Privacy Preserving Dynamic Data Release Against Synonymous Linkage Based on Micro Aggregation

Yan Yan (yanyan@lut.edu.cn)
Lanzhou University of Technology
Eyeleko Herman
Lanzhou University of Technology
Adnan Mahmood
Macquarie University
Jing Li
Lanzhou University of Technology
Zhuoyue Dong
Lanzhou University of Technology
Fei Xu
Lanzhou University of Technology

Research Article

Keywords: privacy preserving data publishing, K-anonymity, l-diversity, synonymous linkage, micro aggregation

Posted Date: August 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-828185/v1

Privacy Preserving Dynamic Data Release Against Synonymous Linkage Based on Micro Aggregation

Yan Yan[1*] , Eyeleko Anselme Herman[1], Adnan Mahmood[2], Jing Li[1], Zhuoyue Dong[1], Fei Xu[1]

1. School of Computer and Communication, Lanzhou University of Technology
Lanzhou, 730050, China

2. Department of Computing, Faculty of Science and Engineering, Macquarie University
Sydney, NSW 2109, Australia

*corresponding author: yanyan@lut.edu.cn

Telephone: +86 18894013443

Funding: the research at hand was supported by the National Nature Science Foundation of China (No. 61762059).

Conflicts of interest: the authors declare no conflicts of interest, and agree to the arrangement of the author list.

Ethical statement: all authors declare that this research does not involve human and/or animal research, and does not violate morals and ethics.

Author contribution: methodology and validation, Y.Y.; investigation and formal analysis, E.A. and A.M.; data curation and algorithm simulation, J.L., Z.D., and F.X.; original draft preparation, Y.Y., A.M. and E.A.; writing, review and editing, Y.Y., A.M. and E.A.; visualization, J.L., Z.D., and F.X.

Availability of data and material: the dataset used in this paper is selected from the Adult dataset from the UCI machine learning repository, which is available online: http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

Privacy Preserving Dynamic Data Release Against Synonymous Linkage Based on Micro Aggregation

Yan Yan[1*], **Eyeleko Anselme Herman**[1], **Adnan Mahmood**[2], **Jing Li**[1], **Zhuoyue Dong**[1], **Fei Xu**[1]
[1] School of Computer and Communication, Lanzhou University of Technology
Lanzhou, 730050, China
[2] Department of Computing, Faculty of Science and Engineering, Macquarie University
Sydney, NSW 2109, Australia
*Correspondence: yanyan@lut.edu.cn

ABSTRACT

The rapid development of the mobile Internet coupled with the widespread use of intelligent terminals have intensified the digitization of personal information and accelerated the evolution of the era of big data. The sharing and publishing of various big data brings convenience and also increases the risk of personal privacy leakage. In order to reduce users' privacy leakage that may be caused by data release, many privacy preserving data publishing methods have been proposed by scientists in both academic and industry in the recent years. However, non-numerical sensitive information has natural semantic relevance,and therefore, synonymous linkages may still exist and cause serious privacy disclosures in privacy protection methods based on an anonymous model. To address this issue, this paper proposes a privacy preserving dynamic data publishing method based on micro aggregation. A series of indicators are accordingly designed to evaluate the synonymous linkages between the non-numerical sensitive values which in turn facilitate in improving the clustering effect of the micro-aggregation anonymous method. The dynamic update program is introduced into the proposed micro-aggregation method to realize the dynamic release and update of data. Experimental analysis suggests that the proposed method provides better privacy protection effect and availability of published data in contrast to the state-of-the-art methods.

keywords: privacy preserving data publishing; K-anonymity; l-diversity; synonymous linkage; micro aggregation.

1 Introduction

Big data is rich in sources, diverse in variety, large in volume,and dynamically updated, and therefore, it contains immense value and information. Big data mining and analysis can be applied to social demographic surveys, public health research, transportation planning, social public opinion analysis, business model surveys and adjustments, agricultural output prediction, biological information analysis, and many other areas of technological innovation and application. Therefore, it has recently attracted the attention of governments, industries, and research departments around the world [1-3]. However, big data is a double-edged sword. Big data publishing without reasonable privacy protection is likely to cause the leakage of sensitive information which may endanger the personal safety of users and of their property, affect personal reputation and physical and mental health, or lead to discriminatory treatments [4-6].

Traditionally, privacy preserving data publishing was realized by deleting identifying attributes that can uniquely identify an individual. Sweeney and Samarati proved that the data subsequent to deleting of the identity attribute may still disclosure an individual's privacy through linking attacks. Therefore, they proposed the K-anonymity privacy protection model [7][8] to deal with privacy leakage. Subsequent theoretical and practical results suggest that although the K-anonymity privacy model cuts off the connection between an individual and his/her record in the published data, there still exists some connections between an individual and his/her sensitive information. For instance,if all the records within the same equivalence class have the same or similar sensitive values, the attacker will directly obtain the sensitive information of all the individuals in the same equivalence class. The l-diversity privacy protection model proposed by Machanavajjhala et al. [9] requires that each equivalence class contains at least l different sensitive values, thereby reducing the ability of the malicious attackers to infer sensitive information. However, non-numerical sensitive information has natural semantic relevance and it is,therefore,

Table 1. A micro patient table.

Name	*Age*	*Sex*	*Zipcode*	*Disease*
Alice	58	F	10030	anemia
Peter	49	M	10037	enteritis
Wilson	49	M	10022	anemia
Jim	51	M	10029	lymphoma
Rose	50	F	10033	leukemia
Jenny	33	F	10019	enteritis
John	36	M	10013	bronchitis
Karine	37	F	10010	anemia
Bob	51	M	10024	leukemia

Table 2. The 3-anonymous edition of Table 1

GID	*Age*	*Sex*	*Zipcode*	*Disease*
1	[33-37]	*	1001*	anemia
1	[33-37]	*	1001*	enteritis
1	[33-37]	*	1001*	bronchitis
2	[49-51]	M	1002*	anemia
2	[49-51]	M	1002*	leukemia
2	[49-51]	M	1002*	lymphoma
3	[49-58]	*	1003*	anemia
3	[49-58]	*	1003*	leukemia
3	[49-58]	*	1003*	enteritis

difficult to avoid synonymous linkage through different sensitive values. For example, Table 1 depicts a micro patient table with quasi-identifiers $\{$Age, Sex, Zipcode$\}$ and sensitive attribute $\{$Disease$\}$. Table 2 is a 3-anonymous edition of Table 1 since each group contains at least 3 different records. If an attacker is able to determine that Wilson belongs to group 2 through some background knowledge, then he/she can definitely infer that Wilson has blood disease. Primarily since "anemia", "leukemia", and "lymphoma" all belong to $\{$Disease$\}$.

In order to prevent this kind of privacy leakage caused by synonymous linkage of sensitive values, this paper proposed a privacy preserving dynamic data release algorithm based on micro aggregation. Our principal contributions are as follows: (1) A series of indicators are designed to evaluate the synonymous linkages between the non-numerical sensitive values in turn facilitating an improvement in the clustering effect of the proposed micro-aggregation anonymous method; (2) The improved micro aggregation algorithm is proposed in a did to enhance the privacy protection effect of the published data by minimizing the distance between records and the total number of linkages, and for maximizing an increase of entropy; (3) The dynamic update program is introduced into the proposed micro-aggregation method to realize the dynamic release and update of data.

The rest of the paper is organized as follows. Section 2 provides an overview of some related studies pertinent to privacy preserving data publishing methods for static and dynamic datasets. Section 3 introduces the basic indicators used in the traditional micro-aggregation method. Section 4 depicts the salient ideas and design indicators of our proposed algorithm. The proposed dynamic data release algorithm against synonymous linkage (DRASL) has been put forward in Section 5. Section 6 depicts the experimental results, whereas, Section 7 concludes our paper.

2 Related Work

Over the past few years, privacy preserving data publishing technology has aroused widespread attention of researchers and achieved many results. Among them, the K-anonymity [7][8] privacy protection model is the most widely used one. The core idea of the K-anonymity model is to express the precise value of the quasi-identifier(QI) attribute in a generalized form. It defines the quasi-identifiers (QI) as the attributes that cannot directly identify a unique individual, but is sufficiently relevant and can be combined with other attributes to identify a specific individual. The records in the original data table can be segregated into multiple equivalence classes by generalizing the exact value of the quasi-identifiers into a certain value range. Each equivalence class contains at least K ($K \geq 2$) records with the same quasi-identifier values, and a certain individual represented by a record cannot be distinguished from other $(K-1)$ records, so as to achieve the purpose of privacy protection. The l-diversity anonymous model improves the privacy protection ability of the traditional K-anonymity model and is widely used in the privacy protection of static data publishing. Several improved algorithms based on this model have been well studied in the literature [10-12]. However, most of the data publishing methods based on the l-diversity anonymous model adopt the generalization operation on quasi-identifiers. The implementation process of generalization requires large computational cost and leads to significant decrease on the availability of published data.

Actually, the partition of equivalence class on quasi-identifiers can also be achieved by the clustering or aggregation, i.e., a cluster with K records can be generated according to the similarity of the quasi-identifiers. Lin et al. [14] proposed an efficient clustering method for K-anonymization, which segregate all the records into different subsets and adjust each of the subset to make sure it contains no less than K records. Meyerson et al.[15] proposed a K-center clustering method, which uses outliers primarily basing on a 2-approximation. Ceccarello et al. [16] proposed a K-center clustering method on MapReduce and Streaming. Zheng et al. [17] suggested to achieve K-anonymity through an improved clustering, wherein

the clustering process was optimized by considering the overall distribution of quasi-identifier groups in a multidimensional space. Siddula et al. [18] use an enhanced equi-cardinal clustering to achieve k-anonymity and provide node, edge, and attribute privacy for the social network. Navid et al. [19] envisaged to protect users' privacy through the anonymization of social network graphs. The proposed method of them optimized the clustering process of the K-anonymity method by means of the particle swarm optimization algorithm. Karuna et al. [20] proposed a method to protect the privacy of data maintained in cloud, which cluster the records using an adaptive k-anonymity algorithm. Yan et al. [21] proposed a weighted K-member clustering algorithm, which designed a series of weight indicators and enhanced the clustering effect and reduced the unnecessary computation during clustering. Nithya et al [22] proposed a predictive delimiter for multiple sensitive attributes aiming at the hidden knowledge in the combination of attributes. This method reallocates the sensitive attributes in the way that the attacker is unable to disclose the sensitive information associated with individuals. However, the algorithm is ineffective for incremental data and fails to protect individual privacy when the sensitive values are related or linked to a generalized value. Shi et al. [23] proposed a privacy preserving algorithm based on micro aggregation with dynamic sensitive attribute updating. Distance metrics and information entropy were used to aggregate data into equivalence groups, which ensure the protection of individual privacy while minimizing information loss. Abidi et al. [25] introduced a new microaggregation method based on fuzzy possibilistic clustering, which proposes to study the distribution of confidential attributes within each sub-dataset and the privacy parameter K is determined by preserving the diversity of confidential attributes within the anonymized microdata. Ana et al. [26] proposed a k-anonymous microaggregation method via linear discriminant analysis. By transforming the original data records to a different data space, the proposed method enables to build microcells more tailored to an intrinsic classification threshold. Ana et al. [27] proposed several strategies to simplify the distance calculations and element sorting operations for data microaggregation. Esteve et al. [28] proposed an optimized prepartitioning strategy to reduce the running time of K-anonymous microaggregation on large datasets. Zouinina et al. [29] managed to achieve K-anonymity by using topological collaborative clustering.

3 Prior Knowledge

In order to facilitate the understanding of subsequent definitions and descriptions, we first provide a unified explanation of the mathematical notations defined and employed in this paper (as depicted in Table 3).

3.1 Distance metric for micro aggregation

The micro aggregation method partitions the data records into different equivalence classes in accordance with the principle of maximum intra-class similarity and minimum inter-class similarity. Distance metric is usually used to evaluate the similarity between different records. In a relational database, a record often corresponds to an entity composed up of different type of attributes. Attributes in their own essence are used to describe the properties of a certain entity and primarily include continuous attributes and discrete attributes.

Continuous attributes are quantitative attributes as they may take on any value within a finite or infinite continuous interval. Age, Height, Weight, Temperature, etc., are all examples of continuous attributes. Discrete attributes refer to attributes with a finite number of discrete values and can be further classified into nominal attributes and ordinal attributes. The discrete nominal attributes cannot be ordered and cannot be measured and include two categories: (1) there are some semantic correlations between the discrete nominal attribute values, the taxonomy tree can be used to define the distance between those values,and (2) the discrete values of a nominal attribute have no relationships whatsoever, proximity measure can be adopted to estimate the distance between such attribute values. The discrete ordinal attribute is an attribute whose possible values have a meaningful order or ranking amongst them but the magnitude between different values is not known.

Table 4 depicts a micro data table with different kinds of attributes, wherein *Age* is a continuous attribute, *Zipcode* is a discrete nominal attribute with semantic correlations, *Sex and Religion* are discrete nominal attributes with non-semantic correlations,and *Capitalgain* is a discrete ordinal attribute with 3 values {moderate, good, excellent}. A series of distance metrics have been defined to assess the relations of records with multiple attributes.

Definition 1 (*Distance for continuous attribute [27]*). For any continuous attribute C in data table T, the distance between two values $v_i, v_j \in C$ can be defined as:

$$d_C(v_i, v_j) = \frac{|v_i - v_j|}{max(C) - min(C)} \tag{1}$$

where, $max(C)$ and $min(C)$ refers to the maximum and minimum value of a continuous attribute C.

Definition 2 (*Distance for semantic correlation nominal attribute [17]*). For any semantic correlation nominal attribute

Table 3. Description of mathematical notations

Symbol	***Description***
QIA	Quasi-identifier attributes
SA_r	A sensitive attribute value in a record r
SA_G	A set of sensitive attribute values in a group G
D_{SA}_random	A random sensitive attribute value $\in D_{SA}$
GID	An equivalent group of table T
C_i	The i^{th} continuous attribute $(i = 1, \ldots, m)$
r	A record included in table T
$r[QID]$	The value of r in quasi-identifierQID
$d_C(v_i, v_j)$	Distance between continuous values v_i and v_j
N_{sj}	The j^{th} semantic correlation nominal attribute $(j = 1, \ldots, n)$
$Tree_{N_s}$	Taxonomy tree defined for semantic correlation nominal attribute N_s
$\|Tree_{N_s}\|$	The total number of leaf nodes for $Tree_{N_s}$
$Parent(v_i, v_j)$	A common parent node of v_i and v_j according to their taxonomy tree
$\|Parent(v_i, v_j)\|$	The total number of leaf nodes with the root $Parent(v_i, v_j)$
$d_{N_s}(v_i, v_j)$	Semantic correlation nominal attribute values v_i and v_j
N_g	The g^{th} non-semantic correlation nominal attribute $(g = 1, \ldots, x)$
p	The total number of non-semantic correlation nominal attributes N_g
$match(v_i, v_j)$	The number of matches between v_i and v_j for attribute N
$d_N(v_i, v_j)$	Distance between non-semantic correlation nominal values v_i and v_j
O_h	The h^{th} ordinal attribute $(h = 1, \ldots, y)$
$\|O\|$	The number of distinct values in ordinal attribute O
$\phi(v)$	The normalize ranking of ordinal value v
$rank(\cdot)$	The ranking function
$d_O(v_i, v_j)$	Distance between ordinal values v_i and v_j
$d(r_1, r_2)$	Distance between records r_1 and r_2
$H(GID)$	The information entropy of an equivalent group GID)
GID'	The union of equivalent group GID with an added record r from table T
$\widehat{H}(GID, GID')$	The entropy increase between equivalent groups GID and GID'
μ_{GID}	The centroid of the equivalent group GID
$f(GID, GID')$	The micro aggregation clustering metric between GID and GID'
$\|GID_j\|$	The number of records in an equivalent group GID_j
$\|GID_j(v)\|$	The number of records $\in GID_j$ with sensitive value v
$\gamma(v_i, v_j)$	The common linked value between v_i and v_j of sensitive attribute SA
$Link_{SA}(v_i, v_j)$	The number of synonymous linkages between v_i and v_j of sensitive attribute SA
U	The strictly upper triangular matrix that contains all set of values $(v_i, v_j)_{i \neq j} \in (v_1, \cdots, v_n)$
$Tlink_{SA}(v_1, \cdots, v_n)$	The total number of synonymous linkages in the set $(v_1, \cdots, v_n)$
$Pr_{SA}(v_1, \cdots, v_n)$	The probability mass synonymous linkage of set $(v_1, \cdots, v_n)$
fg_r	The forged record

Table 4. A micro data table of mixed attributes

ID	*Age*	*Zipcode*	*Sex*	*Religion*	*Capitalgain*
1	33	10010	F	Buddhism	good
2	35	10019	M	Catholicism	excellent
3	49	10020	F	Islam	good
4	51	10022	M	Catholicism	moderate

N_s in data table T, the distance between two values $v_i, v_j \in N_s$ can be denoted as:

$$d_{N_s}(v_i, v_j) = \begin{cases} 0, & v_i = v_j \\ \frac{|Parent(v_i, v_j)|}{|Tree_{N_s}|}, & v_i \neq v_j \end{cases} \tag{2}$$

where $Tree_{N_s}$ refers to the taxonomy tree for semantic correlation nominal attribute N_s, $|Tree_{N_s}|$ is the total number of leaf nodes for $Tree_{N_s}$. $Parent(v_i, v_j)$ is the common parent node of v_i and v_j according to $Tree_{N_s}$, and $|Parent(v_i, v_j)|$ represents the total number of leaf nodes with the root $Parent(v_i, v_j)$.

Definition 3 (*Distance for non-semantic correlation nominal attribute [30]*). For any non-semantic correlation nominal attribute N in data table T, the distance between two values $v_i, v_j \in N$ can be denoted as:

$$d_N(v_i, v_j) = \frac{p - match(v_i, v_j)}{p} \tag{3}$$

where p is the total number of non-semantic correlation nominal values exists in N, $match(v_i, v_j)$ is the number of matches between v_i and v_j.

Definition 4 (*Distance for ordinal attribute [30]*). For any ordinal attribute O in data table T, the distance between two values $v_i, v_j \in O$ can be defined as:

$$d_O(v_i, v_j) = |\phi(v_i) - \phi(v_j)| \tag{4}$$

with :

$$\phi(v) = \frac{rank(v) - 1}{|O| - 1} \tag{5}$$

where $rank(v)$ is the rank of value v in the ordinal attribute O in ascendant order, and $|O|$ is the number of distinct values in ordinal attribute O.

Definition 5 (*Distance between two records [27]*). For a data table T with continuous attributes $C_i (i = 1, \ldots, m)$, semantic correlation nominal attribute $N_{s_j} (j = 1, \ldots, n)$, non-semantic correlation nominal attribute $N_g (g = 1, \ldots, x)$ and ordinal attributes $O_h (h = 1, \ldots, y)$, the distance between two records $r_1, r_2 \in T$ is defined as:

$$d(r_1, r_2) = \frac{1}{|QIA|} \left(\sum_{i=1}^{m} d_C(r_1(C_i), r_2(C_i)) + \sum_{j=1}^{n} d_{N_s}(r_1(N_{sj}), r_2(N_{sj})) + \sum_{g=1}^{x} d_N(r_1(N_g), r_2(N_g)) + \sum_{h=1}^{y} d_O(r_1(O_h), r_2(O_h)) \right) \tag{6}$$

where $d_C(r_1, r_2)$, $d_{N_s}(r_1, r_2)$, $d_N(r_1, r_2)$ and $d_O(r_1, r_2)$ are the corresponding continuous, nominal and ordinal distance functions defined in Definitions 1-4. $|QIA|$ is the number of quasi-identifiers in data table T.

Example 1. Let's consider the micro data shown in Table 4. Figure 1 shows the taxonomy tree of semantic correlation nominal attribute *Zipcode*. The discrete ordinal attribute *Capitalgain* has 3 values {moderate, good, excellent}, where $rank(moderate) = 1$, $rank(good) = 2$ and $rank(excellent) = 3$. According to Definition 5, the distance between r_1 and r_2 is $d(r_1, r_2) = \frac{1}{5} \times (\frac{|33-35|}{51-33} + \frac{2}{4} + \frac{2-0}{2} + |\frac{2-1}{3-1} - \frac{3-1}{3-1}|) = \frac{19}{45}$, and the distance between r_2 and r_4 is $d(r_2, r_4) = \frac{1}{5} \times (\frac{|35-51|}{51-33} + \frac{4}{4} + \frac{2-2}{2} + |\frac{3-1}{3-1} - \frac{1-1}{3-1}|) = \frac{26}{45}$.

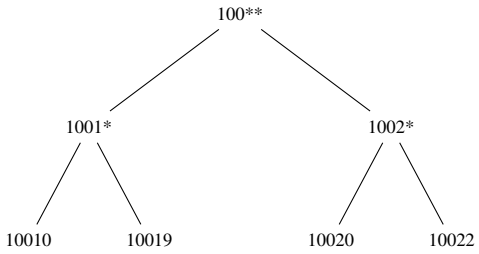


Fig. 1. Taxonomy tree of attribute Zipcode

3.2 Entropy metric for micro aggregation

In clustering-based micro aggregation, we hope to group similar data records together to form an equivalent group during the clustering stage, while ensuring the protection of sensitive values in the equivalent group. The solution is to minimize the distance between the quasi-identifier attributes in the equivalence group and maximize the diversity of the sensitive attributes in each equivalence group. Therefore, information entropy and entropy increase have been used as the indicators of clustering for micro aggregation to evaluate the degree of diversity within the equivalence group during the clustering process.

Definition 6 (*Entropy increase [23]*). Given an equivalent group GID, GID' represents the equivalent group after adding a record r, the increase of information entropy can be defined as:

$$\widehat{H}(GID, GID') = H(GID) - H(GID') \tag{7}$$

with :

$$H(GID) = \sum_{i=1}^{n} p_i log p_i \tag{8}$$

where $G' = G \cup r$ is the union of equivalent group G with the added record r, $H(G)$ is the information entropy of G, and p_i are the probabilities of sensitive values of G.

Definition 7 (*Micro aggregation metric [23]*). The micro aggregation clustering metric is defined as a function to decide which record is the best choice to join the equivalent group during micro aggregation clustering. The function can be be defined as:

$$f(GID, GID') = -\alpha \widehat{H}(GID, GID') - \beta d(\mu_{GID}, r) \tag{9}$$

where GID and GID' represent the equivalent group before and after adding a new record r, d and $\widehat{H}$ are respectively the distance and entropy increase metrics defined in Definition 5 and 6, μ_{GID} is the centroid of the equivalent group GID. α and β are the weight parameters used to adjust the proportion of the entropy increase index and the distance index, which satisfy the condition $\alpha + \beta = 1$.

Example 2. Still take Table 4 as an example. Suppose that the first two records r_1 and r_2 have already clustered to form the equivalent group. For the rest records r_3 and r_4, which one is more suitable to join the equivalent group next? The centroid of the equivalent group is $\mu_{GID} = (34, 10010, F, Buddhism, good)$. $G' = GID \cup r_3$ and $G'' = GID \cup r_4$ are the new equivalent groups after adding record r_3 and r_4 respectively. According to Definition 7, there is $f(GID, G') = -0.6[-(\frac{1}{2}log\frac{1}{2} + \frac{1}{2}log\frac{1}{2}) + (\frac{1}{3}log\frac{1}{3} + \frac{1}{3}log\frac{1}{3} + \frac{1}{3}log\frac{1}{3})] - 0.4[\frac{1}{5}(\frac{|34-49|}{51-33} + \frac{4}{4} + \frac{2-1}{2} + |\frac{2-1}{3-1} - \frac{2-1}{3-1}|)] = -0.081$, and $f(GID, G'') = -0.6[-(\frac{1}{2}log\frac{1}{2} + \frac{1}{2}log\frac{1}{2}) + (\frac{2}{3}log\frac{1}{3} + \frac{1}{3}log\frac{1}{3})] - 0.4[\frac{1}{5}(\frac{|34-51|}{51-33} + \frac{4}{4} + \frac{2-0}{2} + |\frac{2-1}{3-1} - \frac{1-1}{3-1}|)] = -0.29$. $f(GID, G')$ is superior to $f(GID, G'')$, therefore, record r_3 is more suitable to join the equivalent group.

4 Privacy Protection Micro Aggregation against Synonymous Linkage

The K-anonymity privacy protection method based on micro aggregation avoids the generalization operation on the quasi-identifiers, therefore, the availability of the published data is guaranteed. However, as we discussed in the introduction, the attacker may not be able to identify the record of the target victim accurately, but could infer the victim's sensitive value from the published data through the synonymous linkage between the sensitive values associated to the same equivalence group. In order to solve this problem, we propose the privacy preserving data publishing method against synonymous linkage

Table 5. Predefined catalogue of sensitive attribute *Disease*

Disease			
Blood cancer	anemia	Cardiovascular	coronary artery
	leukemia		heart attack
	lymphoma		stroke
	myeloma		marfan syndrome
Lung cancer	lung adenocarcinoma	Digestive system	enteritis
	Lung cancer		gastritis
	oat-cell cancer		GERD
	mesothelioma		stomach flu
Brain cancer	glioblastoma	Respiratory system	bronchitis
	astrocytoma		pneumonia
	meningioma		emphysema
	acoustic neuroma		COPD
Parasitic protozoan	malaria	Skin	acne
	chagas disease		pemphigus
	sleeping sickness		psoriasis
	trypanosomiasis		rosacea

by using micro aggregation method. The main idea of our proposed method is to take the semantic relations of sensitive values into consideration, minimize the number of synonymous linkage so that the majority set of sensitive values within the same equivalent group cannot be linked to a same generalized sensitive value.

4.1 Predefined catalogue for the sensitive attribute Disease

This section formalizes our new method based on the privacy protection requirements of synonymous linkage. In order to facilitate the discussion, we use *Disease* as the sensitive attribute in this paper, which is a typical non-numerical attribute of semantic associations. According to human disease classifications data from Britannica (https://www.britannica.com/science /human-disease/Classifications-of-diseases), we predefined a catalogue of related diseases for the sensitive attribute *Disease* (shown in Table 5).

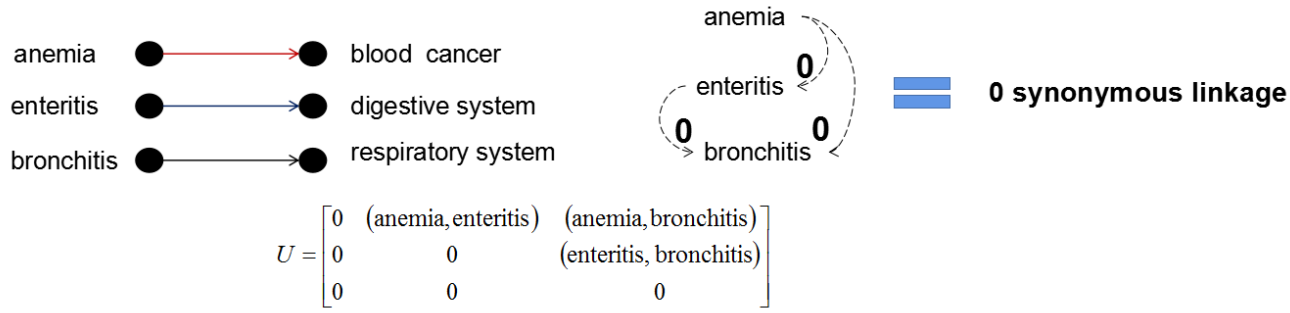
4.2 New definitions

Definition 8 (*The number of synonymous linkage between two values*). For any sensitive attribute SA, the number of linkage between two values $v_i, v_j \in SA$ can be defined as :

$$Link_{SA}(v_i, v_j) = \begin{cases} 2, & v_i = v_j \\ 1, & v_i \neq v_j \text{ and } \gamma(v_i, v_j) \neq \emptyset \\ 0, & otherwise \end{cases} \tag{10}$$

where $\gamma(v_i, v_j)$ is the common linked sensitive attribute value of v_i and v_j in the hierarchical catalogue of general common sensitive attributes.

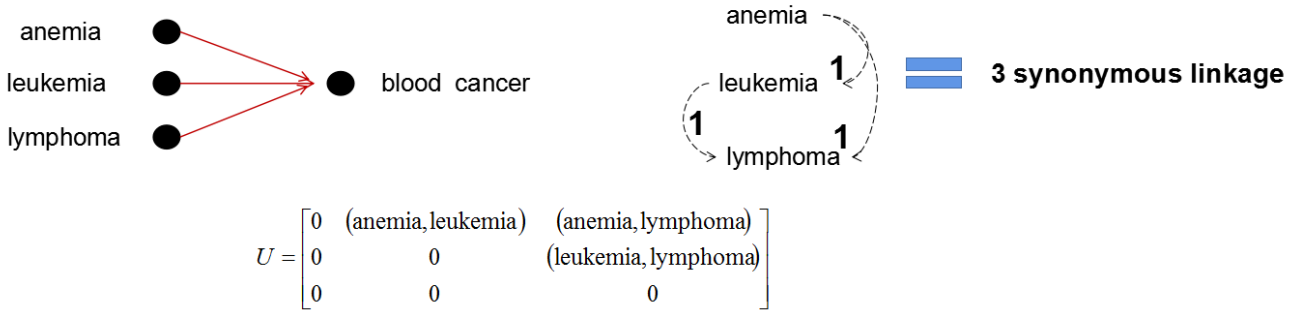
Given n sensitive values $(v_1, \cdots, v_n) \in D_{SA}$ where D_{SA} denotes the domain of sensitive attribute, finding the number of synonymous linkages of between values $(v_1, \cdots, v_n)$ comes down to determining the number of synonymous linkages for all values $(v_i, v_j)_{i \neq j} \in (v_1, \cdots, v_n)$. So, all values $(v_i, v_j)_{i \neq j} \in (v_1, \cdots, v_n)$ can be extend as elements of a strictly upper triangular matrix. Let $U = [(v_i, v_j)]$ such that $(v_i, v_j) = 0$ for $i \geq j$ be the strictly upper triangular matrix that contains all set of values



$$U=\begin{bmatrix} 0 & (\text{anemia},\text{enteritis}) & (\text{anemia},\text{bronchitis}) \\ 0 & 0 & (\text{enteritis},\text{bronchitis}) \\ 0 & 0 & 0 \end{bmatrix}$$

$$Tlink_{SA}(\text{anemia},\text{enteritis},\text{bronchitis})=Link_{SA}(\text{anemia},\text{enteritis})+Link_{SA}(\text{anemia},\text{bronchitis})+Link_{SA}(\text{enteritis},\text{bronchitis})=0$$

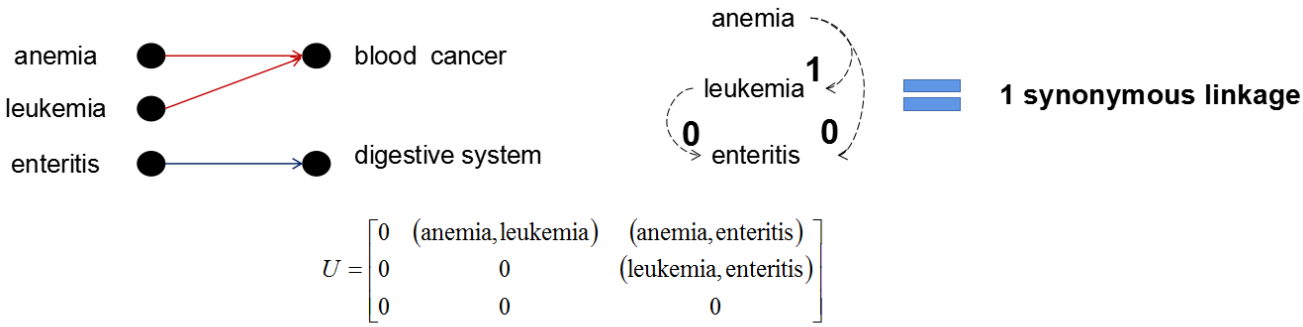
Fig. 2. The total number of synonymous linkage of equivalent group 1 according to Table 2.



$$U=\begin{bmatrix} 0 & (\text{anemia},\text{leukemia}) & (\text{anemia},\text{lymphoma}) \\ 0 & 0 & (\text{leukemia},\text{lymphoma}) \\ 0 & 0 & 0 \end{bmatrix}$$

$$Tlink_{SA}(\text{anemia},\text{leukemia},\text{lymphoma})=Link_{SA}(\text{anemia},\text{leukemia})+Link_{SA}(\text{anemia},\text{lymphoma})+Link_{SA}(\text{leukemia},\text{lymphoma})=3$$

Fig. 3. The total number of synonymous linkage of equivalent group 2 according to Table 2.



$$U=\begin{bmatrix} 0 & (\text{anemia},\text{leukemia}) & (\text{anemia},\text{enteritis}) \\ 0 & 0 & (\text{leukemia},\text{enteritis}) \\ 0 & 0 & 0 \end{bmatrix}$$

$$Tlink_{SA}(\text{anemia},\text{leukemia},\text{enteritis})=Link_{SA}(\text{anemia},\text{leukemia})+Link_{SA}(\text{anemia},\text{enteritis})+Link_{SA}(\text{leukemia},\text{enteritis})=1$$

Fig. 4. The total number of synonymous linkage of equivalent group 3 according to Table 2.

$(v_i,v_j)_{i\neq j}\in(v_1,\cdots,v_n)$, $U=[(v_i,v_j)]$ can be fined as:

$$U=\begin{pmatrix} 0 & (v_1,v_2) & \cdots & (v_1,v_n) \\ 0 & 0 & \cdots & (v_2,v_n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{11}$$

Definition 9 (*Total number of synonymous linkage*). For a set of sensitive values $(v_1,\cdots,v_n)\in D_{SA}$, the total number of

linkage can be defined as :

$$Tlink_{SA}(v_1,\cdots,v_n)=\sum_{i,j=1}^{n} Link_{SA}(U_{i,j}) \tag{12}$$

where $(v_1,\cdots,v_n)$ are the n sensitive values $\in D_{SA}$, $U_{i,j}$ are the strictly upper triangular matrix that contains all set of values $(v_i,v_j)_{i\neq j} \in (v_1,\cdots,v_n)$, $Link_{SA}(U_{i,j})$ is the number of synonymous linkages between each values in $U_{i,j}$.

Example 3. Let's considerate the previous Table 2, according to Definition 9, the total number of synonymous linkage in each equivalent group is $Tlink_{SA}(GID\_1)=0$, $Tlink_{SA}(GID\_2)=3$, and $Tlink_{SA}(GID\_3)=1$. The synonymous linkage relations of the sensitive values in each equivalent group are shown in Figure 2 to Figure 4 respectively.

Definition 10 (*Probability mass synonymous linkage*). For any n sensitive values $(v_1,\cdots,v_n) \in D_{SA}$, we define the probability mass synonymous linkage of set $(v_1,\cdots,v_n)$ as :

$$Pr_{SA}(v_1,\cdots,v_n)=\frac{Tlink_{SA}(v_1,\cdots,v_n)}{n(n-1)} \tag{13}$$

where $Tlink_{SA}(v_1,\cdots,v_n)$ is the total number of synonymous linkage in the set $(v_1,\cdots,v_n)$, n is the total number of sensitive values $\in (v_1,\cdots,v_n)$, $n(n-1)$ is the maximum number of synonymous linkages in the set $(v_1,\cdots,v_n)$ if all values are the same.

Definition 11 (*Micro aggregation metric against synonymous linkage*). In order to minimize the synonymous linkage of sensitive values during the micro aggregation process, we introduced the probability mass synonymous linkage on the basis of traditional micro aggregation clustering metric. The new function can be defined as:

$$f_{link_{SA}}(GID,GID')=f(GID,GID')+\alpha Pr_{SA}(GID')=-\alpha(\widehat{H}(GID,GID')-\beta d(\mu_{GID},r)+\alpha Pr_{SA}(GID')) \tag{14}$$

where $f(GID,GID')$ is the micro aggregation clustering function defined in Definitions 7, Pr_{SA} is the probability mass synonymous linkage defined in Definitions 10. The functions of parameters α and β are the same as those in Definition 7.

5 Privacy Preserving Dynamic Data Release based on Micro Aggregation

In order to avoid the privacy leakage caused by semantic correlation between sensitive values within the same equivalence class, we proposes a new publishing method based on the improved micro aggregation metric against synonymous linkage. In addition, the dynamic update program is introduced into the proposed algorithm to realize the insertion, deletion and modification of data, which makes the proposed method applicable for both the static and the dynamic data publishing scenario.

5.1 Micro aggregation publishing algorithm for the first release

Algorithm 1 depicts the static release of the proposed dynamic data release algorithm against synonymous linkage (DRASL). For a input data table T and a parameter K, the static release algorithm returns the anonymous data table T^* and the clustered equivalent groups GID according to the predefined catalogue of sensitive values. The algorithm starts by initiating an empty list of equivalent groups GID and an empty anonymous table T^* (**Line 4 and 5**). **Lines 6-22** illustrates the main process of micro aggregation, where the improved micro aggregation metric against synonymous linkage (Definition 11) was adopted as the criteria to find a best record and insert it into the current equivalent group.

5.2 Micro aggregation dynamic adjustment for record insertion

Big data publishing is a dynamic update process, therefore, we should take into consideration the various changes of data records that may occur during the publishing. Algorithm 2 depicts the dynamic adjustment record insertion process of the proposed DRASL algorithm. The main steps were derived from the previous method in [24], but using the improved micro aggregation metric against synonymous linkage (Definition 11) as the criteria to find a best equivalent group for the insert record.

In the case of dynamic update for record insertion, the method proposed in [24] generates forged records and insert them into equivalent groups so as to prevent sensitive information disclosure. However, there are still chances for an attacker to infer the sensitive value of an individual. **Situation 1**: for an equivalent group without forged record, when a new record r is added, a forged record fg_r is generated randomly so that the sensitive value of fg_r is different with that of r. However, if the

Algorithm 1 DRASL for the static release

Input: Data table T; parameter K; predefined catalogue of sensitive values;
Output: Clustered equivalent groups GID; anonymous data table T^*

1: **if** $(|T| \leq K)$ **then**
2: Return
3: **end if**
4: Let $GID = \emptyset$ //Create an empty list of equivalent groups
5: Let $T^* = \emptyset$ //Create an empty anonymous table of T
6: **while** $(|T| > K)$ **do**
7: Select a record r from T randomly
8: $T = T - \{r\}$
9: $gid = \{r\}$
10: **while** $(|gid| < K)$ **do**
11: Find a record $r^{'} \in T$ s.t. $\max\{f_{link_{SA}}(gid, (gid \cup \{r^{'}\}))\}$
12: Find a group $gid_j \in GID$ s.t. $\max\{f_{link_{SA}}(gid, (gid \cup \{gid_j\}))\}$
13: **if** $(f_{link_{SA}}(gid, (gid \cup \{r^{'}\})) > f_{link_{SA}}(gid, (gid \cup gid_j)))$ **then**
14: $T = T - \{r^{'}\}$
15: $gid = gid \cup \{r^{'}\}$
16: **else**
17: $GID = GID - gid_j$
18: $gid = gid \cup gid_j$
19: **end if**
20: **end while**
21: $GID = GID \cup gid$
22: **end while**
23: Create an empty list Q
24: **while** $(|T| \neq 0)$ **do**
25: Select a record r from T randomly
26: $T = T - \{r\}$
27: **for** each group $gid_j \in GID$ **do**
28: $Q \longleftarrow f_{link_{SA}}(gid_j, (gid_j \cup \{r\}))$
29: **end for**
30: $j =$ the sequence number of the element with the maximal value in Q
31: $gid_j = gid_j \cup \{r\}$
32: **end while**
33: $T^* \longleftarrow generalization(GID)$
34: **return** GID, T^*

sensitive value of fg_r is synonymous linked with that of r, then, the new record is exposed to synonymous attack. **Situation 2**: for an equivalent group already with a forged record fg_r, when a new record r is added, the sensitive value of fg_r will be updated into another value randomly. However, if the updated sensitive value of fg_r is synonymous linked with that of the new record r, the new record is exposed to synonymous attack.

Example 4. Take the clustered equivalent groups shown in Table 6 for example, Table 7 is the updated version after a new record $r = (23, 15032, bronchitis)$ (shown in red) has been inserted to the equivalent group GID_1. According to situation 1, the dynamic adjustment algorithm generated a forged record $fg_r = ([21-23], [12***-15**], pneumonia)$ (shown in blue) in the group GID_1 where the new record is belong. The sensitive values of fg_r and r are different, however, by comparing the differences between Table 6 and Table 7, the attacker still can conclude that the individual corresponding to the newly added record has some respiratory system disease, because "bronchitis" and "pneumonia" are synonymous linked to the respiratory system disease. The same problem may also occur in **Situation 2**.

In order to solve this issue, we propose another dynamic adjustment method to protect sensitive values after Algorithm 2 (as shown in Algorithm 3). **Lines 3-4** deal with the situation that there is already a forged record fg_r in the group gid_j. The algorithm randomly changes the sensitive value of the forged record into a new value $D_{SA}\_random \in D_{SA}$, so that the sensitive value of the forged record and the new record are different and have no synonymous linkage ($Link_{SA}(D_{SA}\_random, SA_r) = 0$). **Lines 5-6** aims at the situation that there is no forged record in the group gid_j. The algorithm generates a new forged record $fg_r \in gid_j$ with a random sensitive value $D_{SA}\_random \in D_{SA}$ so that the sensitive value of the forged record and the new record are different and have no synonymous linkage ($Link_{SA}(D_{SA}\_random, SA_r) = 0$).

Algorithm 2 DRASL dynamic adjustment for record insertion

Input: Clustered equivalent groups GID; new record r; predefined catalogue of sensitive values;
Output: Updated cluster equivalent groups GID'

Let $H = \emptyset$ //Create an empty cash table H
while (there is a new record r) **do**
  $gid = \{r\}$
  $r \rightarrow H$ //Store r in H
  $GID_H \longleftarrow recall\_Algorithm\ 1(H)$
  Find a group $gid_j \in GID$ s.t. $\max\{f_{link_{SA}}(gid, (gid \cup \{gid_j\}))\}$
  Find a group $gidh_i \in GID_H$ s.t. $\max\{f_{link_{SA}}(gid, (gid \cup \{gidh_i\}))\}$
  if $(f_{link_{SA}}(gid, (gid \cup \{gid_j\})) > f_{link_{SA}}(gid, (gid \cup \{gidh_i\})))$ **then**
    $GID = GID - gid_j$
    $gid_j = gid_j \cup \{r\}$
    $GID' = GID \cup gid_j$
  else
    $r_v = \{GID \cap gidh_i\}$ //records that include in both GID and $gidh_i$
    $GID = GID - \{r_v\}$
    $GID_H = GID_H - gidh_i$
    $GID' = GID \cup gidh_i$
  end if
end while
return GID'

Table 6. Clustered equivalent groups from a micro patient table.

GID	*Age*	*Zipcode*	*Disease*
1	[21-22]	[12***–14***]	enteritis
1	[21-22]	[12***–14***]	leukemia
2	[26-28]	[18***–25***]	bronchitis
2	[26-28]	[18***–25***]	anemia

Table 7. Updated equivalent groups after new record insertion.

GID	*Age*	*Zipcode*	*Disease*
1	[21-23]	[12***–15***]	enteritis
1	[21-23]	[12***–15***]	leukemia
1	[21-23]	[12***–15***]	bronchitis
1	[21-23]	[12***–15***]	pneumonia
2	[26-28]	[18***–25***]	bronchitis
2	[26-28]	[18***–25***]	anemia

Algorithm 3 DRASL dynamic adjustment for the protection of sensitive values

Input: Clustered equivalent groups GID; predefined catalogue of sensitive values;
Output: Updated clustered equivalent groups GID with forged record

while (Group change $\Rightarrow r \rightarrow SA_r \in D_{SA}$) **do**
  Let $gid_j \in GID$ be the Group where the change occurs
  if (there is already $fg_r \subseteq gid_j$) **then**
    $fg_r \rightarrow fg_r(D_{SA}\_random \in D_{SA})\ s.t.\ D_{SA}\_random \neq SA_r\ \&\ Link_{SA}(D_{SA}\_random, SA_r) = 0)$
  else
    $gid_j \cup gid_j\{fg_r \rightarrow D_{SA}\_random \in D_{SA}\}$ s.t. $D_{SA}\_random \neq SA_r\ \&\ Link_{SA}(D_{SA}\_random, SA_r) = 0$
  end if
end while
return GID

Example 5. Let's consider the previous clustered equivalent groups shown in Table 6. After applying the proposed Algorithm 3, the results have changed (as shown in Table 8). Compared to previous results after new record has been added (as shown in Table 7), Algorithm 3 generated a forged record $fg_r = ([21-23], [12***-15***], acne)$. The sensitive values of fg_r and the new added record r are not only different but also not synonymous linked, because there is $Link_{SA}(bronchitis, acne) = 0$. Therefore, the privacy protection effect on the published data has been enhanced.

Table 8. New record insertion after using Algorithm 3.

GID	*Age*	*Zipcode*	*Disease*
1	[21-23]	[12***–15***]	enteritis
1	[21-23]	[12***–15***]	leukemia
1	[21-23]	[12***–15***]	bronchitis
1	[21-23]	[12***–15***]	acne
2	[26-28]	[18***–25***]	bronchitis
2	[26-28]	[18***–25***]	anemia

5.3 Micro aggregation dynamic adjustment for record deletion

Algorithm 4 presents the pseudo code of the dynamic adjustment record deletion process of the proposed DRASL algorithm. **Line 1** defines the equivalent group $gid_j \in GID$ contains the record to be deleted. **Line 2** removes the record r from its original equivalent group gid_j. In **Lines 3-5**, when the size of the equivalent group gid_j after record deletion reaches or exceeds K, Algorithm 3 will be recalled and the deleted record r and equivalent group gid_j will be used as the input to carry out the updating process. The equivalent group gid_j will be added into the set of clustered equivalent group GID. **Lines 6-18** are the key processes of dynamic adjustment for record deletion. When the size of the equivalent group gid_j after record deletion is less than K, the algorithm deletes gid_j from the clustered equivalent group GID (**Line 7**). In **Lines 9-10**, a record r will be randomly selected and removed from gid_j. Then, **Lines 11-13** evaluate which equivalent group is most suitable for the record r to join in according to the improved micro aggregation metric defined in Definition 11. After that, the record r is added to the corresponding group G_i (**Lines 14-15**), and Algorithm 3 will be recalled to update the clustered equivalent groups GID (**Lines 16-17**). This kind of loop will keep on running as long as the group gid_j is not empty.

Algorithm 4 DRASL dynamic adjustment for record deletion

Input: Clustered equivalent groups GID; deleted record r; parameter K; predefined catalogue of sensitive values;
Output: Updated clustered equivalent groups GID

1: Let $gid_j \in GID$ be the equivalent group currently containing the deleted record r
2: $gid_j = gid_j - \{r\}$
3: **if** $(|gid_j| \geq K)$ **then**
4: $gid_j \longleftarrow recall\_Algorithm\ 3(gid_j, r)$
5: $GID = GID \cup gid_j$
6: **else**
7: $GID = GID - gid_j$ //retrieve the group gid_j from the set GID
8: **while** $(|gid_j| \neq 0)$ **do**
9: Select a record r from gid_j randomly
10: $gid_j = gid_j - \{r\}$
11: **for** each group $G_i \in GID$ **do**
12: $Q \longleftarrow f_{link_{SA}}(G_i, (G_i \cup \{r\}))$
13: **end for**
14: $i =$ the sequence number of the element with the maximal value in Q
15: $G_i = G_i \cup \{r\}$
16: $G_i \longleftarrow recall\_Algorithm\ 3(G_i, r)$
17: $GID = GID \cup G_i$
18: **end while**
19: **end if**
20: **return** GID

5.4 Micro aggregation dynamic adjustment for record modification

Previous researches deal with the process of record modification by deleting old record and then inserting the new one. In this section, we propose the micro aggregation dynamic adjustment algorithm for record modification, as shown in Algorithm 5. **Line 1** defines the equivalent group $gid_j \in GID$ contains the record to be modified. **Lines 3-7** describe

the modification process when the quasi-identifier attributes of the record changed from $r[QIA]$ to $r^{'}[QIA]$. **Line 4** callback Algorithm 4 to delete r from gid_j, **Line 5** update the equivalent group GID, and **Line 6** callback Algorithm 2 to insert the modified record $r^{'}$ into the most suitable equivalent group in GID. **Lines 8-15** illustrate the modification process when the sensitive value of the record changed from SA_r to $SA_{r'}$. **Lines 9-11** check if the modify record is in the cash table H, then it update the sensitive value SA_r to $SA_{r'}$. Otherwise, in **Line 12**, the sensitive value of the record SA_r is modified to $SA_{r'}$ directly from its corresponding group $gid_j \in GID$. Then, **Line 13** callback Algorithm 3 to protect the modified sensitive value $SA_{r'}$, and **Line 14** adds gid_j to the set of clustered equivalent group GID.

Algorithm 5 DRASL dynamic adjustment for record modification

Input: Clustered equivalent groups GID; modify record r; parameter K; predefined catalogue of sensitive values;
Output: Updated clustered equivalent groups GID

1: Let $gid_j \in GID$ be the equivalent group currently containing the modify record r
2: **while** (record change $r \rightarrow r^{'}$) **do**
3: **while** (modification includes only quasi-identifiers $r[QID]$) **do**
4: $gid_j \longleftarrow recall\_Algorithm\ 4(gid_j, r)$
5: $GID = GID \cup gid_j$
6: $GID \longleftarrow recall\_Algorithm\ 2(GID, r^{'})$
7: **end while**
8: **while** (modification includes only sensitive value SA_r) **do**
9: **if** (Cash table H contains the modify record r) **then**
10: $H = H \cup H\{r \rightarrow SA_r = SA_{r'}\}$
11: **end if**
12: $gid_j = gid_j \cup gid_j\{r \rightarrow SA_r = SA_{r'}\}$
13: $gid_j \longleftarrow recall\_Algorithm\ 3(gid_j, r^{'})$
14: $GID = GID \cup gid_j$
15: **end while**
16: **end while**
17: **return** GID

6 Experimental Results

In order to evaluate the effectiveness and efficiency of the proposed algorithm, we compare and analyze the proposed DRASL algorithm with some existing K-anonymity algorithms from the aspects of privacy protection effect, availability of published data, and execution time. The baseline methods include, but are not limited to, the one pass K-means algorithm (OKA) [14], the improved K-anonymity algorithm based on clustering (IKA) [17], and the data privacy protection algorithm based on micro aggregation (DPP) [23].

All the algorithms were implemented in Python and carried out on Huawei Elastic Cloud Server 8vCPUs —32GB— pl2.2xlarge.4 under Windows Server 2016 Standand 64bit for T4 with TESLA. The dataset used for the experiments were composed of seven quasi-identifier attributes and one sensitive attribute. The quasi-identifier attributes were originally selected from the Adult dataset (https://archive.ics.uci.edu/ml/datasets/Adult) from the UCI machine learning repository, where we retain only the attributes *Age*, *Workclass*, *Occupation*, *Education*, *Capitalgain*, *Race* and *Gender*. The sensitive value is *Disease*, we randomly generate sensitive values from 32 different diseases based on the predefined catalogue for sensitive attribute (shown in Table 5), and assign a disease to each record in the dataset. Inaccurate records such as missing values and duplicate records were removed from the dataset.

6.1 Privacy protection effect

Privacy preserving data publishing method based on the K-anonymity model mainly protects the user's sensitive information through the "group masking effect", which can reduce the possibility of the attacker obtaining sensitive information of a certain individual. However, there are always some correlations among the values belonging to the same type of sensitive attribute. Even if all the sensitive values in the same group are different from each other, the semantic relevance between them is inevitable. Therefore, the attacker may not precisely identify the record of the target victim, but could infer the victim's sensitive value via the semantic relevance within the same published group. This is the synonymous linking phenomenon between sensitive values discussed in this paper. The stronger the synonymous linkage is, the weaker is the "group masking effect" and the larger is the possibility of privacy disclosure.

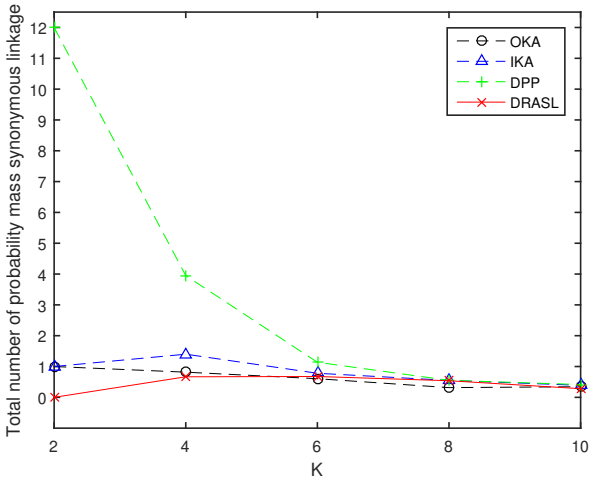


Fig. 5. Total number of probability mass synonymous linkage for static release.

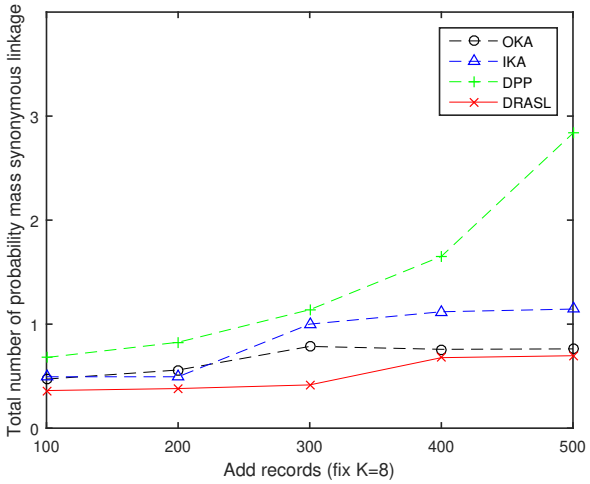


Fig. 6. Total number of probability mass synonymous linkage for records insertion.

In this paper, we use the total number of probability mass synonymous linkage to evaluate the privacy protection effect for sensitive attribute on the published data. Let GID be the final set which contains the all the equivalent groups, $|GID|$ is total number of equivalent groups, and $Pr_{SA}(SA_{G_i})$ is the probability mass synonymous linkage of equivalent group G_i. The total number of probability mass synonymous linkage for the set GID can be defined as:

$$Total\_Pr_{SA}(SA_{GID}) = \sum_{i=1}^{|GID|} Pr_{SA}(SA_{G_i}) \tag{15}$$

Figure 5-8 depict the total number of probability mass synonymous linkage of all the algorithms in terms of static publishing, records insertion, deletion, and modification. The weight parameters $\alpha = 0.6$ and $\beta = 0.4$ are set to adjust the proportion of the entropy increase index and the distance index. For the static data release, we can observe from Figure 5 that with the increase of the K value, the proposed DRASL algorithm is superior to other algorithms in almost all cases.

In the case of dynamic update for records insertion and deletion, we add/delete various proportions of records consecutively and set $K = 8$. In the case of dynamic update for records modification, we modify various proportions of records consecutively, where half of the modifications occur on the quasi-identifiers and the other half on the sensitive values. The parameter is also set with $K = 8$. In all the above dynamic update situations, the proposed DRASL algorithm has lower total number of probability mass synonymous linkage as compared to the other algorithms. The primary reason is that the proposed DRASL algorithm is based on the criterion that minimizing synonymous semantic linkage during the process of selecting records and adjusting the aggregation of equivalence groups. Therefore, the proposed DRASL algorithm has better ability to prevent synonymous attack and provide better privacy protection effect on the published data.

6.2 Availability of published data

Privacy preserving data publishing method based on the K-anonymity model reduces the availability of published data to a certain extent. The primary reason is that the generalization operation carried out on the quasi-identifiers directly reduced the accuracy of the published data. The greater the degree of generalization is, the lower is the availability of published data. In this paper, we use the average of information loss to evaluate the availability of published data generated by different clustering and micro aggregation algorithms.

Let G be an equivalent group and $|G|$ is the total number of records, the amount of information loss that occurs in G can be defined as:

$$IL(G) = \frac{1}{|G|} \sum_{i=1}^{|G|} d(r_i, \mu_G) \tag{16}$$

where $d(r_i, \mu_G)$ is the distance between records r_i and the centroid of the equivalent group G according to Definition 5.

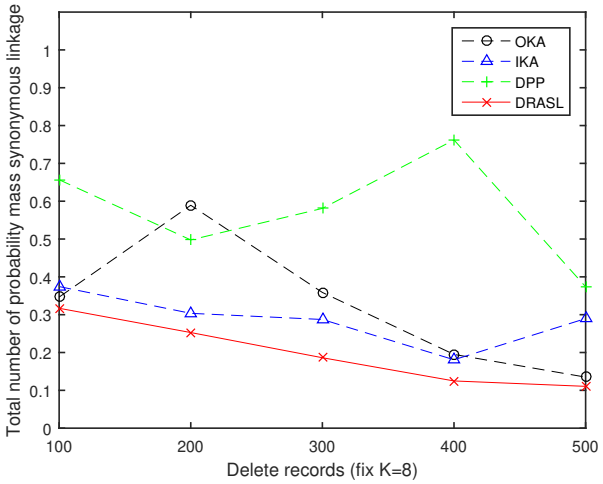


Fig. 7. Total number of probability mass synonymous linkage for records deletion.

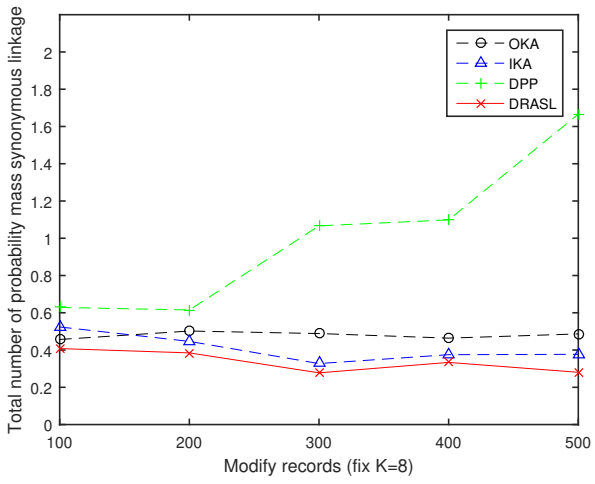


Fig. 8. Total number of probability mass synonymous linkage for records modification.

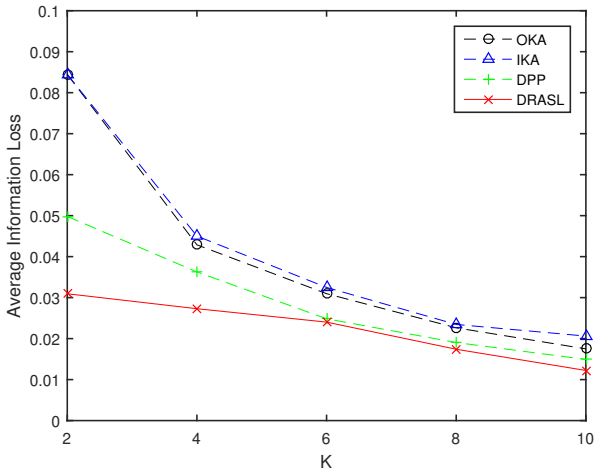


Fig. 9. Average information loss for static release.

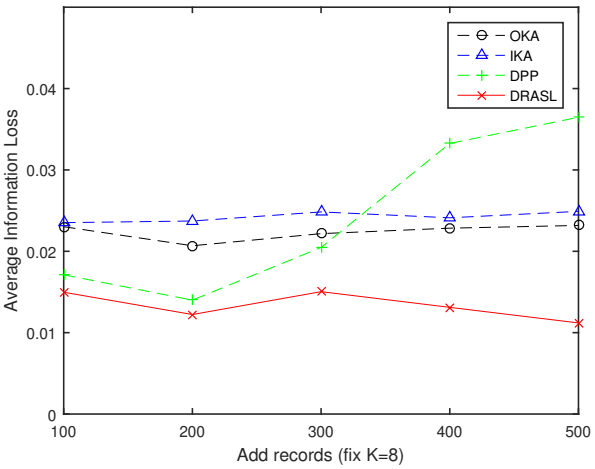


Fig. 10. Average information loss for records insertion.

Let GID be the set of all the equivalent groups and $|GID|$ is the total number of equivalent groups in the set GID, the average information loss of the set GID is defined as:

$$Average\_IL(GID) = \frac{1}{|T|} \sum_{i=1}^{|GID|} IL(G_i) \tag{17}$$

where $|T|$ is total number of records in the data table T.

Figure 9-12 portray the average information loss of all the algorithms in terms of static publishing, records insertion, deletion, and modification. All the parameters and the ratio of record insertion, deletion and modification are consistent with section 6.1. It's obviously that the proposed DRASL algorithm outperforms other algorithms in the aspect of average information loss under all the situations. The reason is that the micro aggregation process has fully consider the distance between original records, which facilitate to minimize the impact of generalization operations and improve the availability of published data.

Both of the DPP algorithm and the proposed DRASL algorithm use the insertion of forged records to realize the dynamic update and adjustment of the released data. Table 9 compares the number of forged records during the dynamic update process of the two algorithms. To be fair, the amount of records dynamically updated by the two algorithms remain the same and keep the parameter K=8. As noticed in Table 9, the number of forged records of the proposed DRASL algorithm are obviously less than that of the DPP algorithm. This also proves that the proposed DRASL algorithm introduces less interference during the process of data dynamic updating and provide better availability on the published data.

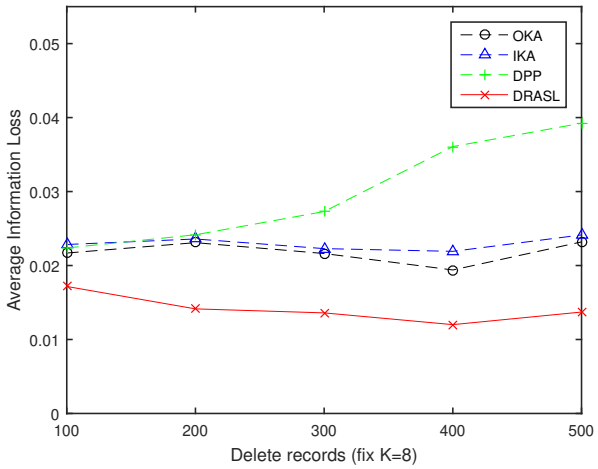


Fig. 11. Average information loss for records deletion.

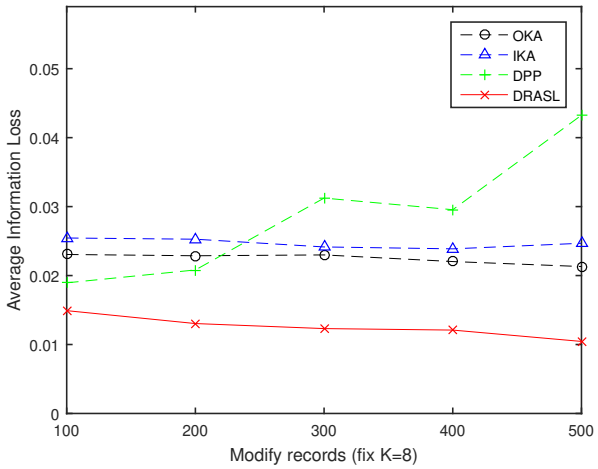


Fig. 12. Average information loss for records modification.

Table 9. Comparison of forged records after dynamic update.

Changed records	Forged records for insertion		Forged records for deletion		Forged records for modification	
	DPP	DRASL	DPP	DRASL	DPP	DRASL
100	6	3	6	5	6	5
200	6	5	6	4	6	4
300	10	7	6	3	9	4
400	16	8	6	2	9	4
500	21	6	6	2	14	4

6.3 Comparison of execution time

Essentially, the four algorithms involved in the comparison are all based on clustering methods to achieve K anonymous privacy protection for data release. However, their specific clustering process and evaluation indicators are different from each other. For a dataset with x records, the OKA algorithm first splits all the records into $\lceil \frac{x}{K} \rceil$ subsets, and then compares the loss of information and adjust the records in each subset to achieve K-anonymity. Therefore, it has a time complexity of $O(\frac{x^2}{K})$.

For the IKA algorithm, the calculation of the distance between the first centroid and the remaining records should be carried out for $(x-1)$ times so as to construct the first cluster. Then, it needs $2 \times (x-K-2)$ to repeat the same calculation and construct the second cluster. In order to get the third cluster, $3 \times (x-2K-3)$ calculations have to be spend. Therefore, for the published table with $\lceil \frac{x}{K} \rceil$ clusters, the overall time complexity is $(x-1)+2 \times (x-K-2)+\cdots+(\lceil \frac{x}{K} \rceil -1) \times (x-(\lceil \frac{x}{K} \rceil -2)K-(\lceil \frac{x}{K} \rceil -1)) \approx O(x^2)$.

Most of the time complexity of the DPP algorithm is used to compare the distances between records and the increase of entropy of the equivalent groups so as to find proper equivalent groups. Each of the equivalent group begins with an randomly selected record and continuously select the most appropriate record to join the group with the criteria of minimizing the increase of entropy. As the clustering progresses, each equivalent group needs to add records until the number of records reaches at least K. So the number of record selections will increase from $(x-1)$ to K. For a dataset with x records, the time complexity of DPP algorithm is about $O(x^2)$.

The proposed DRASL algorithm follows the same process as the DPP algorithm, but it take the comprehensive effect of the distance between records, the increase of entropy, and the number of synonymous linkages between sensitive values into consideration during the clustering process. Each of the equivalent group begins with an randomly selected record and and selects at least $(K-1)$ appropriate records close to the centroid by minimizing the distance between the records and the centroid as well as minimizing the number of synonymous linkage between sensitive values while maximizing the increase of entropy. When constructing equivalent groups, the proposed DRASL algorithm not only checks the number of synonymous linkage between the sensitive values of each equivalent group but also considers the sensitive values of other remaining records. Therefore, the number of record selections is doubled, i.e., from $(2x-1)$ to K. For a dataset with x records, the time complexity of the proposed DRASL algorithm is about $O(2x^2) \approx O(x^2)$.

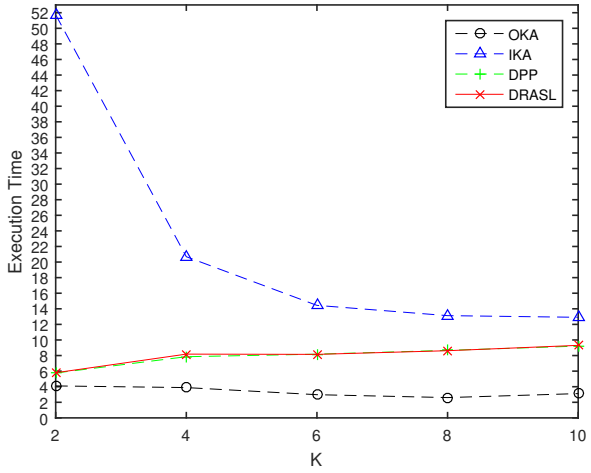


Fig. 13. Execution time for static release.

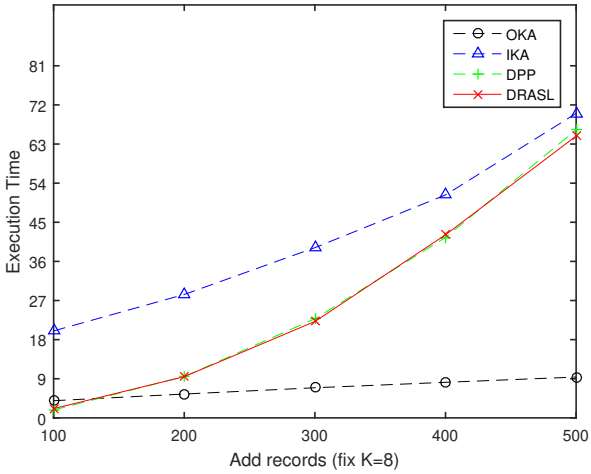


Fig. 14. Execution time for records insertion.

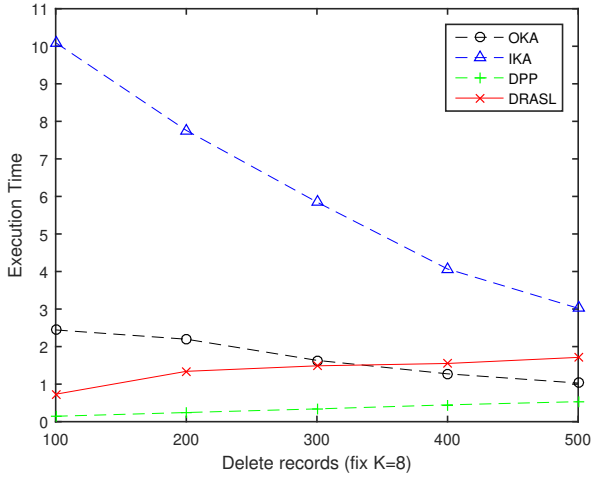


Fig. 15. Execution time for records deletion.

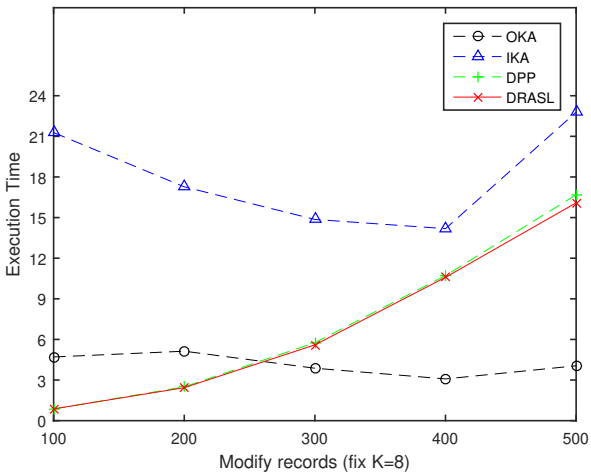


Fig. 16. Execution time for records modification.

Figure 13-16 portray the execution time of all the algorithms in terms of static publishing, records insertion, deletion, and modification. The method and proportion of record insertion, deletion, and modification remain the same as in section 6.1. In most of the cases, the execution time of the proposed DRASL algorithm is close to that of the DPP algorithm, which is second only to the OKA algorithm among all the comparison algorithms.

7 Conclusion

The research on privacy preserving data publishing is indispensable for the further innovation and development of big data technology. However, most of the anonymous data publishing methods based on the K anonymity model and its improvement strategy adopt the generalization operation on quasi-identifiers, which requires large computational cost and leads to significant decrease on the availability of published data. This paper addresses the problem of privacy leakage caused by synonymous linkage between sensitive values and proposes a dynamic data publishing algorithm based on micro aggregation. A series of indicators are designed to evaluate the synonymous linkage between non-numerical sensitive values and facilitate to improve the clustering effect of the proposed micro-aggregation anonymous method. The dynamic update program is introduced into the proposed micro-aggregation method to realize the dynamic release and update of data. Experimental analysis suggests that the proposed method provide better privacy protection effect and availability of published data as compared with some existing methods.

Acknowledgements

The research at hand was supported by the National Nature Science Foundation of China (No. 61762059).

References

[1] Ge, M.Z.; Bangui, H.; Buhnova, B. Big data for internet of things: A survey. Future Gener. Comput. Syst. 2018, 87, 601–614.

[2] Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. IEEE Trans. Intell. Transp. Syst. 2019, 20, 383–398.

[3] Qi, C. Big data management in the mining industry. Int J Miner Metall Mater. 2020, 27, 131-–139.

[4] J. Shamsi, M. Khojaye. Understanding Privacy Violations in Big Data Systems. IT Professional, 2018, 20(3): 73–81.

[5] Lv, Z.; Qiao, L. Analysis of healthcare big data. Future Generation Computer Systems, 2020, 109, 103–110.

[6] D. Anupam, S. K. Sarma, S. Deka. Data Security with DNA Cryptography. Proceedings of the World Congress on Engineering 2019, 2019: 246–251.

[7] Samarati P, Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. SRI Computer Science Laboratory, 1998: 1–19.

[8] Sweeney L. K-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557–570.

[9] Machanavajjhala, A.; Gehrke, J.; Kifer, D. L-diversity: Privacy beyond k-anonymity. Proceedings of the International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, April 2006, 3-–7.

[10] Palanisamy, B.; Liu, L.; Zhou, Y.; Wang, Q. Privacy-preserving publishing of multilevel utility-controlled graph datasets. ACM Trans. on Internet Techn. 2018, 18, 1–21.

[11] Temuujin, O.; Ahn, J.; Han, J.; Im, D. H. Efficient L-Diversity Algorithm for Preserving Privacy of Dynamically Published Datasets. Int. J. IEEE Access, 2019, 7, 122878–122888.

[12] Xiao, Y. and Li, H. Privacy Preserving Data Publishing for Multiple Sensitive Attributes Based on Security Level. Int. J. Inf. 2020, 11.

[13] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Introduction to Data Mining. Publisher: Pearson, 2nd edition, 2019.

[14] Lin J., MengCheng W. An Efficient Clustering Method for k-Anonymization. Proceedings of the 11th International Conference on Extending Database Technology, Nantes, France, 2008, 46–50.

[15] Meyerson A, Williams R. The non-uniform k-center problem. Proceedings of the 43rd International Colloquium on Automata Languages and Programming (ICALP 2016), Rome, Italy, 2016, 223–228.

[16] Ceccarello M, Pietracaprina A, Pucci G. Solving k-center clustering (with outliers) in MapReduce and streaming, almost as accurately as sequentially. Proceedings of the VLDB Endowment, 2019, 12(7), 766–778.

[17] Zheng W.T., Zhongyue W., Tongtong L.v., Ma Y., Jia C. K-anonymity Algorithm Based on Improved Clustering. Proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2018), Guangzhou, China, November, 2018, 462–476.

[18] Siddula M., Li Y., Cheng X., Tian Z., Cai Z. Anonymization in online social networks based on enhanced equi-cardinal clustering. IEEE Transactions on Computational Social Systems, 2019, 6(4), 809–820.

[19] Navid Y., Mohammad F., Babak A. Evolutionary algorithms for k-anonymity in social networks based on clustering approach. The Computer Journal, 2020, 63(7), 1039–1062.

[20] Karuna A., Sumalatha L. Adaptive k-anonymity approach for privacy preserving in cloud. Arabian Journal for Science and Engineering, 2020, 45, 2425–2432.

[21] Yan, Y., Herman, E.A., Mahmood, A. Feng T., Xie P.S. A weighted K-member clustering algorithm for K-anonymization. Computing, 2021, https://doi.org/10.1007/s00607-021-00922-0

[22] Nithya, M. and Sheela, T. Predictive delimiter for multiple sensitive attribute publishing. Int. J. Cluster Comput. 2019, 12297–12304.

[23] Shi, Y.; Zhang, Z.; Chao, H. C.; Shen, B. Data Privacy Protection Based on Micro Aggregation with Dynamic Sensitive Attribute Updating.Int. J. Sensors. 2018, 2307.

[24] Li S., Guo X. Distributed k-clustering for data with heavy noise. Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18), Montréal, Canada, 2018, 7849–7857.

[25] Abidi B., Ben Yahia, S., Perera, C. Hybrid microaggregation for privacy preserving data mining. J Ambient Intell Human Comput. 2020, 11, 23—38.

[26] Ana R., David R., José E., Jordi F., Luis U. Preserving empirical data utility in k-anonymous microaggregation via linear discriminant analysis. Engineering Applications of Artificial Intelligence, 2020, 94, 103787.

[27] Ana R., José E., David R., Ahmad M., Javier P., Jordi F. The Fast Maximum Distance to Average Vector (F-MDAV): An algorithm for k-anonymous microaggregation in big data. Engineering Applications of Artificial Intelligence, 2020, 90, 103531.

[28] Esteve P., David R., Ana R., José E., Ahmad M., Jordi F. Mathematically optimized, recursive prepartitioning strategies for k-anonymous microaggregation of large-scale datasets. Expert Systems with Applications, 2020, 144, 113086.
[29] Zouinina S., Bennani Y., Rogovschi N., Lyhyaoui A. Data Anonymization through Collaborative Multi-view Microaggregation. Journal of Intelligent Systems, 2021, 30(1): 327–345.
[30] Han, J.; Kamber, M.; Pei, J. Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann: 225 Wyman Street, Waltham, MA 02451, USA, 2011, 39–44.