Heterogeneous Data Release for Cluster Analysis with Differential Privacy

Rong Wang[a], Benjamin C. M. Fung[b], Yan Zhu[a]

[a] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, Sichuan, China*

[b] *School of Information Studies, McGill University, Montreal, QC, Canada, H3A 1X1*

Abstract

Many models have been proposed to preserve data privacy for different data publishing scenarios. Among these models, ϵ-differential privacy has drawn increasing attention in recent years due to its rigorous privacy guarantees. While many existing solutions using ϵ-differential privacy deal with relational data and set-valued data separately, most of the real-life data, such as electronic health records, are in heterogeneous form. Privacy protection on heterogeneous data has not been widely studied. Furthermore, many existing works in privacy protection consider preserving the utility for the tasks of frequent itemset mining or classification analysis, but few works have focused on data publication for cluster analysis. In this paper, we propose the first differentially-private solution to release heterogeneous data for cluster analysis. The challenge facing us is how to mask raw data without any explicit guidance. Our approach addresses this challenge by converting a clustering problem to a classification problem, in which class labels can be used to encode the cluster structure of the raw data and assist the masking process. The approach generalizes the raw data probabilistically and adds noise to them for satisfying ϵ-differential privacy. Through extensive experiments on real-life datasets, we validate the performance of our approach.

Keywords: data publishing, heterogeneous data, differential privacy, cluster

Email addresses: `wangrong.kiko@qq.com` (Rong Wang), `ben.fung@mcgill.ca` (Benjamin C. M. Fung), `yzhu@swjtu.edu.cn` (Yan Zhu)

The first author conducted the research during the visit at McGill University.

analysis

1. Introduction

As information becomes a kind of strategic resource in the era of big data, many organizations, such as government agencies and hospitals, release their data (e.g., census data or medical records) to third parties in order to reveal the hidden value of the data [1, 2]. However, directly releasing raw data may unavoidably leak data privacy and may even violate privacy laws [3, 4]. To address this problem, privacy-preserving data publishing (PPDP) [5] has been studied extensively, with the goal of protecting private information by distorting the raw data before publication while preserving as much utility of the perturbed data as possible for subsequent data analysis.

Because of its strong privacy guarantee, ϵ-differential privacy [6, 7] has received increasing attention in the literature. As the structure of the collected data becomes much richer, many differentially-private approaches [8, 9, 10, 11] that handle relational data or set-valued data individually are non-effective. Relational data refer to the data in which records have a single value for each attribute, and set-valued data refer to the data in which records have one or more values for each attribute. Many real-life data are typically composed of relational data and set-valued data, and they are called *heterogeneous data*. For example, a patient who goes to the hospital for the first time may be asked to fill out a form that requires his/her gender (relational), age (relational), medical history (set-valued), etc. The information is stored as a heterogeneous data record in the hospital's database to assist physicians in diagnosing and treating. For heterogeneous data publishing, one naive approach is to vertically divide the raw data into different subsets such that each subset has only one type of data structure, and then to apply existing approaches on these subsets independently. However, most data publishing scenarios require that the entire data be released together so that the associations among different data types can be retained. On the other hand, many privacy-preserving works consider preserving the utility for frequent itemset mining [12, 13, 14] or classification analysis [15, 16, 17], but a very limited number of works have focused on privacy protection for cluster analysis. Thus, we cover these gaps with a differentially-private approach to release heterogeneous data for cluster analysis.

Consider our data release scenario as follows. The data owner wants to release heterogeneous data (e.g., Table 1) to the data recipient for clustering. If the data owner releases the raw data directly, the individual privacy of the data may be leaked. Thus, private information should be masked before being released. Note that the data owner wants to release data records to the data recipient, instead of clustering results, because unlike association rules and classifiers, releasing the clustering results (e.g., clusters with their centroids and sizes) may not provide enough information for further analysis. For example, the data recipient may browse into the clustered records to find their inherent relationship. Releasing data records not only satisfies the demand for clustering, but also gives the data recipient greater flexibility in conducting his specific data analysis.

Table 1. Patients' heterogeneous data (Each row of the table corresponds to a patient. The attributes *Age*, *Sex*, and *ICD codes* are numerical, categorical, and set-valued, respectively.)

ID	Age	Sex	ICD codes	Class
1	21	M	21	0
2	44	M	11, 12	0
3	72	F	12	1
4	25	M	11	0
5	19	F	11, 21	0
6	36	F	21, 22	1
7	32	M	12, 21, 22	1
8	45	F	22	1
9	63	M	12, 21	1
10	28	F	11, 21, 22	1

In this paper, we present a differentially-private algorithm to protect individual privacy while preserving as much information as possible for cluster analysis. To tackle the challenge of lacking proper guidance for the masking process, our approach converts the clustering problem into a classification problem. That is, it groups the raw data into clusters and utilizes cluster/class labels to encode the cluster structure of the data. It then generalizes the raw data iteratively while preserving the cluster structure. At each iteration, the approach selects a general value in a probabilistic manner and specializes the value to a more specific one. The process is repeated until certain conditions are reached. Finally, noise is added to further guarantee ϵ-differential privacy. The contributions of this paper are summarized as follows:

- We formally define the problem of *differentially-private heterogeneous data release for cluster analysis.* This paper is the first work that tackles this problem and addresses the challenges of heterogeneity and lack of guidance in the anonymization process for cluster analysis.

- We propose a customizable approach to heterogeneous data anonymization for cluster analysis. Users can choose different clustering algorithms and algorithmic parameters to get their desired results. Also, a distance metric that considers both relational and set-valued attributes is tailored for heterogeneous data clustering.

- To satisfy the differential privacy principle, we propose an algorithm to simultaneously handle relational and set-valued data in a non-deterministic fashion. Data of different types are anonymized in a similar way, which is computationally efficient.

- We extensively evaluate the performance of the proposed cluster-oriented approach on real-life datasets. The results suggest that our approach can generate anonymous data of better utility compared to the general method that does not consider the task of cluster analysis during anonymization.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Preliminaries including the problem statement are presented in Section 3. The proposed approach is described in Section 4, and experimental results are presented in Section 5. A discussion of the approach is given in Section 6. Section 7 concludes the paper.

2. Related Work

2.1. Anonymization of Different Types of Data

Relational Data Anonymization. Many privacy models had been proposed to anonymize relational data, such as k-anonymity [18, 19], l-diversity [20], and t-closeness [21]. Recently, researchers extend these models to provide stricter privacy protections. Amiri et al. [22] hide the correlations between identifying attributes and sensitive attributes and generate k-anonymous β-likeness data to prevent identity and attribute disclosures. Agarwal et al. [23] propose a privacy model called (P, U)-sensitive k-anonymity to protect sensitive records instead of sensitive attributes. Zhu et al. [24] present an independent l-diversity principle to prevent corruption attacks even if adversaries have known the corresponding data publishing strategy. Soria-Comas et al. [25] propose two cluster-based algorithms using microaggregation to attain anonymized data that satisfy t-closeness. Wang et al. [26] also focus on the t-closeness principle and protect the privacy of multiple sensitive attributes. Instead of employing such syntactic privacy models, we adopt differential privacy [6] in this paper because it is independent of any adversary's knowledge and can provide a provable privacy guarantee. Mohammed et al. [27] show that differentially-private data can be published via the addition of uncertainty during the generalization process. Inspired by [27], we extend our research scope from relational data to heterogeneous data and combine the generalization technique with differential privacy.

Set-Valued Data Anonymization. Terrovitis et al. [28] propose a k^m-anonymity model for set-valued data. They limit the maximum knowledge of the adversaries and guarantee that any set of m or less items corresponds to at least k records that contain the set in the released data. Bewong et al. [29] present a clustering method based on a distance function that considers both the similarity and the disclosure risk of transaction records. They prove that when the total distance of inter-clusters is minimized, data anonymization can be achieved with minimal utility loss. For privacy-preserving set-valued data publishing, Zhang et al. [30] use a data partition technique to break associations among identifying attributes and add noise to the final query results. Gunawan et al. [31] propose an approach to prevent set-valued data from identity linkage attacks while maintaining data utility and data property. Their approach consists of two steps, i.e., grouping records based on adversaries' knowledge and selecting surrogate items to replace the items in the adversaries' knowledge. We refer to [32] for a broad review of the anonymization of relational data and set-valued data.

Heterogeneous Data Anonymization. Poulis et al. [33] propose a (k, k^m)-anonymity to prevent an adversary, who knows an individual's information including relational attributes and at most m items of the set-valued attribute, from linking any individual to the corresponding record in the released data. However, (k, k^m)-anonymity cannot protect the privacy of individuals with more than m items in the set-valued attribute. To address the drawback of (k, k^m)-anonymity, Wang et al. [34] introduce a (k, ρ)-anonymity model by differentiating sensitive and non-sensitive items in the set-valued attribute. Their proposed model prevents attribute disclosures by satisfying the diversity constraint. Wang et al. [35] propose a graph-based multifold model to anonymize data with relational attributes and a set-valued attribute. They protect the associations between two objects in the raw data by masking sensitive relational attributes and modeling association rules as an uncertain graph. Gong et al. [36] propose a privacy model called (k, l)-diversity to address the disclosure risk of the raw data in which an individual may correspond to multiple records. The data they processed can be converted into heterogeneous data including relational and set-valued attributes. Other works focusing on heterogeneous data can be found in [37, 38]. However, none of these privacy-preserving works consider the problem of cluster analysis on heterogeneous data, which is the primary contribution of this paper.

2.2. Data Mining with Differential Privacy

Many data mining problems with differential privacy have been studied. Maruseac et al. [14] combine the exponential mechanism of differential privacy with reservoir sampling to mine high-confidence association rules privately. Gong et al. [39] present a differentially-private regression analysis model. They transform the objective function into the form of polynomial and add noise to the coefficients of the polynomial representation. Sun et al. [15] combine differential privacy with a decision tree to provide privacy preservation of classifiers. They also use the differentially-private mini-batch gradient descent algorithm to protect the privacy of training data. Zhang et al. [17] propose a differential privacy support vector machine (SVM) based on dual variable perturbation. Their algorithm solves the dual problem of SVM first and adds Laplace random noise to the corresponding dual variables of each support vector to be released. Su et al. [40] address the problem of differentially-private k-means clustering. They propose a non-interactive approach that can output a synopsis of the input dataset for k-means. Their approach divides the input dataset into equal-size cells and adds Laplace noise to the size of each cell. Nguyen [41] also develops a non-interactive differentially-private approach for the cluster analysis. Compared to [40], his approach focuses on the k-modes algorithm and adds geometric noise to the final cluster centroids. Other works [16, 42, 43] also consider the problem of data mining with differential privacy. While these works are specific to certain data mining algorithms, our approach can be applied with different clustering algorithms.

3. Preliminaries

Table 2 summarizes some notations used in the following.

Table 2. Notations

Notations	Explanation	Notations	Explanation
$\mathfrak{R}$	universe	r	record
D	dataset	n	size of dataset
$\hat{D}$	neighbor dataset	d	number of attributes
D^*	labeled dataset	d_{num}	number of numerical attributes
D'	anonymized dataset	h	number of specializations
A	set of attributes	ϵ	privacy budget
$\mathcal{M}$	mechanism	f, u	function
$\mathbb{U}$	cluster structure	Δf, Δu	global sensitivity
$\mathbb{T}$	cluster structure	T_i	i^{th} cluster in $\mathbb{T}$
$\mathbb{P}$	cluster structure	P_j	j^{th} cluster in $\mathbb{P}$
$\mathbb{C}_1$, $\mathbb{C}_2$	cluster structure	x, p, c	attribute value

3.1. Differential Privacy

Let $\mathfrak{R}$ represent a finite data universe and r represent a record with d attributes. A dataset D is a set of n records sampled from universe $\mathfrak{R}$. Two datasets D and $\hat{D}$ are defined as neighboring datasets if and only if either $\hat{D} = D + r$ or $D = \hat{D} + r$, where $D + r$ (or $\hat{D} + r$) denotes the dataset resulted from adding the record r to the dataset D (or $\hat{D}$). The definition of differential privacy is as follows.

Definition 1 (*ϵ-Differential Privacy* [6])**.** A randomized mechanism $\mathcal{M}$ is differentially private if for any pair of neighboring datasets D and $\hat{D}$, and for any set of possible sanitized outputs Ω,

$$Pr[\mathcal{M}(D) \in \Omega] \leq exp(\epsilon) \times Pr[\mathcal{M}(\hat{D}) \in \Omega]. \tag{1}$$

The parameter ϵ, called privacy budget, is used to control the level of privacy guarantees achieved by mechanism $\mathcal{M}$. A smaller ϵ means a stronger privacy level. ϵ defaults to a positive number and its value is usually small, such as 0.1, 0.5, and 0.8 [44].

The magnitude of added noise depends not only on the privacy budget ϵ but on the global sensitivity of a randomized function. Global sensitivity reflects the maximum difference of outputs of a function on two neighboring datasets.

Definition 2 (*Global Sensitivity* [6])**.** Given a randomized function $f : D \to \mathfrak{R}$, the global sensitivity of f is

$$\Delta f = \max \| f(D) - f(\hat{D}) \|_1, \tag{2}$$

for any pair of neighboring datasets D and $\hat{D}$.

Laplace mechanism and exponential mechanism are two common principal mechanisms to achieve differential privacy.

Definition 3 (*Laplace Mechanism* [45])**.** Given a dataset D, privacy budget ϵ, and a randomized function $f : D \to \mathfrak{R}$, which global sensitivity is Δf, a mechanism $\mathcal{M}(D) = f(D) + Lap(\Delta f / \epsilon)$ satisfies ϵ-differential privacy.

Definition 4 (*Exponential Mechanism* [46])**.** Given a dataset D, output range T, privacy budget ϵ, and a utility function $u : (D, T) \to \mathfrak{R}$, a mechanism $\mathcal{M}$ that selects an output $t \in T$ with probability proportional to $exp(\frac{\epsilon u(D,t)}{2\Delta u})$ satisfies ϵ-differential privacy.

There are two important properties of differential privacy. They play a vital role in judging whether a mechanism satisfies differential privacy.

Property 1 (Sequential Composition [47])**.** Let $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m\}$ be a set of privacy mechanisms. If each $\mathcal{M}_i$ provides ϵ_i-differential privacy and $\mathcal{M}$ is sequentially performed on a dataset, $\mathcal{M}$ will provide $(\sum_i^m \epsilon_i)$-differential privacy.

The sequential composition suggests that the privacy budget and noise accumulate linearly when a series of differential privacy is applied to the same dataset.

Property 2 (Parallel Composition [47])**.** Let $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m\}$ be a set of privacy mechanisms. If each $\mathcal{M}_i$ provides ϵ_i-differential privacy on a disjointed subset of a dataset, $\mathcal{M}$ will provide $(\max\{\epsilon_1, \epsilon_2, \cdots, \epsilon_m\})$-differential privacy.

The parallel composition suggests that the degree of privacy protection depends upon the maximum value of ϵ_i when a series of differential privacy is applied to different subsets of a dataset.

3.2. Problem Statement

Suppose that the data owner wants to release person-specific data (e.g., Table 1) to the data recipient for cluster analysis. The raw data can be defined as a set of records $D = \{r_1, r_2, \cdots, r_n\}$, where each record r_i $(1 \leq i \leq n)$ represents the information of an individual with d attributes $A = \{A_1, A_2, \cdots, A_d\}$. We assume that each attribute A_j $(1 \leq j \leq d)$ can be categorical, numerical or set-valued and that a taxonomy tree is given for each categorical or set-valued attribute. Note that explicit identifiers, such as *name* and *driver's license number*, should be removed before publication and are not discussed in the following.

In this paper, we focus on differentially-private heterogeneous data release for cluster analysis. The task of cluster analysis is to divide objects into groups such that similar objects are in the same group and dissimilar objects are in different groups. The clustering result can be represented by a cluster structure.

Definition 5 (*Cluster Structure*)**.** Let g be the number of clusters. The cluster structure of a dataset $D = \{r_1, r_2, \cdots, r_n\}$ is defined as a matrix $\mathbb{U}_{n\times g}$, where each element $e_{i,j} \in \{1, 0\}$ $(1 \leq i \leq n, 1 \leq j \leq g)$ denotes the clustering assignment of record r_i to the j^{th} cluster; that is, record r_i belongs to the j^{th} cluster while $e_{i,j}$ is equal to 1, and do not while $e_{i,j}$ is equal to 0.

Based on the above assumptions, our problem statement can be defined as:

Definition 6 (*Differentially-Private Heterogeneous Data Release for Cluster Analysis*)**.** Given a dataset $D = \{r_1, r_2, \cdots, r_n\}$ and privacy budget ϵ, the problem of anonymization on heterogeneous data for cluster analysis is to anonymize D on attributes of different types such that the anonymized dataset $D' = \{r'_1, r'_2, \cdots\}$ (1) satisfies ϵ-differential privacy and (2) maintains the similarity of the cluster structure of D as much as possible.

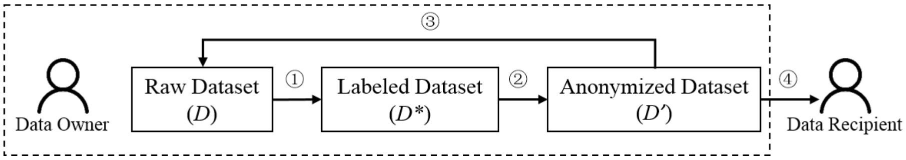


Fig. 1. Overview of the proposed approach

Metrics of the similarity of two cluster structures will be discussed in Section 5.3.

4. Proposed Approach

In this section, we first present an overview of our approach to the problem of heterogeneous data anonymization for cluster analysis. We then elaborate details of the proposed differentially-private algorithm. Finally, we analyze the privacy guarantee and the time complexity of the algorithm.

4.1. Overview

Fig. 1 gives an overview of the proposed approach. In step ①, the data owner employs a clustering algorithm on the raw dataset D to obtain the initial cluster structure. Records in the same cluster are assigned the same cluster label. Compared with D, the labeled dataset D^* has $d+1$ attributes $A^* = \{A_1, A_2, \cdots, A_d, Class\}$, where $Class$ denotes the $Class$ attribute; namely, in addition to the d original attributes from D, each record r_i in D^* has a cluster/class label. Therefore, preserving the cluster structure of D means preserving the ability to identify these class labels during anonymization. In step ②, the proposed differentially-private algorithm is executed on D^* to obtain the anonymized dataset D'. If the utility of D' is unsatisfactory, the data owner can return to the first step and tune algorithmic parameters, such as taxonomy trees, choice of clustering algorithms, and the number of clusters (see step ③). Repeat steps ①-③ until D' with the desired utility is obtained. In step ④, the data owner releases D' to a data recipient.

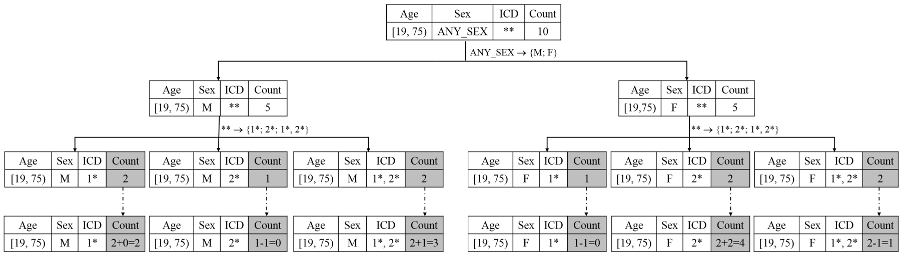


Fig. 2. Example of a partition tree

4.2. Proposed Differentially-Private Generalization Algorithm

We propose a differentially-private generalization algorithm called *DPHeter* for heterogeneous data, which is significantly modified based on the top-down specialization (TDS) technique [48] due to its efficiency. The specialization starts with the most general state and goes down iteratively by replacing some values with more specific values until reaching the predefined number of specializations. A specialization, denoted by $p \to Children(p)$, replaces a parent value p with its directly connected child values $Children(p)$ according to the corresponding taxonomy tree. For example, in Fig. 3 $Children([19, 75)) = \{[19, 45), [45, 75)\}$, $Children(ANY\_SEX) = \{M, F\}$, and $Children(**) = \{1*, 2*\}$. We use the terms "child nodes" and "child values" interchangeably. We also refer the parent value that can be replaced with its directly connected child values to as "cut" in the following.

Example 1. Fig. 2 shows a process of the TDS technique on the data of Table 1. At first, each value is generalized to the topmost value of its corresponding taxonomy tree shown in Fig. 3, and the initial $\cup Cut$ is $\{[19, 75), ANY\_SEX, **\}$. Suppose that the ANY_SEX cut is selected to split downwards. Then the root of the partition tree in Fig. 2 will have two new child nodes because of $ANY\_SEX \to M, F$, and the current $\cup Cut$ is updated to $\{[19, 75), M, F, **\}$. □

To ensure that the specialization process satisfies ϵ-differential privacy, the key is to make sure every step in the anonymization process is differentially-

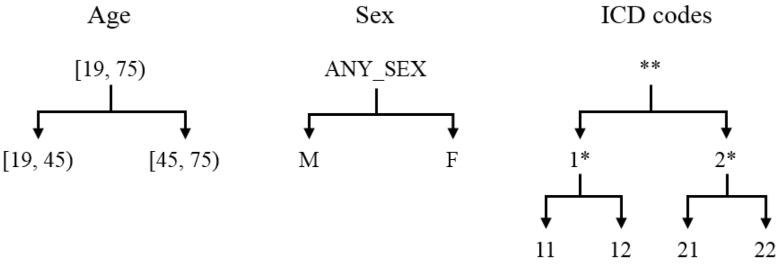


Fig. 3. Taxonomy trees of the attributes *Age*, *Sex*, and *ICD codes*

private. The essential steps include cut selection and record partition.

4.2.1. Selection of Cuts

We choose the exponential mechanism (Definition 4) to select cuts because the mechanism is designed for discrete alternatives. According to Definition 4, a utility function is required. In this paper, we adopt the information gain between attributes and class labels as our utility function. This is because each specialization on a cut tends to increase information by producing specific attribute values and the information gain has the ability to make the class labels "more predictable" based on these values.

The entropy of an attribute value x in a dataset D is calculated as:

$$H_x(D) = -\sum_{cls \in \Omega(Class)} \frac{|D_x^{cls}|}{|D_x|} \times log_2 \frac{|D_x^{cls}|}{|D_x|}, \tag{3}$$

where $\Omega(Class)$ is the domain of the *Class* attribute in D, D_x is the set of data records in D whose attribute values can be generalized to x, D_x^{cls} is the set of data records in D_x that contain the class label cls, and $|\cdot|$ is the size of a dataset.

The utility function/score of an attribute value p that is generalized to its child values is defined as:

$$u(p) = H_p(D) - \sum_{c \in Children(p)} \frac{|D_c|}{|D_p|} H_c(D). \tag{4}$$

where $Children(p)$ is the child values of p, and $D_p = \sum_{c \in Children(p)} D_c$.

The global sensitivity (Definition 2) of $u(p)$ is $log_2|\Omega(Class)|$, where $|\Omega(Class)|$ is the domain size of the *Class* attribute. This is because $H_p(D)$ is in the range of $[0, log_2|\Omega(Class)|]$, and $\sum_{c \in Children(p)} \frac{|D_c|}{|D_p|} H_c(D)$ is in the range of $[0, H_p(D)]$. Thus, the change of $u(p)$ is not greater than $log_2|\Omega(Class)|$ no matter whether adding or removing any record.

In each round of specialization, we first compute the utility score of each cut candidate according to (4), and then probabilistically choose a cut to split downwards according to the exponential mechanism.

Example 2. Continue to consider the data of Table 1 and taxonomy trees in Fig. 3. $Children(ANY\_SEX) = \{M, F\}$. According to (4), the utility score of ANY_SEX is calculated as:

$$H_{ANY\_SEX}(D) = -(\frac{4}{10} \times log_2 \frac{4}{10} + \frac{6}{10} \times log_2 \frac{6}{10}) = 0.9709$$

$$H_M(D) = -(\frac{3}{5} \times log_2 \frac{3}{5} + \frac{2}{5} \times log_2 \frac{2}{5}) = 0.9709$$

$$H_F(D) = -(\frac{4}{5} \times log_2 \frac{4}{5} + \frac{1}{5} \times log_2 \frac{1}{5}) = 0.7219$$

$$u(ANY\_SEX) = H_{ANY\_SEX}(D) - [\frac{5}{10} \times H_M(D) + \frac{5}{10} \times H_F(D)] = 0.1245$$

Similar to the above calculation, $u([19, 75)) = 0.2812$, and $u(**) = 0.0954$. According to the exponential mechanism, the possibility of $[19, 75)$, ANY_SEX, or $**$ being selected as the current cut is $56.12\%(\frac{0.2812}{0.2812+0.1245+0.0954} \approx 0.5612)$, $24.85\%(\frac{0.1245}{0.2812+0.1245+0.0954} \approx 0.2485)$, or $19.04\%(\frac{0.0954}{0.2812+0.1245+0.0954} \approx 0.1904)$, respectively. □

4.2.2. Partition of Records

After a cut is selected, the raw records are divided into different groups. The partition strategy for categorical attributes is fixed because of their predefined taxonomy trees, so the global sensitivity of the partition function of categorical attributes is 1. Thus, the step of record partitioning satisfies differential privacy, according to the currently selected categorical cut and the corresponding taxonomy tree.

Compared with categorical attributes, the difference in the specialization on set-valued attributes is the existence of the combination of child nodes. Suppose a set-valued cut p is selected and it has t child nodes in its corresponding taxonomy tree. The specialization on p will produce a total of $(2^t - 1)$ child groups. To improve the efficiency of *DPHeter*, empty child groups should be pruned as early as possible. Because of the indeterminacy required by differential privacy, we treat a child group as "non-empty" by verifying whether its noisy size (generated by the Laplace mechanism) is greater than a threshold. That is, if the noisy size of a sub-partition is greater than a threshold, the sub-partition is preserved; otherwise, it is treated as "empty" and should be pruned. The threshold can be controlled by the data owner.

As mentioned in [48], there is no need to provide taxonomy trees for numerical attributes. If a numerical cut is selected to split downwards, its corresponding taxonomy tree will be dynamically generated or expanded when searching for a split value of the cut. A split value should not be randomly selected for a cut because the probability of choosing the same value from a dataset not containing this value is 0. This means the selection on a split value for a numerical attribute is probabilistic. We again use the exponential mechanism. The utility score of each attribute value in the range of the numerical cut is computed, and the exponential mechanism is used to select an attribute value as the split value for the numerical cut. The probability of selecting an attribute value c as the split value for a numerical cut p is defined as:

$$Pr[\text{split value} \leftarrow c] = \frac{exp(\frac{\epsilon}{2\Delta u}u(c))}{\sum_{x_i \in I(p)} exp(\frac{\epsilon}{2\Delta u}u(x_i))}, \tag{5}$$

where ϵ is the privacy budget assigned for this selection, Δu is the global sensitivity of Equation (4), $u(c)$ (or $u(x_i)$) is the utility score of c (or x_i), and $I(p)$ is the set of attribute values in the range of cut p.

Example 3. The *Age* attribute in Table 1 is initially generalized to the [19, 75) cut. If the [19, 75) cut is selected to split, we calculate the utility score of each attribute value located in the range of [19, 75) and probabilistically select a value as the split value of the [19, 75) cut. Consider the first value 21 and the *Class* attribute in Table 1; then, based on Equation (4), $u(21)$ is calculated as follows:

$$H_{21}(D) = -(\frac{4}{10} \times log_2 \frac{4}{10} + \frac{6}{10} \times log_2 \frac{6}{10}) = 0.9709$$

$$H_{[19,21)}(D) = -(\frac{1}{1} \times log_2 \frac{1}{1} + \frac{0}{1} \times log_2 \frac{0}{1}) = 0$$

$$H_{[21,75)}(D) = -(\frac{3}{9} \times log_2 \frac{3}{9} + \frac{6}{9} \times log_2 \frac{6}{9}) = 0.9182$$

$$u(21) = H_{21}(D) - [\frac{1}{10} \times H_{[19,21)}(D) + \frac{9}{10} \times H_{[21,75)}(D)] = 0.1445$$

After all $u(\cdot)$ are calculated, Equation (5) is used to calculate the probability for each value to be selected as the real split value. Suppose 45 is selected as the real split value of the [19, 75) cut, then the taxonomy tree of the *Age* attribute will be expanded as the one shown in Fig. 3. □

4.2.3. Implementation

DPHeter is depicted in Algorithm 1. First, split values are initialized for d_{num} numerical attributes, where d_{num} is the number of numerical attributes (Line 4). Then, for each round of specialization, a cut is probabilistically selected (Line 7). If the cut is set-valued, its non-empty child nodes should be verified to determine whether they are really "non-empty" (Lines 8-10); if the cut is numerical, a split value is chosen for it (Lines 12-13). Note that the two situations are mutually exclusive. The exact number of records in each leaf partition node cannot be directly published because for different data, the number may be different. This difference can be masked by adding noise to the number of records in each node (Lines 16-18). For example, the dotted arrows in Fig. 2 describe this step. We use $\frac{\epsilon}{2}$ to guide the partition process and the rest $\frac{\epsilon}{2}$ to obtain the noisy size of leaf partition nodes. $\frac{\epsilon}{2}$ is distributed evenly to all steps of the partition process, so the privacy budgets assigned for these steps is set to $\frac{\epsilon}{2(d_{num}+2h)}$ (Line 3).

4.2.4. Analysis of the Privacy

Theorem 1. *DPHeter satisfies ϵ-differential privacy.*

Proof. In Line 5, *DPHeter* initializes a split value for each of d_{num} numerical attributes by the exponential mechanism. The privacy budget cost by each

Algorithm 1: *DPHeter*

```
Input:  $D$: raw dataset
        $\epsilon$: privacy budget
        $h$: number of specializations
Output: $D'$: anonymized dataset
initialize each value in $D$ to the root value of its corresponding taxonomy tree;
$\cup Cut_0 = \{$all root values$\}$;
$\epsilon' = \frac{\epsilon}{2(d_{num}+2h)}$;
choose a split value for each numerical attribute with the probability calculated by (5);
compute the utility score of each candidate $\forall v \in \cup Cut_0$ according to (4);
for $i$=1 to $h$ do
    select $p \in \cup Cut_{i-1}$ with the probability proportional to $exp(\frac{\epsilon'}{2\Delta u}u(v))$;
    if $p$ is set-valued then
        compute a noisy size for each of its non-empty child nodes using the Laplace
          mechanism with $\epsilon'$;
        the non-empty child nodes of noisy sizes $> 1/\epsilon'$ are determined “true non-empty”;
    specialize $p$ on $D$ and update $\cup Cut_i$;
    if $p$ is numerical then
        choose its split value with the probability calculated by (5);
    update the utility score for each new $Cut$ added to $\cup Cut_i$ according to (4);
$D' = \emptyset$;
for $\forall node \in \{$leaf nodes of the partition tree$\}$ do
    add noise to the number of records in $node$ using the Laplace mechanism with $\frac{\epsilon}{2}$;
    $D' = D' \cup \{$records in $node\}$;
return $D'$;
```

exponential mechanism is ϵ', so Line 5 guarantees $\epsilon' \times d_{num}$-differential privacy according to the sequential composition property (Property 1). In Line 7, *DPHeter* selects a cut to split using the exponential mechanism, and this step satisfies ϵ'-differential privacy. In Lines 8-10, the set-valued cut produces "non-empty" partition nodes by the Laplace mechanism with ϵ' privacy budget. In Lines 12-13, *DPHeter* probabilistically selects a split value for a new numerical cut by the exponential mechanism. Thus, if the cut is categorical, set-valued, or numerical, the required privacy budget cost by one specialization (Lines 6-14) will be ϵ', $2\epsilon'$, or $2\epsilon'$, respectively. The algorithm finally returns the fuzzy number of records in each group in Lines 16-18 by the Laplace mechanism. These steps guarantee $\frac{\epsilon}{2}$-differential privacy.

Each non-deterministic step of *DPHeter* is differentially-private, and the total privacy budget is not greater than ϵ. Therefore, *DPHeter* satisfies ϵ-differential privacy due to the sequential composition property. □

4.2.5. Analysis of the Time Complexity

Theorem 2. *The time complexity of DPHeter is bounded by* $O(h \times nlog_2^n)$.

Proof. In Algorithm 1, choosing a split value for a numerical attribute requires $O(nlog_2^n)$, where n is the size of the input dataset. Line 4 determines split values for d_{num} numerical attributes, which takes $O(d_{num} \times nlog_2^n)$. Line 5 is done by scanning the input dataset with d attributes, which takes $O(d \times nlog_2^n)$. Then instead of scanning all data records, *DPHeter* calculates utility scores based on some information maintained for candidates in $\cup Cut_i$; thus, Line 14 only requires $O(n)$. According to Definition 4, the cost of the exponential mechanism is proportional to the number of discrete cuts, from which the mechanism chooses a cut; thus, the cost of Line 7 is $O(|\cup Cut_i|)$, where $|\cup Cut_i|$ is the size of $\cup Cut_i$. Lines 12-13 selects a split value for a numerical attribute and requires $O(nlog_2^n)$. Usually $|\cup Cut_i|$ is much smaller than n. Thus, the cost of Lines 6-14 is $O(h \times nlog_2^n)$.

Other lines of Algorithm 1 can be done in constant $O(1)$ time. Hence, the total runtime of *DPHeter* is $O(h \times nlog_2^n)$. □

5. Experimental Evaluation

In this section, we evaluate the performance of our approach. First, we study the quality of the clusters by satisfying different differential privacies. Second, we evaluate the quality of the clusters of the anonymized dataset generated by our approach and those generated by a general method without focusing on cluster analysis during anonymization. Third, we investigate the impact of using different clustering algorithms before and after anonymization. Fourth, we evaluate the scalability of our approach.

All experiments were performed on a PC with a 3.4 GHz @Intel core i7 CPU and 16 GB of RAM running Windows 10 (64-bit). Each result presented below is the average over 5 runs.

5.1. Datasets

Two publicly available datasets, i.e., *Adult* and *MIMIC*-III, were used in our experiments. The *Adult*[3] dataset contains census records, and the literature indicates that it has been used extensively for testing anonymization approaches [24, 8, 22, 42, 16, 26]. In our experiments, we removed the class label and used this dataset for cluster analysis. In order to synthesize a heterogeneous dataset, we assumed an individual can have multiple occupations, and then we combined records with the same attribute values, with the exception of *occupation*, into one record, thereby making the *occupation* attribute as set-valued. For the synthesis processing, we abandoned three numerical attributes (i.e., *fnlwg*, *capital-gain*, and *capital-loss*) because they could result in fewer heterogeneous records. Thus, we retained 28,308 records with seven categorical attributes, six numerical attributes, and one set-valued attribute. For simplicity, we also called the synthesized dataset *Adult*.

The second dataset, *MIMIC*-III [49], is an important public source for healthcare research. It consists of some tables of clinical notes, including nursing

[3] https://archive.ics.uci.edu/ml/datasets/adult

records and discharge summaries. Specifically, we joined three tables, i.e., *ADMISSIONS*, *PATIENTS*, and *DIAGNOSES_ICD*, together based on the *subject_id* column they shared. Then, we combined multiple ICD-9 codes of the same *subject_id* into one row. We retrieved 48,612 records and selected seven categorical attributes (i.e., *gender*, *marital status*, *religion*, *ethnicity*, *admission type*, *insurance method*, and *admission source*) and one set-valued attribute (i.e., *ICD-9 codes*). *MIMIC* is the abbreviated form of *MIMIC*-III in the following.

5.2. Clustering Algorithms

We chose k-means [50] and bisecting k-means [51] to get clusters in steps ① and ③ in Fig. 1 because they contain only one algorithmic parameter, i.e., the number of clusters, k. Rather than considering different combinations of clustering parameters, we focus on the evaluation of the performance of our approach. Any clustering algorithm requires certain method of measuring the distance or the similarity between objects. We here introduce a semantic distance metric of two heterogeneous records. If we let x_1, x_2 denote two attribute values from the same domain, the distance between x_1 and x_2 is calculated as:

$$dist(x_1, x_2) = \frac{path(x_1, x_2)}{2H}, \tag{6}$$

where $path(x_1, x_2)$ is the length of the shortest path between x_1 and x_2, and H is the height of the corresponding taxonomy tree. The advantage of the normalized definition is that all leaf nodes of the taxonomy tree can have different depths. The distance between two heterogeneous records, i.e., r_1 and r_2, is defined as:

$$dist(r_1, r_2) = \sum_{i=1}^{d} w_i \times dist(x_1^i, x_2^i), \tag{7}$$

where d is the number of attributes, and w_i $(0 < w_i < 1)$ is the weight for the i^{th} attribute for flexibility. In our experiments, we set all w_i $(1 \leq i \leq d, \sum_{i=1}^{d} w_i = 1)$ as equal.

5.3. Metrics

The goal of PPDP is to protect the private information of the raw dataset while preserving considerable data utility. We measured the data utility by the similarity of the cluster structures before and after anonymization. That is, the more similar the cluster structures before and after anonymization are, the higher the utility of the anonymized dataset is. In our experiments, two metrics, i.e., *F-Measure* and *MatchPoint*, were used to measure the similarity of two cluster structures.

5.3.1. F-Measure

F-Measure [52] is used extensively to evaluate the similarity of two cluster structures. Consider two cluster structures $\mathbb{T}$ and $\mathbb{P}$, and treat each cluster T_i in $\mathbb{T}$ as a "true cluster", and treat each cluster P_j in $\mathbb{P}$ as a "prediction cluster". Let num_{ij} denote the number of records contained in both T_i and P_j, and let $|\cdot|$ denote the number of objects in a cluster. The *Precision*, *Recall*, and *F-Measure* of T_i and P_j are calculated as:

$$Precision(T_i, P_j) = \frac{num_{ij}}{|P_j|}, \tag{8}$$

which is the ratio of true relevant records in the prediction cluster divided by all records in the prediction cluster,

$$Recall(T_i, P_j) = \frac{num_{ij}}{|T_i|}, \tag{9}$$

which is the ratio of true relevant records in the prediction cluster divided by all records in the true cluster, and

$$F(T_i, P_j) = 2 \times \frac{Precision(T_i, P_j) \times Recall(T_i, P_j)}{Precision(T_i, P_j) + Recall(T_i, P_j)}, \tag{10}$$

which measures the accuracy of the prediction of cluster P_j, which describes the true cluster T_i based on *Precision* and *Recall*.

The successful prediction of a true cluster T_i is measured by the "best" prediction cluster P_j for T_i, i.e., P_j maximizes $F(T_i, P_j)$. Thus, the sum of the weighted maximum *F-Measures* is used to evaluate the quality of $\mathbb{P}$, and the overall *F-Measures* of $\mathbb{P}$ are calculated as:

$$F\text{-}Measures(\mathbb{P}) = \sum_{T_i \in \mathbb{T}} \frac{|T_i|}{|D|} \max_{P_j \in \mathbb{P}} F(T_i, P_j), \tag{11}$$

where $|D|$ is the number of records in the raw dataset, D. *F-Measures*($\mathbb{P}$) is in the range of [0, 1]. The larger the value of *F-Measures*($\mathbb{P}$) is, the more similar are the two cluster structures that are compared.

5.3.2. MatchPoint

Two cluster structures $\mathbb{C}_1$ and $\mathbb{C}_2$ are treated as similar if (1) two records that stay in the same cluster in $\mathbb{C}_1$ are kept together in $\mathbb{C}_2$, and (2) two records that stay in different clusters in $\mathbb{C}_1$ are divided into different clusters in $\mathbb{C}_2$ [53]. For each cluster structure, a square matrix $Matrix(\cdot)$ is generated to represent the relationship between each pair of records. That is, the $(i, j)^{th}$ element in $Matrix(\cdot)$ is equal to 1 if the i^{th} record and the j^{th} record are in the same cluster; otherwise it is equal to 0. Then, *MatchPoint* is defined to represent the percentage of the same values appearing in $Matrix(\mathbb{C}_1)$ and $Matrix(\mathbb{C}_2)$:

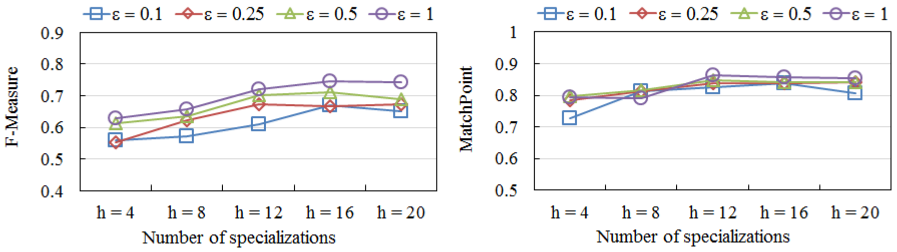
$$MatchPoint(Matrix(\mathbb{C}_1), Matrix(\mathbb{C}_2)) = \frac{\sum\limits_{1 \le i,j \le |D|} m_{ij}}{|D|^2}, \tag{12}$$

where m_{ij} is equal to 1 if the values of the $(i, j)^{th}$ element in $Matrix(\mathbb{C}_1)$ and $Matrix(\mathbb{C}_2)$ are the same; otherwise, m_{ij} is equal to 0, and $|D|$ is the number of records in the raw dataset, D. *MatchPoint* is in the range of [0, 1]. The larger the value of *MatchPoint* is, the more similar are the two cluster structures that are compared.

5.4. Analysis of the Results

5.4.1. Data Utility and Privacy

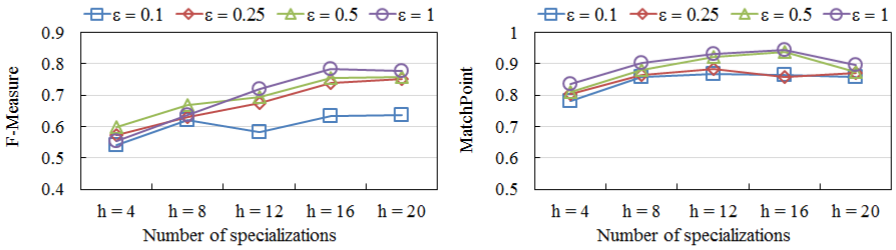
In this experiment, we varied the privacy budget ϵ, the number of specializations h, and the number of clusters k, to observe *F-Measure* and *MatchPoint*. Figs. 4-7 show the results on *Adult*. Among these results, Fig. 5a shows that the minimum *F-Measure* was 0.5408 when $\epsilon = 0.1$ and $h = 4$. Fig. 5a also shows that the maximum *F-Measure* was 0.7840 when $\epsilon = 1$ and $h = 16$. Compared with the *F-Measure*, the spans of the *MatchPoint* values for different values of ϵ and h were smaller, roughly in the range of [0.7270, 0.9314]. There was an obvious trend that indicated that the values of *F-Measure* increased as ϵ increased since a higher ϵ resulted in less perturbation and less noise. In addition, *F-Measure* and *MatchPoint* also increased as h increased because more detailed information was preserved in the anonymized dataset for clustering. However, starting from a certain level of h, *F-Measure* and *MatchPoint* remained the same or decreased as h increased further. This is because a higher value of h corresponded to more leaf nodes in the partition tree, and the greater the number of leaf nodes became, the more noise was produced from the Laplace mechanism that was acting on the number of records in these leaf nodes. Figs. 8-11 show the similar trends of the *F-Measure* and *MatchPoint* values for *MIMIC*, with the only difference being in the case of the values of ϵ and h when getting the best performance. These results demonstrate that *DPHeter* can keep a similar cluster structure of the raw dataset after anonymization even for different anonymity requirements.



(a) *F-Measure*

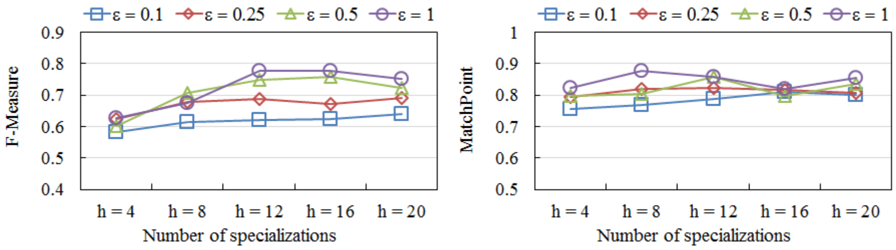
(b) *MatchPoint*

Fig. 4. Data utility of the anonymized *Adult* over 3-means



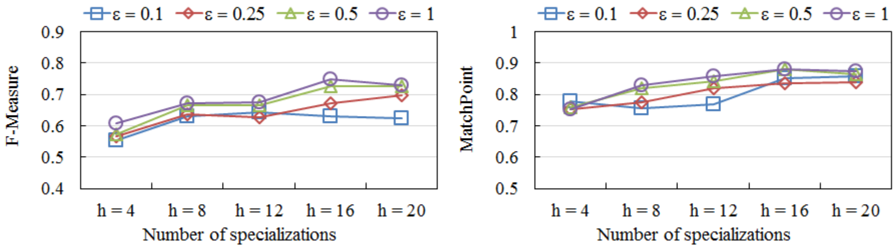
(a) *F-Measure* (b) *MatchPoint*

Fig. 5. Data utility of the anonymized *Adult* over 5-means



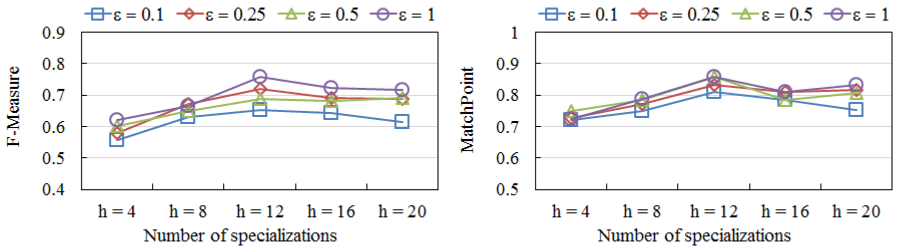
(a) *F-Measure* (b) *MatchPoint*

Fig. 6. Data utility of the anonymized *Adult* over bisecting 3-means



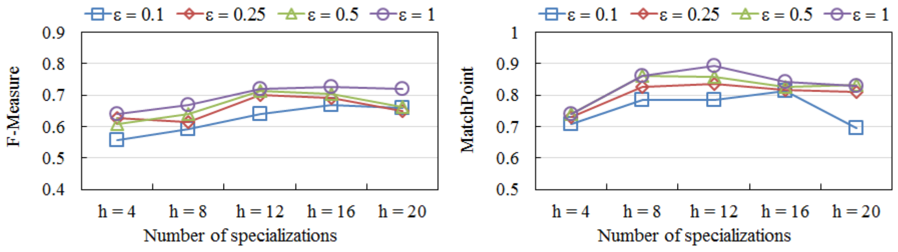
(a) *F-Measure* (b) *MatchPoint*

Fig. 7. Data utility of the anonymized *Adult* over bisecting 5-means



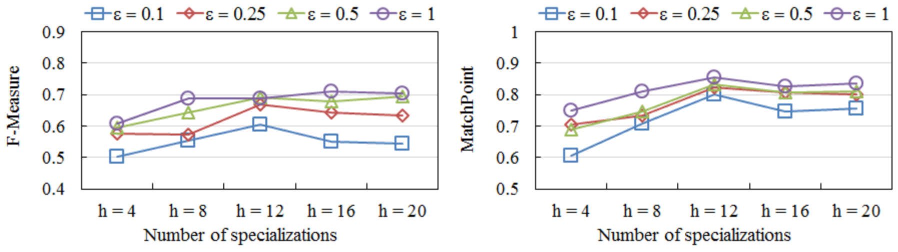
(a) *F-Measure* (b) *MatchPoint*

Fig. 8. Data utility of the anonymized *MIMIC* over 3-means



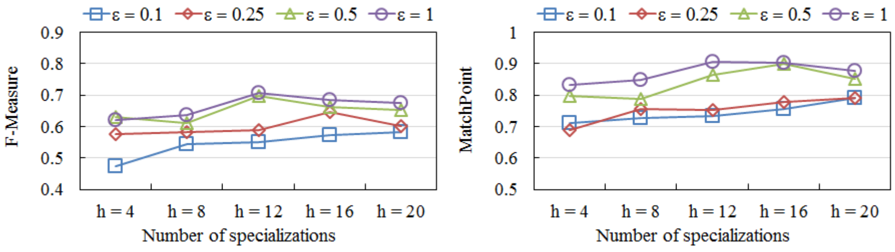
(a) *F-Measure* (b) *MatchPoint*

Fig. 9. Data utility of the anonymized *MIMIC* over 5-means



(a) *F-Measure* (b) *MatchPoint*

Fig. 10. Data utility of the anonymized *MIMIC* over bisecting 3-means



(a) *F-Measure*

(b) *MatchPoint*

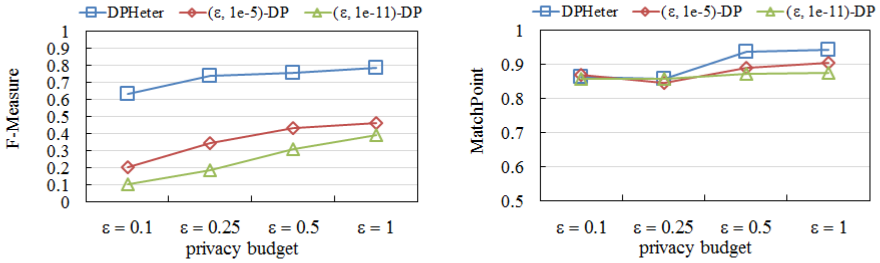
Fig. 11. Data utility of the anonymized *MIMIC* over bisecting 5-means

5.4.2. Data Utility over Different Anonymization Algorithms

To verify whether the cluster quality of our cluster-oriented algorithm is better than that of a generally differentially-private method without such focus, we compared our algorithm with the (ϵ, δ)-differential privacy [54] in the ARX tool [55]. The (ϵ, δ)-differential privacy is a relaxation version of ϵ-differential privacy since the former allows an error probability bounded by δ. Because only relational data can be input to ARX, we first converted heterogeneous *Adult* and *MIMIC* into relational data. Specifically, a binary attribute would be created for each value of set-valued attributes. For example, if an attribute is set-valued and has 2 values, i.e., x_1 and x_2, then the pattern for records will be "0 1", "1 0", or "1 1". Such conversion is only executed for ARX, not for *DPHeter*. We set δ = 1E-5 and δ = 1E-11 for (ϵ, δ)-differential privacy because the two values are the maximal and minimal acceptable values, respectively, in the case of the tool. We fixed $h = 16$ for *DPHeter*.

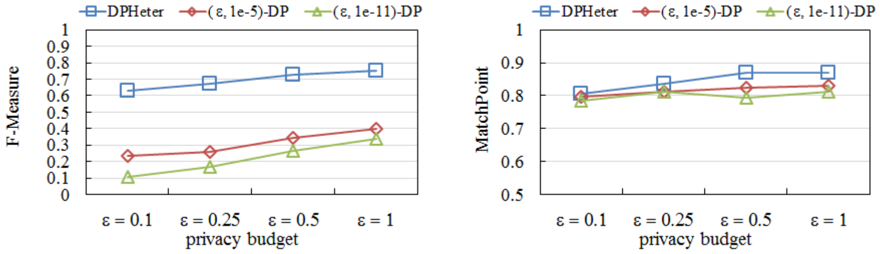
Figs. 12-15 show the results. These figures suggest that the values of *F-Measure* of *DPHeter* clearly were better than those of (ϵ, δ)-differential privacy over every privacy budget. For example, in Figs. 12a and 14a, even when $\epsilon = 0.1$, our *F-Measure* for *Adult* was 0.6331, and the one for *MIMIC* was 0.6428, while the *F-Measures* of (ϵ, δ)-differential privacy for *Adult* and *MIMIC* were only 0.2015 and 0.3328 respectively when δ = 1E-5. However, the differences between the *MatchPoint* values were smaller. This is because the cases in which two records stayed in different clusters before and after anonymization also contributed to the value of *MatchPoint*.

We also conducted a series of one-tailed t-tests on pairs of test cases when $0.1 \leq \epsilon \leq 1$ to evaluate the improvement of *DPHeter* over the (ϵ, 1E-5)-differential privacy in the ARX tool. The results shown in Table 3 demonstrate that the improvement of *DPHeter* was statistically significant at $\alpha = 5\%$. From these results, it can be claimed that our approach outperformed the general anonymization method in terms of cluster quality. In addition, the improvement was unlikely to have happened by accident.



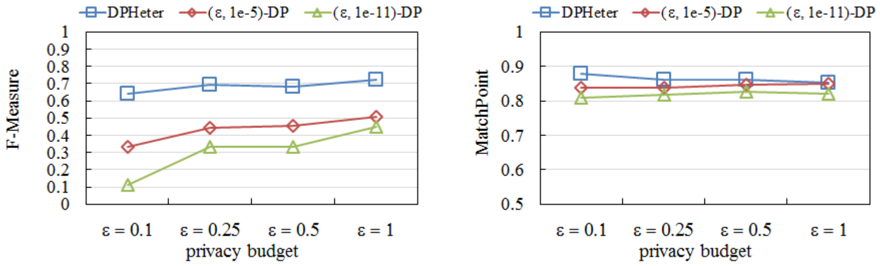
(a) *F-Measure* (b) *MatchPoint*

Fig. 12. Different anonymization algorithms on *Adult* using 5-means



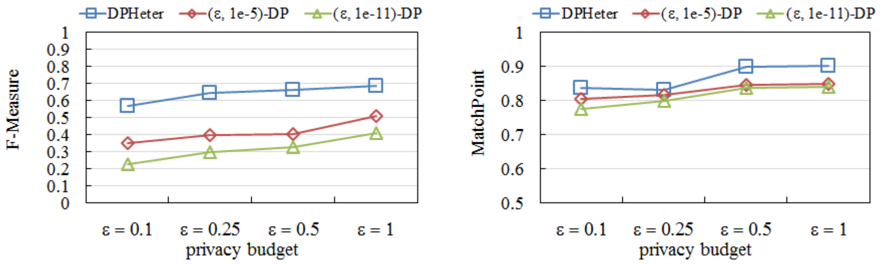
(a) *F-Measure* (b) *MatchPoint*

Fig. 13. Different anonymization algorithms on *Adult* using bisecting 5-means



(a) *F-Measure* (b) *MatchPoint*

Fig. 14. Different anonymization algorithms on *MIMIC* using 5-means



(a) *F-Measure* (b) *MatchPoint*

Fig. 15. Different anonymization algorithms on *MIMIC* using bisecting 5-means

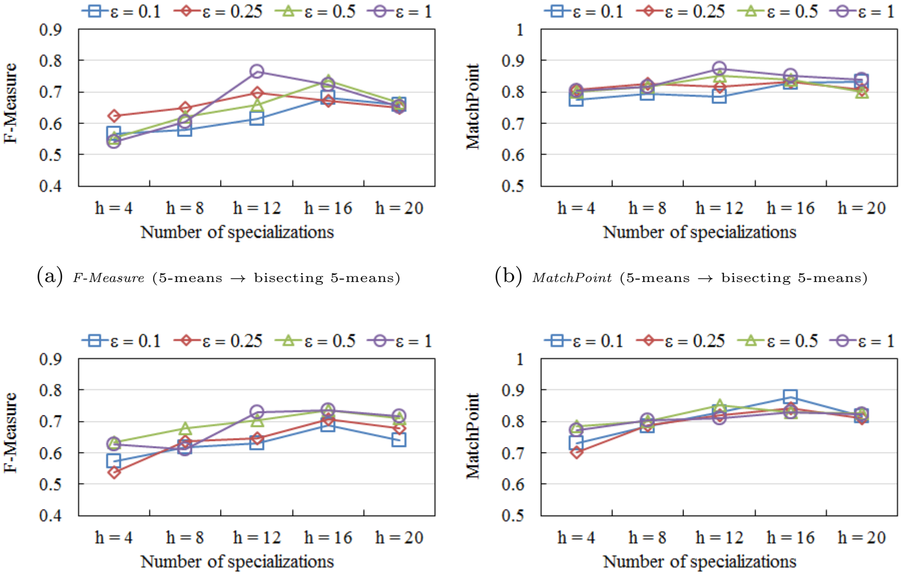
Table 3. The p-values for one-tailed t-tests on *F-Measure* and *MatchPoint*

	5-means		bisecting 5-means	
	F-Measure	*MatchPoint*	*F-Measure*	*MatchPoint*
Adult	7.47E-8	2.21E-5	8.26E-9	6.94E-5
MIMIC	3.93E-8	9.13E-6	8.03E-8	7.23E-5

5.4.3. Data Utility over Different Clustering Algorithms

In this experiment, we studied the data utility in the case that the data recipient applies a different clustering algorithm from the one used by the data owner. That is, different clustering algorithms were used in steps ① and ③ in Fig. 2. We applied 5-means and bisecting 5-means in two different orders, denoted by (bisecting 5-means → 5-means) and (5-means → bisecting 5-means). Figs. 16-17 show the data utility of the anonymized *Adult* and *MIMIC*, respectively. Except for the cases of $\epsilon = 0.25$ and $h = 4$ in Fig. 17c, all values of *F-Measure* were higher than 0.5. Specifically, the largest *F-Measure* values were 0.7660 and 0.7577 for *Adult* and *MIMIC* in Figs. 16a and 17a, respectively. All values of *MatchPoint* were higher than 0.7206, and the average *MatchPoint* values were 0.8127 and 0.8401 for *Adult* and *MIMIC*, respectively. These results suggest that *DPHeter* can obtain good data utility even using different clustering algorithms. Note that the distance metric between records used in these different clustering algorithms should remain the same or be similar. Otherwise, the cluster structures produced by different clustering algorithms may be totally different.

Compared with the experimental results in subsection 5.4.2, the data utility over different clustering algorithms may not be very stable. For example, in Fig. 17c the average value of *F-Measure* at $h = 20$ was only 0.5719, which was smaller than the average values at $h = 12$ and $h = 16$. It can be concluded that the cluster structure produced by one clustering algorithm may be different from the structure produced by another clustering algorithm because of their different search criteria.



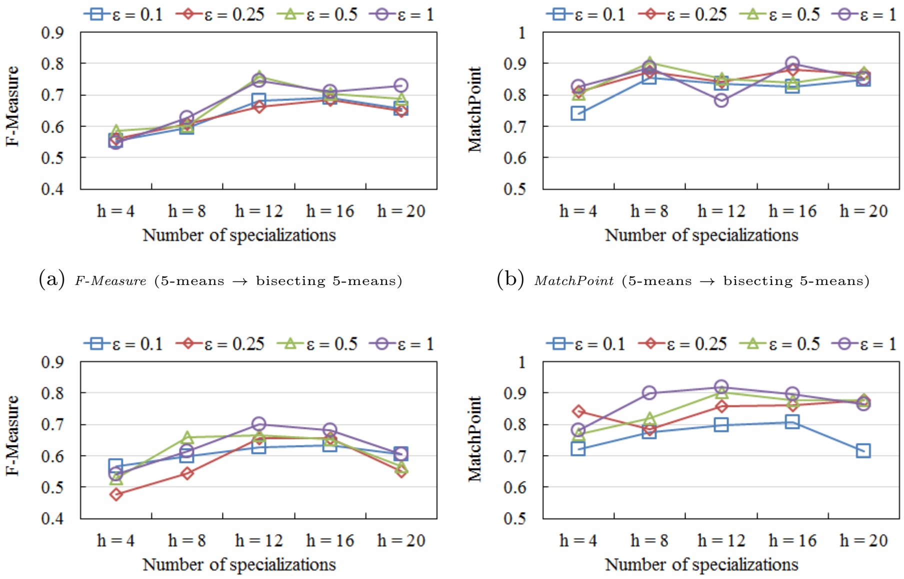
(a) *F-Measure* (5-means → bisecting 5-means)

(b) *MatchPoint* (5-means → bisecting 5-means)

(c) *F-Measure* (bisecting 5-means → 5-means)

(d) *MatchPoint* (bisecting 5-means → 5-means)

Fig. 16. Different clustering algorithms on *Adult*



(a) *F-Measure* (5-means → bisecting 5-means)

(b) *MatchPoint* (5-means → bisecting 5-means)

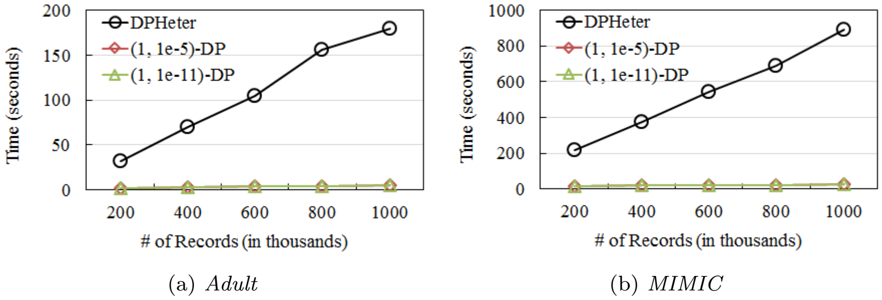
(c) *F-Measure* (bisecting 5-means → 5-means)

(d) *MatchPoint* (bisecting 5-means → 5-means)

Fig. 17. Different clustering algorithms on *MIMIC*

5.4.4. Scalability

DPHeter was compared with the $(\epsilon,\ \delta)$-differential privacy in ARX in terms of scalability. Similar to the experiments in Section 5.4.2, we set δ = 1E-5 and δ = 1E-11 for (ϵ, δ)-differential privacy and $h = 16$ for *DPHeter*. We also fixed $\epsilon = 1$ and did 5-means clustering. We generated multiple versions of *Adult* and *MIMIC* by randomly duplicating their records. For comparison, Fig. 18 shows the results of *DPHeter* and ARX on *Adult* and *MIMIC* with 200,000 to 1000,000 data records. This figure shows that ARX is more efficient than *DPHeter* in terms of runtime because ARX does not consider data analysis tasks. When searching for split values for the numerical attributes, *DPHeter* calculates the utility scores of all possible numerical values in the current value range. When splitting set-valued attributes, *DPHeter* considers a combination of child nodes of the current parent node according to the taxonomy tree. We accelerated the running speed of *DPHeter* by maintaining and updating information, which was required by each utility score calculation, instead of repeatedly scanning all data records. Also, it was evident that the time spent on *MIMIC* was more than that the time spent on *Adult*. This is because there are thousands of ICD-9 codes in *MIMIC*, and the corresponding taxonomy tree is much larger than that of the *occupation* attribute in *Adult*, which means more calculation time is required when the ICD-9 code attribute is selected to split.



(a) *Adult*

(b) *MIMIC*

Fig. 18. Scalability on *Adult* and *MIMIC*

6. Discussion

Adaptability of DPHeter. Although only k-means and bisecting k-means were used in Section 5 to evaluate the performance of *DPHeter*, other clustering algorithms, such as DBSCAN [56], can be integrated into our approach; namely, other clustering algorithms can be applied to steps ① and ③ in Fig. 1. Our proposed approach provides a flexible framework in which the clustering algorithms can be viewed as “plug-in” components. *DPHeter* utilizes the clustering results to anonymize the raw data, not the clustering algorithms. However, it is worth noting that the distance metric used for clustering before and after data anonymization should remain the same, or at least be similar, for better data utility. Otherwise, the cluster structures produced by different clustering strategies may be totally different. We also emphasize that the focus of *DPHeter* is on preserving the similarity of cluster structures before and after data publication. If the raw data are not suitable for cluster analysis or produce a poor clustering result by some clustering algorithm, *DPHeter* cannot help the data or their anonymous version yield a better one.

Optimality of DPHeter. *DPHeter* produces a sub-optimal solution rather than the optimal solution. This is because it utilizes the exponential mechanism to probabilistically select split values for numerical attributes and cuts to be specialized, which means different selections might provide better anonymization. However, this also is an inherent limitation of differential privacy techniques.

7. Conclusions and Future Work

In this paper, we introduced an approach to release heterogeneous data for cluster analysis. The proposed approach utilizes cluster labels to encode the cluster structure and combines the generalization technique with output perturbation to mask raw data. The experimental results showed that the utility of the anonymized data produced by our cluster-oriented approach was significantly better than that of the anonymized data produced by the method without initially considering cluster analysis.

We have planned some directions for our future work. First, we will extend our differentially-private centralized algorithm to the scenario of distributed data publications for cluster analysis. Secure protocols must be studied to exchange information among different parties. Second, the generalization technique is context-dependent and cannot handle high-dimensional data. Other anonymization techniques combined with differential privacy are worth considering.

References

[1] M. Janssen, Y. Charalabidis, A. Zuiderwijk, Benefits, adoption barriers and myths of open data and open government, Information Systems Management 29 (4) (2012) 258–268.

[2] P. Doshi, T. Jefferson, C. D. Mar, The imperative to share clinical study reports: Recommendations from the tamiflu experience, PLoS Medicine 9 (4) (2012) e1001201.

[3] The HIPAA Privacy Rule, Available at `https://www.hhs.gov/hipaa/for-professionals/privacy/index.html`, online; Accessed: April 06, 2019.

[4] The PIPEDA Privacy Law, Available at `https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/`, online; Accessed: April 06, 2019.

[5] A. H. Rashid, N. B. M. Yasin, Privacy preserving data publishing, International Journal of Physical Sciences 10 (7) (2015) 239–247.

[6] C. Dwork, Differential privacy, in: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Vol. 4052 of Lecture Notes in Computer Science, Springer, 2006, pp. 1–12.

[7] T. Zhu, G. Li, W. Zhou, S. Y. Philip, Differentially private data publishing and analysis: A survey, IEEE Transactions on Knowledge and Data Engineering 29 (8) (2017) 1619–1638.

[8] D. Sánchez, J. Domingo-Ferrer, S. Martínez, J. Soria-Comas, Utility-preserving differentially private data releases via individual ranking microaggregation, Information Fusion 30 (2016) 1–14.

[9] A. Friedman, A. Schuster, Data mining with differential privacy, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 493–502.

[10] O. Jia, Y. Jian, S. Liu, Y. Liu, An effective differential privacy transaction data publication strategy, Journal of Computer Research and Development 51 (10) (2014) 2195–2205.

[11] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, L. Xiong, Publishing set-valued data via differential privacy, Proceedings of the VLDB Endowment 4 (11) (2011) 1087–1098.

[12] J. Lee, C. W. Clifton, Top-k frequent itemsets via differentially private fp-trees, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 931–940.

[13] T. Wang, N. Li, S. Jha, Locally differentially private frequent itemset mining, in: Proceedings of the 2018 IEEE Symposium on Security and Privacy, IEEE, 2018, pp. 127–143.

[14] M. Maruseac, G. Ghinita, Precision-enhanced differentially-private mining of high-confidence association rules, IEEE Transactions on Dependable and Secure Computing (2018) 1–1, Early Access.

[15] Z. Sun, Y. Wang, M. Shu, R. Liu, H. Zhao, Differential privacy for data and model publishing of medical data, IEEE Access 7 (2019) 152103–152114.

[16] D. Su, J. Cao, N. Li, M. Lyu, PrivPfC: Differentially private data publication for classification, The VLDB Journal 27 (2) (2018) 201–223.

[17] Y. Zhang, Z. Hao, S. Wang, A differential privacy support vector machine classifier based on dual variable perturbation, IEEE Access 7 (2019) 98238–98251.

[18] P. Samarati, Protecting respondents identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13 (6) (2001) 1010–1027.

[19] L. Sweeney, k-Anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (5) (2002) 557–570.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, l-Diversity: Privacy beyond k-anonymity, in: Proceedings of the 22nd International Conference on Data Engineering, IEEE, 2006, pp. 24–24.

[21] N. Li, T. Li, S. Venkatasubramanian, t-Closeness: Privacy beyond k-anonymity and l-diversity, in: Proceedings of the 23rd International Conference on Data Engineering, IEEE, 2007, pp. 106–115.

[22] F. Amiri, N. Yazdani, A. Shakery, A. H. Chinaei, Hierarchical anonymization algorithms against background knowledge attack in data releasing, Knowledge-Based Systems 101 (2016) 71–89.

[23] S. Agarwal, S. Sachdeva, An enhanced method for privacy-preserving data publishing, in: Innovations in Computational Intelligence, Springer, 2018, pp. 61–75.

[24] H. Zhu, S. Tian, L. Kevin, Privacy-preserving data publication with features of independent l-diversity, The Computer Journal 58 (2015) 549–571.

[25] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, S. Martinez, t-Closeness through microaggregation: Strict privacy with enhanced utility preservation, IEEE Transactions on Knowledge and Data Engineering 27 (11) (2015) 3098–3110.

[26] R. Wang, Y. Zhu, T.-S. Chen, C.-C. Chang, Privacy-preserving algorithms for multiple sensitive attributes satisfying t-closeness, Journal of Computer Science and Technology 33 (6) (2018) 1231–1242.

[27] N. Mohammed, R. Chen, B. C. M. Fung, P. S. Yu, Differentially private data release for data mining, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 4052, ACM, 2011, pp. 493–501.

[28] M. Terrovitis, N. Mamoulis, P. Kalnis, Local and global recoding methods for anonymizing set-valued data, The VLDB Journal 20 (1) (2011) 83–106.

[29] M. Bewong, J. Liu, L. Liu, J. Li, Utility aware clustering for publishing transactional data, in: Proceedings of the 21st Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2017, pp. 481–494.

[30] H. Zhang, Z. Zhou, L. Ye, X. Du, Towards privacy preserving publishing of set-valued data on hybrid cloud, IEEE Transactions on Cloud Computing 6 (2) (2018) 316–329.

[31] D. Gunawan, M. Mambo, Set-valued data anonymization maintaining data utility and data property, in: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, ACM, 2018, pp. 1–8.

[32] V. Puri, S. Sachdeva, P. Kaur, Privacy preserving publication of relational and transaction data: Survey on the anonymization of patient data, Computer Science Review 32 (2019) 45–61.

[33] G. Poulis, G. Loukides, A. Gkoulalas-Divanis, S. Skiadopoulos, Anonymizing data with relational and transaction attributes, in: Proceedings of the 2013 Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 353–369.

[34] J. Wang, S. Zhou, J. Wu, C. Liu, A new approach for anonymizing relational and transaction data, in: Proceedings of the 2nd International Conference on Healthcare Science and Engineering, Springer, 2018, pp. 251–261.

[35] L.-E. Wang, X. Li, A graph-based multifold model for anonymizing data with attributes of multiple types, Computers & Security 72 (2018) 122–135.

[36] Q. Gong, J. Luo, M. Yang, W. Ni, X.-B. Li, Anonymizing 1: M microdata with high utility, Knowledge-Based systems 115 (2017) 15–26.

[37] T. Kanwal, S. A. A. Shaukat, A. Anjum, K.-K. R. Choo, A. Khan, N. Ahmad, M. Ahmad, S. U. Khan, et al., Privacy-preserving model and generalization correlation attacks for 1: M data with multiple sensitive attributes, Information Sciences 488 (2019) 238–256.

[38] N. Mohammed, X. Jiang, R. Chen, B. C. M. Fung, L. Ohno-Machado, Privacy-preserving heterogeneous health data sharing, Journal of the American Medical Informatics Association 20 (3) (2013) 462–469.

[39] M. Gong, K. Pan, Y. Xie, Differential privacy preservation in regression analysis based on relevance, Knowledge-Based Systems 173 (2019) 140–149.

[40] 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, ACM, 2016, pp. 26–37.

[41] H. H. Nguyen, Privacy-preserving mechanisms for k-modes clustering, Computers & Security 78 (2018) 60–75.

[42] X. Liu, Q. Li, T. Li, D. Chen, Differentially private classification with decision tree ensemble, Applied Soft Computing 62 (2018) 807–816.

[43] Z. Lu, H. Shen, A convergent differentially private k-means clustering algorithm, in: Proceedings of the 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 612–624.

[44] J. Wang, S. Liu, Y. Li, A review of differential privacy in individual data release, International Journal of Distributed Sensor Networks 11 (10) (2015) 259682.

[45] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography, Springer, 2006, pp. 265–284.

[46] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2007, pp. 94–103.

[47] F. McSherry, Privacy integrated queries: An extensible platform for privacy-preserving data analysis, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, ACM, 2009, pp. 19–30.

[48] B. C. M. Fung, K. Wang, P. S. Yu, Anonymizing classification data for privacy preservation, IEEE Transactions on Knowledge and Data Engineering 19 (5) (2007) 711–725.

[49] A. E. Johnson, T. J. Pollard, L. Shen, L. wei H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, MIMIC-III, a freely accessible critical care database, Scientific Data 3 (2016) 160035.

[50] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[51] L. Kaufman, P. J. Rousseeuw, Finding groups in data: An introduction to cluster analysis, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[52] C. V. Rijsbergen, Information Retrieval, 2nd Edition, Butterworths, London, 1979.

[53] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1) (1985) 193–218.

[54] N. Li, W. Qardaji, D. Su, On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ACM, 2012, pp. 32–33.

[55] F. Prasser, J. Gaupp, Z. Wan, W. Xia, Y. Vorobeychik, M. Kantarcioglu, K. Kuhn, B. Malin, An open source tool for game theoretic health data de-identification, in: Proceedings of the American Medical Informatics Association 2017 Annual Symposium, 2017, pp. 1430–1439.

[56] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Vol. 96, 1996, pp. 226–231.