varepsilon = 2.2$, and then changes very steadily. In Abalone data set, FMI with Chebyshev distance has a tendency to fluctuate slightly in a certain range at first, reach a critical value at $\varepsilon = 2.1$, and then maintain a stable state, while FMI with Euclidean distance always has a small fluctuation. In the Ecoli data set, the FMI of Chebyshev distance and Euclidean distance both increase in a certain range at first, but the FMI of Chebyshev distance changes in a small range, reaches a critical value at $\varepsilon = 0.4$ and then maintains a stable state. It can be seen that the performance of DP-chDPC algorithm on high-dimensional data sets is consistent with that on low-dimensional data sets, which can not only effectively reduce the loss of clustering accuracy after the algorithm adds noise, but also improve the stability of the algorithm.

E. COMPARATIVE ANALYSIS OF CLUSTERING RESULTS

In this experiment, the clustering results of DP-DPC algorithm and DP-chDPC algorithm are compared and analyzed, as shown in Figure 4 and Figure 5. The first parameter in both Figure 4 and Figure 5 represents the number of clustering centers.

Through comparative analysis, the clustering result of DP-chDPC algorithm is better than that of DP-DPC algorithm on the whole. Aggregation data set has 7 cluster centers, 5 cluster centers were obtained by DP-DPC algorithm, and 6 cluster centers were obtained by DP-chDPC algorithm. The clustering result of DP-chDPC algorithm is closer to the original data set, and the clustering result is better. Compound data set has 6 cluster centers, 3 cluster centers were obtained by DP-DPC algorithm, and 4 cluster centers were obtained by DP-chDPC algorithm. The clustering result of DP-chDPC algorithm is closer to the original data set, and the clustering result is better. Flame dataset has two cluster centers, and two cluster centers can be obtained by DP-chDPC algorithm and DP-DPC algorithm, but it can be seen from the graph results that the DP-chDPC algorithm has better clustering results. For Jain, Pathbased and R15 data sets, the number of clustering centers obtained by DP-chDPC algorithm and DP-DPC algorithm is the same, respectively 2,3,15, which is consistent with the number of cluster centers in the original data set, and the graphs obtained by clustering are similar, so the clustering results of the two algorithms are basically the same.

F. COMPARISON ANALYSIS WITH OTHER ALGORITHMS

To further verify the effectiveness of the DP-chDPC algorithm, the DP-chDPC algorithm is compared with other density peak clustering algorithms based on differential privacy protection for experiments. The DP-DPC, DP-ADPC, and DP-KNNDPC algorithms are included. Among them, the DP-DPC algorithm is described in detail in Algorithm 3 in Section III of this paper, the DP-KNNDPC algorithm is derived from the literature [25], and the DP-ADPC algorithm is derived from the literature [27]. We use ARI, AMI and FMI as the evaluation criteria for clustering accuracy, and the privacy preserving budget ε is taken as (0.1, 10) with a step size of 0.1, and calculate the average value of all corresponding ARI, AMI and FMI, and then repeat the same experiment 10 times, and take the average value of 10 times of ARI, AMI and FMI as the final result, which can not only reduce the random noise caused by random error, but also can balance the privacy protection strength of the algorithm with the clustering accuracy. The parameter ϵ indicates the value of the privacy protection budget when the FMI reaches the smooth state. "-" indicates that the FMI does not reach the steady state. Par represents the value of the parameters after each algorithm has been tuned. The DP-ADPC algorithm needs to set the value of truncation percentage p and the DP-KNNDPC algorithm needs to set the value of the number of nearest neighbors K.

As shown in Table 6, for the synthetic dataset, the clustering accuracy of DP-chDPC algorithm is higher on Aggregation, Compound dataset and significantly better than other algorithms. On the Flame dataset, the DP-chDPC algorithm performs better on ARI, FMI, and the DP-ADPC algorithm performs better on AMI. On the Jain dataset, the DP-ADPC algorithm performs better on ARI, AMI, and the DP-chDPC

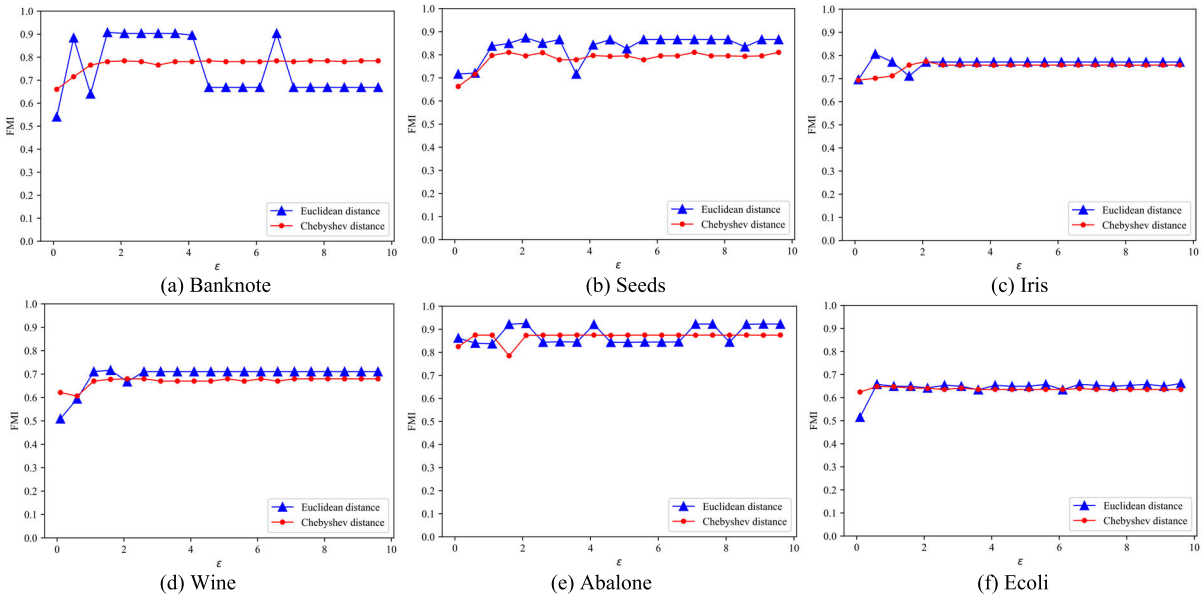


FIGURE 3. Trends of FMI in UCI datasets.

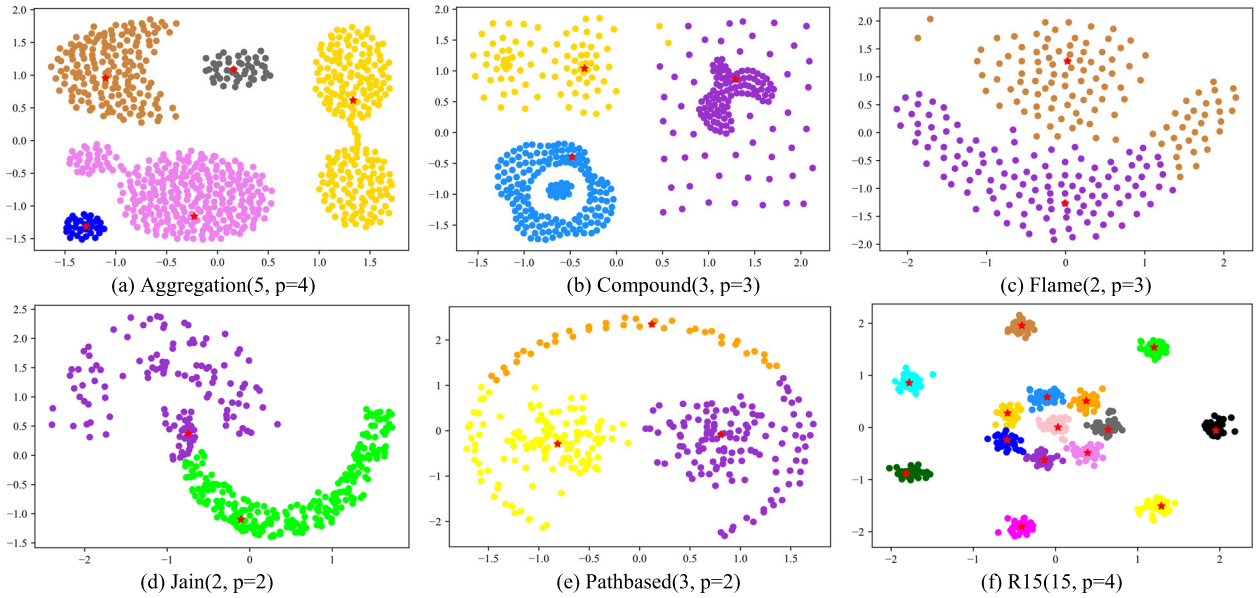
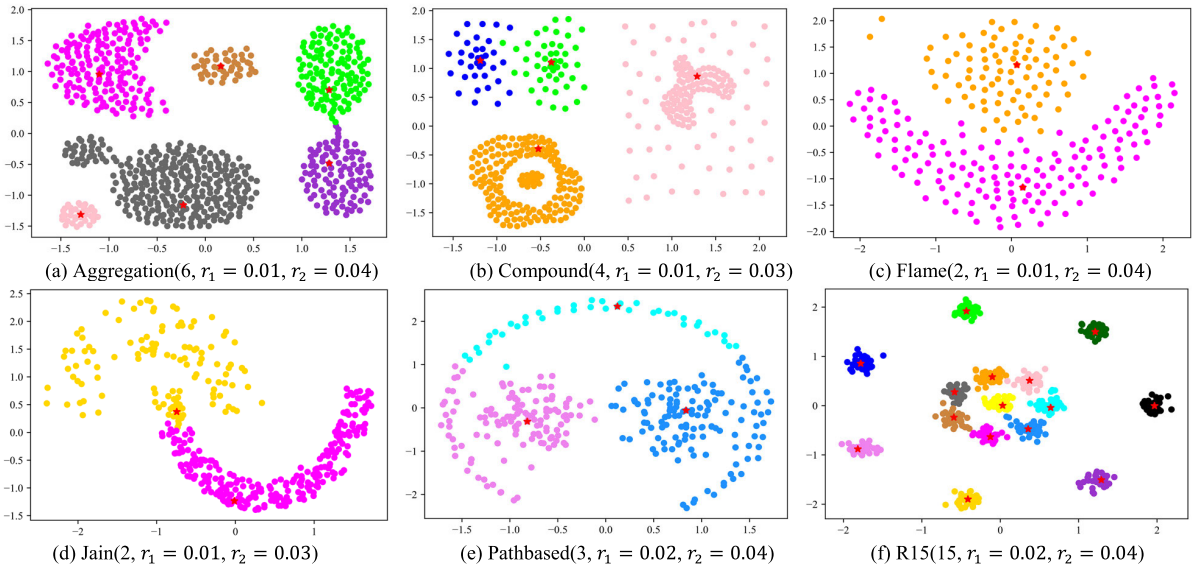


FIGURE 4. Clustering results of DP-DPC algorithm.

algorithm performs better on FMI. On the Pathbased dataset, the DP-chDPC algorithm performs better on ARI, AMI, and the DP-ADPC algorithm performs better on FMI. On the R15 dataset, the DP-chDPC algorithm performs better on AMI, FMI, and the DP-DPC algorithm performs better on ARI.

For the UCI dataset, on the Banknote dataset, the DP-chDPC algorithm performs better on AMI, FMI, and the DP-DPC algorithm performs better on ARI. On the Seeds dataset, the DP-chDPC algorithm performs better on ARI, the DP-KNNDPC algorithm performs better on AMI, and the DP-DPC algorithm performs better on FMI. On the Iris dataset, the DP-chDPC algorithm performs better on FMI, the DP-DPC algorithm performs better on ARI, and the DP-ADPC algorithm performs better on AMI. On the wine dataset, the DP-chDPC algorithm performs better on ARI, and the DP-DPC algorithm performs better on AMI and FMI. On the Abalone dataset, the clustering accuracy of the DP-chDPC algorithm is higher and significantly better than the other algorithms. On the Ecoli dataset, the DP-chDPC algorithm performs better on FMI, and the DP-ADPC algorithm performs better on ARI, AMI.



(a) Aggregation(6, $r_1 = 0.01, r_2 = 0.04$) (b) Compound(4, $r_1 = 0.01, r_2 = 0.03$) (c) Flame(2, $r_1 = 0.01, r_2 = 0.04$)

(d) Jain(2, $r_1 = 0.01, r_2 = 0.03$) (e) Pathbased(3, $r_1 = 0.02, r_2 = 0.04$) (f) R15(15, $r_1 = 0.02, r_2 = 0.04$)

FIGURE 5. Clustering results of DP-chDPC algorithm.

TABLE 6. Algorithm comparison results.

Algorithm	ϵ	*ARI*	*AMI*	*FMI*	*Par*	ϵ	*ARI*	*AMI*	*FMI*	*Par*
	Aggregation					*Compound*				
DP-DPC	1.1	0.7141	0.8135	0.7539	4	1.8	0.5013	0.6837	0.6178	3
DP-ADPC	3	0.0898	0.1163	0.5064	4	-	0.4366	0.5853	0.6763	6.5
DP-KNNDPC	5	0.2871	0.4753	0.4821	6	9	0.3761	0.4386	0.5297	8
DP-chDPC	0.6	**0.8204**	**0.8971**	**0.8725**	0.01, 0.04	1.9	**0.7263**	**0.7752**	**0.8055**	0.01, 0.03
	Flame					*Jain*				
DP-DPC	2.4	0.7966	0.7842	0.9061	3	-	0.5786	0.5438	0.8206	2
DP-ADPC	-	0.8321	**0.8665**	0.9141	3	-	**0.6754**	**0.6257**	0.8301	0.5
DP-KNNDPC	-	0.8149	0.8073	0.8948	4	7	0.5859	0.5982	0.8107	5
DP-chDPC	1.8	**0.8726**	0.8593	**0.9205**	0.01, 0.04	1.9	0.6207	0.5669	**0.8363**	0.01, 0.03
	Pathbased					*R15*				
DP-DPC	1.3	0.4684	0.5386	0.6566	2	0.7	**0.9775**	0.9728	0.9739	4
DP-ADPC	3	0.4209	0.5091	**0.6673**	2.1	6	0.2244	0.4262	0.4179	4
DP-KNNDPC	2.3	0.4482	0.5295	0.6437	9	3	0.6394	0.6529	0.7297	12
DP-chDPC	1.3	**0.4759**	**0.5422**	0.6623	0.02, 0.04	0.3	0.9721	**0.9793**	**0.9757**	0.02, 0.04
	Banknote					*Seeds*				
DP-DPC	-	**0.7482**	0.7292	0.7861	2	2.1	0.7308	0.7043	**0.8156**	2
DP-ADPC	2	0.3248	0.3599	0.5954	0.5	5	0.4887	0.5659	0.7184	0.5
DP-KNNDPC	5	0.6382	0.6738	0.7561	4	4.2	0.7246	**0.7493**	0.8086	6
DP-chDPC	1.8	0.7234	**0.7489**	**0.7935**	0.02, 0.04	1.6	**0.7351**	0.6746	0.7939	0.01, 0.03
	Iris					*Wine*				
DP-DPC	2.1	**0.6177**	0.6926	0.7429	3	2.2	0.6394	**0.6678**	**0.7351**	2
DP-ADPC	6	0.5606	**0.7039**	0.7447	0.1	6	0.4582	0.5912	0.7142	14.1
DP-KNNDPC	8	0.2681	0.3074	0.3885	6	-	0.1082	0.2973	0.4657	8
DP-chDPC	2.1	0.5393	0.6786	**0.7502**	0.02, 0.04	1.6	**0.6417**	0.6382	0.6739	0.01, 0.04
	Abalone					*Ecoli*				
DP-DPC	-	0.5713	0.7114	0.6972	2	0.4	0.3575	0.4273	0.6219	2
DP-ADPC	1	0.1950	0.1806	0.5291	2.9	4.5	**0.3781**	**0.4457**	0.6371	0.5
DP-KNNDPC	8	0.4862	0.5061	0.5349	5	8	0.1972	0.2085	0.2365	6
DP-chDPC	2.1	**0.8007**	**0.8186**	**0.8607**	0.02, 0.04	0.4	0.3573	0.4158	**0.6387**	0.01, 0.03

For the parameter ϵ, the DP-DPC algorithm does not reach the steady state on the Jain, Banknote, and Abalone datasets. the DP-ADPC algorithm does not reach the steady state on the Compound, Flame, and Jain datasets. the DP-KNNDPC

algorithm does not reach the steady state on the Flame, and Wine datasets. The DP-chDPC algorithm was able to achieve steady state on all datasets and took lower values, indicating that the algorithm is more stable than the other algorithms.

It can be seen that the clustering results of the DP-chDPC algorithm perform better on the low-dimensional dataset, and the clustering results on the high-dimensional dataset still need to be improved. However, the stability of DP-chDPC algorithm is higher than other algorithms.

V. CONCLUSION

In this paper, we propose a density peak clustering algorithm DP-chDPC for differential privacy preservation. The algorithm uses Chebyshev distance instead of Euclidean distance to calculate the distance matrix, which reduces the interference of noise on clustering, reduces the loss of clustering accuracy, and improves the stability of the algorithm. Through the comparative analysis with Euclidean distance and other algorithms, it can be seen that the DP-chDPC algorithm has good privacy preserving performance, and it can effectively reduce the loss of clustering accuracy after the algorithm adds noise, and improve the stability of the algorithm on both low-dimensional data sets and high-dimensional data sets. And the clustering results on low-dimensional data sets also perform better. However, the problem of poor adaptation of the DPC algorithm to high-dimensional data still needs to be studied, so how to improve the adaptation of the algorithm on high-dimensional data sets is the focus of future research, and its application to practical problems is also an important research direction.

REFERENCES

[1] I. K. Nti, J. A. Quarcoo, J. Aning, and G. K. Fosu, "A mini-review of machine learning in big data analytics: Applications, challenges, and prospects," *Big Data Mining Anal.*, vol. 5, no. 2, pp. 81–97, Jun. 2022.

[2] G. Czibula, G. Ciubotariu, M. Maier, and H. Lisei, "IntelliDaM: A machine learning-based framework for enhancing the performance of decision-making processes. A case study for educational data mining," *IEEE Access*, vol. 10, pp. 80651–80666, 2022.

[3] X. Zheng, L. Zhang, K. Li, and L. Xi, "Efficient publication of distributed and overlapping graph data under differential privacy," *Tsinghua Sci. Technol.*, vol. 27, no. 2, pp. 235–243, Apr. 2022.

[4] X. Pang, Z. Wang, D. Liu, J. C. S. Lui, Q. Wang, and J. Ren, "Towards personalized privacy-preserving truth discovery over crowdsourced data streams," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 327–340, Feb. 2022.

[5] B. Aslam, A. Maqsoom, A. H. Cheema, F. Ullah, A. Alharbi, and M. Imran, "Water quality management using hybrid machine learning and data mining algorithms: An indexing approach," *IEEE Access*, vol. 10, pp. 119692–119705, 2022.

[6] W. Liu, R. Yin, and P. Zhu, "Deep learning approach for sensor data prediction and sensor fault diagnosis in wind turbine blade," *IEEE Access*, vol. 10, pp. 117225–117234, 2022.

[7] F. Song, T. Ma, Y. Tian, and M. Al-Rodhaan, "A new method of privacy protection: Random k-anonymous," *IEEE Access*, vol. 7, pp. 75434–75445, 2019.

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

[9] L. Sun, G. Ping, and X. Ye, "PrivBV: Distance-aware encoding for distributed data with local differential privacy," *Tsinghua Sci. Technol.*, vol. 27, no. 2, pp. 412–421, Apr. 2022.

[10] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3628–3636, Aug. 2018.

[11] H. Jiang, J. Pei, D. Yu, J. Yu, B. Gong, and X. Cheng, "Applications of differential privacy in social network analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 108–127, Jan. 2023.

[12] Y. Wang, D. Wang, Y. Zhou, X. Zhang, and C. Quek, "VDPC: Variational density peak clustering algorithm," *Inf. Sci.*, vol. 621, pp. 627–651, Apr. 2023.

[13] Y. Wang, W. Pang, and J. Zhou, "An improved density peak clustering algorithm guided by pseudo labels," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109374.

[14] Y. Wang, W. Pang, and Z. Jiao, "An adaptive mutual K-nearest neighbors clustering algorithm based on maximizing mutual information," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109273.

[15] F. Li, M. Zhou, S. Li, and T. Yang, "A new density peak clustering algorithm based on cluster fusion strategy," *IEEE Access*, vol. 10, pp. 98034–98047, 2022.

[16] S. Ding, W. Du, X. Xu, T. Shi, Y. Wang, and C. Li, "An improved density peaks clustering algorithm based on natural neighbor with a merging strategy," *Inf. Sci.*, vol. 624, pp. 252–276, May 2023.

[17] Y. Zou and Z. Wang, "ConDPC: Data connectivity-based density peak clustering," *Appl. Sci.*, vol. 12, no. 24, p. 12812, Dec. 2022.

[18] L. Yin, Y. Wang, H. Chen, and W. Deng, "An improved density peak clustering algorithm for multi-density data," *Sensors*, vol. 22, no. 22, p. 8814, Nov. 2022.

[19] Y. Li, L. Sun, and Y. Tang, "DPC-FSC: An approach of fuzzy semantic cells to density peaks clustering," *Inf. Sci.*, vol. 616, pp. 88–107, Nov. 2022.

[20] M. Liu and Q. Wang, "Based on density reachable peak density clustering algorithm," *Comput. Simul.*, vol. 39, no. 11, pp. 371–375, 2022.

[21] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," *Pods*, vol. 5, pp. 128–138, Jan. 2005.

[22] W. Wu and H. Huang, "Research on DP-DBScan clustering algorithm based on differential privacy protection," *Comput. Eng. Sci.*, vol. 37, no. 4, pp. 830–834, 2015.

[23] 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.

[24] H. Wang, L. Ge, S. Wang, L. Wang, Y. Zhang, and J. Liang, "Improvement of differential privacy protection algorithm based on OPTIMS clustering," *Comput. Appl.*, vol. 38, no. 1, pp. 73–78, 2018.

[25] L. Sun, S. Bao, S. Ci, X. Zheng, L. Guo, and Y. Luo, "Differential privacy-preserving density peaks clustering based on shared near neighbors similarity," *IEEE Access*, vol. 7, pp. 89427–89440, 2019.

[26] Y. Chen, Y. Du, and X. Cao, "Density peak clustering algorithm based on differential privacy preserving," in *Science of Cyber Security*. Cham, Switzerland: Springer, 2019, pp. 20–32.

[27] H. Chen, Y. Zhou, K. Mei, N. Wang, and G. Cai, "A new density peak clustering algorithm with adaptive clustering center based on differential privacy," *IEEE Access*, vol. 11, pp. 1418–1431, 2023.

[28] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, Berlin, Germany: Springer, 2008, pp. 1–19.

[29] Y. Li, X. Chen, L. Liu, Y. An, and Z. Li, "Random forest algorithm for differential privacy protection," *Comput. Eng.*, vol. 46, no. 1, pp. 93–101, 2020.

[30] M. Bi, Y. Wang, Z. Cai, and X. Tong, "A privacy-preserving mechanism based on local differential privacy in edge computing," *China Commun.*, vol. 17, no. 9, pp. 50–65, Sep. 2020.

[31] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Aug. 2017.

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

[33] 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.

[34] C. Ren, L. Sun, Y. Yu, and Q. Wu, "Effective density peaks clustering algorithm based on the layered K-nearest neighbors and subcluster merging," *IEEE Access*, vol. 8, pp. 123449–123468, 2020.

[35] C. Wu, J. Lee, T. Isokawa, J. Yao, and Y. Xia, "Efficient clustering method based on density peaks with symmetric neighborhood relationship," *IEEE Access*, vol. 7, pp. 60684–60696, 2019.

[36] Y. Shi, Z. Yu, W. Cao, C. L. P. Chen, H. Wong, and G. Han, "Fast and effective active clustering ensemble based on density peak," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3593–3607, Aug. 2021.
[37] D. Huang, C. Wang, H. Peng, J. Lai, and C. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 508–520, Jan. 2021.
[38] Z. Liu, C. Wu, Q. Peng, J. Lee, and Y. Xia, "Local peaks-based clustering algorithm in symmetric neighborhood graph," *IEEE Access*, vol. 8, pp. 1600–1612, 2020.

HUA CHEN received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2012. Since 2015, she has been an Associate Professor with the School of Science, Hubei University of Technology. Her research interests include machine learning, information security, and cryptography.

KEHUI MEI received the B.E. degree in mathematics and applied mathematics from the School of Mathematics and Computer Science, Jianghan University, in 2020. He is currently pursuing the master's degree in applied statistics with the School of Science, Hubei University of Technology. His research interests include data mining and privacy protection.

YUAN ZHOU received the B.E. degree in applied statistics from the School of Science, Hubei University of Technology, in 2020, where she is currently pursuing the master's degree in applied statistics. Her research interests include data mining and privacy protection.

NAN WANG received the B.E. degree from the School of Science, Hubei University of Technology, in 2020, where she is currently pursuing the master's degree. Her research interests include data mining and machine learning.

MENGDI TANG received the B.E. degree from the School of Science, Inner Mongolia Agricultural University, in 2021. She is currently pursuing the master's degree in applied statistics with the School of Science, Hubei University of Technology. Her research interests include data mining and privacy protection.

GUANGXING CAI has been a Professor with the Hubei University of Technology for many years. His research interests include mathematics, information and coding, and cryptography.