



Article Designing a Novel Approach Using a Greedy and Information-Theoretic Clustering-Based Algorithm for Anonymizing Microdata Sets

Reza Ahmadi Khatir[†], Habib Izadkhah^{*,†} and Jafar Razmara[†]

Department of Computer Science, University of Tabriz, Tabriz 5166616471, Iran; r.ahmadikhatir@tabrizu.ac.ir (R.A.K.); razmara@tabrizu.ac.ir (J.R.)

* Correspondence: izadkhah@tabrizu.ac.ir

⁺ These authors contributed equally to this work.

Abstract: Data anonymization is a technique that safeguards individuals' privacy by modifying attribute values in published data. However, increased modifications enhance privacy but diminish the utility of published data, necessitating a balance between privacy and utility levels. K-Anonymity is a crucial anonymization technique that generates k-anonymous clusters, where the probability of disclosing a record is 1/k. However, k-anonymity fails to protect against attribute disclosure when the diversity of sensitive values within the anonymous cluster is insufficient. Several techniques have been proposed to address this issue, among which *t*-closeness is considered one of the most robust privacy techniques. In this paper, we propose a novel approach employing a greedy and information-theoretic clustering-based algorithm to achieve strict privacy protection. The proposed anonymization algorithm commences by clustering the data based on both the similarity of quasiidentifier values and the diversity of sensitive attribute values. In the subsequent adjustment phase, the algorithm splits and merges the clusters to ensure that they each possess at least k members and adhere to the t-closeness requirements. Finally, the algorithm replaces the quasi-identifier values of the records in each cluster with the values of the cluster center to attain *k*-anonymity and *t*-closeness. Experimental results on three microdata sets from Facebook, Twitter, and Google+ demonstrate the proposed algorithm's ability to preserve the utility of released data by minimizing the modifications of attribute values while satisfying the k-anonymity and t-closeness constraints.

Keywords: information theory; entropy; data anonymization; clustering; privacy-preserving; individuals' privacy

1. Introduction

Data privacy protection is a critically important area of research, especially as data collection from individuals increases and proliferates rapidly. Data often contains sensitive information that could pose privacy risks if released without proper processing. To ensure the privacy of individuals, many countries have enacted privacy protection laws, such as the General Data Protection Regulation (GDPR) in the European Union, the Personal Data Protection Act (PDPA) in Singapore, and the Personal Information Protection Law (PIPL) in China. These laws aim to regulate data handling practices and safeguard individuals' personal information. As data collection and usage continue to grow exponentially, effective privacy protection techniques remain vital to uphold individuals' rights while enabling data-driven innovation.

Preserving data privacy presents several challenges, the most significant being the prevention of information leakage during data release. Data can contain different types of information; therefore, it is crucial to identify confidential data and assess the potential consequences of a data breach. Identifying and classifying sensitive data are key steps in developing an effective data privacy protection strategy. Once the sensitive data has been



Citation: Khatir, R.A.; Izadkhah, H.; Razmara, J. Designing a Novel Approach Using a Greedy and Information-Theoretic Clustering-Based Algorithm for Anonymizing Microdata Sets. *Entropy* **2023**, *25*, 1613. https:// doi.org/10.3390/e25121613

Academic Editor: Shu-Chuan Chu

Received: 19 October 2023 Revised: 16 November 2023 Accepted: 22 November 2023 Published: 1 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). identified, organizations can determine the appropriate data privacy protection measures to implement. To protect individuals' privacy, it may be necessary to anonymize the data while considering the type of data, the intended use, and the potential privacy risks.

Data anonymization is a practical and widely adopted approach to protecting individual privacy when releasing data. Anonymization techniques aim to reduce the risk of unauthorized disclosure by converting personal information into anonymous data. This balanced approach to safeguarding privacy mitigates the original details in the dataset to some extent [1]. Anonymizing data involves modifying quasi-identifier values to reduce specificity, enhancing privacy protection while preserving data utility [2].

Data containing confidential information about individuals is prone to privacy breaches, so it is essential to protect sensitive information from privacy threats [3]. One way to protect personal privacy is to erase direct identifier attributes, such as names and addresses, before releasing the original data to the public. This process, known as de-identification, is insufficient to protect privacy, as a study by Sweeney [4] showed that 87% of Americans can still be uniquely identified based on just three quasi-identifiers: gender, age, and postal code. To prevent privacy threats, researchers have proposed several privacy models, including k-anonymity (KA) [5], l-diversity (LD) [6], and t-closeness (TC) [7]. In the following, we briefly overview these models and their anonymization operations.

Anonymization techniques have become increasingly important for ensuring privacy in data publishing. Among these techniques, KA is widely used by researchers [5]. KA aims to protect privacy by modifying quasi-identifiers to ensure that each record is indistinguishable from at least *k*-1 other records. It is important that individual records remain faithful to the original data while meeting privacy requirements after anonymization [8]. Generalization, suppression, and perturbation are the main anonymization operations used to achieve k-anonymization algorithms:

- Generalization replaces some quasi-identifier values with parent values in the equivalent class, while specialization is the opposite operation. Generalization preserves data privacy but reduces data accuracy [3]. Excessive generalization results in significant information loss (IL) and reduced data quality [9].
- Suppression replaces specific quasi-identifier values with asterisks (*), while the disclosure is the opposite. Suppression causes more IL than generalization. For example, suppose we have a dataset of people's ages. We could generalize this dataset by replacing all ages with a range of ages, such as 20–30, 30–40, and so on. This would reduce the accuracy of the dataset, but it would still preserve some information about the people's ages. On the other hand, if we suppress data points from the dataset, we would be losing information altogether.
- Perturbation involves modifying a feature's value by replacing it with another value to minimize the difference between the original and modified datasets. One of the most popular perturbation techniques is microaggregation, which clusters records in the dataset with a restriction on cluster size. Microaggregation replaces the values of each record in the anonymized dataset with the values of the cluster center to which the record belongs. This approach is more effective than generalization and suppression [10] and simplifies further data analysis.

Table 1 shows a sample of de-identified original data, and Table 2 shows its 3anonymized version. The Weight, Gender, and Age attributes are set as quasi-identifier attributes, and disease is the sensitive attribute. All records in a cluster share the same values for the quasi-identifier attributes, making them indistinguishable from each other. However, the 3-anonymity model in Table 2 does not prevent attribute disclosure. For example, if an adversary knows that a person's record is in cluster C1 in Table 3, they can infer that the person has pneumonia.

No.	Weight	Sex Age		Disease
1	73	Male	33	Pneumonia
2	56	Female	36	Pneumonia
3	82	Male	31	Pneumonia
4	71	Female	44	Bronchitis
5	51	Female	47	Colon cancer
6	68	Female	40	Flu
7	70	Female	55	Colitis
8	69	Male	60	Colon cancer
9	48	Female	59	Flu

Table 1. Patients' original data.

Table 2. 3-Anonymity version of original data.

Cluster	Weight	Sex	Age	Disease
	70.33	Male	[30–39]	Pneumonia
C1	70.33	Male	[30–39]	Pneumonia
	70.33	Male	[30–39]	Pneumonia
	63.33	Female	[40-49]	Bronchitis
C2	63.33	Female	[40-49]	Colon cancer
	63.33	Female	[40-49]	Flu
	62.33	Female	[50–59]	Colitis
C3	62.33	Female	[50–59]	Colon cancer
	62.33	Female	[50–59]	Flu

Table 3. 3-Anonymity 2-Diversity version of original data.

Cluster	Weight	Sex	Age	Disease
	64.66	Female	[40-49]	Pneumonia
C1	64.66	Female	[40-49]	Colon cancer
	64.66	Female	[40-49]	Colitis
	57.33	Female	[40-49]	Pneumonia
C2	57.33	Female	[40-49]	Flu
	57.33	Female	[40-49]	Flu
	74.00	Male	[40-49]	Pneumonia
C3	74.00	Male	[40-49]	Colon cancer
	74.00	Male	[40-49]	Bronchitis

While KA is effective in preventing identity disclosure, it remains vulnerable to attribute disclosure attacks. Specifically, if the range of sensitive attribute values within a k-anonymous cluster is limited, there is a risk of inferring the value of individuals' sensitive attributes. Several extended KA models have been proposed to address this issue [11], including LD [6], which ensures a minimum degree of diversity (*l*) of sensitive values in each cluster.

Table 3 shows 3-anonymity and 2-diversity to the original data presented in Table 1. All records within each cluster in Table 3 have the same value for the quasi-identifier but contain distinct values for the sensitive attribute. This means that even if an adversary knows the cluster containing a target individual's record, they cannot narrow down or disclose the person's specific sensitive value (disease).

However, LD also has drawbacks [12]. For example, if an attacker knows that a person's record is in cluster C2 in Table 3, they can infer that the person has the Flu with a probability of 66%. To address these limitations, Li et al. [7] introduced the concept of TC.

To meet TC requirements, the distribution of sensitive information must be similar for each group of indistinguishable records and the entire original dataset. Table 4 shows an anonymized version of the original data from Table 1 that satisfies 3-anonymity and 0.33-closeness. This anonymized table is safe for public release because it protects the privacy of the individuals in the dataset. In addition to protecting against the attacks mentioned above, TC also offers safeguards against potential skewness attacks and similarity attacks, as detailed in [7].

Cluster	Weight	Sex	Age	Disease
	74.00	Male	[40-49]	Pneumonia
C1	74.00	Male	[40-49]	Colon cancer
	74.00	Male	[40-49]	Bronchitis
	61.00	Female	[40-49]	Pneumonia
	61.00	Female	[40-49]	Flu
C^{2}	61.00	Female	[40-49]	Flu
02	61.00	Female	[40-49]	Pneumonia
	61.00	Female	[40-49]	Colon cancer
	61.00	Female	[40-49]	Colitis

Table 4. A 0.33-Closeness version of original data.

TC provides stronger privacy guarantees compared to other techniques like LD. It protects against attribute disclosure by restricting the distance between sensitive attribute distributions in any cluster and the original distribution. The parameter *t* effectively controls the balance between privacy protection and data utility. Decreasing *t* enhances privacy by requiring sensitive attribute distributions to more closely mirror the original distribution. However, it can reduce utility if overly strict clustering is needed to satisfy the tighter threshold. Conversely, increasing *t* may compromise privacy protection as it allows more divergence from the original distribution. However, it can improve utility by permitting more flexible clustering.

The Earth Mover's Distance (EMD) [13] is used to calculate the TC metric using variational distance [7,14] as follows:

$$EMD(P,Q) = \frac{1}{2} \sum_{i=1}^{m} |p_i - q_i|$$
(1)

where p_i and q_i are the probabilities of distributions *P* and *Q*, respectively. It is assumed that |P| = |Q| = m and both *P* and *Q* are normalized distributions, that is $\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} q_i = 1$.

As Figure 1 shows, there is an inverse relationship between data privacy and data utility, meaning that improving one inevitably reduces the other [15]. Therefore, a fundamental goal of anonymization is to balance these two factors when creating an anonymized dataset suitable for publication. This balance requires determining an acceptable trade-off between privacy and utility.



Data Utility

Figure 1. Trade-off between privacy and utility level.

Table 5 summarizes the key abbreviations used throughout this paper for ease of reference.

Description	Abbreviation
K-Anonymity	KA
L-Diversity	LD
T-Closeness	TC
Information Loss	IL
Earth Mover's Distance	EMD
Normalized Avarege QI-group	CAVG
Average Distortion Ratio	ADR
Greedy Clustering-based Anonymization Algorithm	GCAA

Table 5. The main abbreviation used in the paper.

The Contribution

In recent years, researchers have explored clustering-based anonymization methods to protect people's privacy from potential intruders [16–20]. These methods involve identifying important attributes to anonymize and using clustering techniques to group records according to their characteristics and similarities. The desired level of privacy protection can be achieved by modifying the attribute values of records within each cluster [21].

Byun et al. formulated the KA problem as a clustering problem and proposed the *k*-member algorithm to achieve optimal KA [20]. To enhance this algorithm, Honda et al. [22] developed a fuzzy variant of the *k*-member algorithm. Rahimi et al. [23] successfully implemented all three KA, LD, and TC restrictions on the original microdata using the X-means clustering algorithm. Langari et al. [24] also satisfied all three aforementioned constraints by formulating the anonymization problem as an optimization problem and subsequently employing an evolutionary algorithm and modifications to sensitive attribute values. Abbasi et al. [16] introduced a clustering process using the K-means++ method to achieve optimal KA.

Most real-world datasets contain non-numerical data (categorical data), which presents challenges for anonymization using generalization and suppression since categorical features lack natural distance metrics. For example, quantifying the distance between two different occupations, such as 'dentist' and 'farmer', is difficult. Therefore, most existing anonymization methods often focus on numerical data. This paper designs a novel approach for anonymizing categorical data using a greedy and information-theoretic clustering-based algorithm to protect microdata privacy against identity and attribute disclosure attacks.

The algorithm aims to minimize mutual information loss of records within each cluster while ensuring strict privacy protection. To this end, the clustering-based anonymization algorithm operates on a binary data table converted from the original data table. The output is an anonymized binary data table where some attribute values have been modified to achieve strict privacy protection. The proposed method is specifically designed to anonymize non-numerical microdata so it can be publicly released for research and knowledge discovery while protecting privacy. We evaluated the proposed algorithm against four different approaches, including state-of-the-art methods, on three different datasets. The results demonstrated its superiority in achieving strict privacy protection while preserving data utility. It ensures that each cluster satisfies both KA and TC, protecting the anonymized data against identity and attribute disclosure attacks.

To summarize, the key contributions of this study are:

- Proposing a 3-phase anonymization algorithm that aims to achieve strict privacy protection while maximizing the data utility.
- To form clusters, we used the loss of mutual information and proposed an agglomerative clustering based on information bottleneck.
- Considering the diversity of sensitive values in addition to the similarity of quasiidentifier values during clustering to reduce the cost of implementing privacy constraints.

The paper is structured as follows: Section 2 introduces the key fundamental concepts necessary to understand privacy-preserving data anonymization, while Section 3 reviews existing methods. Section 4 then provides a detailed explanation of the proposed anonymization algorithm. In Section 5, an empirical performance evaluation and comparison of GCAA is conducted against four state-of-the-art benchmark methods. Both the privacy protections and utility preservation of the different algorithms are analyzed and discussed. Finally, Section 6 offers conclusions and outlines potential future work.

2. Basic Definition

Datasets can be modeled as tables, with each row containing a record of an individual and each column containing values related to their features. Features in tables to be anonymized are often classified into four main categories [25].

- Direct identifiers: Such as names, national ID numbers, and email addresses, allow for direct identification of individuals. Thus, it is essential to remove them from the original data table before anonymizing the data.
- Quasi-identifiers: They are attributes that do not directly identify an individual but can be used to do so when linked to external data sources.
- Sensitive Attributes: They are confidential information that individuals may not want to share publicly. Disclosing this information to third parties can be harmful and may be misused. Examples of sensitive attributes are disease, income, and political or religious views.
- Non-Sensitive Attributes: They are any information that does not fall into the other categories. Examples of non-sensitive attributes include physical characteristics such as eye or hair color. Non-sensitive attributes are only released if they are essential for further analysis.

Orooji et al. [26] classified disclosure into two types: identity disclosure and attribute disclosure.

- *Identity Disclosure*: Individuals can be identified from a dataset with accurate identity disclosure. External information sources, such as voter lists, can be used to identify a person, as shown in Figure 2.
- Attribute Disclosure: It occurs when an attacker can link a person to a sensitive attribute in a dataset. This is often the result of identity disclosure, which uniquely identifies an individual and makes it easier to associate their sensitive attributes [27].

By linking a person to a record in the anonymized dataset using external sources, an attacker can identify that record and associate the values of its sensitive attributes with the person's identity, a process known as re-identification. The KA is proposed to avoid re-identification.



Figure 2. Identity disclosure using external information.

2.1. k-Anonymity

Given a data table *T*, the KA model requires that every record in the modified data table T^* be indistinguishable from at least *k*-1 other records with respect to their quasiidentifiers. To achieve KA, the original dataset is partitioned into equivalence classes, each containing at least *k* members sharing the same quasi-identifier values. These equivalence classes can be referred to as QI-groups [28] or QI-clusters [29].

2.2. *l-Diversity*

The LD is a more robust privacy notion than KA, addressing some of the latter's limitations. A cluster is *l*-diverse if it contains at least *l* distinct values of the sensitive attribute. A table is considered *l*-diverse if all its clusters satisfy the LD criterion. The LD principle ensures that each cluster has at least *l* distinct values for the sensitive attribute [7].

2.3. t-Closeness

The primary objective of the TC is to restrict the information an attacker can extract about sensitive attribute values within a particular cluster. It achieves this by requiring the distribution of sensitive attribute values within each cluster to be sufficiently similar to the distribution of sensitive attribute values in the entire dataset. The TC privacy metric is successful when all clusters satisfy the constraint. A cluster satisfies TC if the distance between the two distributions is smaller than t.

2.4. Information Theory Concept

Because our new privacy framework uses an information theory metric, we briefly overview the basic concepts of information theory [30] used in this paper.

• Information Entropy: Information entropy, also known as Shannon entropy, is the average amount of information in each event. The more uncertain an event is, the higher its entropy; the less uncertain an event is, the lower its entropy. Let *X* be a discrete random variable that takes its values from the set *X'*. If $P(x_i)$ is the probability density function of values x_i ($x_i \in X'$), the entropy of variable *X* is defined as follows:

$$H(X) = -\sum_{x_i \in X'} P(x_i) \log P(x_i)$$
⁽²⁾

• *Conditional Entropy*: It measures the uncertainty associated with a random variable given the value of another random variable. Let *Y* be another discrete random variable with values y_j taken from the set *Y'*. The conditional entropy H(Y|X) is calculated as follows:

$$H(Y|X) = -\sum_{x_i \in X'} P(x_i) \sum_{y_i \in Y'} P(y_i|x_i) \log P(y_i|x_i)$$
(3)

Conditional entropy measures the additional information required to predict the value of *Y* given the value of *X*, beyond what can be predicted from *X* alone.

As shown in Figure 3, it can be used to measure the uncertainty in predicting the value of a discrete random variable when the value of another variable is known. Mutual information measures the extent to which knowing the value of one variable enables the prediction of another.



Figure 3. Relation between mutual information and conditional entropy.

Mutual information: Mutual information is expressed as follows:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X)$$
(4)

where I(X; Y) = 0 if and only if X and Y are independent random variables.

This concept enables us to identify which clusters can be merged at each stage of the clustering process to minimize mutual information loss.

Kolbeck-Leibler (KL) divergence: KL divergence, also known as relative entropy, is a metric that measures the dissimilarity between two probability distributions. It provides a criterion for determining how different one distribution is from another. A KL divergence value of 0 indicates that the two distributions are similar in behavior, though not necessarily identical. Conversely, a value of 1 for this measure implies that the two distributions exhibit opposite behaviors. Given two probability distributions, P and Q, defined over a set F, the KL divergence quantifies how much they diverge from each other.

$$D_{KL}[P||Q] = \sum_{f \in F} P(f) \log \frac{P(f)}{Q(f)}$$
(5)

3. Related Work

Most state-of-the-art anonymization techniques use a clustering-based KA algorithm to achieve privacy preservation. These techniques partition input data into distinct clusters, containing a minimum of *k* records, before applying anonymization operations to create an anonymized output. The following section investigates several representative approaches to privacy-preserving.

LeFevre et al. [31] introduced the *Incognito* algorithm, a bottom-up approach that utilizes global generalization to generate an optimal *k*-anonymous dataset. However, this method has a high level of IL due to the full-domain generalization technique, which can negatively impact the usefulness of the anonymized data. To address this issue, Ref. [32] proposed a top-down greedy algorithm called *Mondrian*, which uses local recoding to generate an optimal *k*-anonymous dataset.

Byun et al. [20] proposed a clustering-based approach to solve the KA problem using a *k*-member greedy algorithm to find the optimal solution. The method partitions the dataset into clusters with at least *k* members and generalizes the quasi-identifier values in each cluster to achieve KA. However, it is sensitive to outlier records because it selects the furthest record from the previous cluster to create a new one, which leads to a high amount of IL. Honda et al. [22] introduced a fuzzy version of this approach that considers the degree of fuzzy membership when assigning records to clusters. However, this approach may overfit the data to satisfy KA, a significant drawback.

Abbasi et al. [16] proposed a clustering-based approach for privacy preservation in healthcare datasets. They first used the normal distribution function to identify and remove less frequent data, which is more vulnerable to privacy breaches. Then, they used the K-means++ algorithm to generate clusters during the clustering process. Finally, they applied the KA algorithm to each cluster separately to achieve the desired level of privacy preservation.

Truta et al. [29] proposed a *p*-sensitive *k*-anonymous technique to protect against homogeneity and background knowledge attacks. The algorithm ensures that each equivalent class has at least *p* distinct values ($p \le k$). However, this technique leads to a high degree of IL when the distribution of sensitive attribute values is not uniform. To enhance this approach, Sun et al. [28] proposed two improved versions of the *p*-sensitive *k*-anonymity technique: p^+ -sensitive *k*-anonymity and (p, α)-sensitive *k*-anonymity. Their approach first categorizes the sensitive attributes and then focuses more on the categories of sensitive values rather than their specific values. However, this method may still result in a high amount of IL in particular scenarios.

Solanas et al. [33] proposed a heuristic microaggregation-based algorithm to apply *p*-sensitive *k*-anonymity to the original dataset. Their algorithm first creates clusters with at least *k* members that satisfy the *p*-sensitive *k*-anonymity property. To achieve this, it randomly selects a record and then adds the closest records with different sensitive values to form the cluster. Finally, the algorithm anonymizes the original dataset by replacing the values of each record with the values of the cluster centroid to which the record belongs. This method reduces the loss of information and provides higher data utility than other *p*-sensitive *k*-anonymity techniques.

Existing approaches that incorporate diversity in sensitive values have limitations, especially in protecting individuals' privacy from attribute disclosure attacks. Researchers have suggested a privacy-preserving model known as TC to address this drawback.

Rahime et al. [23] proposed a 3-layer clustering algorithm that applied KA, LD, and TC constraints for anonymizing the original dataset. This approach protected individuals' privacy against identity and attribute disclosure attacks, but its main weakness was the high amount of IL.

Langari et al. [24] proposed a hybrid approach for anonymizing graph and tabular data. Their method first partitions the data into clusters with at least *k* members using the fuzzy clustering technique and then uses the Firefly evolutionary algorithm to optimize the anonymization and clustering parameters simultaneously. LD and TC were implemented using the modification of sensitive values. However, its scalability is questionable due to the use of an evolutionary algorithm.

Our Contributions against the Existing Methodology

Table 6 summarizes several state-of-the-art algorithms. These algorithms use different anonymization techniques and operations. According to the table, most algorithms were designed to satisfy only the KA constraint and do not adequately protect anonymized data against attribute disclosure attacks. Algorithms that provide more protection by satisfying LD or TC constraints suffer from a high amount of IL or low data utility. This is because almost all of these algorithms only consider the similarity of quasi-identifiers when creating clusters. Therefore, to satisfy the privacy limitation, they are forced to merge more clusters, which in the worst case, could result in a single cluster with zero data utility.

Most anonymization techniques use generalization operations, which tend to incur high levels of IL (except for the algorithms in [24,33]). However, anonymization inherently reduces data usefulness to some degree by obscuring details. Therefore, it is crucial to

consider the trade-off between privacy and IL to maintain the utility of anonymized data for knowledge extraction.

Paper	Privacy Model	Operation	Key Demerits
[29]	KA	Generalization	Insufficient to protect against attribute disclosure (AD) attacks High amount of IL
[33]	KA, LD	Microaggregation	Insufficient to protect against AD attacks
[22]	KA	Generalization	Insufficient to protect against AD attacks Overfitting to data
[23]	KA, LD, TC	Generalization	High amount of IL Low amount of data utility
[16]	KA	Generalization	Insufficient to protect against AD attacks
[24]	KA, LD, TC	Perturbation	High computational complexity Lack of scalability

Table 6. The review of state-of-the-art privacy-preserving algorithms.

Compared to the proposed method, most investigated approaches perform poorly in maintaining the trade-off between privacy protection and data usefulness. Additionally, unlike the proposed method, some of these techniques remove data points from the table as outliers, while others change the values of the sensitive attributes to meet the LD and TC requirements. This leads to a significant decrease in the usefulness of the data.

To overcome the limitations of prior work, a new greedy, information-theoretic clustering algorithm is proposed for microdata anonymization. It differs from existing techniques in several important ways:

- The proposed algorithm clusters data points by considering both the similarity of quasi-identifier values and the diversity of sensitive values within each cluster. This approach ensures that clusters have high intra-cluster similarity for quasi-identifiers and high intra-cluster diversity for sensitive attribute, which reduces both the modification of values and the merging of clusters as a result of increasing the usefulness of the data.
- The proposed algorithm protects released data from identity and attribute disclosure attacks by satisfying the KA and TC privacy constraints. These constraints ensure that individuals are indistinguishable from the anonymized data and that the distribution of sensitive attribute values is preserved.
- As shown in Figure 1, there is a trade-off between data privacy and data utility. Existing methods prioritize data privacy at the expense of data utility, but our algorithm achieves strict privacy by adhering to KA and TC while minimizing IL. This approach strikes a balance between data privacy and utility, ensuring that the anonymized data remains useful for its intended purposes.

4. Proposed Approach

Creating a dataset that satisfies the TC constraint through generalization involves solving an optimization problem to determine the minimum generalization required to meet the TC requirement. A common strategy for this is an iterative process of progressively generalizing one or more attributes until the resulting dataset meets the TC constraint.

Traditional methods for achieving TC first cluster records into groups containing at least *k* members to ensure KA. KA can be achieved through generalization techniques that replace quasi-identifier values with broader categories or microaggregation techniques that substitute original values with statistical approximations. Next, the methods merge clusters of records to construct a *t*-close dataset. The clusters are progressively merged to minimize the distance between sensitive attribute distributions across merged clusters. The distance metric guides the selection of cluster mergers, identifying combinations that most

reduce divergence between clusters' sensitive field distributions. This process continues by merging clusters until all satisfy the threshold for TC.

Generating *t*-close datasets using the traditional methods is costly, primarily due to the reduction in data utility. A key factor is the lack of consideration for sensitive attribute value diversity during initial clustering. Real-world datasets often contain records with similar quasi-identifiers but different sensitive attribute values. To enhance data utility, our proposed approach generates clusters having more similar quasi-identifiers and more diverse sensitive values. Additionally, merging clusters to achieve *t*-closeness can severely compromise data utility, sometimes rendering it entirely useless (all clusters merged into one). A more effective strategy could be to swap records between clusters instead of merging clusters, which could potentially mitigate this problem.

Pre-processing, clustering, and adjustment are the three phases of the proposed approach to generating anonymized datasets. In the pre-processing phase, the original data table *T* is converted into a binary table, T_b , with values of 1 or 0. T_b is then normalized to the range [0, 1] and divided into two subtables: quasi-identifier, T_{QI} , and sensitive attribute, T_{SA} . In the clustering phase, a bottom-up clustering algorithm is used to generate clusters with similar quasi-identifier values but distinct sensitive values within each cluster. Finally, to create the anonymized table T^* , which satisfies the KA and TC requirements for strict privacy, we apply an adjustment process. The quasi-identifier values in each cluster are replaced with the centroid quasi-identifier value to anonymize the data. Figure 4 shows the overall workflow of the proposed approach.

4.1. Pre-Processing

The original input data table *T* contains *n* records from the set \mathcal{R} , each of them having *m* values of attributes from the set \mathcal{F} which is represented by the matrix *M*.

First of all, attribute values are converted to binary values. Let feature F_j from the set \mathcal{F} have a value domain $F_j = \{v_{j1}, v_{j2}, \dots, v_{jd_j}\}$ (for each $j = 1, 2, \dots, m$). Therefore, each record $r \in R$ takes exactly one value from the set F_j for its *j*th feature. Let $F = F_1 \cup F_2 \cup \ldots \cup F_m$ be the set of all possible values of the attributes in the original table. Also, let $d = d_1 + d_2 + \cdots + d_m$ be equal to the size of *F*.

In this case, the original matrix converted to a matrix M_b with the size $n \times d$, where each record $r \in R$ in this matrix is a vector with binary values of features size d. If the record r contains the attribute value f, the value of $M_b[r, f]$ equals 1, otherwise is equal to 0. For example, according to Table 7, the first feature is the *Gender*, and its values domain is {*Male*, *Female*}. In the binary table, if the person is male, only the attribute *Male* value is 1 and *Female* value is 0, and vice versa.

		Quasi-Identifiers			ve Attributes
ID	Gender	Status	Nationality	Disease	Salary
1	Male	Married	Russian	HIV	>50 K
2	Female	Single	American	Flu	>50 K
3	Female	Married	Iranian	Cancer	≤50 K
4	Male	Married	Iranian	HIV	\leq 50 K
5	Male	Single	American	Cancer	>50 K

Table 7. Original data table (*T*).

Next, we identify quasi-identifiers and sensitive attributes in the data table and partition the feature set \mathcal{F} into two sets, \mathcal{F}_{QI} and \mathcal{F}_{SA} . We then construct two sub-tables, T_{QI} and T_{SA} , from the original data table, ensuring that each record retains its unique *ID* across both sub-tables. In the next step, we normalize the values in both sub-tables to lie within the interval [0, 1], so that the sum of the values for each record in each sub-table equals 1.



Figure 4. Overall flowchart of the proposed algorithm.

Let *R* and *F* (F_{QI} and F_{SA}) be discrete random variables whose values are taken from sets *R* and *F*, respectively. In this case, each record in the two sub-tables has the conditional probability distribution $P(f_{QI}|r)$ (for all $f_{QI} \in F_{QI}$) and $P(f_{SA}|r)$ (for all $f_{SA} \in F_{SA}$), respectively. The normalized sub-tables of the original Table 7 are shown in Tables 8 and 9.

Table 8. The normalized quasi-identifiers version (T_{QI}) of Table 7.

	The Quasi-Identifiers Values Domain						
ID	Male	Female	Married	Single	Russian	American	Iranian
1	1/3	0	1/3	0	1/3	0	0
2	0	1/3	0	1/3	0	1/3	0
3	0	1/3	1/3	0	0	0	1/3
4	1/3	0	1/3	0	0	0	1/3
5	1/3	0	0	1/3	0	1/3	0

The Sensitive Attributes Values Domain						
ID	HIV	Flu	Cancer	>50 K	\leq 50 K	
1	1/2	0	0	1/2	0	
2	0	1/2	0	1/2	0	
3	0	0	1/2	0	1/2	
4	1/2	0	0	0	1/2	
5	0	0	1/2	1/2	0	

Table 9. The normalized sensitive attribute version (T_{SA}) of Table 7.

4.2. Clustering

The two sub-tables generated in the previous phase, T_{QI} and T_{SA} , serve as input to the clustering phase. The proposed bottom-up clustering algorithm simultaneously considers both T_{QI} and T_{SA} to group the records. The goal is to form clusters where quasiidentifier values are highly similar within each cluster, but sensitive values remain diverse. Algorithm 1 outlines the bottom-up clustering process.

Algorithm 1 The Pseudocode for the Clustering Process

- 1: **Input:** Microdata table *T* with $n \times m$ dimensions, parameters *K*, *T*, β , and γ
- 2: **Output:** Clusters $C_1, C_2, ..., C_c$ (c < n)
- 3: Let *n*: the number of records, *m*: be the number of features, and *d*: the size of the feature values domain, *S*: the distribution of sensitive values in the dataset.
- 4: Convert data table *T* to binary table T_b with $n \times d$ dimensions
- 5: Select quasi-identifiers and sensitive attributes, then establish two sub-tables T_{QI} and T_{SA} with unique *ID*
- 6: Initially, let each record be a distinct cluster, $C_n = (c_1, c_2, ..., c_n)$
- 7: Calculate $P(c_i)$ based on Equation (6) for i = 1, 2, ..., n
- 8: while $\delta I(c_{n+1}) \delta I(c_n) \leq \gamma \operatorname{do}$
- 9: Calculate $\delta I(c_i, c_j)$ based on Equation (13) for all $i \neq j$
- 10: Select a pair of clusters c_x , c_y that have minimum $\delta I(c_x, c_y)$
- 11: **if** $|c_x| + |c_y| \le K + \beta$ then
- 12: **if** $EMD(c_x \cup c_y, S) < min(EMD(c_x, S), EMD(c_y, S))$ **then**
- 13: Merge c_x and c_y , and update both sub-tables based on Equations (8) and (9) and go to Line 7
- 14: **else**
- 15: Select the next pair of clusters and go to Line 12
- 16: **end if**
- 17: else
- 18: Select the next pair of clusters and go to Line 11
- 19: **end if**
- 20: end while
- 21: Return obtained clusters C_1, C_2, \ldots, C_c

A *k*-clustering C_k partitions the records of data table *T* into *k* clusters $C_k = \{c_1, c_2, c_3, ..., c_k\}$, where each cluster $c_i \in C_k$ has three properties as follows: (1) $|c_i| \neq 0$, (2) $c_i \cap c_i = 0$ for all $i, j, i \neq j$, (3) $\bigcup_{i=1}^k c_i = T$. We define the probability of each record *r* in data table *T* as p(r) = 1/n, where *n* is the number of records in *T*. Then, for each cluster $c \in c_k$ we have:

$$P(c) = \sum_{r \in c} P(r) = \frac{|c|}{n} \tag{6}$$

For example, from Tables 8 and 9, one can conclude that $P(r_i) = \frac{1}{5}$ for i = 1, 2, ..., 5. In addition, $P(f_{OI}|r_1)$ and $P(f_{SA}|r_i)$ are the feature vectors for the quasi-identifier and sensitive attribute of record r_1 , respectively. They are equal to [1/3, 0, 1/3, 0, 1/3, 0, 0] and [1/2, 0, 0, 1/2].

Initially, each record is assumed to be an independent cluster. We denote this partition as C_n , where C_n is a partition with *n* clusters $\{c_1, c_2, ..., c_n\}$ such that:

$$P(c_i) = 1/n, (i = 1, 2, \dots, n)$$
(7)

At each iteration, the proposed algorithm seeks to generate a new cluster with similar quasi-identifier values and distinct sensitive values from the existing clusters. To this end, we employ the Agglomerative Information Bottleneck algorithm [34]. The algorithm starts with a C_n partition. In this case, $I(F; R) = I(F; C_n)$, and in each step, it merges two clusters, reducing the number of clusters by one. According to the definition of mutual information, this causes an increase in the uncertainty of predicting the features with respect to clusters, or, equivalently, a reduction of the information that clusters contain about the values of the features, i.e., $I(F; C_{n-1}) \leq I(F; C_n)$. We use δI to represent this information reduction. Therefore, at each step, the algorithm attempts to merge the pair of clusters with the minimum δI .

After merging two clusters c_i and c_j , new cluster c_{ij} has:

$$P(c_{ij}) = P(c_i) + P(c_j)$$
(8)

$$P(f|c_{ij}) = \frac{P(c_i)}{P(c_{ij})} P(f|c_i) + \frac{P(c_j)}{P(c_{ij})} P(f|c_j)$$
(9)

for all $f \in F$. Tishaby et al. [35] showed that:

$$\delta I(c_i, c_i) = P(c_{ij}) \times JSD[P(f|c_i), P(f|c_j)]$$
⁽¹⁰⁾

for all $f \in F$. *JSD* in Equation (10) is the Jensen-Shannon divergence defined below:

$$JSD[P(f|c_i), P(f|c_j)] = \frac{P(c_i)}{P(c_{ij})} D_{KL}[P(f|c_i)||\bar{P}] + \frac{P(c_j)}{P(c_{ij})} D_{KL}[P(f|c_j)||\bar{P})]$$
(11)

where D_{KL} is KL divergence, and \overline{P} is avarge distribution of $P(f|c_i)$ and $P(f|c_j)$ and calculated by Equation (9). Finally, the $\delta I(c_i, c_j)$ which $i \neq j$ is calculated with the following formula:

$$\delta I(c_i, c_j) = P(c_i) \sum_{f \in F} P(f|c_i) \log \frac{P(f|c_i)}{P(f|c_{ij})} + P(c_j) \sum_{f \in F} P(f|c_j) \log \frac{P(f|c_j)}{P(f|c_{ij})}$$
(12)

where $\delta I(c_i, c_j) = \delta I(c_j, c_i)$.

At each step, the proposed algorithm calculates δI of each pair of clusters c_i, c_j ($i \neq j$) for both sub-tables T_{QI} and T_{SA} as $\delta I_{QI}(c_i, c_j)$ and $\delta I_{SA}(c_i, c_j)$, respectively. Then $\delta I_F(c_i, c_j)$ is calculated according to the following formula:

$$\delta I_F(c_i, c_j) = \delta I_{QI}(c_i, c_j) - \delta I_{SA}(c_i, c_j)$$
(13)

At each step, the algorithm aims to merge the pair of clusters with the minimum δI_F value. However, if two clusters have identical quasi-identifiers (i.e., $\delta I_{QI} = 0$), the algorithm selects the pair of clusters with the highest diversity in sensitive values for merging. By following this strategy, we can step-by-step create clusters of records with similar quasi-identifiers but different sensitive values. This process reduces IL and improves the quality of anonymous data.

To create balanced clusters and minimize changes during TC enforcement, our algorithm imposes two additional restrictions on merging. To speed up the process and create more evenly sized clusters, we remove any cluster that has between k and $k + \beta$ (i.e., $k \le |c| \le k + \beta$) data points from the clustering process. To make it easier to enforce TC during the adjustment phase, two clusters are merged only if they meet the following criteria:

$$EMD(S_{c_{ii}}, S) < min(EMD(S_{c_i}, S), EMD(S_{c_i}, S))$$
(14)

where S is the distribution of sensitive values in the entire dataset, and S_{c_i} is the distribution of sensitive values in cluster *i*.

The algorithm runs until it reaches a stopping point, defined as follows: if the change in mutual information after each merging step δI_F exceeds the threshold γ , the merging process stops, and the final clusters are formed.

4.3. Adjustment

The adjustment phase is a crucial step in the proposed algorithm to satisfy the KA and TC constraints. It takes as input the set of clusters generated in the clustering phase, denoted as C_1, C_2, \ldots, C_c , and outputs a new set of clusters, denoted as $C_1^*, C_2^*, \ldots, C_c^*$ $(c^* \leq c)$, that all satisfy the KA and TC constraints while preserving the data quality. The resulting anonymized table T^* can be easily constructed using the modified clusters. Algorithms 2 and 3 present the pseudocode for the adjustment phase.

Algorithm 2 The Pseudocode for the Adjustment Phase 1: **Input:** Clusters C_1, C_2, \ldots, C_c , parameter *K* and *T* 2: **Output:** Clusters $C_1^*, C_2^*, \ldots, C_{c^*}^*$ ($c^* \leq c$), Anonymized table T^* 3: Let D_1 , D_2 , D_{21} , and D_{22} be empty sets 4: Let SortedList: Sort all clusters based on their EMD in descending order 5: **for** i = 1, 2, ..., c **do** if C_i satisfy KA and TC then 6: 7: ADD C_i To D_1 8: else 9: ADD C_i To D_2 10: end if 11: end for for $j = 1, 2, ..., |D_2|$ do 12: 13: if $|C_i| \geq K$ then ADD C_i To D_{21} 14: 15: else ADD C_i To D_{22} 16: end if 17: 18: end for 19: **for** all clusters c in D_{22} **do** 20: while *c* is not empty **do** Randomly select a record *r* From *c* 21: $c = c \setminus \{r\}$ 22: 23: *p* = *Findthebestcluster*(SortedList, *r*) 24: $p = p \cup \{r\}$ end while 25: 26: end for 27: **for** all clusters c in D_{21} **do** while *c* is not empty **do** 28: Randomly select a record r from c29: $c = c \setminus \{r\}$ 30: *p* = *Findthebestcluster*(SortedList, *r*) 31: $p = p \cup \{r\}$ 32: end while 33: 34: end for

Algorithm 3 Function: Findthebestcluster(C,r)

- 1: **Input:** A set of clusters C that sorted based on their *EMD* in descending order, and a record r
- 2: **Output:** A cluster $c \in C$ such that the $EMD(c \cup \{r\}, S)$ is decreased
- 3: Let n = |C|, best = null
- 4: **for** i = 1, 2, ..., n **do**
- 5: **if** $EMD(c_i \cup r, S) < EMD(c_i, S)$ **then**
- 6: best = c_i with the lowest *ADR* value
- 7: break
- 8: end if
- 9: end for
- 10: Return best

The input set of clusters is partitioned into two subsets: D_1 , which contains the clusters that satisfy both KA and TC, and D_2 , which contains the clusters that do not meet one or both of these constraints. D_2 is further partitioned into two subsets: D_{21} , which contains clusters with more than *k* records, and D_{22} , which contains clusters with less than or equal to *k* records.

The algorithm then proceeds to modify the clusters in D_{22} by transferring their records to clusters in D_{21} using the *Findthebestcluster* function to minimize the EMD between the merged clusters. If the EMD of the modified cluster is less than or equal to the predefined threshold *T*, the cluster is moved from D_{21} to D_1 .

The process continues until D_{22} is empty. If D_{21} is not empty, the algorithm selects the cluster with the highest EMD value and transfers its records to the best cluster based on the *Findthebestcluster* function until all clusters in D_{21} satisfy the TC constraint and are moved to D_1 .

5. Performance Evaluation and Discussion

To evaluate the proposed algorithm, we use two metrics to measure the quality of anonymized datasets.

• Normalized Avarege QI-group (CAVG): The metric was proposed by [32]. It measures the quality of the KA constraint. The closer the value to 1, the better the result. The normalized average group is defined as follows:

$$CAVG = \frac{N}{C \times k} \tag{15}$$

where C is the number of clusters and *k* is the privacy parameter.

• Average Distortion Ratio (ADR): This metric measures the amount of distortion (IL) introduced by the anonymization process. If a feature value is not changed, there is no distortion. If a feature value is changed, the distortion increases by 1. The metric is defined as follows:

$$ADR = \frac{\sum_{i=1}^{N} \sum_{j=1}^{d} F_{ij}}{N \times d}$$
(16)

where $F_{ij} = 1$ if *j*th features of record *i* is modified and 0 otherwise, *N* is the number of records, and *d* is the domain of the feature values.

To assess the effectiveness of the algorithm in protecting against identity and attribute disclosure attacks, we use two penalty criteria: PKA and PTC. These criteria are defined as follows:

- P_{KA} is incremented by 1 if any cluster does not satisfy the KA constraint.
- P_{TC} is incremented by 1 if any cluster does not satisfy the TC constraint.

These criteria enable an evaluation of the algorithm's ability to satisfy the desired privacy constraints while minimizing the risk of information disclosure.

The proposed anonymization algorithm was developed in Python version 3 and implemented on a personal computer with a 2.6 GHz (Core i7) processor and 16 GB of memory. The effectiveness of the proposed Greedy Clustering-based Anonymization Algorithm (GCAA) was compared to four other clustering-based anonymization techniques: K-anonymity L-diversity T-closeness 3Layer (KLT3L) [23], K-member Fuzzy Clustering Firefly Algorithm (KFCFA) [24], P-Sensitive K-Anonymity (PSKA) [28], and Fuzzy K-Member Clustering (FKMC) [22]. The comparison was performed in terms of their ability to protect against identity and attribute disclosure attacks.

To ensure a fair comparison with existing methods, we designed our experiment to match the setup of [24] as closely as possible. We used three microdata sets (https://snap.stanford.edu/data/, accessed on 10 February 2023) that were also used in the [24]. Each dataset contained a table of records, with each record having a set of features. The details of the microdata sets are summarized in Table 10.

Table 10. Microdata sets characterstics.

Dataset	Number of Records	Number of Features	The Domain of Features Values
Facebook	347	19	224
Twitter	244	28	1364
Google+	352	6	289

The level of protection against intruder attacks increases with higher values of k and t but at the cost of increased I. To assess the performance of our algorithm under different conditions, we considered three different privacy scenarios, each with different parameter values. A summary of the scenarios is presented in Table 11.

Considering the privacy requirements and the potential severity of intruder attacks, we chose different values for the anonymization parameters *k* and *t* for each privacy scenario. As shown in Table 11, we considered three scenarios, ranging from simple to strict, with corresponding values of *l* set to 3, 4, and 6, respectively. We also set the parameters β and γ to $\lfloor K/2 \rfloor$ and 0.5, respectively. To terminate the clustering process, we monitored the difference between two consecutive δI values for merging. Clustering was stopped when this value exceeded 0.5, at which point the current clusters were considered to be the final clusters.

Table 11. Different privacy scenarios.

Privacy Scenario	Parameter k	Parameter t
Simple Privacy	4	0.7
Medium Privacy	6	0.5
Strict Privacy	8	0.3

5.1. Experimental Result for Simple Privacy Scenario

Table 12 displays the results obtained for a simple privacy scenario on the Facebook micro dataset using the proposed algorithm. In this scenario, the algorithm generated 70 clusters from the data. Cluster sizes ranged from a minimum of four records to a maximum of eight records. All clusters satisfied the LD requirement, containing at least *l* distinct sensitive attribute values. Sensitive attribute diversity within clusters varied between 3 and 7 different values.

Importantly, all clusters generated by the algorithm satisfied both KA and TC privacy criteria. This is because the clustering process prioritized forming groups with a high diversity of sensitive values. Additionally, none of the 70 clusters produced were susceptible to attribute disclosure attacks. By ensuring sensitive attribute diversity within each cluster, the proposed method successfully anonymized the data while preventing homogeneity and skewness of sensitive values that could otherwise compromise privacy according to the attribute disclosure attack model.

It is worth noting that, unlike other techniques, our proposed approach did not remove any records from the original data table as outliers. These results demonstrate the effectiveness of the algorithm in achieving both KA and TC privacy goals while preserving data utility, as shown by retaining all records. A key advantage of our approach compared to other anonymization methods is the ability to satisfy strong privacy standards without eliminating any records. By clustering records rather than discarding outliers, more of the original information can be retained for analysis purposes. Maintaining all records has important implications for data utility, as no information is lost due to outlier removal. This highlights the strengths of the proposed method.

Table 12 shows that while the FKMC method performed best in terms of CAVG and ADR, it suffered from a significantly high P_{TC} error. Although PSKA and FKMC had no P_{KA} errors, they remained susceptible to attribute disclosure due to their non-zero P_{TC} penalty errors. Only the KLT3L and KFCFA methods generated clusters that fully satisfied the TC constraint, making them truly comparable to the proposed GCAA approach in terms of privacy protection.

Metrics	Description	GCAA (The Proposed Algorithm)	KLT3L	KFCFA	PSKA	FKMC
С	Number of Clusters	70	16	49	18	86
CAVG	Equation (13)	1.239	5.42	1.77	4.81	1.008
ADR	Equation (14)	0.0197	0.07	0.005	0.001	0.001
P_{KA}	Penalty of KA	0	0	0	0	0
P_{TC}	Penalty of TC	0	0	0	17	71
Time	Average Running Time	16	2.3	46	3.1	3.4

Table 12. The obtained result for a simple privacy scenario in the Facebook dataset.

The proposed GCAA method generates more balanced clusters than KLT3L and KFCFA, which improves CAVG by 78% and 33%, respectively, compared to KLT3L and KFCFA. Moreover, GCAA significantly reduces ADR compared to KLT3L, although it does not outperform KFCFA. KFCFA changes sensitive values to achieve TC, which significantly reduces data utility, whereas GCAA does not require such changes.

Table 12 also reports the average CPU running time of each model. The proposed GCAA algorithm showed computational performance that was nearly three times faster than the KFCFA approach. This significant difference can be attributed to KFCFA's use of an evolutionary algorithm during its optimization process, which induces greater computational overhead compared to GCAA.

Figures 5–7 show a comparative analysis of C, CAVG, and ADR for three microdata sets. The results show that the proposed GCAA algorithm performs better than KLT3L on all three datasets and outperforms KFCFA in most cases.



Figure 5. Comparison of C for simple privacy scenario.



Figure 6. Comparison of CAVG for simple privacy scenario.



Figure 7. Comparison of ADR for simple privacy scenario.

Figure 5 reveals that GCAA produces more balanced clusters than KLT3L for all three datasets, resulting in higher C values. Moreover, Figure 6 shows that GCAA consistently outperforms KLT3L in terms of CAVG across all datasets. Finally, Figure 7 demonstrates that GCAA significantly reduces ADR compared to KLT3L on all three datasets.

In addition, GCAA achieves competitive results with KFCFA, outperforming it in terms of C and CAVG on two datasets, as shown in Figures 5 and 6. While KFCFA performs slightly better than GCAA in terms of ADR on one dataset, Figure 7 shows that GCAA still achieves excellent results with a lower ADR than KFCFA on the other two datasets.

5.2. Experimental Result for Medium Privacy Scenario

To limit the anonymization process, the experimenters increased *k* to 6 and decreased *t* to 0.5. The results were compared across different models, as shown in Table 13.

Similar to the previous scenario, FKMC outperformed other methods in terms of C and CAVG. At the same time, GCAA, KLT3L, and KFCFA were superior to PSKA and FKMC due to more robust privacy protection. These methods satisfied the TC constraint in all clusters, resulting in zero P_{TC} error. KLT3L merged many clusters to satisfy the TC constraint, resulting in a reduction in the number of final clusters to 12. This resulted in a larger CAVG of 4.82, which is almost four times larger than GCAA, and an ADR that was twice as large. KFCFA outperformed the proposed GCAA in this experiment, as shown in Figures 8 and 9.

These figures compare the C and CAVG criteria for GCAA, KLT3L, and KFCFA in the medium privacy scenario on three microdata sets. The proposed GCAA had a lower CAVG error in all datasets than KLT3L. KFCFA performed better than GCAA on the Facebook and Twitter datasets, but GCAA had fewer errors than KFCFA on the Google+ dataset. Figure 10 shows a similar comparison for ADR criteria, with the proposed GCAA performing better than KLT3L in most cases.

Metrics	GCAA	KLT3L	KFCFA	PSKA	FKMC
С	43	12	49	17	57
CAVG	1.344	4.81	1.18	3.40	1.01
ADR	0.0242	0.046	0.021	0.001	0.001
P_{KA}	0	0	0	0	0
P_{TC}	0	0	0	17	37

Table 13. The obtained result for a medium privacy scenario in the Facebook dataset.



Figure 8. Comparison of C for medium privacy scenario.



Figure 9. Comparison of CAVG for medium privacy scenario.



Figure 10. Comparison of ADR for medium privacy scenario.

5.3. Experimental Result for Strict Privacy Scenario

To evaluate the algorithms under an even more restrictive privacy setting, the experimenters increased the KA parameter k to 8 and decreased the TC parameter t to 0.3. Table 14 shows the results of different approaches on the Facebook microdata set. KLT3L and KFCFA generated 4 and 25 clusters, respectively, with CAVGs of 10.48 and 1.735. The

proposed GCAA algorithm improved the ADR error by 70.5% and 0.7% compared to KLT3L and KFCFA, respectively.

Metrics	GCAA	KLT3L	KFCFA	PSKA	FKMC
С	25	4	31	10	43
CAVG	1.735	10.84	1.399	4.33	1.0081
ADR	0.0268	0.091	0.027	0.001	0.001
P_{KA}	0	0	0	0	0
P_{TC}	0	0	0	10	43

Table 14. The obtained result for a strict privacy scenario in the Facebook dataset.

Figures 11–13 compare the C, CAVG, and ADR results obtained from the proposed GCAA, KLT3L, and KFCFA under the strict privacy scenario on three microdata sets. The proposed approach showed superior performance by generating more balanced clusters on the Twitter and Google+ datasets, resulting in less error than the other two methods for CAVG and ADR criteria. Although KFCFA produced more clusters on the Facebook dataset than the proposed GCAA, the ADR error was not lower because KFCFA significantly changed the feature values to achieve TC.



Figure 11. Comparison of C for strict privacy scenario.



Figure 12. Comparison of CAVG for strict privacy scenario.



Figure 13. Comparison of ADR for strict privacy scenario.

5.4. Experimental Result for Different Values of Parameter T

To investigate the impact of the parameter t, a set of experiments was carried out where the parameter t was systematically varied from 0.3 to 0.7, while holding k and l fixed at 6 and 4, respectively. In this analysis, the Google+ microdata set was used to observe the effect of changing t on the C, CAVG, and ADR criteria. Since decreasing t complicates the problem of achieving the TC constraint, it is expected to increase various error rates. As a result, with a decrease in t, CAVG increases, indicating the appearance of larger clusters.

Figures 14 and 15 show that the proposed GCAA performs better than KLT3L and KFCFA in terms of C and CAVG metrics, especially in scenarios where privacy is very important. However, as seen in Figure 16, ADR increases for all methods as *t* decreases. Despite its shortcomings, the proposed GCAA still outperforms KLT3L and KFCFA in strict privacy settings.



Figure 14. Comparison of the C for different values of T in the Google+ dataset.



Figure 15. Comparison of the CAVG for different values of *T* in the Google+ dataset.





5.5. Discussion

In terms of privacy, the proposed GCAA strategy along with KLT3L and KFCFA fully addresses the risks of identity and attribute disclosure attacks by satisfying both the KA and TC requirements. The PSKA and FKMC methods only partially mitigate these risks, as they do not perfectly satisfy TC. However, the resistance of GCAA and some other algorithms to similarity attacks could be improved. Specifically, they currently lack a criterion for determining the semantic similarity of attribute values during clustering. For example, in a medical context, the algorithms have no way to judge that diseases like colon cancer and liver cancer should be considered highly similar rather than diverse, for the purpose of TC. Not accounting for such attribute-relatedness weakens protections against similarity attacks.

In terms of data utility, FKMC and PSKA methods had the lowest amount of IL, as measured by ADR, and therefore the best data utility. The proposed method and KFCFA performed similarly well, but the KLT method lost a significant amount of IL. This suggests that the proposed method is effective at preserving the usefulness of anonymized data for future knowledge extraction.

In terms of computational efficiency, the clustering phase has a time complexity of $O(n^2 \times d^2 \times \log n)$, where *d* is the size of the attribute's domain and *n* is the number of records. However, if n >> d, it can be concluded that the time complexity is $O(n^2 \times \log n)$. The proposed adjustment phase for splitting and merging clusters has a time complexity of $O(n^2)$.

In terms of average execution time, the proposed method outperforms KFCFA but underperforms other methods. KFCFA is slow, unscalable, and computationally expensive due to its evolutionary nature. The proposed method is also slower than other methods due to its time complexity. Because it provides a high level of privacy protection and data utility, the proposed method is suitable for anonymizing publishable data in inactive mode, where the data owner publishes the entire anonymized dataset after making some modifications to the original data [36]. In this case, the data owner prioritizes the level of privacy and utility of anonymized data.

These findings are consistent with the results of other experiments, highlighting the effectiveness of the proposed GCAA approach in achieving high privacy protection while minimizing the distortion of the original data. A brief review of the above, key findings emerge:

- 1. As privacy protection increases, the data becomes more distorted, which reduces its usefulness. This is consistent with Figure 1.
- 2. The proposed GCAA method, as well as the KLT3L and KFCFA methods, can protect data from identity and attribute disclosure attacks by meeting both KA and TC.
- 3. The KLT3L algorithm has higher error measures (CAVG and ADR) than the proposed methods because it needs to merge more clusters and make more errors to meet the TC constraint.

- 4. The proposed GCAA outperforms the KFCFA method in some cases, while the KFCFA method produces better results in other cases. However, changing the values of sensitive features to meet the TC constraint can reduce the usefulness of the data.
- 5. The results demonstrate the effectiveness of the proposed GCAA algorithm. The main reasons are: (1) It considers the diversity of sensitive feature values and the similarity of quasi-identifier values simultaneously during clustering (2) It generates evenly sized clusters that are consistent with the KA and TC constraints. (3) It uses an adjustment algorithm to produce an anonymized dataset with maximum data utility.

6. Conclusions and Future Works

The proposed Greedy Clustering-based Anonymization Algorithm (GCAA) provides a novel contribution to the field of data privacy. It uses information theory concepts and performs bottom-up hierarchical clustering to form clusters with similar quasi-identifier values and diverse sensitive attribute values. The algorithm iteratively merges pairs of clusters with the minimum loss of mutual information and reduces the EMD criterion. Clusters with a cardinality between *k* and $k + \beta$ removed from the clustering process to ensure that balanced clusters satisfying the KA and TC constraints are formed.

In the adjustment phase, the GCAA algorithm enforces the KA and TC constraints by splitting and merging clusters that do not meet these requirements while reducing the EMD. We evaluated GCAA's performance based on the CAVG and ADR criteria and compared it with four other methods: KLT3L, KFCFA, PSKA, and FFKMC. Our experimental results demonstrate that GCAA efficiently forms balanced clusters while providing robust privacy protection and preserving data quality. GCAA is a faster method than KFCFA that provides similar levels of privacy protection and IL. However, the proposed algorithm may require a longer execution time than some other methods due to its time complexity.

The proposed algorithm exhibits several limitations. Firstly, it is tailored for nonnumeric data, rendering it unsuitable for numerical data anonymization. Secondly, the algorithm is confined to single-sensitive attribute scenarios. Finally, the algorithm is restricted to tabular data and is ill-suited for graphical or structural data analysis. GCAA has practical applications in anonymizing scalable microdata where privacy preservation is crucial while retaining useful information for data mining purposes.

The primary objectives of our future research to enhance the proposed method include:

- Defining a criterion to evaluate the similarity between attribute values based on the proposed structure, thereby reinforcing the algorithm against similarity attacks.
- Extending the proposed method to handle datasets with multiple sensitive attributes.
- Expanding the algorithm to anonymize numerical data in addition to non-numerical data.
- Other suitable clustering methods for categorical data could also be considered, such as partitioning around medoids using the Gower coefficient [37]. This method is well-suited for datasets containing binary values, similar to the data used in our experiments. Evaluating the performance of partition around medoids and comparing it to the proposed approach may provide additional insights.
- Reducing the time complexity of the algorithm.

Author Contributions: Conceptualization, R.A.K. and H.I.; Software, R.A.K.; Validation, R.A.K., H.I. and J.R.; Formal analysis, R.A.K.; Investigation, R.A.K., H.I. and J.R.; Data curation, R.A.K.; Writing—original draft, R.A.K. and H.I.; Writing—review & editing, H.I. and J.R.; Visualization, R.A.K.; Supervision, H.I. and J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Srijayanthi, S.; Sethukarasi, T. Design of privacy preserving model based on clustering involved anonymization along with feature selection. *Comput. Secur.* 2023, *126*, 103027. [CrossRef]
- 2. Majeed, A.; Lee, S. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE Access* **2020**, *9*, 8512–8545. [CrossRef]
- 3. Aldeen, Y.A.A.S.; Salleh, M.; Aljeroudi, Y. An innovative privacy preserving technique for incremental datasets on cloud computing. *J. Biomed. Inform.* **2016**, *62*, 107–116. [CrossRef]
- 4. Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness* -Knowl.-Based Syst. 2002, 10, 571–588. [CrossRef]
- Sweeney, L. k-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzziness-Knowl.-Based Syst. 2002, 10, 557–570. [CrossRef]
- 6. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data* (*TKDD*) **2007**, 1, 3-es. [CrossRef]
- Li, N.; Li, T.; Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2006; pp. 106–115.
- 8. Belsis, P.; Pantziou, G. A k-anonymity privacy-preserving approach in wireless medical monitoring environments. *Pers. Ubiquitous Comput.* **2014**, *18*, 61–74. [CrossRef]
- Zheng, W.; Wang, Z.; Lv, T.; Ma, Y.; Jia, C. K-anonymity algorithm based on improved clustering. In Proceedings of the Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018, Guangzhou, China, 15–17 November 2018; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2018; pp. 462–476.
- 10. Soria-Comas, J.; Domingo-Ferrer, J.; Sanchez, D.; Martinez, S. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3098–3110. [CrossRef]
- 11. di Vimercati, S.D.C.; Foresti, S.; Livraga, G.; Samarati, P. k-Anonymity: From Theory to Applications. *Trans. Data Priv.* 2023, 16, 25–49.
- Lee, H.; Kim, S.; Kim, J.W.; Chung, Y.D. Utility-preserving anonymization for health data publishing. BMC Med. Inform. Decis. Mak. 2017, 17, 1–12. [CrossRef] [PubMed]
- 13. Rubner, Y.; Tomasi, C.; Guibas, L.J. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99. [CrossRef]
- 14. Li, N.; Li, T.; Venkatasubramanian, S. Closeness: A new privacy measure for data publishing. *IEEE Trans. Knowl. Data Eng.* **2009**, 22, 943–956.
- 15. Mehmood, A.; Natgunanathan, I.; Xiang, Y.; Hua, G.; Guo, S. Protection of big data privacy. *IEEE Access* 2016, 4, 1821–1834. [CrossRef]
- 16. Abbasi, A.; Mohammadi, B. A clustering-based anonymization approach for privacy-preserving in the healthcare cloud. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6487. [CrossRef]
- Arava, K.; Lingamgunta, S. Adaptive k-anonymity approach for privacy preserving in cloud. *Arab. J. Sci. Eng.* 2020, 45, 2425–2432. [CrossRef]
- Madan, S.; Goswami, P. A privacy preserving scheme for big data publishing in the cloud using k-anonymization and hybridized optimization algorithm. In Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 21–22 December 2018; pp. 1–7.
- Ni, S.; Xie, M.; Qian, Q. Clustering Based K-anonymity Algorithm for Privacy Preservation. *Int. J. Netw. Secur.* 2017, *19*, 1062–1071.
 Byun, J.W.; Kamra, A.; Bertino, E.; Li, N. Efficient k-anonymization using clustering techniques. In Proceedings of the Advances in Databases: Concepts, Systems and Applications: 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, 9–12 April 2007; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2007; pp. 188–200.
- 21. Fouad, M.R.; Elbassioni, K.; Bertino, E. A supermodularity-based differential privacy preserving algorithm for data anonymization. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1591–1601. [CrossRef]
- Honda, K.; Kawano, A.; Notsu, A.; Ichihashi, H. A fuzzy variant of k-member clustering for collaborative filtering with data anonymization. In Proceedings of the 2012 IEEE International Conference on Fuzzy Systems, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–6.
- 23. Rahimi, M.; Bateni, M.; Mohammadinejad, H. Extended k-anonymity model for privacy preserving on micro data. *Int. J. Comput. Netw. Inf. Secur.* **2015**, *7*, 42–51. [CrossRef]
- 24. Langari, R.K.; Sardar, S.; Mousavi, S.A.A.; Radfar, R. Combined fuzzy clustering and firefly algorithm for privacy preserving in social networks. *Expert Syst. Appl.* **2020**, *141*, 112968. [CrossRef]
- 25. Eyupoglu, C.; Aydin, M.A.; Zaim, A.H.; Sertbas, A. An efficient big data anonymization algorithm based on chaos and perturbation techniques. *Entropy* **2018**, *20*, 373. [CrossRef]
- 26. Orooji, M.; Knapp, G.M. A novel microdata privacy disclosure risk measure. arXiv 2019, arXiv:1901.07311.
- 27. Kiabod, M.; Dehkordi, M.N.; Barekatain, B. TSRAM: A time-saving k-degree anonymization method in social network. *Expert Syst. Appl.* **2019**, *125*, 378–396. [CrossRef]
- 28. Sun, X.; Sun, L.; Wang, H. Extended k-anonymity models against sensitive attribute disclosure. *Comput. Commun.* 2011, 34, 526–535. [CrossRef]

- Truta, T.M.; Vinay, B. Privacy protection: P-sensitive k-anonymity property. In Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA, 3–7 April 2006; p. 94.
- 30. Ben-Naim, A. Information Theory: Part I: An Introduction to the Fundamental Concepts; World Scientific: Singapore, 2017; Volume 1.
- 31. LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Incognito: Efficient full-domain k-anonymity. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 14–16 June 2005; pp. 49–60.
- LeFevre, K.; DeWitt, D.J.; Ramakrishnan, R. Mondrian multidimensional k-anonymity. In Proceedings of the 22nd International Conference on data Engineering (ICDE'06), Atlanta, GA, USA, 3–7 April 2006; p. 25.
- Solanas, A.; Sebé, F.; Domingo-Ferrer, J. Micro-aggregation-based heuristics for p-sensitive k-anonymity: One step beyond. In Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society, Nantes, France, 29 March 2008; pp. 61–69.
- 34. Slonim, N.; Tishby, N. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems* 12; MIT Press: Cambridge, MA, USA, 1999; Volume 12.
- 35. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. arXiv 2000, arXiv:physics/0004057.
- Mohammed, N.; Chen, R.; Fung, B.C.; Yu, P.S. Differentially private data release for data mining. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 493–501.
- 37. Alkhushayni, S.; Choi, T.; Alzaleq, D. Data analysis using representation theory and clustering algorithms. *WSEAS Trans. Comput.* **2020**, *19*, 310–320. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.