Information Sciences 681 (2024) 121211

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Density-based clustering with differential privacy

Fuyu Wu, Mingjing Du *, Qiang Zhi

Jiangsu Key Laboratory of Educational Intelligent Technology, School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China

ARTICLE INFO

Keywords:
Density-based clustering
Differential privacy
Laplace mechanism
Data mining
Privacy budget allocation

ABSTRACT

In recent years, differentially private clustering has received increasing attention. However, most existing differentially private clustering algorithms cannot achieve better results when handling non-convex datasets. To enhance knowledge extraction from data while protecting users' sensitive information, we propose a density-based clustering algorithm with differential privacy. Specifically, we incorporate differential privacy mechanisms into the density-based clustering paradigm to enhance the effectiveness of differentially private clustering on non-convex datasets. Firstly, to avoid privacy leakage, we employ the Laplace mechanism for inject noise into the density during the density estimation stage. Then, we design a privacy budget allocation scheme in the cluster expansion stage to make it harder for attackers to access private information. Theoretical analysis demonstrates that our algorithm satisfies ϵ-differential privacy. Experimental outcomes in synthetic and real-world datasets show that our introduced algorithm can obtain high-quality clustering results when dealing with non-convex datasets. In the approximation experiments, it is evident that our algorithm outperforms others in terms of approximation.

1. Introduction

Clustering is a fundamental issue in unsupervised learning [1]. The objective of clustering involves partitioning the group of datasets such that points close to one another are grouped into the same cluster, whereas those that are distant are allocated to separate clusters. Clustering techniques have widespread applications in various areas, including market segmentation, recommender systems, and educational data mining, among others [2,3]. These applications often involve sensitive user information, which could be unintentionally disclosed by non-private clustering algorithms.

The growing concern about user privacy motivates exploring privacy-preserving algorithms [4,5]. Differential privacy (DP), a rigorous mathematical theory of privacy protection, has attracted sustained research interest in recent years. In practice, this technology is integrated into the core offerings of prominent companies such as Google, Apple, and Microsoft, among others [6]. Informally, the notion of differential privacy is that the algorithm's output should be mostly unchanged when any one input is changed, thereby making it difficult to infer any individual's sensitive information. Differentially private algorithms operate in two primary modes: local and central. In the local model, each user will randomize the data on a local level and then send the noisy data to the server. Conversely, the central model involves a trusted server to which users directly submit their data for analysis within the server's environment. These researches focus on the clustering problem based on central differential privacy (CDP) [7,8].

* Corresponding author.
E-mail addresses: fuyu@jsnu.edu.cn (F. Wu), dumj@jsnu.edu.cn (M. Du), zhi@jsnu.edu.cn (Q. Zhi).

https://doi.org/10.1016/j.ins.2024.121211
Received 15 April 2024; Received in revised form 10 July 2024; Accepted 18 July 2024

Available online 24 July 2024

Current research on differentially private clustering has garnered significant attention. Most of the existing studies focus on partition-based clustering algorithms. These algorithms have near-linear running time and obtain an approximation factor that is identical to the optimal non-private algorithms. The Ghazi algorithm [9] uses the DensestBall technique to reduce the approximation rate to the same as the optimal non-private clustering algorithm. The Vincent algorithm [10] uses the randomly shifted quadtrees technique to reduce the running time to close to the near-linear time $\tilde{O}(nkd)$. The Alisa algorithm [11] utilizes location-sensitive hashing techniques to reduce communications for local differential privacy models to once.

However, these partition-based private algorithms rely on iterative optimization, which can result in over-segmentation of the privacy budget if the iteration count becomes too high. This leads to high noise injection and degraded clustering performance. In addition, real-world datasets tend to be arbitrarily shaped and are called non-convex datasets. When dealing with non-convex datasets, partition-based clustering is not suitable for dealing with such datasets. These algorithms inherit the intrinsic limitations of the partition-based clustering paradigm, which can lead to erroneous clustering.

To solve the issues, we develop density-based clustering with differential privacy, named DBDP. First, in the density estimation phase, we add some noise into density by using the Laplace mechanism. Then, in the cluster expansion phase, we design a new privacy budget assignment solution, which cause the privacy budget to decay exponentially.

The major contributions of this article are as follows.

- We propose a new algorithm for density-based clustering with differential privacy. It can detect clusters with non-convex shapes while ensuring privacy preservation.
- We integrate the Laplace mechanism into the density estimation phase of the original algorithm. Meanwhile, we devise a new privacy budget assignment solution. These techniques endow the proposed algorithm with privacy-preserving capabilities.
- We theoretically prove that DBDP satisfies ϵ-differential privacy. In addition, the results of our experiments not only demonstrate the efficacy of our algorithm but also verify a high approximation between it and the original algorithm.

The following structure is adopted for the subsequent sections of this paper. In Section 2, we briefly summarize related research. In Section 3, preliminary details are provided on differential privacy and DBSCAN clustering. Section 4 delineates the proposed algorithm and its comprehensive analysis. Section 5 shows results from experiments with synthetic datasets and real-world datasets. In Section 6, our conclusion is presented.

2. Previous work

The research on differential privacy in clustering algorithms is widespread [12–19]. Currently, the majority of studies concentrate on the partition-based clustering paradigm.

There are two main directions in partition-based private clustering. The first direction involves improving classical k-means algorithms for satisfying definitions of differential privacy. To address the issue of privacy leakage in k-means, the SuLQ framework is presented by Blum et al. [20]. This method introduces the Laplace mechanism [21] in differential privacy, where noise is added to the center point at each iteration to satisfy differential privacy. However, the SuLQ framework suffers from the problem of privacy budget over-allocation. To solve this problem, Su et al. [22] propose the EUKGM algorithm. This algorithm introduces the idea of approximating the initial center point to achieve the purpose of reducing the number of iterations. Although these algorithms use differential privacy budget allocation schemes, the size of the increased noise is still determined by the algorithm's iteration count. Since the iteration count is unpredictable, the injected noise increases with the number of iterations, which ultimately leads to a degradation of the clustering quality.

The second direction is to use a one-cluster-like approach to build private coresets [23–25]. Coresets refer to selecting n representative points in the whole dataset to represent the whole dataset. Feldman et al. [26] introduce a differential privacy concept based on coresets and propose a private coresets approach. However, the method suffers from poor clustering in high-dimensional Euclidean space. In order to solve a problem, Balcan et al. [27] introduce the Johnson-Lindenstrauss (JL) theorem to reduce the dimension of the dataset to $O(logn)$. In addition, they construct private coresets using a hyper-rectangular partitioning method to optimize the approximation ratio of the algorithm. However, the method still suffers from excessive multiplication error in the approximation ratio [28,29]. To solve this problem, Ghazi et al. [9] introduce the DensestBall technique to reduce the multiplication error of the algorithm to the constant level. To address the issue of the high time complexity of DensestBall when dealing with real data, Vincent et al. [10] introduce the hierarchically separated tree (HST) method. Their work makes the time complexity of the algorithm close to linear time. Although these studies make significant breakthroughs in the approximation ratio and time complexity of the algorithm, they still do not give good clustering results when dealing with non-convex datasets.

Density-based private clustering has also seen a small amount of research work recently. Wu et al. [30] propose the DP_DBSCAN algorithm, which adds Laplace noise in the distance metric. Ni et al. [31] introduce the DP_MCDBSCAN technique, building upon the foundation of DP_DBSCAN. This method selects multiple initial core objects for clustering using the farthest distance method to mitigate the effects of random selection on clustering outcomes. Although differentially private density clustering has made significant strides, such algorithms add Laplace noise to the distance metric. This approach introduces inaccuracies in the distance metric between data points, which can lead to unsatisfactory clustering results. In contrast, our method adds Laplace noise to the density metric stage for the purpose of privacy data protection.

Table 1
Notations.

Notations	Descriptions
$\mathbf{X}$	Dataset of points
$\mathbf{D}$	Domain of datasets
n	Number of data points
d	Dimension of the dataset
$\mathbf{x}_i$	i-th data point in the dataset
eps	Distance threshold
$minPts$	Density threshold within a given radius
dist($\mathbf{x}$,$\mathbf{y}$)	Distance between data points $\mathbf{x}$ and $\mathbf{y}$
ϵ	Privacy budget
Δf	Global sensitivity
$\mathbb{C}$	Clustering results
$\mathbf{C}_k$	k-th cluster

3. Preliminaries

In this section, we introduce the concepts of differential privacy and DBSCAN. The notations used throughout this paper are delineated in Table 1 for convenient reference.

3.1. Differential privacy

We then recall a definition and a fundamental property of differential privacy [32]. Datasets $\mathbf{X}$ and $\mathbf{X}'$ are considered neighbors when $\mathbf{X}'$ is obtained by either removing or adding a single data point from $\mathbf{X}$.

Definition 1. Differential Privacy (DP). Let $\epsilon, \delta \in \mathbb{R}_{\geq 0}$ and $n \in \mathbb{N}$. A randomized algorithm $\mathcal{A}$ taking as input a dataset is said to be (ϵ, δ)-differentially private if for any two neighboring datasets $\mathbf{X}$ and $\mathbf{X}'$, and for any subset S of outputs of $\mathcal{A}$, the following condition holds:

$$Pr[\mathcal{A}(\mathbf{X}) \in S] \leq e^{\epsilon} \cdot Pr[\mathcal{A}(\mathbf{X}') \in S] + \delta \tag{1}$$

where $Pr[\cdot]$ refers to the probability. It represents the likelihood of an event occurring. The parameters ϵ and δ govern the algorithm's privacy guarantee. If $\delta = 0$, then $\mathcal{A}$ is said to be ϵ-differential privacy. We assume throughout that $0 < \epsilon \leq O(1)$, and when used, $\delta > 0$.

The differential privacy mechanism is based on the sensitivity of the query function to evaluate the amount of added noise. The definition of sensitivity is as follows.

Definition 2. Sensitivity. Global sensitivity [6] is the nature of the query function f itself, regardless of the size of the dataset. Sensitivity refers to the maximum change for the query results by deleting any records in the dataset. For the query function $f : \mathbf{D} \rightarrow \mathbb{R}^d$, the sensitivity of f is defined as follows:

$$\Delta f = \max_{\mathbf{X},\mathbf{X}'} ||f(\mathbf{X}) - f(\mathbf{X}')||_1 \tag{2}$$

In differential privacy, Dwork [32] proposes the Laplace mechanism to achieve ϵ-differential privacy protection by adding random noise following the Laplace distribution to query results.

Theorem 1. *Laplace mechanism. A random variable is distributed as Lap(**y**) if its probability density function is* $Lap(\mathbf{y}) = \frac{1}{2\lambda} exp(-\frac{|\mathbf{y}|}{\lambda})$. *Let* $\epsilon > 0$, *and assume* $f : \mathbf{D} \rightarrow \mathbb{R}^d$ *has sensitivity* Δf. *The mechanism with input* $\mathrm{S} \in \mathbf{X}$ *and output* $f(\mathrm{S}) + (Lap(\frac{\Delta f}{\epsilon}))^d$ *is* $(\epsilon, 0)$*-differential privacy.*

To solve complex privacy protection problems, differentially private algorithms are used. The privacy budget ϵ should be assigned to every step of this algorithm in order to control the overall level of privacy. Two compositional properties of differential privacy are sequential composability and parallel composability.

(i) Sequential composability: The algorithm $\mathcal{A}$ applies successively two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, which are respectively ϵ_1-DP and ϵ_2-DP. The resulting algorithm $\mathcal{A}$ is $(\epsilon_1 + \epsilon_2)$-DP. If $\mathcal{A}_1$ and $\mathcal{A}_2$ run on two distinct parts of the dataset, then $\mathcal{A}$ is $\max(\epsilon_1, \epsilon_2)$-DP.
(ii) Parallel composability: If $\mathcal{A}$: $\mathbf{D}_1 \times \mathbf{D}_2 \rightarrow \mathbf{Z}$ satisfies that for all $\mathbf{X} \in \mathbf{D_1}$, the algorithm $\mathcal{A}(\mathbf{X}, \cdot)$ is ϵ_1-DP, and some algorithm $\mathcal{B} : \mathbf{D}_2 \rightarrow \mathbf{D}_1$ is ϵ_2-DP, then the algorithm $X \rightarrow \mathcal{A}(\mathcal{B}(\mathbf{X}), \mathbf{X})$ is $(\epsilon_1 + \epsilon_2)$-DP.

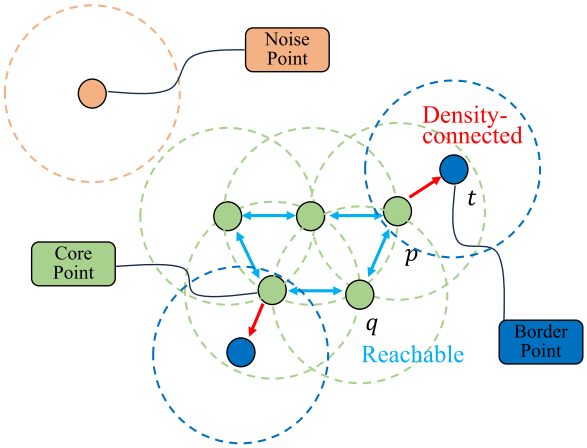

Fig. 1. Illustration of a few concepts in DBSCAN.

3.2. DBSCAN algorithm

Partition-based clustering and model-based clustering algorithms inherently generate spherical clusters, rendering renders them inadequate for handling clusters with arbitrary shapes. Additionally, these algorithms require the a priori determination of the number of clusters, limiting their flexibility and adaptability to varying datasets. Although hierarchical clustering algorithms can potentially identify clusters with various shapes, their flexibility in this regard is still somewhat limited. Furthermore, hierarchical clustering algorithms is inherently sensitive to outliers since they relies on distance metrics to construct the clustering hierarchy. Density-based clustering algorithms such as DBSCAN have become increasingly popular [33,34]. It could recognize the arbitrarily formed clusters and those with noise [35,36] (i.e., outliers). DBSCAN relies on two crucial parameters.

(i) The parameter eps defines a radius, where two points are neighbors if their distance is less than or equal to eps.
(ii) The parameter $minPts$ sets the minimum number of neighbors within a specified radius.

Based on these parameters, DBSCAN establishes a set of criteria. [37,38]:

- **Core point**: A point qualifies as a core point if it has a radius of eps and is surrounded by at least $minPts$ points (inclusive of itself).
- **Reachability**: A point $\mathbf{x}_q$ is directly reachable from $\mathbf{x}_p$ if the distance between the point $\mathbf{x}_q$ and the core point $\mathbf{x}_p$ is within eps.
- **Density-connectedness**: Two points $\mathbf{x}_p$ and $\mathbf{x}_q$ exhibit density connectedness if there exists a direct or transitive path from $\mathbf{x}_p$ to $\mathbf{x}_q$.
- **Border point**: A point is a border point if it is reachable from a core point and the area around it has fewer points than $minPts$.
- **Noise point**: If a point is neither a core point nor a point that can be reached from a core point, then that point is a noise point.

To facilitate understanding of the above concepts, we provide an example in Fig. 1. We set $minPts$ to 3. Core points, depicted in green, possess a minimum of three neighboring points in a radius of eps. This area is shown with the green circles in the figure. The blue points are border points because they are reachable from a core point and have fewer than 3 points within their neighborhood. The yellow point is classified as a noise point since it lies outside the core and is unreachable from all core points. The blue double arrows signify that the data points at each end are directly reachable; for instance, q can be reached directly from p. A single red arrow denotes that the data points are density-connected, as in the case of p being directly reachable from t, and q being transitively reachable from t.

4. Differentially private DBSCAN

One of the crucial techniques in density-based methods is DBSCAN [39,40]. In DBSCAN, clusters can be assumed as exhibiting connectivity within areas of high density, which are separated by areas of low density. Hence, an attacker can exploit the information on radius and density to deduce members' details, thereby compromising privacy. We have modified this approach to incorporate privacy considerations as follows.

4.1. Overview

As shown in Fig. 2, DBDP includes three stages: density estimation, core point identification, and cluster expansion. Below is a presentation of each step in the overview.

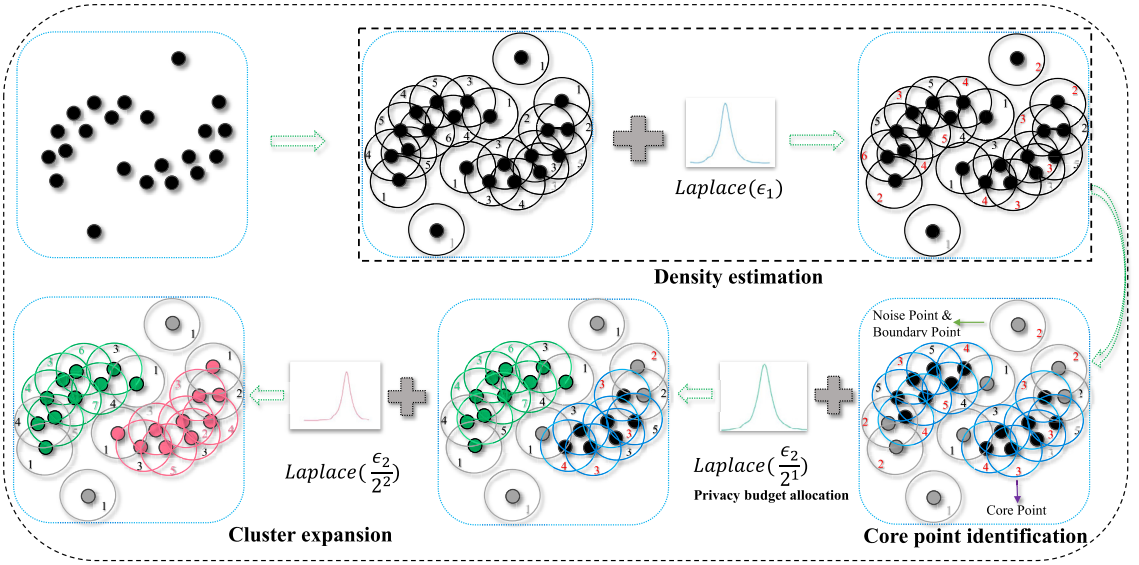

Fig. 2. The architecture of DBDP.

Density estimation: To construct a differentially private DBSCAN, the DBDP algorithm initially sets some parameters, such as the distance threshold (eps), the smallest neighborhood number ($minPts$), and the privacy budget (ϵ). Next, the DBDP algorithm counts points in the user-defined radius eps as the density of each data point. Finally, the DBDP algorithm utilizes the Laplace mechanism to add Laplace noise to the density values of each point.

Core point identification: In this step, we identify core points based on the density obtained from the added Laplace noise, $Lap(\frac{\Delta f}{\epsilon_1})$. It is important to note that at this stage, in addition to core points, there are also non-core points that include both the final noise points and boundary points. Specifically, a point is classified as a core point if its density, after adding Laplace noise, exceeds the threshold $minPts$.

Cluster expansion: In this step, the DBDP algorithm connects all the core points with reachability to expand clusters. During the division of clusters, we give different privacy budget levels, to achieve privacy protection.

4.2. Algorithm

To overcome the limitations of differentially private partition-based clustering, we propose a DBDP algorithm. Our algorithm runs in polynomial-time private clustering algorithm, and can effectively handle non-convex datasets while maintaining high precision in the clustering results.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^d$ be a dataset consisting of n data points, where each data point has d features, i.e., $\mathbf{x}_i = x_{i1}, x_{i2}, ..., x_{id}$.

In the dataset $\mathbf{X}$ with the d-dimensional space, for two points $\mathbf{x}_i$ and $\mathbf{x}_j$, we let $\text{dist}(\mathbf{x}_i, \mathbf{x}_j) := ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{\sum_{l=1}^{d}(x_{il} - x_{jl})^2}$ be the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. We calculate the Euclidean distance for every pair of points.

In order to provide a clearer description of the density estimation and the core point identification in the proposed algorithm, we give the following definitions.

Definition 3. Density. Let $eps > 0$ and $\mathbf{x}_i \in \mathbf{X}$. The density of $\mathbf{x}_i$ is:

$$Den_{NS}(\mathbf{x}_i) = |\{\mathbf{x}_j \in \mathbf{X} | \text{dist}(\mathbf{x}_i, \mathbf{x}_j) \leq eps\}| + Lap(\frac{\Delta f}{\epsilon_1}) \tag{3}$$

Definition 4. Core point. A point $\mathbf{x}_i$ is a core point if its density is greater than or equal to the density threshold $minPts$, i.e. $Den_{NS}(\mathbf{x}_i) \geq minPts$. Conversely, it is a non-core point, including noise points and boundary points.

Algorithm 1 summarizes the DBDP algorithm.

We randomly select a point $\mathbf{x}_i$ from the dataset $\mathbf{X}$. Then, we calculate its eps-neighborhood as neighborPts which includes all other points whose Euclidean distance from $\mathbf{x}_i$ is less than or equal to eps. Then we use the Laplace mechanism to add noise $Lap(\frac{\Delta f}{\epsilon_1})$ to the size of neighborPts as $Den_{NS}(\mathbf{x}_i)$. If $minPts > Den_{NS}(\mathbf{x}_i)$, this point $\mathbf{x}_i$ will be considered as a non-core point (Algorithm 1, lines 2-7); otherwise, core point to be considered. If the point is core point, then we call the cluster expansion(Algorithm 1, lines 8-19).

We assign the point $\mathbf{x}_i$ to the cluster $\mathbf{C}_k$ (the k-th cluster) and randomly choose a point $\mathbf{x}_i'$ from the eps-neighborhood of $\mathbf{x}_i$. The eps-neighborhood of $\mathbf{x}_i'$, denoted as neighborPts′, is determined. We apply the Laplacian mechanism to add noise, calculated as $Lap(\frac{2^k \Delta f}{\epsilon_2})$,

to the number of data points in neighborPts′ and denote it as $Num_{cLap}(\mathbf{x}'_i)$. (Algorithm 1, lines 11-15). If $minPts \leq Num_{cLap}(\mathbf{x}'_i)$, this point $\mathbf{x}'_i$ will be considered a core point and neighborPts′ will be merged into neighborPts. This process continues until no more points are in neighborPts (Algorithm 1, lines 16-17).

The privacy budget ϵ allocation is important. We divide ϵ into ϵ_1 and ϵ_2, and ϵ_2 is subsequently divided into $\frac{\epsilon_2}{2^1}, \frac{\epsilon_2}{2^2}, \ldots, \frac{\epsilon_2}{2^k}$. The privacy budget ϵ_1 is used for distinguishing core and non-core points. The privacy budget ϵ_2 is used for cluster expansion. Specifically, in the cluster expansion stage, each class is assigned a certain privacy budget. As shown in Fig. 2, the privacy budget allocated to the first class is $\frac{\epsilon_2}{2^1}$, and the second class is assigned a privacy budget of $\frac{\epsilon_2}{2^2}$. Applying varying privacy budgets results in different levels of noise being introduced. This method prevents the accumulation of privacy leakage through multiple queries, thereby enhancing privacy.

Algorithm 1 DBDP algorithm.

Input: $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, *eps*, $MinPts$, ϵ
Output: $\mathbb{C} = \{\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_k\}$
$k = 1$, $\epsilon = \epsilon_1 + \epsilon_2$
for *each unvisited point* $\mathbf{x}_i$ *in dataset* $\mathbf{X}$ **do**
  mark $\mathbf{x}_i$ as visited
  $NeighborPts$ = all points within $\mathbf{x}_i$ *eps*-neighborhood
  $Den_{NS}(\mathbf{x}_i) = \text{sizeof}(NeighborPts) + \text{Lap}(\frac{\Delta f}{\epsilon_1})$
  if $Den_{NS}(\mathbf{x}_i) < minPts$ **then**
    mark $\mathbf{x}_i$ as non-core
  else
    $k = k+1$
    add $\mathbf{x}_i$ to cluster $\mathbf{C}_k$
    for *each point* $\mathbf{x}'_i$ *in* $NeighborPts$ **do**
      if $\mathbf{x}'_i$ *is not visited* **then**
        mark $\mathbf{x}'_i$ as visited
        $NeighborPts'$ = all points within $\mathbf{x}'_i$ *eps*-neighborhood
        $Den_{NS}(\mathbf{x}'_i) = \text{sizeof}(NeighborPts') + \text{Lap}(\frac{\Delta f \cdot 2^k}{\epsilon_2})$
        if $Den_{NS}(\mathbf{x}'_i) \geq minPts$ **then**
          $NeighborPts = NeighborPts \cup NeighborPts'$
      if $\mathbf{x}'_i$ *is not yet member of any cluster* **then**
        add $\mathbf{x}'_i$ to cluster $\mathbf{C}_k$

4.3. Privacy and analysis of DBDP

This subsection offers theoretical validations supporting DBDP's privacy and practicality guarantees.

Theorem 2. *The global sensitivity of DBDP is* $\Delta f = 1$.

Proof. In the DBDP framework, the query function is designed to count the number of points within the *eps*-neighborhood of a given point, denoted as f : *eps* neighborhood query. Consider datasets $\mathbf{X}$ and $\mathbf{X}'$ that makes a difference of one point $\mathbf{x}_t$, i.e., $\mathbf{X} = \mathbf{X}' \cup \{\mathbf{x}_t\}$. Without loss of generality, let $\mathbf{x}_i$ and $\mathbf{x}_j$ are point common to both datasets, i.e., $\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathbf{X} \cap \mathbf{X}'$. The point $\mathbf{x}_t$ is an *eps* neighbor of $\mathbf{x}_i$ and $\mathbf{x}_t$ is not an *eps* neighbor of $\mathbf{x}_j$. When an *eps*-neighborhood query is conducted for $\mathbf{x}_i$, the discrepancy in the query results is 1, i.e., $||f_{\mathbf{X}}(\mathbf{x}_i) - f_{\mathbf{X}'}(\mathbf{x}_i)|| = 1$. When an *eps*-neighborhood query is conducted for $\mathbf{x}_j$, and the discrepancy in the query results is 0, i.e., $||f_{\mathbf{X}}(\mathbf{x}_j) - f_{\mathbf{X}'}(\mathbf{x}_j)|| = 0$. For all points, the query result gap is either 1 or 0, i.e., the largest gap in query results across all data points is 1. Consequently, from the definition of sensitivity, the global sensitivity can be derived as $\Delta f = 1$.

We demonstrate that our proposed DBDP satisfies differential privacy in the following Theorem 3.

Theorem 3. *The algorithm DBDP satisfies* $m\epsilon$*-DP.*

Proof. The DBDP algorithm achieves differential privacy by employing the Laplace mechanism in two distinct stages. Firstly, it satisfies $m\epsilon_1$-DP during the core point identification stage. Secondly, $m\epsilon_2$-DP is maintained in the cluster expansion stage.

Consider datasets $\mathbf{X}$ and $\mathbf{X}'$ such that $\mathbf{X} = \mathbf{X}' \cup \{\mathbf{x}_t\}$. Discrepancies between two neighboring datasets can impact the density estimation (or *eps*-neighborhood queries) of m data points, where m represents the maximum density within the dataset. We first provide the proof of this assertion.

Without loss of generality, suppose that the data point $\mathbf{x}_1$ can influence the *eps*-neighborhood of h points, indicating that the point resides within the *eps* radius of these h points. Due to the symmetry of the *eps*-neighborhoods, we infer that the *eps*-neighborhood of $\mathbf{x}_1$ includes these h points, thus having a density value of h. In a dataset characterized by a maximum density of m, the removal of a data point could potentially affect the *eps*-neighborhood outcomes for up to m points.

The above conclusion indicates that the density impact on the entire dataset is equivalent to the density impact on the m data points, as expressed mathematically below.

$$\frac{Pr[M(\mathbf{X})=\mathbf{S}]}{Pr[M(\mathbf{X}')=\mathbf{S}]}=\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_1)=S_1]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_1)=S_1]}\wedge\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_2)=S_2]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_2)=S_2]}\wedge \cdots\wedge\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_m)=S_m]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_m)=S_m]}\wedge\cdots\wedge\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_n)=S_n]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_n)=S_n]} \tag{4}$$

where $Pr[\cdot]$ represents the probability density function. The output results of the density estimation for $\mathbf{X}$ and $\mathbf{X}'$ are denoted by $M[\mathbf{X}]$ and $M[\mathbf{X}']$, respectively. Additionally, $\mathbf{S}$ represents any possible density, with $f_{\mathbf{X}}(\cdot)$ being the true density function.

Next, we need to give an upper bound for Eq. (4). Without loss of generality, it can start by solving for the upper bound of one of the terms $\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_1)=S_1]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_1)=S_1]}$. The derivation is as follows.

$$\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_1)=S_1]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_1)=S_1]}=\frac{\frac{\epsilon_1}{2\Delta f}exp(-\frac{\epsilon_1||S_1-f_{\mathbf{X}}(\mathbf{x}_1)||}{\Delta f})}{\frac{\epsilon_1}{2\Delta f}exp(-\frac{\epsilon_1||S_1-f_{\mathbf{X}'}(\mathbf{x}_1)||}{\Delta f})} \tag{5}$$

$$=exp(\frac{\epsilon_1(||S_1-f_{\mathbf{X}}(\mathbf{x}_1)||-||S_1-f_{\mathbf{X}'}(\mathbf{x}_1)||)}{\Delta f}) \tag{6}$$

$$\leq exp(\frac{\epsilon_1(||f_{\mathbf{X}}(\mathbf{x}_1)-f_{\mathbf{X}'}(\mathbf{x}_1)||)}{\Delta f}) \tag{7}$$

$$\leq e^{\epsilon_1} \tag{8}$$

Eq. (5) is using $Pr[M_{\mathbf{X}}(\mathbf{x}_1)=S_1]=Pr[Lap(\Delta f/\epsilon_1)=S-f_{\mathbf{X}}(\mathbf{x}_1)]$ and for the same reason $Pr[M_{\mathbf{X}'}(\mathbf{x}_1)=S_1]=Pr[Lap(\Delta f/\epsilon_1)=S-f_{\mathbf{X}'}(\mathbf{x}_1)]$; Eq. (6) utilizes the power function property; Eq. (7) is using the properties of trigonometric inequalities; Eq. (8) is using the definition of global sensitivity: $\Delta f=\max_{\mathbf{X},\mathbf{X}'}||f(\mathbf{X})-f(\mathbf{X}')||\geq||f_{\mathbf{X}}(\mathbf{x}_1)-f_{\mathbf{X}'}(\mathbf{x}_1)||$.

From Eqs. (5)-(8), it can be inferred that any point $\mathbf{x}_i\in\{\mathbf{x}_1\ldots\mathbf{x}_m\}$ satisfies $\frac{Pr[M_{\mathbf{X}}(\mathbf{x}_i)=S_i]}{Pr[M_{\mathbf{X}'}(\mathbf{x}_i)=S_i]}\leq e^{\epsilon_1}$. Bringing this into Eq. (4) gives the following result.

$$\frac{Pr[M(\mathbf{X})=\mathbf{S}]}{Pr[M(\mathbf{X}')=\mathbf{S}]}\leq\underbrace{e^{\epsilon_1}*e^{\epsilon_1}\cdots*e^{\epsilon_1}}_{m}*\underbrace{e^0*\cdots*e^0}_{n-m}=e^{m\epsilon_1} \tag{9}$$

Therefore, the core point identification stage satisfies $m\epsilon_1$-DP.

Next, we give a proof of the cluster expansion stage. Given that we employ the Laplace mechanism for density computation in this stage, the proof process remains largely unchanged. The main difference lies in the assignment of the privacy budget, which is distributed across each cluster $\frac{\epsilon_2}{2^1},\frac{\epsilon_2}{2^2},\ldots,\frac{\epsilon_2}{2^k}$, where k is the number of clusters. This allocation ensures that each cluster satisfies $m(\epsilon_2/2^k)$-DP, with the aggregate across all clusters not exceeding $m\epsilon_2$ differential privacy.

In summary, the DBDP algorithm preserves $m(\epsilon_1+\epsilon_2)$-DP, $\epsilon=\epsilon_1+\epsilon_2$. Therefore, we conclude that the DBDP algorithm preserves $m\epsilon$-DP.

Theorem 4. *The algorithm DBDP has a time complexity of $O(nlog(n))$, where n represents the number of data points.*

Proof. In practical situations, datasets are usually non-uniformly distributed, so the average time complexity of DBSCAN is often close to $O(nlog(n))$, which is more efficient than the worst-case scenario($O(n^2)$). The algorithm DBDP uses the Laplace mechanism to satisfy differential privacy, which has a time complexity $O(1)$, so the time complexity for DBDP is $O(nlog(n))$.

5. Experimental results

We investigate our algorithm's performance as well as examine the privacy budget's impact by comparing it to four private and one non-private clustering algorithms across eight datasets.

5.1. Experiment setup

Datasets. We evaluate performance using five synthetic datasets and five real-world datasets. The synthetic datasets include Moons, T0, Yinyang, T4, and T7. The real-world datasets include Haberman, Wine, Pageblocks, Penbased, and Htru2. These datasets are from the baseline clustering datasets[1] and UCI Machine Learning Repository.[2] Table 2 provides details of these datasets.

[1] https://github.com/milaan9/Clustering-Datasets.
[2] https://archive.ics.uci.edu/.

Table 2
The dataset utilized in the experiment.

Name	#Instance	#Feature	#Class
Moons	1000	2	2
T0	2000	2	3
Yinyang	3200	2	5
T4	7326	2	6
T7	9208	2	9
Haberman	306	3	2
Wine	175	13	3
Pageblocks	5473	10	5
Penbased	10992	16	10
Htru2	17898	8	2

Baselines. *Non-private baseline.* We compare the outcomes of DBDP with those of a non-private DBSCAN, utilizing the implementation provided by Python's scikit-learn library. We use the default parameter settings and run each dataset multiple times to obtain optimal performance.

Private baseline. To our knowledge, the majority of differentially private clustering algorithms are fundamentally based on the k-means paradigm. We performed a comparison of our algorithm with three private baselines: Cohen-Addad [10], Balcan [27], and PrivKmeans [22]. In addition, we also compare our algorithm with the DP_DBSCAN [30] algorithm, which is based on density paradigm.

- The Cohen-Addad algorithm partitions the data points according to their LSH outputs, generates differentially private k-means, counts for each partition, and then runs a (non-private) k-means algorithm on the means with the counts as weights.
- The Balcan algorithm suggests first to selecting a small subset of candidate centers, which includes candidate centers with minimal k-mean loss. Subsequently, a non-private clustering algorithm is executed for these candidate centers.
- The PrivKmeans algorithm uses Laplace or Gaussian noise in Lloyd's iteration.
- The DP_DBSCAN algorithm adds Laplace noise in the distance metric to satisfy differential privacy.

Evaluation Metrics. In the experiment, we select the Micro-F1 Score (F1), Normalized Mutual Information (NMI), and the Fowlkes–Mallows Index (FMI) as the evaluation criteria. F1, NMI, and FMI are widely used in clustering analysis. These evaluation metrics range from 0 to 1, with higher values indicating better performance. The evaluation of F1 is defined as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

where $Precision$ is the percentage of samples classified into specific category that truly within this category and $Recall$ represents the portion of actual instances belonging to a certain class that are correctly identified.

The evaluation of NMI is defined as:

$$NMI = \frac{I(\mathbf{L}, \mathbf{P})}{\sqrt{H(\mathbf{L})H(\mathbf{P})}} \tag{11}$$

where $\mathbf{L}$ and $\mathbf{P}$ represent the true and predicted category labels. $H(\mathbf{L})$ and $H(\mathbf{P})$ are their entropies respectively, and $I(\mathbf{L}, \mathbf{P})$ is their cross entropy.

The evaluation of FMI is defined as:

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \tag{12}$$

where TP represents the number of sample pairs that are actually in a category and predicted to be in a category, FN represents the number of sample pairs that are actually in a category but predicted not to be in a category, FP represents the number of sample pairs that are not in a category but predicted in a category, and FN represents the number of sample pairs that are not actually in a category and predicted not to be in a category.

5.2. Results on all datasets

These algorithms have several parameters. The specific range of values for each parameter is displayed in Table 3. eps is the distance threshold. $minPts$ represents the smallest count of neighbors within a specified radius. K is the amount of clusters and we set K to the true number of classes.

On the density-based clustering paradigm, we select the optimal results within a specified parameter range and execute the algorithm multiple times to compute the average result. On the partition-based clustering paradigm, we set the parameter k to match the true number of classes. We set the privacy budget ϵ in private algorithms to 10. We normalize all datasets. Based on Figs. 3-7, our algorithm demonstrates the capability to detect clusters of varying sizes and shapes.

Table 3
Parameter settings.

Algorithm	Parameters
DBDP	*eps: [0.01:0.01:2.0]; minPts: [2:1:200]*
DBSCAN	*eps: [0.01:0.01:2.0]; minPts: [2:1:200]*
DP_DBSCAN	*eps: [0.01:0.001:2.0]; minPts: [2:1:200]*
Cohen-Addad	*K: True number of classes*
Balcan	*K: True number of classes*
PrivKmeans	*K: True number of classes*

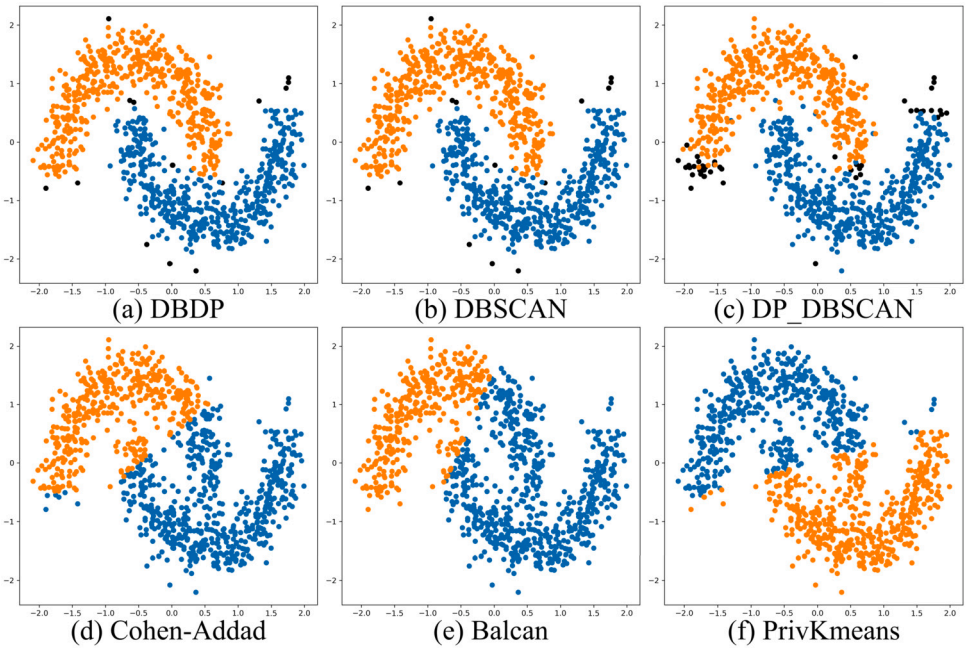

Fig. 3. Clustering results on Moons.

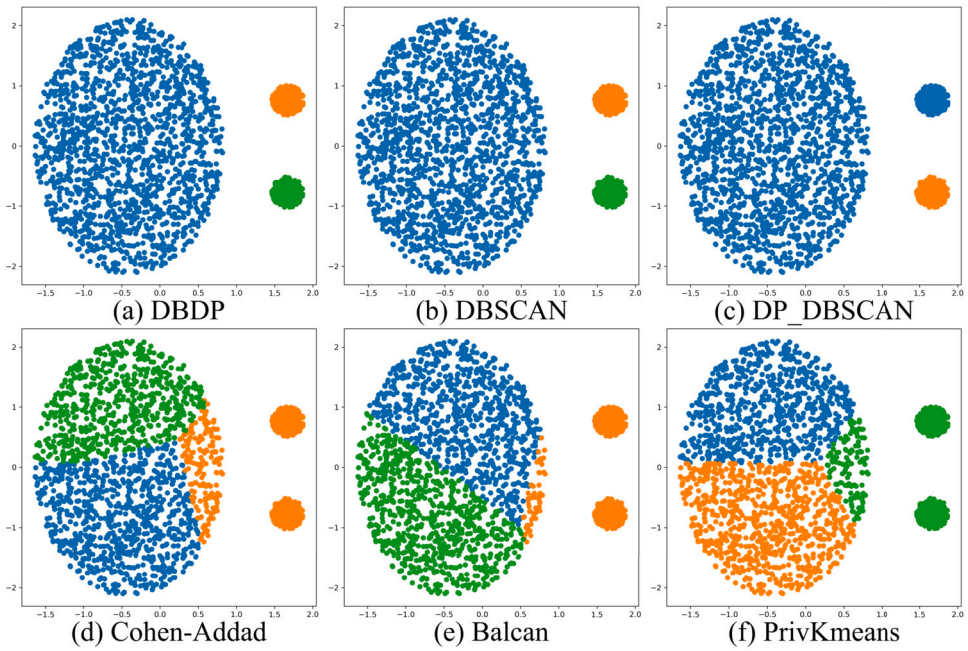

Fig. 4. Clustering results on T0.

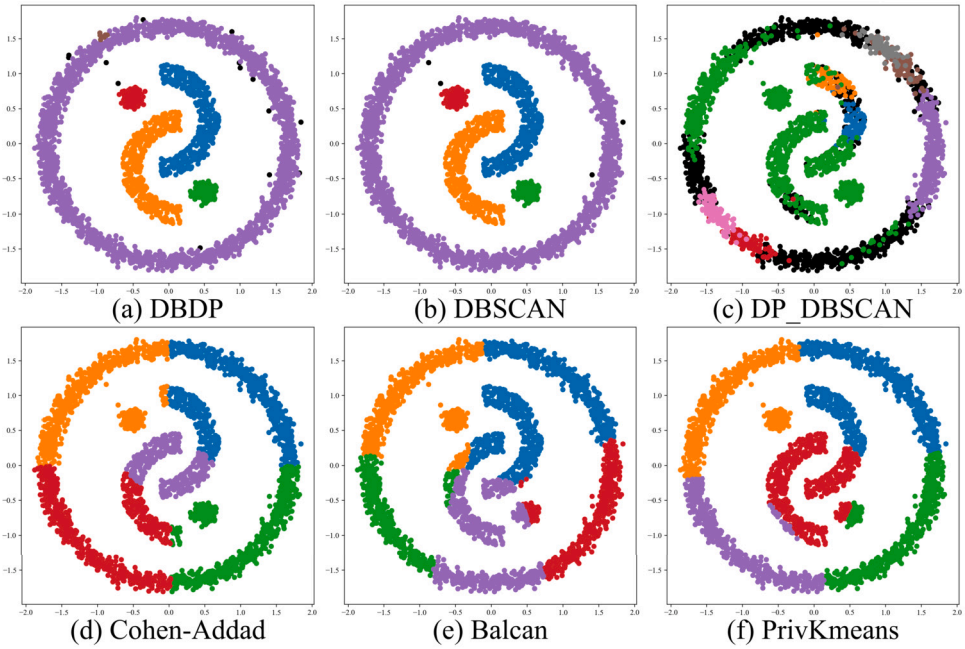

Fig. 5. Clustering results on Yinyang.

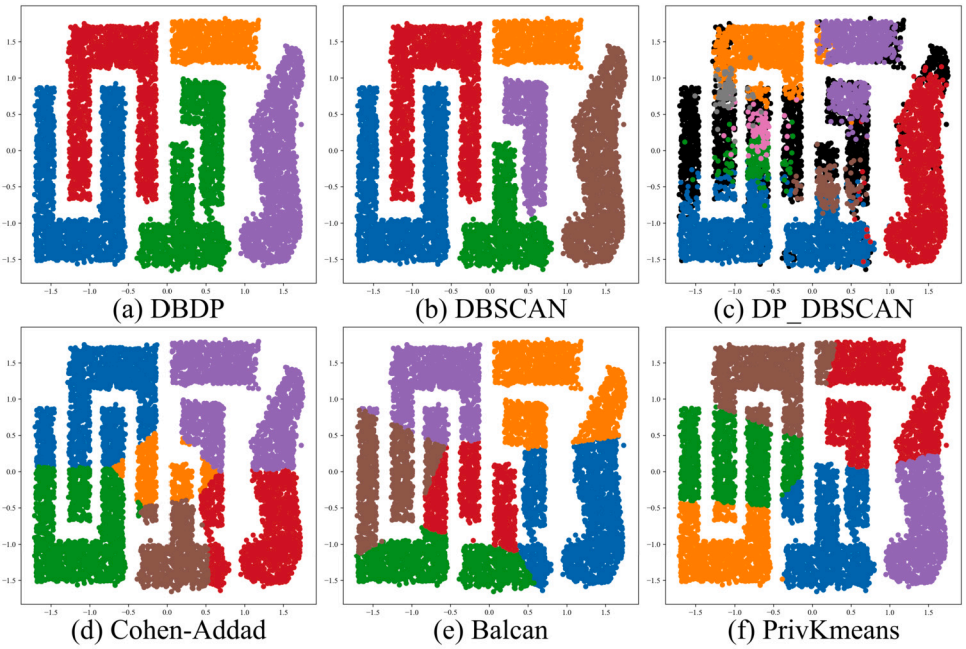

Fig. 6. Clustering results on T4.

In Fig. 3, the data structure is composed of two crescent-shaped clusters with different densities. DP_DBSCAN can recognize clusters with arbitrary shapes, but some points are incorrectly classified as noise points. Cohen-Addad, Balcan, and PrivKmeans are based on the partition-based clustering paradigm, which cannot handle arbitrarily shaped datasets well, such as crescent-shaped ones. Therefore, they cluster the dataset incorrectly into two spherical-like classes in the final result. Whereas, DBDP is based on the density-based clustering paradigm, which can handle arbitrarily shaped datasets well and adaptively determines the number of clusters. The DBDP algorithm ultimately achieves highly consistent clustering results with DBSCAN while ensuring privacy.

In Fig. 4, the T0 dataset consists of three circular clusters. Notably, this dataset is imbalanced, with the majority of data concentrated in one class and only a small portion distributed across the other two classes. As shown in the first row of Fig. 4, the three density-based clustering algorithms correctly identify the overall structure of all clusters and produce accurate clustering results. In

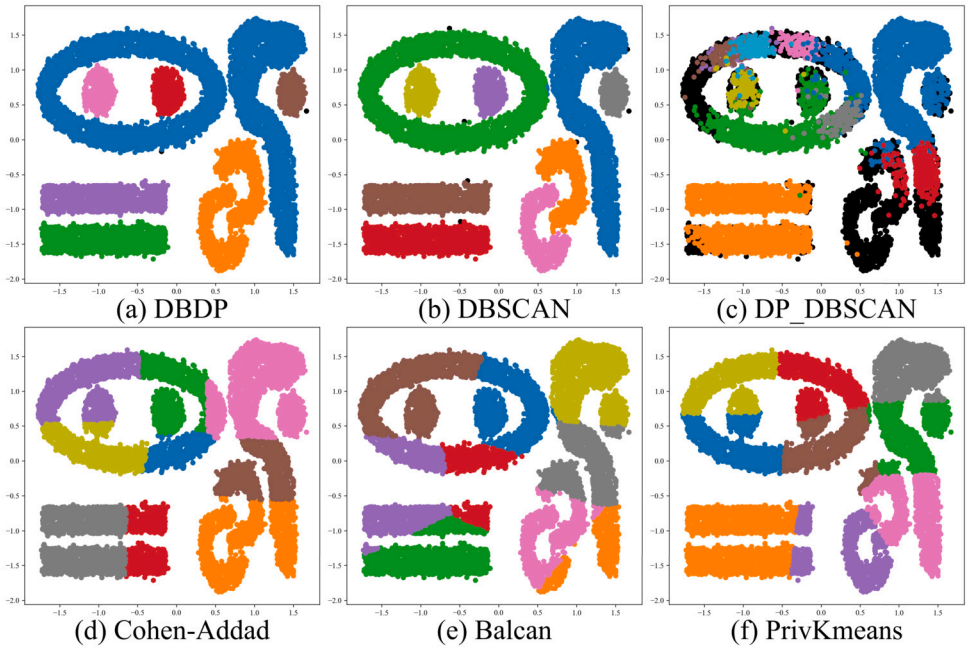

Fig. 7. Clustering results on T7.

contrast, the three k-means algorithms (Cohen-Added, Balcan, and PrivKmeans) in the second row the still produce erroneous clustering results despite handling with convex datasets. This is because the inherent uniform effect in k-means. Specifically, the k-means paradigm tends to distribute data points across clusters in a roughly equal manner due to the fact that it is designed to minimize the intra-cluster sum of squares. In contrast, DBSCAN and the two privacy-preserving DBSCAN algorithms (DBDP and DP_DBSCAN) accurately detect all cluster structures. This is because these algorithms operate on the principles of density reachability and connectivity rather than the minimization of the intra-cluster sum of squares.

In Fig. 5, the Yinyang dataset resembles a yin-yang diagram, which consists of five differently shaped clusters. Our algorithm identifies the overall structure of all clusters correctly and produces a clustering result that is highly consistent with DBSCAN.

In Figs. 6 and 7, both T4 and T7 are datasets consisting of multiple shapes. Cohen-Addad, Balcan, and PrivKmeans fail to handle datasets with such complex-shaped clusters and produce poor clustering results. DP_DBSC AN incorrectly categorizes many points as noise points. DBSCAN and DBDP achieve satisfactory clustering results and their results are highly consistent.

Table 4 presents the quantitative outcomes for our algorithm and the five baselines on all datasets, including five synthetic and five real-world datasets, using F1, NMI, and FMI scores as evaluation metrics. The results on the real datasets are the best results obtained by each algorithm on each dataset, using different parameter values. Table 4 presents the highest and second-highest scores for each indicator in the respective dataset, denoted by bold and underlined formatting.

As shown in Table 4, DBDP achieves clustering performance comparable to that of DBSCAN on the five synthetic datasets. Evidently, the clustering performance of our algorithm are significantly better than other privacy algorithms (DP_DBSCAN, Cohen-Addad, Balcan, and PrivKmeans) on these synthetic datasets. On five real-world datasets (Haberman, Wine, Pageblocks, Penbased, and Htru2), DBDP aligns closely with the performance of the DBSCAN algorithm and even outperforms it in certain cases. Nevertheless, other private algorithms exhibit lower clustering performance compared to our algorithm. This is due to the fact that these real-world datasets have very complex data distributions. DBDP and DBSCAN algorithms can handle these complex distributions better than other competitors. Differentially private clustering algorithms based on the K-means paradigm (Cohen-Added, Balcan and PrivKmeans) suffer from the issue of dividing non-spherical-like clusters into multiple spherical-like clusters when dealing with non-convex datasets, leading to poor clustering results. The DP_DBSCAN algorithm, which uses the Laplace mechanism over distance, misclassifies many normal points as noise, thereby affecting overall clustering performance. In summary, DBDP still has a clustering performance comparable to that of DBSCAN despite the incorporation of the differential privacy mechanism.

5.3. Approximation experiment

This subsection evaluates the approximation of private algorithms to their respective original counterparts. Here, we focus on the approximations of DBDP and DP_DBSCAN to their foundational algorithm, DBSCAN, as well as the approximations of Cohen-Addad, Balcan, and PrivKmeans to their original algorithm, K-Means. To facilitate the quantification and comparison of approximations, we develop an approximation metric scheme. We employ NMI to measure the clustering results of the two algorithms on the same dataset, serving as the approximation score (AS) between them. Obviously, the AS value range is [0,1]. The higher the AS, the better the approximation. For a given dataset, the parameter values eps and $minPts$ in DBDP and DP_DBSCAN are aligned with the optimal

Table 4
Quantitative results in experiments.

Algorithm	F1	NMI	FMI	Prams	F1	NMI	FMI	Prams
	Moons				T0			
DBDP	0.965	0.822	0.950	0.3/25	**1.000**	**1.000**	**1.000**	0.3/6
DBSCAN	**0.969**	**0.864**	**0.961**	0.3/25	**1.000**	**1.000**	**1.000**	0.5/40
DP_DBSCAN	0.718	0.752	0.921	0.26/32	0.260	0.900	0.896	0.26/42
Cohen-Addad	0.453	0.373	0.729	2	0.282	0.265	0.513	3
Balcan	0.438	0.448	0.776	2	0.497	0.486	0.663	3
PrivKmeans	0.413	0.375	0.736	2	0.053	0.431	0.635	3
	Yinyang				T4			
DBDP	0.990	0.972	0.985	0.1/6	**1.000**	0.999	**1.000**	0.1/14
DBSCAN	**0.999**	**0.993**	**0.998**	0.1/6	**1.000**	**1.000**	**1.000**	0.1/10
DP_DBSCAN	0.148	0.347	0.491	0.18/78	0.311	0.546	0.517	0.08/68
Cohen-Addad	0.078	0.355	0.396	5	0.141	0.539	0.522	6
Balcan	0.125	0.323	0.517	5	0.105	0.491	0.557	6
PrivKmeans	0.072	0.248	0.334	5	0.195	0.562	0.549	6
	T7				Haberman			
DBDP	**0.998**	0.993	**0.997**	0.1/32	0.735	**0.115**	**0.781**	2.3/6
DBSCAN	**0.998**	**0.994**	0.996	0.1/32	**0.739**	**0.115**	**0.781**	2.3/6
DP_DBSCAN	0.180	0.180	0.426	0.09/104	0.137	0.101	0.626	0.33/2
Cohen-Addad	0.187	0.616	0.457	9	0.196	0.011	0.608	2
Balcan	0.147	0.464	0.448	9	0.581	0.017	**0.781**	2
PrivKmeans	0.195	0.642	0.479	9	0.582	0.017	0.644	2
	Wine				Pageblocks			
DBDP	0.933	**0.833**	**0.896**	0.3/6	**0.913**	**0.242**	**0.915**	**1.3/28**
DBSCAN	**0.938**	0.809	0.878	0.3/6	**0.913**	**0.242**	**0.915**	1.3/10
DP_DBSCAN	0.657	0.758	0.852	0.2/5	0.902	**0.242**	**0.915**	0.69/30
Cohen-Addad	0.101	0.582	0.684	3	0.364	0.032	0.525	3
Balcan	0.659	0.657	0.581	3	0.392	0.093	0.723	3
PrivKmeans	0.607	0.625	0.729	3	0.452	0.164	0.655	3
	Penbased				Htru2			
DBDP	**0.887**	0.881	**0.831**	0.3/40	**0.964**	0.601	0.967	0.1/25
DBSCAN	**0.887**	**0.881**	**0.831**	0.3/20	**0.964**	**0.621**	**0.968**	0.1/25
DP_DBSCAN	0.051	0.682	0.529	0.03/190	0.908	0.014	0.913	1.4/98
Cohen-Addad	0.090	0.804	0.721	10	0.345	0.120	0.681	2
Balcan	0.736	0.757	0.749	10	0.537	0.065	0.651	2
PrivKmeans	0.014	0.841	0.764	10	0.259	0.163	0.734	2

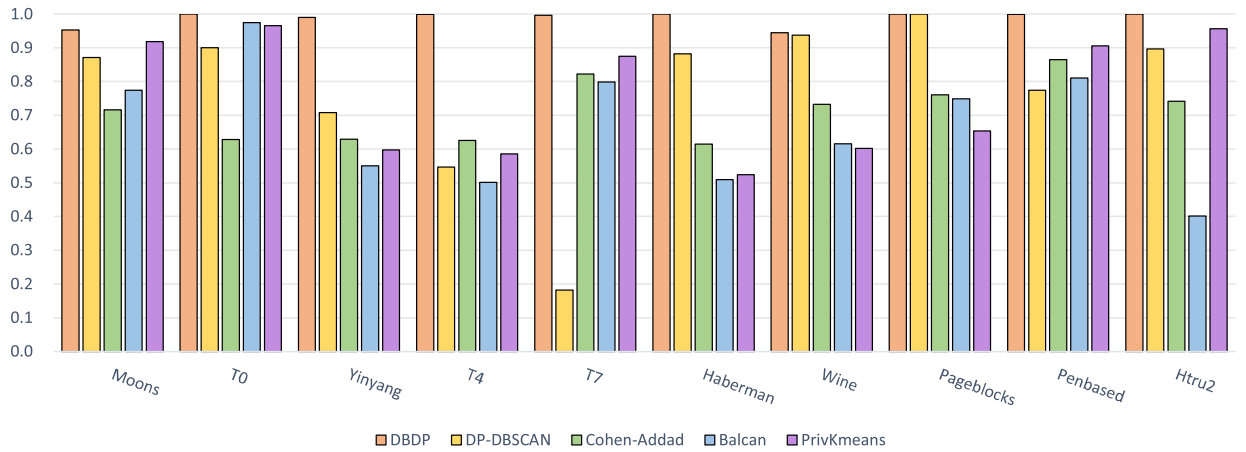

Fig. 8. AS values on all datasets.

configurations of the original DBSCAN algorithm when computing their respective approximations. Additionally, the privacy budget is set to 10. In partition-based clustering, we set K to the number of true clusters in the dataset and the privacy budget to 10. Fig. 8 shows the results on five synthetic datasets and five real-world datasets. The vertical axis represents the approximation score. From Fig. 8, it can be seen that the DBDP algorithm achieves the highest AS on each dataset, with the AS values of the DBDP algorithm on Yinyang, T4, T7, and Penbased reaching 0.99. This indicates that the DBDP algorithm approaches the non-private clustering results as closely as possible while maintaining privacy.

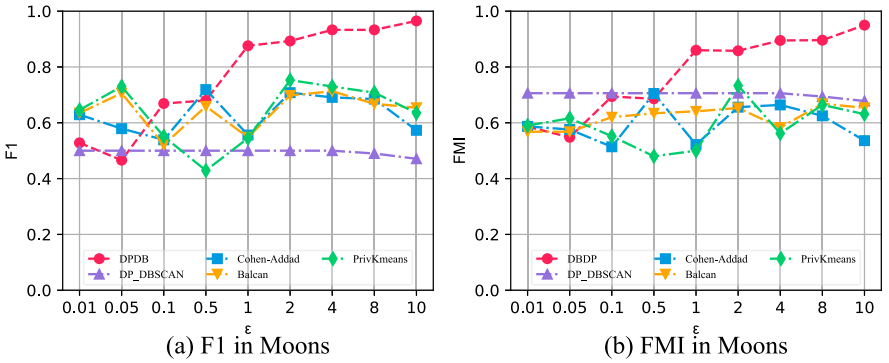

Fig. 9. Effect of privacy budget on Moons.

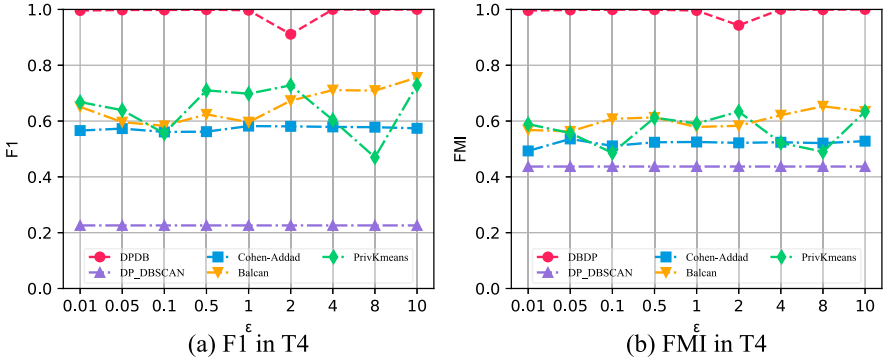

Fig. 10. Effect of privacy budget on T4.

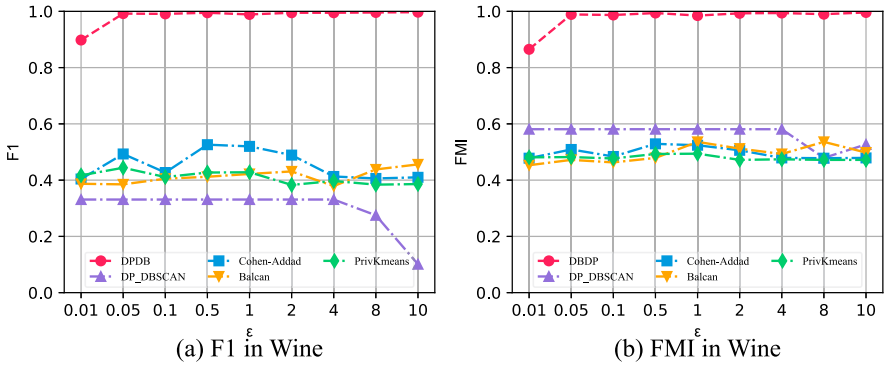

Fig. 11. Effect of privacy budget on Wine.

5.4. Effect of privacy budget

This subsection investigates the implications of privacy budgets. We utilize two synthetic datasets, Moons and T4, and two real-world datasets, Wine and Haberman. Figs. 9-12 depict the F1 and FMI metrics variation for each approach across four different datasets. As shown in Fig. 9, the F1 and FMI rates consistently increase as the privacy-preserving level ϵ varies from 0.01 to 10.0. This implies that lower privacy levels reduce usefulness. In Figs. 10-12, DBDP exhibits remarkably smooth performance, whereas the other algorithms display significant fluctuations. Overall, DBDP is less sensitive to the privacy budget and demonstrates a notably stable performance across various parameter values.

5.5. Running time

This subsection examines the algorithms' runtime performance. Experiments are conducted to compare their runtimes across the datasets detailed in Section 5.1. As shown in Fig. 13, the large figure displays the runtime of all algorithms. For clarity, the small figure excludes the runtime of the DP_DBSCAN algorithm. The horizontal axis represents the names of the datasets, comprising five

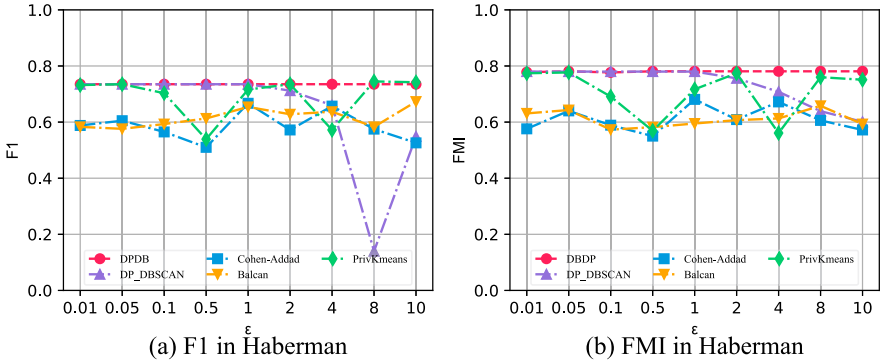

Fig. 12. Effect of privacy budget on Haberman.

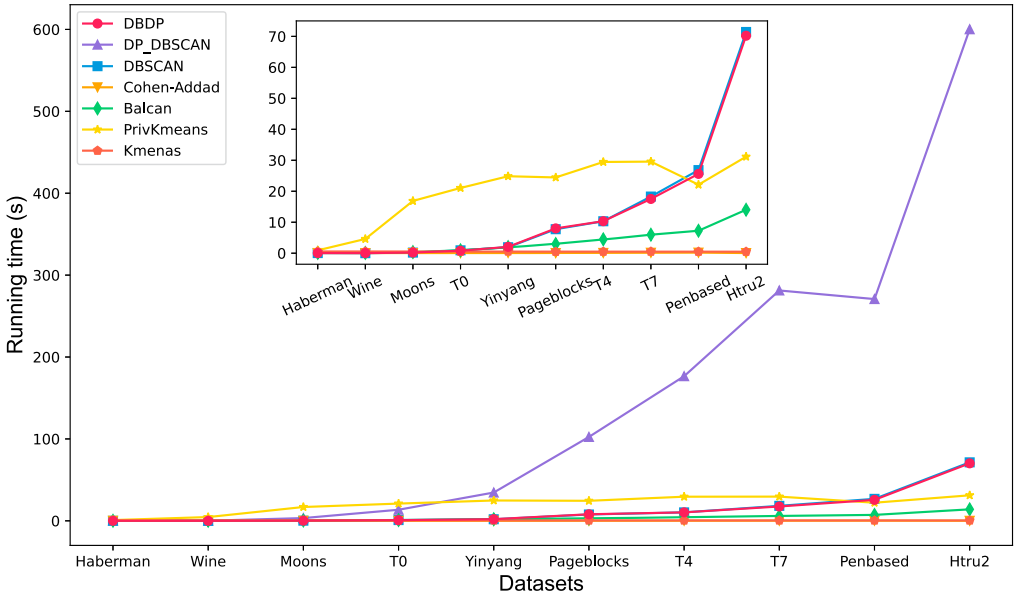

Fig. 13. Running time on all datasets.

real and five synthetic datasets. The vertical axis represents the time in seconds. Each point in the figure represents the running time of each algorithm on each dataset. The running time of DBDP is faster than DP_DBSCAN, comparable to that of DBSCAN, and slower than that of *k*-means paradigm. Compared to the density-based differentially private clustering algorithm DP_DBSCAN, DBDP is more efficient. This is because our algorithm injects noise satisfying differential privacy during the density estimation stage, rather than during the distance metric. Although DBDP modifies the density estimation method in the original DBSCAN, it does not reduce execution efficiency compared to the non-privacy density clustering algorithm DBSCAN. While our algorithm's running time is slightly slower than that of the differentially private algorithm in the *k*-means paradigm, this is due to the inherent properties of the DBSCAN algorithm. However, it is noteworthy that DBDP mostly has a better running time than PrivKmeans. This demonstrates that the DBDP algorithm successfully incorporates privacy-preserving mechanisms into DBSCAN without sacrificing efficiency.

6. Conclusion

We introduce a new private clustering algorithm DBDP. The algorithm is based on the DBSCAN paradigm. This approach inherits the benefits of DBSCAN, particularly its ability to identify clusters with arbitrary shapes, while satisfying the differential privacy. Specifically, we introduce a modification to the DBSCAN algorithm by incorporating Laplace noise and demonstrate that this algorithm adheres to the principles of differential privacy. In addition, we experimentally demonstrate that our algorithm outperforms current differentially private clustering algorithms and has performance comparable to the original DBSCAN. This algorithm is more suitable for non-convex shape data, making it applicable to a wider range of scenarios compared to other differentially private clustering algorithms. The proposed algorithm enhances data sharing and collaboration, thereby advancing data-driven research and cooperation. Additionally, the use of DBDP mitigates algorithmic bias, promoting fairness. The main drawback of our algorithm is the excessive computation time required for handling large and complex datasets. Future research directions include exploring data

compression techniques to address the challenges of large and complex datasets. Another research direction involves using optimization algorithms to enhance the efficiency of DBDP, such as the geyser-inspired algorithm [41] and the prairie dog optimization algorithm [42].

CRediT authorship contribution statement

Fuyu Wu: Writing – original draft, Visualization, Software, Resources, Methodology, Formal analysis, Conceptualization. **Mingjing Du:** Writing – original draft, Validation, Supervision, Project administration, Funding acquisition. **Qiang Zhi:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62006104), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. SJCX23_1377). The authors thank Wang Nana for assistance with differential privacy.

References

[1] H. Barlow, Unsupervised learning, Neural Comput. 1 (3) (1989) 295–311.
[2] J. Sun, M. Du, C. Sun, Y. Dong, Efficient online stream clustering based on fast peeling of boundary micro-cluster, IEEE Trans. Neural Netw. Learn. Syst. (2024) 1–14.
[3] J. Sun, M. Du, Z. Lew, Y. Dong, Twstream: three-way stream clustering, IEEE Trans. Fuzzy Syst. (2024) 1–13.
[4] T. Zhu, G. Li, W. Zhou, S. Philip, Differentially private data publishing and analysis: a survey, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1619–1638.
[5] M. Gong, Y. Xie, K. Pan, K. Feng, A. Qin, A survey on differentially private machine learning, IEEE Comput. Intell. Mag. 15 (2) (2020) 49–64.
[6] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends® Theor. Comput. Sci. 9 (3–4) (2014) 211–407.
[7] U. Stemmer, Locally private k-means clustering, J. Mach. Learn. Res. 22 (1) (2021) 7964–7993.
[8] M. Yang, I. Tjuawinata, K. Lam, K-means clustering with local d-privacy for privacy-preserving data analysis, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2524–2537.
[9] B. Ghazi, R. Kumar, P. Manurangsi, Differentially private clustering: tight approximation ratios, Adv. Neural Inf. Process. Syst. 33 (2020) 4040–4054.
[10] V. Cohen-Addad, A. Epasto, S. Lattanzi, V. Mirrokni, A. Munoz Medina, D. Saulpic, C. Schwiegelshohn, S. Vassilvitskii, Scalable differentially private clustering via hierarchically separated trees, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 221–230.
[11] C. Alisa, K. Pritish, Practical differentially private clustering, https://blog.research.google, 2021.
[12] Z. Guan, Z. Lv, X. Sun, L. Wu, J. Wu, X. Du, M. Guizani, A differentially private big data nonparametric Bayesian clustering algorithm in smart grid, IEEE Trans. Netw. Sci. Eng. 7 (4) (2020) 2631–2641.
[13] Z. Lu, H. Shen, A convergent differentially private k-means clustering algorithm, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2019, pp. 612–624.
[14] P. Jain, A. Thakurta, Differentially private learning with kernels, in: Proceedings of the 30th International Conference on International Conference on Machine Learning, 2013, pp. 118–126.
[15] T. Ni, M. Qiao, Z. Chen, S. Zhang, H. Zhong, Utility-efficient differentially private k-means clustering based on cluster merging, Neurocomputing 424 (2021) 205–214.
[16] F. McSherry, I. Mironov, Differentially private recommender systems: building privacy into the Netflix prize contenders, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 627–636.
[17] Z. Lu, H. Shen, Differentially private k-means clustering with convergence guarantee, IEEE Trans. Dependable Secure Comput. 18 (4) (2021) 1541–1552.
[18] N. Fu, W. Ni, H. Hu, S. Zhang, Multidimensional grid-based clustering with local differential privacy, Inf. Sci. 623 (2023) 402–420.
[19] D. Zhang, W. Ni, N. Fu, L. Hou, R. Zhang, Locally differentially private multi-dimensional data collection via Haar transform, Comput. Secur. 130 (2023) 103291.
[20] A. Blum, C. Dwork, F. McSherry, K. Nissim, Practical privacy: the sulq framework, in: Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2005, pp. 128–138.
[21] Y. Wang, Y. Wang, A. Singh, Differentially private subspace clustering, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 1, 2015, pp. 1000–1008.
[22] D. Su, J. Cao, N. Li, E. Bertino, H. Jin, Differentially private k-means clustering, in: Proceedings of the 6th ACM Conference on Data and Application Security and Privacy, 2016, pp. 26–37.
[23] D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang, S. Zhang, A fast granular-ball-based density peaks clustering algorithm for large-scale data, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[24] D. Cheng, S. Liu, S. Xia, G. Wang, Granular-ball computing-based manifold clustering algorithms for ultra-scalable data, Expert Syst. Appl. 247 (2024) 123313.
[25] L. Huang, N.K. Vishnoi, Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal, in: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, 2020, pp. 1416–1429.
[26] D. Feldman, C. Xiang, R. Zhu, D. Rus, Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks, in: Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2017, pp. 3–15.
[27] M. Balcan, T. Dick, Y. Liang, W. Mou, H. Zhang, Differentially private clustering in high-dimensional Euclidean spaces, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 322–331.

[28] U. Stemmer, H. Kaplan, Differentially private k-means with constant multiplicative error, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems December, vol. 31, 2018, pp. 5436–5446.
[29] Z. Huang, J. Liu, Optimal differentially private algorithms for k-means clustering, in: Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, 2018, pp. 395–408.
[30] W. Wu, H. Huang, A dp-dbscan clustering algorithm based on differential privacy preserving, Comput. Eng. Sci. 37 (4) (2015) 830–834.
[31] L. Ni, C. Li, X. Wang, H. Jiang, J. Yu, Dp-mcdbscan: differential privacy preserving multi-core dbscan clustering for network user data, IEEE Access 6 (2018) 21053–21063.
[32] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 1–12.
[33] J. Guan, S. Li, X. He, J. Chen, Clustering by fast detection of main density peaks within a peak digraph, Inf. Sci. 628 (2023) 504–521.
[34] J. Guan, S. Li, J. Zhu, X. He, J. Chen, Fast main density peak clustering within relevant regions via a robust decision graph, Pattern Recognit. (2024) 110458.
[35] H. Jia, Q. Ren, L. Huang, Q. Mao, L. Wang, H. Song, Large-scale non-negative subspace clustering based on Nyström approximation, Inf. Sci. 638 (2023) 118981.
[36] H. Tu, S. Ding, X. Xu, H. Hou, C. Li, L. Ding, Non-iterative border-peeling clustering algorithm based on swap strategy, Inf. Sci. 654 (2024) 119864.
[37] H. Jia, Y. Wu, Q. Mao, Y. Li, H. Song, Adaptive density subgraph clustering, IEEE Trans. Comput. Soc. Syst. (2024).
[38] X. Xu, H. Hou, S. Ding, Semi-supervised deep density clustering, Appl. Soft Comput. 148 (2023) 110903.
[39] J. Hou, H. Yuan, M. Pelillo, Towards parameter-free clustering for real-world data, Pattern Recognit. 134 (2023) 109062.
[40] C. Li, S. Ding, X. Xu, H. Hou, L. Ding, Fast density peaks clustering algorithm based on improved mutual k-nearest-neighbor and sub-cluster merging, Inf. Sci. 647 (2023) 119470.
[41] M. Ghasemi, M. Zare, A. Zahedi, M.A. Akbari, S. Mirjalili, L. Abualigah, Geyser inspired algorithm: a new geological-inspired meta-heuristic for real-parameter and constrained engineering optimization, J. Bionics Eng. 21 (1) (2024) 374–408.
[42] A.E. Ezugwu, J.O. Agushaka, L. Abualigah, S. Mirjalili, A.H. Gandomi, Prairie dog optimization algorithm, Neural Comput. Appl. 34 (22) (2022) 20017–20065.