Hindawi
Security and Communication Networks
Volume 2021, Article ID 6612923, 17 pages
https://doi.org/10.1155/2021/6612923

Research Article

Proximity Measurement for Hierarchical Categorical Attributes in Big Data

Zakariae El Ouazzani,[1] **An Braeken**,[2] **and Hanan El Bakkali**[1]

[1]*Rabat-IT Center, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco*
[2]*Industrial Engineering Department (INDI), Vrije Universiteit Brussel (VUB), Brussels, Belgium*

Correspondence should be addressed to Zakariae El Ouazzani; zakariae.elouazzani@gmail.com

Received 17 November 2020; Revised 8 June 2021; Accepted 25 June 2021; Published 6 July 2021

Academic Editor: Fulvio Valenza

Nearly most of the organizations store massive amounts of data in large databases for research, statistics, and mining purposes. In most cases, much of the accumulated data contain sensitive information belonging to individuals which may breach privacy. Hence, ensuring privacy in big data is considered a very important issue. The concept of privacy aims to protect sensitive information from various attacks that may violate the identity of individuals. Anonymization techniques are considered the best way to ensure privacy in big data. Various works have been already realized, taking into account horizontal clustering. The *L*-diversity technique is one of those techniques dealing with sensitive numerical and categorical attributes. However, the majority of anonymization techniques using *L*-diversity principle for hierarchical data cannot resist the similarity attack and therefore cannot ensure privacy carefully. In order to prevent the similarity attack while preserving data utility, a hybrid technique dealing with categorical attributes is proposed in this paper. Furthermore, we highlighted all the steps of our proposed algorithm with detailed comments. Moreover, the algorithm is implemented and evaluated according to a well-known information loss-based criterion which is Normalized Certainty Penalty (NCP). The obtained results show a good balance between privacy and data utility.

1. Introduction

Nowadays, big data are very helpful in many sectors, like online banking, natural science, and preventive medicine [1]. Organizations collect huge amounts of data from various data sets for scientific research purposes and also to improve the life quality in various sectors. Medical data in particular are known to contain large-scale information gathered from heterogeneous data sets. The Electronic Health Records (EHRs), for example, store information belonging to specific patients. Nevertheless, such information contains sensitive values, which may breach the privacy of individuals. Privacy is usually defined as having the capacity to protect sensitive information from being violated [2]. One trivial way of ensuring privacy is removing Personally Identifying Information (PII) from published data sets [3]. However, there is a need for more sophisticated anonymization techniques to ensure privacy while preserving data utility in published data sets [4].

There are many anonymization techniques in the literature [5–8]. Some of them use the clustering principle. For example, *L*-diversity and *T*-closeness techniques use horizontal partitioning when treating numerical and categorical sensitive attributes. The *L*-diversity technique splits the data set into several buckets including only distinct values while guaranteeing that there are at least *L* "well-represented" sensitive records [9]. However, the current anonymization approaches taking into account *L*-diversity for hierarchical data suffer from the similarity attack and cannot prevent the disclosure of data very well [10]. In spite of having buckets that contain only distinct values, these values may be semantically similar; thus, through a similarity attack, an adversary could easily violate the privacy of a certain individual by knowing his or her sensitive attributes' values.

In order to prevent the similarity attack, we had previously proposed a technique called "variable *T*-closeness for sensitive numerical attributes" [11]. However, the technique

only processes sensitive numerical attributes. In this paper, we propose a hybrid technique based on proximity measurement in order to deal with categorical attributes. Our proposed algorithm has been inspired by the T-closeness principle, permutation, and K-anonymity techniques. The basic idea of T-closeness principle is to measure the distance between the distribution of sensitive attributes' values in each bucket and the overall distribution of this attribute in the data set [12]. The measurement could be done through a distance metric such as Earth Mover's Distance (EMD) [13]. Then, taking into account this distance, the distribution of values into each bucket is scattered in such a way that each bucket in the data set does not include values belonging to any specific category.

In our proposed algorithm, there is dynamism in the way of dealing with the hierarchy, which includes categorical values. In order to resist the similarity attack, the algorithm will apply the anonymization process on buckets that includes values belonging to a specific category. Thus, the adversary will be prevented from knowing, for example, the exact type of a patient's disease. Our proposed algorithm will be evaluated based on an effective measurement, which is NCP in order to measure the degree of information loss [14–18].

The remainder of the paper is organized as follows. In Section 2, we define the problem position and give some useful definitions. In Section 3, we present some related work treating nonnumerical sensitive attributes in order to ensure privacy in big data. Next, in Section 4, we present our proposed algorithm applied to sensitive categorical attributes. Later, in Section 5, we introduce the notion of the privacy measurement NCP. After that, in Section 6, we present experimental results with an evaluation of the proposed algorithm and we discuss the obtained results. Finally, we conclude our paper and give some perspectives in Section 7.

2. Preliminaries

In this section, some useful definitions related to Quasi-Identifier (QI) and sensitive attributes, K-anonymity, and L-diversity techniques are given. Next, we will define the problem position.

2.1. Background and Definitions

2.1.1. Quasi-Identifier and Sensitive Attributes. In domain knowledge, dealing with categorical attributes is a hard task because every single algorithm will accept only numerical values. Categorical attributes could be represented in the literature into two forms which are QI and sensitive attributes. The first one represents a set of information that contains personal details other than identifiers [19, 20]. In addition, QI attributes like "Age" and "Gender" attributes do not identify a record directly and are separate by themselves [21, 22]. Even though the individual identifiers are removed, QI attributes can be utilized by a malicious person to track an individual and to reveal its identity when we gather these attributes together with the assistance of other pieces of background knowledge information [23, 24]. The second type is considered as private personal attributes [22, 25]. For instance, in a hospital data set, "Disease" is a sensitive attribute; in a financial data set, "CCV" number is a sensitive attribute, and the "annual income" is a sensitive attribute in a census data set [21, 23]. Moreover, sensitive attributes (SA) should be hidden while publishing and sharing data [21, 24]. Sensitive attributes could be exploited as well by third parties like data analysts, marketers, or even social media. Any third party with malicious intentions on users' sensitive information can be viewed as adversaries and they can breach user privacy by collecting sensitive data [26, 27]. Thus, sensitive attributes require more protection rather than QI attributes [28].

2.1.2. K-Anonymity Technique. K-anonymity is the basis of many anonymization researches so far. The key concept of this technique has been used in various domains of privacy models [29]. K-anonymity hides certain key information and anonymizes the QI attributes belonging to users in a data set in order to ensure privacy in big data [30]. In addition, this technique prevents the attacker from identifying a person in a data set by reducing the identification probability to $1/K$. Thus, the higher the value of the threshold K, the lower the probability of identification. The K-anonymity is realized if each record is similar to at least $K-1$ other records within each bucket in the data set [19, 30]. Although K-anonymity resists linkability attack, it does not include protection against attacks based on background knowledge and homogeneity attack [31, 32]. Moreover, it may lead to distortions of data and hence greater information loss [2].

2.1.3. L-Diversity Technique. In the K-anonymity technique, sensitive attributes are not treated; thus, individuals can be identified if their corresponding sensitive values are similar within each bucket [33]. In order to overcome the lack of diversity due to the application of the K-anonymity technique, a privacy model known as L-diversity is proposed in the literature. This technique prevents the adversary from inferring the sensitive information of individuals [33]. L-diversity is an anonymization technique used to ensure privacy in big data by reducing the granularity of data representation. In literature, there are three main L-diversity models: Distinct, Entropy, and Recursive. In this paper, we adopt the distinct L-diversity principle which ensures that each bucket (equivalence class) includes at least L "well-represented" distinct sensitive values [22, 23]. However, this technique is vulnerable to a similarity attack [2, 31, 32]. A similarity attack occurs when values within an equivalence class are the same or even when there is a semantic significance between distinct sensitive values in the same equivalence class [23, 34]. Therefore, an adversary can deduce the different possibilities of sensitive values related to a specific individual [35].

2.1.4. Similarity Attack. A similarity attack occurs when values within an equivalence class are the same or even when there is a semantic significance between distinct sensitive

values in the same equivalence class [23, 34]. Therefore, an adversary can deduce the different possibilities of sensitive values related to a specific individual [35]. In this paper, we treat categorical sensitive attributes where the values are presented in the form of a hierarchy. Since the values are categorical, we assign a number to every value in the hierarchy in such a way that we can calculate the distance between every two consecutive values within each equivalence class in the data set. We notice that the distance is calculated after giving an ascending order to the values within each equivalence class. In Section 4, we present in detail the way our algorithm resists similarity attack. Besides, Section 6 will highlight through an example our proposed algorithm processing and also the resistance to the similarity attack.

2.2. Problem Position. In the literature, as described above, there are two main types of data including QI and sensitive attributes. Thus, a data set containing these two types of data has to be anonymized before publication.

However, even if the K-anonymity and the distinct L-diversity principles were sequentially applied on QI and sensitive attributes respectively, an adversary would easily retrieve valuable information from the anonymized data set since L-diversity cannot resist a similarity attack as shown in Figure 1.

Therefore, our proposed algorithm will resist the similarity attack by breaking the semantic relation between sensitive values within each bucket in the data set and by consequence preventing the adversary from extracting valuable information from the anonymized data set as shown in Figure 2.

3. Related Work

A huge amount of nonnumerical data is accumulated and shared by organizations. The Electronic Health Records (EHRs) system, for example, deals with categorical attributes such as disease, symptom, and diagnosis methods [36]. Moreover, most of the anonymization techniques using L-diversity principle for hierarchical data cannot resist the similarity attack. For instance, the Maximum Delay Anonymous Clustering Feature (MDACF) tree data publishing algorithm was proposed in [5] where an anonymous CF tree is produced. Besides, L-diversity and K-anonymity principles are used to anonymize the cluster. The MDACF algorithm cannot resist the similarity attack since it satisfies K-anonymity while anonymizing the clusters.

Furthermore, Cui et al. [6] proposed a hierarchical multiple sensitive attributes algorithm based on the L-diversity principle. The algorithm establishes a hierarchical strategy based on the frequencies of the set of attributes. Although the privacy is ensured, the proposed technique in [6] cannot resist the similarity attack because each equivalence class of the anonymized data set includes identical values. Besides, Ozalp et al. [7] extend two anonymization techniques ensuring privacy in tabular data including K-anonymity and L-diversity by applying them to hierarchical data. Ozalp et al. [7] proposed a clustering procedure called ClusTree, which takes as input a data set of hierarchical records. Since the resulting buckets after the anonymization process do not care about the semantic relation between values, the proposed technique is not able to resist the similarity attack.

In addition, Huang [8] used in the anonymization process two techniques including K-anonymity and L-diversity in order to ensure privacy. Huang [8] selects the attribute with the greatest width when anonymizing the data in order to reduce the amount of information loss. However, in the anonymization process, the data are divided into equivalence classes with the same root, which means that there is a strong semantic relation between values within each equivalence class and consequently the proposed technique in [8] cannot resist the similarity attack. Additionally, Poovammal and Ponnavaikko [37] presented a technique, which takes into account a taxonomy tree of sensitive categorical attributes. The mapping table technique is realized belonging to the domain knowledge and the individual specifications. In other words, the privacy and disclosure levels related to each sensitive categorical attribute are determined by the user [37]. Again, the proposed technique cannot resist the similarity attack since the semantic relation between values is not taken into consideration during the anonymization process. Moreover, Wang et al. [38] proposed a K-anonymity clustering technique taking into account an analytic hierarchy. The main idea of the algorithm is to split the data set into several buckets in accordance with the degree of similarity between values. The clustering method is done in such a way that values within each bucket have higher similarity, which means that the proposed algorithm in [38] cannot resist the similarity attack.

Otherwise, Wang et al. [10] suggested a multilevel privacy-preserving approach based on fuzzy sets to ensure privacy. The proposed algorithm treats both numerical and categorical sensitive attributes by converting the categorical attribute value into a numerical attribute value based on its occurrence frequency. Unlike the previous algorithms, the suggested one in [10] can resist the similarity attack by partitioning sensitive values into five levels and then setting a sensitivity level for each sensitive value. Based on semantic rules, Mubark et al. [39] proposed a technique dealing with categorical data. The technique is based on the L-diversity technique and the semantic extraction belonging to both a repository and data owner semantic rules. The suggested algorithm in [39] also resists the similarity attack by using each semantic rule belonging to specific data that may breach privacy during the anonymization process.

Moreover, Jia and Chen in [40] proposed a (l, m, d)-anonymity model treating multiple sensitive attributes able to resist the similarity attack, where m represents the dimension of the sensitive attributes. The suggested model uses the semantic hierarchical tree to calculate and analyze the semantic dissimilarity between sensitive attribute values, and each equivalence class must include at least L sensitive attribute values satisfying d-different on each dimension sensitive attribute. At the same time, in order to make the

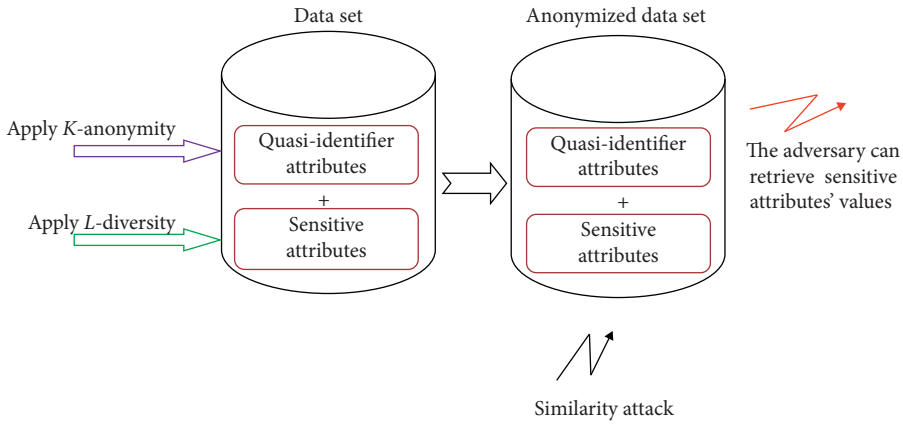

Figure 1: Data disclosure due to similarity attack.

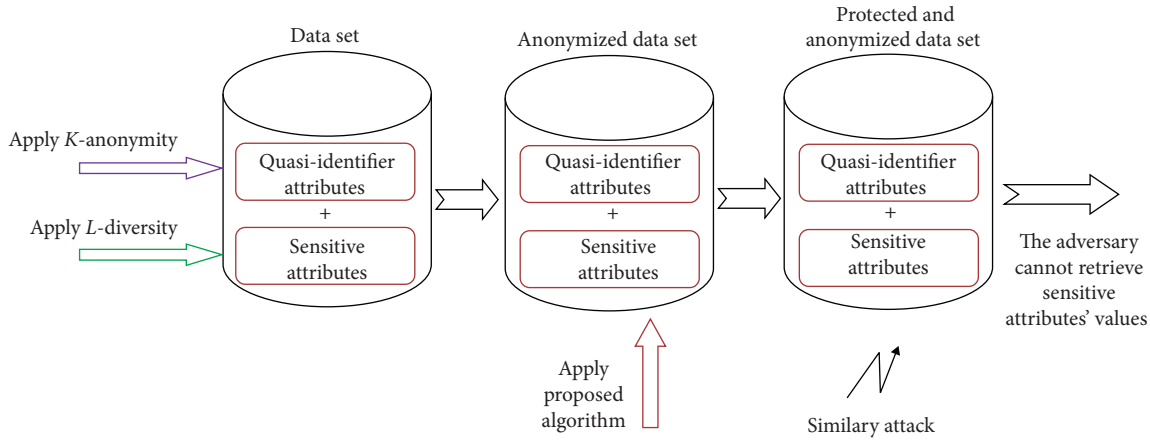

Figure 2: Our proposed anonymization process resisting similarity attack.

published data highly available, the proposed model adopts the distance-based measurement process to divide the equivalence classes. The authors in [40] assume that if the distance from the public ancestor nodes to the leaf nodes is 1, then they can conclude that the degree of dissimilarity is very small, so there are similar diseases. However, the choice of the values is not clear in the algorithm and the way the values are organized in the new equivalence classes is not well presented. In addition, the used hierarchy is not completed to detect how the values in the anonymized data set are permuted.

Even, Saeed and Rauf in [41] proposed a privacy-preserve data publishing algorithm by utilizing more than one PPDP approach (Anatomization with anonymization) to be a practical and effective tool for ensuring data privacy against membership, identity, and semantic similarity disclosure attacks. The buckets are formed by recursively selecting L closest tuples from L largest semantic groups rather than sequentially selecting records. Besides, the proposed algorithm in [41] does not mention the way the treated hierarchy helps in selecting the semantically similar values within equivalence classes. In addition, Kayem and Meinel in [42] suggested an efficient algorithm based on clustering as a heuristic classification to ensure that the distance between sensitive attributes and the cluster centroid is no more than a threshold value of T. The degree of similarity between a cluster and a sensitive attribute is computed by using a combination of severity rankings (cost to privacy due to attribute exposure). Besides, the distance between categorical attributes is measured by using Jaccard's coefficient. The authors in [42] assume that a high degree of similarity is captured by a smaller distance from the cluster centroid, and the reverse is true for a low similarity degree. Then, in order to ensure the protection against the similarity attack, the authors in [42] used the T-closeness principle by seeking to distribute tuples in such a way to ensure that the difference in distributions within both the equivalence classes and the whole data set does not exceed the threshold T of T-closeness.

Concerning Saraswathi and Thirukumar in [43], they applied the T-closeness over the Multisensitive Bucketization K-Anonymity Clustering Attribute Hierarchy (MSB-KACA) algorithm. The EMD is calculated for the data set in which the privacy is preserved using MSB-KACA algorithm. The T-closeness principle reorders the tuples in the data set in order to ensure that the sensitive values are equally distributed within the equivalence classes.

Wang et al. in [44] proposed a cluster-based algorithm for multiple sensitive attributes satisfying T-closeness. Concerning categorical attributes, Wang et al. used a

formula with the name of «distCat» which selects the values having the lowest common ancestor node in order to get the semantically similar values among the others existing in each equivalence class. However, an improved fuzzy c-means clustering (FCM) algorithm, called Equi-sized FCM, is used to partition the original records.

In addition, Hao and Ya-Bin in [45] proposed an algorithm based on T-closeness model using the same formula as done in [44] where the detection of semantically similar sensitive categorical values is realized based on calculating the distance between values in each equivalence class. However, the authors in [45] replace the categorical values by more ambiguous values corresponding to the original attribute values. Thus, even if the privacy is ensured, the data utility is not preserved. Besides, Kayem et al. in [46] suggested a modified T-closeness algorithm based on the notion of clustering providing protection against the similarity attack by computing the minimum cluster size required to guarantee a global minimum level of T-closeness for all clusters. The computation is done through measuring a distance that describes the similarity between every two tuples in the data set. When treating the categorical attributes, the measured distance is realized by using Jaccard's coefficient. However, looking only at the similarity of attributes in [46] to decide on classifications can result in a high number of outliers (a loss of information due to misclassification or the inability to classify) that are removed from the generated anonymized data set.

Besides, Wang et al. [47] proposed a maximal-bucket first (MBF) algorithm to achieve (l, e)-diversity. The goal is to split an original data set into various equivalence classes satisfying (l, e)-diversity constraint. First, the MBF algorithm puts all the records with e-similar sensitive values in the same equivalence class. According to a semantic hierarchy, two values are considered similar if they have the common parent on the tree. Otherwise, the two values are comparative dissimilar if they have the common great-grandparent. Second, the MBF algorithm selects records from various equivalence classes to form sequentially equivalence classes based on the size of buckets until the equivalence classes are (l, e)-diverse. The algorithm in [47] repeats the process of constructing equivalence classes until it cannot construct a new equivalence class satisfying (l, e)-diversity constraint. After that, the algorithm joins the remaining records to the generated equivalence classes, provided that the diversity of the equivalence classes is still achieved. Finally, the algorithm removes the remaining records which cannot be joined to any equivalence class.

Lately, the L-diversity is still problematic when there are semantically similar values in the equivalence classes. Thus, a novel privacy indicator, (l, d)-semantic diversity, and an algorithm that anonymizes a database to satisfy (l, d)-semantic diversity were proposed in [48]. The algorithm adds $L - 1$ dummy records to each true record in such a way that an adversary cannot understand the anonymized database in a simple way. Besides, the algorithm calculates the distance between the true and the dummy values to be able to detect the semantic similar values within each equivalence class. However, this distance is useful when treating numerical values. In fact, to define the distance between sensitive attribute values, the authors in [48] used the tree structure used by the International Classification of Diseases (ICD) and maintained by the World Health Organization. In our proposed algorithm PM-HCA, we treated categorical values whether they are numerical or nonnumerical. In addition, we did not add dummy values to the data set before applying the anonymization process in order to preserve the data utility.

Despite the fact that all the cited algorithms treat categorical data through a hierarchy, some of them do not resist the similarity attack. Contrary to the proposed algorithm in [41], we applied in our proposed algorithm the T-closeness principle on an L-diverse table where the threshold L is variable within the equivalence classes in the data set. Even in the exception of the work done in [42], the threshold T of T-closeness technique is variable in our proposed algorithm in such a way that the threshold T is recalculated after every step of the anonymization process. Normally, the EMD measurement is used to calculate the distance between numerical sensitive attributes. However, the authors in [43] assume that the EMD is also useful when treating categorical attributes without assigning numerical codes to each categorical value existing in the data set. In addition, the way the tuples are rearranged is not clear in [43] and may lead to identity disclosure if the permutation process is not well chosen. On the other side, there exists a problem when using the algorithm presented in [44] which is reflected in the fact of randomly assigning coefficients to each data point for being in the clusters, which reduces the data utility. In our proposed algorithm, we do not use fuzzy values when anonymizing the data set so as to preserve the data utility.

Our proposed technique resists the similarity attack by using a simpler method compared to the suggested techniques offering resistance to the similarity attack already mentioned. We do not need to divide sensitive values into levels and assign each value to its level as done in [10] to be able to detect values that are semantically similar. On the other side, we treat the semantic similarity between sensitive categorical values according to the treated hierarchy without transforming the content of the hierarchy into semantic rules as done in [39]. Besides, our proposed algorithm does not remove any record from the data set as done in [47] and it does not add dummy values to the data set before the anonymization process as done in [48]. In addition, our algorithm offers several options in the permutation process in order to ensure privacy as much as possible. Moreover, it reduces the time processing by specifying whether the number of buckets to be permuted is odd or even.

Table 1 is a summary of the anonymization techniques mentioned in the Related Work section. It summarizes the various techniques treating nonnumerical attributes or both numerical and nonnumerical attributes. In addition, we compared the techniques of Table 1 based on two criteria: the resistance to the similarity attack and the information loss. Concerning the second criterion, it represents whether the data are lost after the anonymization process. However, ensuring privacy implicates that the data must be lost. Thus, the challenge is to reduce the information loss as much as

possible. In our proposed anonymization technique, the information loss is reduced and did not even reach 50% as we mentioned in the discussion of Section 6. However, the reason behind every anonymization technique is to ensure privacy while the information loss can easily attend 100%. In the next section, we will discuss the source of inspiration that prompted us to realize the proposed technique. Then, we will present in detail our proposed algorithm called "Proximity Measurement for Hierarchical Categorical Attributes (PM-HCA)."

4. The Proposed Technique

Based on the clustering principle, there are various anonymization techniques, for example, K-anonymity and L-diversity which ensure privacy in big data. The objective of this paper is to ensure privacy in a data set containing sensitive categorical attributes. However, the existing techniques using the principle of L-diversity for hierarchical data may suffer from various limitations and thus cannot ensure privacy carefully [10, 11]. Among L-diversity for hierarchical data technique limitations, it does not take into account the semantic of categorical values.

In a previous work [11], we had suggested an algorithm called "variable T-closeness for sensitive numerical attributes" which addresses the L-diversity for hierarchical data technique limitation when treating sensitive numerical attributes. Our proposed algorithm splits the original data set into buckets in order to reduce the amount of information loss. In addition, we take into consideration an analytic hierarchy while applying a permutation technique during the anonymization process.

The adapted hierarchy of Figure 3 is formed by first selecting 10 diseases representing the leaves of our adapted hierarchy from "Careplans" data set [49]. Then, we tried to connect the leaves through edges to their corresponding parent nodes based on the degree of similarity between these leaves. As a result, we obtain four parent nodes including "Respiratory infection," "lung diseases," "Brain diseases," and "Gut diseases." After that, we have noticed that the previous parent nodes called also categories of diseases could be connected to two other parent nodes including "Vascular lung diseases" and "Gut-brain diseases." In the end, the last two parent nodes are associated with a node called "Respiratory Gut-brain diseases" representing the root node of our adapted hierarchy.

Before starting the algorithm and based on an adapted hierarchy (Figure 3), we convert categorical values into numerical ones. The conversion is done because it is easy to manipulate numerical data and it allows us to determine the degree of proximity between values, which is the main idea of our paper. The algorithm can use several hierarchies corresponding to various sensitive categorical attributes. The hierarchy of Figure 3, showing a tree for diseases, is converted to Table 2, where every disease attribute value is represented by a "Disease code," which is a numerical number of 3 digits corresponding with the depth of the hierarchy.

The conversion process is carried out according to the weights, assigned with red color to every node in the hierarchy as shown in Table 2. A deep assignment starting from the top of the hierarchy is realized in this process. The weights' assignment follows a specific order because our proximity test on categorical values is based on the consecutive character of values within each bucket. Thus, we sort "Disease code" values within each bucket in an ascending order. In the following, we present our proposed algorithm PM-HCA (Algorithm 1).

The complexity of our proposed algorithm PM-HCA is of the order of $O(N)$ where N denotes the number of values in the treated attribute's column. In fact, we treat all the buckets from line 6 to line 17 of the algorithm; then, the complexity will be of the order of $O(N)$. In addition, from line 18 to line 23 of the algorithm, we treat in the worst case all the buckets existing in the data set needing anonymization; then, the complexity will be of the order of $O(N)$. The line 24 remains where we generate the QI attributes. Let us consider that we have q QI attributes; then, the complexity will be of the order of $O(q * N)$. The total complexity of the whole proposed algorithm will be of the order of $O(N + N + q * N)$ which is approximately equal to $O(N)$. We notice that since our proposed algorithm is applied on a data set satisfying L-diversity principle as shown in Figure 2, the number of buckets N is known in advance.

The algorithm has two inputs, the original table with L-diversity property and the hierarchy, and returns the anonymized table. We get the initial buckets of the original table by applying the distinct L-diversity algorithm as mentioned in [9]. First, we start the algorithm by creating a vector, in line 2 of the algorithm, to store the number of buckets, which need anonymization. We notice that the index "j" corresponds to the values within every bucket. Thus, since we do not know the number of values within each bucket, we use the loop "while" and we calculate the difference "L" between every two consecutive "Disease code" values within each bucket until treating all the buckets in the table (from line 5 to 17) while the length of the bucket is still superior to "j." Then, if the difference between two consecutive values in a bucket is different from 1, then the current bucket does not need anonymization. Else, we store the index of the treated bucket in the variable vector. Next, in line 18, we make a test to know whether the number of buckets in vector is even or odd. If the number of buckets needing anonymization decision is even, then we apply the even permutation process where every two consecutive buckets existing in vector will be swapped. In that case, we gain time processing because we use a bucket only once. Else, if the decision variable is odd, then every two consecutive buckets in vector will be permuted and the last bucket will be swapped with an already anonymized bucket. By the end, we generalize the values related to QI attributes to prevent the adversary definitively from deducing any information and consequently ensuring privacy and at the same time resisting similarity attack.

In the next section, we present some useful information about the NCP criterion in order to evaluate the proposed algorithm.

Table 1: A summary of the anonymization techniques mentioned in the related work.

Authors	Used technique	Resistance to similarity attack	Information loss
Zhang et al. [5]	The Maximum Delay Anonymous Clustering Feature (MDACF) tree data publishing algorithm	No	Not decreased
Cui et al. [6]	A hierarchical multiple sensitive attributes algorithm	No	Decreased
Ozalp et al. [7]	ClusTree algorithm	No	Decreased
Huang [8]	K-anonymity and L-diversity algorithms	No	Decreased
Poovammal and Ponnavaikko [37]	The fuzzy based technique in preserving privacy	No	Decreased
Wang et al. [38]	K-anonymity clustering algorithm	No	Not decreased
Wang et al. [10]	A multilevel privacy-preserving approach in hierarchical data based on fuzzy sets	Yes	Not decreased
Mubark et al. [39]	Combination of L-diversity technique and semantic extraction techniques	Yes	Not decreased
Jia and Chen [40]	(l, m, d)-anonymity algorithm treating multiple sensitive attributes	Yes	Decreased
Saeed and Rauf [41]	A privacy-preserve data publishing algorithm	Yes	Decreased
Kayem and Meinel [42]	An efficient algorithm based on clustering as a heuristic classification	Yes	Decreased
Saraswathi and Thirukumar [43]	T-closeness over the Multisensitive Bucketization K-Anonymity Clustering Attribute Hierarchy (MSB-KACA) algorithm	Yes	Decreased
Wang et al. [44]	A cluster-based algorithm for multiple sensitive attributes satisfying T-closeness	Yes	Not decreased
Hao and Ya-Bin [45]	An algorithm based on T-closeness model	Yes	Decreased
Kayem et al. [46]	A modified T-closeness algorithm based on the notion of clustering	Yes	Decreased
Wang et al. [47]	Maximal-bucket first (MBF) algorithm to achieve (l, e)-diversity	Yes	Decreased
Oishi et al. [48]	Algorithm that anonymizes a database to satisfy (l, d)-semantic diversity	Yes	Not decreased

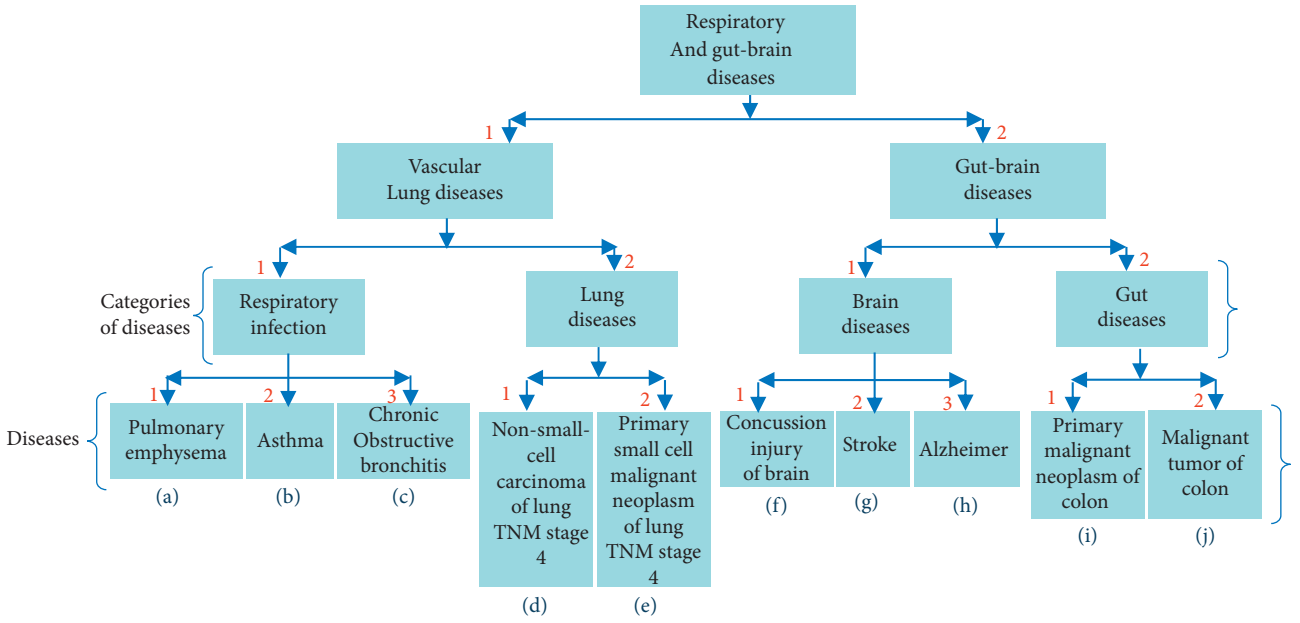

Figure 3: Adapted hierarchy for disease attribute.

5. Evaluation of the Algorithm

When anonymizing a data set, we do not have to focus only on ensuring privacy in big data but also on preserving data utility before publishing the anonymized data set. There are various metrics to quantify the usefulness of the data [15, 16]. In this paper, we will focus on the NCP in order to evaluate the results of our proposed algorithm.

In the literature, several studies have used the NCP to measure the information loss [14–18]. The utility is mostly

Table 2: Weight assignments to disease attribute.

Disease	Short disease representation	Disease code
Pulmonary emphysema	a	111
Asthma	b	112
Chronic obstructive bronchitis	c	113
Non-small-cell carcinoma of lung TNM stage 4	d	121
Primary small cell malignant neoplasm of lung TNM stage 4	e	122
Concussion injury of brain	f	211
Stroke	g	212
Alzheimer	h	213
Primary malignant neoplasm of colon	i	221
Malignant tumor of colon	j	222

```
(1) procedure Anonymization (TableBuckets, Hierarchy) /*TableBuckets: The initial table with different buckets, Hierarchy: the
    hierarchy of categorical attributes*/
(2) vector [N] array of integer //N is the number of buckets in TableBuckets
(3) z ⟵ 0
(4) decision ⟵ 0
(5) i ⟵ 0
(6) while i < N do
(7)       j ⟵ 0
(8)       while j < TableBuckets[i]. length do //for every tuple in TableBuckets[i]
(9)       L ⟵ TableBuckets [i][j + 1].DiseaseCode – TableBuckets[i][j].DiseaseCode
             //Measure the difference between two consecutive tuples in TableBuckets[i]
(10)      if L ≠ 1 then
(11)            //TableBuckets[i] does not need anonymization
(12)            go to 6
(13)      else
(14)         vector [z] ⟵ i //vector stores the indexes of buckets needing anonymization
(15)         z++
(16)         j++
(17)   i++
(18) decision ⟵ z (mod 2) //z begins from 0
(19) if decision/ = 0 then
(20)      Apply odd permutation
(21) else
(22)      Apply even permutation
(23) End If
(24) Generalize QI attributes ensuring the same values within each bucket
```

Algorithm 1: Proximity measurement for hierarchical categorical attributes algorithm (PM-HCA).

measured by computing the information loss. The NCP is considered a very popular measurement [50]. The NCP cost is principally used to measure the degree of generalization of values in the data set [14, 51]. Moreover, NCP can be calculated during the anonymization process; thus, it can be considered as a distance in clustering-based anonymization algorithms [52].

The NCP is defined in [53, 54] with respect to the taxonomy tree of the sensitive categorical attribute as shown in

$$\mathrm{NCP}(i) = \begin{cases} 0, & \mathrm{subtr}(\widetilde{i}) = 1, \\ \dfrac{\mathrm{subtr}(\widetilde{i})}{|x|}, & \text{otherwise}, \end{cases} \quad (1)$$

where «$\widetilde{i}$» indicates the generalized set of elements that is declared as a nonleaf level node in the hierarchy H and to which the set of elements «i» is mapped. And subtr: $\widetilde{i} \longrightarrow [1, |i|]$ is a function that counts the number of descendant values of the generalized set of elements «$\widetilde{i}$» based on the whole hierarchical generalization tree H [53, 54]. Based on the previous definition, the NCP for a data set D is defined in

$$\mathrm{NCP}(D) = \frac{\sum_{\forall i \in I} (\sup(i, D) \times \mathrm{NCP}(i))}{\sum_{\forall i \in I} (\sup(i, D))}. \quad (2)$$

The mapping $i \longrightarrow \mathrm{Sup}\ (i, D)$ represents the number of times the set of elements «i» is repeated in the data set D.

The NCP is an algorithm intended for measuring the amount of information loss, which is powerful and simple to use [14, 18]. In other words, NCP defines the level of generalization related to the anonymized data set [14, 55]. Note that more generalization leads to an important loss of information [41]. Then, NCP works according to how

elements in the data set are generalized. Moreover, NCP allocates significant penalties to elements with the highest generalization in the original data set [54]. The implementation of the algorithm is presented in the next section.

6. Implementation

In this section, we present the results of the implementation of our algorithm PM-HCA. Moreover, we evaluate the algorithm through NCP analysis and we discuss the obtained results. We have developed our algorithm with the Java tool applied on an extract of a real huge data set called "Careplans" related to the health sector. The "Careplans" data set belongs to the "SyntheticMass" database, which contains one million synthetic patient medical records.

6.1. Experimental Results. In our test Table 3, *K*-anonymity is already applied with respect to "Zip code" and "Age" QI attributes. The *K*-anonymity based on suppression is applied on the "Zip code" attribute and *K*-anonymity based on generalization is applied on the "Age" attribute. A distinct *L*-diversity technique is also already applied with respect to the sensitive attributes "Salary" and "Disease" where their corresponding values into each bucket are distinct.

The test Table 3 contains four buckets with different numbers of tuples. Table 3 is satisfying 2-diversity with respect to "Salary" and "Disease" attributes because every bucket includes at least two distinct values. The values corresponding to "Disease" attribute will be converted into codes by using the hierarchy in Figure 3. Table 4 shows the original test table where "Disease" column values are replaced by "Disease code" values sorted in an ascending order.

Since the "Disease code" column is sorted in Table 4, we make a test to know whether the values within each bucket are consecutive or not. Buckets 1, 3, and 4 contain consecutive values in "Disease code" attribute of Table 4 which means that these buckets must be swapped in order to avoid the similarity attack. The even permutation is applied when the number of buckets needing anonymization is even. In this case, the permutation process will be applied on every two consecutive buckets without repeating a bucket another time until processing all the buckets. For example, if we have 4 buckets needing anonymization, then the algorithm will apply two permutations, the first one between bucket 1 and bucket 2 and the second one between bucket 3 and bucket 4. However, the odd permutation is applied when the number of buckets needing anonymization is odd. Then, the permutation process will be applied in a way that just the first bucket and the last one will be used just once, but all intermediate buckets will be used twice. For example, if we have 5 buckets needing anonymization, then the algorithm will apply 4 permutations, the first one between bucket 1 and bucket 2, the second one between bucket 2 and bucket 3, the third one between bucket 3 and bucket 4, and the last permutation will be between bucket 4 and bucket 5. In our case, it is quite difficult because we have an odd number of buckets requiring anonymization. Thus, we opted to make a permutation between the maximum value of bucket 1 and the minimum value of bucket 3. Then, we permute between the maximum value of bucket 3, which is already anonymized with the minimum value in bucket 4. We notice that bucket 3 has been involved twice in the process of permutation. Table 5 shows the result after applying the permutation process.

Now, we can see that the resulting buckets in Table 5 contain distinct values corresponding at least to two different categories of diseases. However, Table 5 still does not resist similarity attack since an adversary may know, for example, that a person called Bob suffers from lung diseases if he had access to Table 6.

Based on Table 6, the adversary knows that Bob's Zip code and Age are 47685 and 52, respectively. Thus, by making a link between Tables 5 and 6, the adversary will find that an individual who is 52 years old certainly suffers from a disease with codes 121 or 122 and consequently Bob suffers from a disease belonging to lung diseases category. So, we have to more generalize the QI attributes related to buckets needing anonymization to prevent the adversary from knowing the exact category of diseases related to a person and hence resisting similarity attack. Table 7 presents the final anonymized table.

The generalization of QI attributes will be applied by creating a new interval in every bucket that was already needing anonymization (in our case, the buckets are 1, 3, and 4). Then, we take the minimum and the maximum value of all the intervals existing in the bucket and we put them in the new created interval. If we take, for example, bucket 4 of Table 5, we will transform the intervals [20; 29] and [50; 59] to a new interval. We will have the values 20 and 59 as the minimum and maximum values, respectively, of the new created interval. Besides the QI attributes' values within each bucket have to be the same to satisfy the *K*-anonymity constraint in the resulting anonymized table.

After ensuring that the *K*-anonymity constraint is satisfied in all the buckets of Table 7 with respect to QI attributes, the adversary would not be able to deduce the real disease of Bob even if he/she knows the values of Bob's Zip code and Age. Since the information that the adversary has about Bob exists in both buckets 3 and 4 by referring to Table 7, Bob's disease corresponds to "Respiratory infection," "lung diseases," and "Brain diseases" categories. Consequently, the proposed algorithm resists similarity attack. In the next part of this section, we will evaluate our proposed algorithm through NCP criterion.

6.2. Evaluation with NCP. Our resulting Table 5 gives good results in terms of anonymization since the *L*-diversity principle is still applied and at the same time Table 5 resists similarity attack. There exist several ways to measure the amount of information loss such as utility loss criterion [56] and NCP [55]. In this paper, we are going to focus on the NCP privacy measurement. In the following, we will present the application of NCP criterion in Table 3 and on the real huge data set "Careplans" with respect to Respiratory Gutbrain diseases category as shown in the hierarchy in Figure 3.

Table 3: The *L*-diversity original test table.

ID	Zip code	Age	Salary	Disease	Disease code	Bucket
1	476**	[20; 29]	3k	Concussion injury of brain	211	1
2	476**	[20; 29]	4k	Alzheimer	213	1
3	476**	[20; 29]	5k	Stroke	212	1
4	479**	[40; 49]	5k	Asthma	112	2
5	479**	[40; 49]	9k	Stroke	212	2
6	479**	[40; 49]	10k	Pulmonary emphysema	111	2
7	479**	[40; 49]	7k	Malignant tumor of colon	222	2
8	472**	[30; 39]	11k	Asthma	112	3
9	472**	[30; 39]	9k	Pulmonary emphysema	111	3
10	472**	[30; 39]	10k	Chronic obstructive bronchitis	113	3
11	476**	[50; 59]	14k	Non-small-cell carcinoma of lung TNM stage 4	121	4
12	476**	[50; 59]	13k	Primary small cell malignant neoplasm of lung TNM stage 4	122	4

Table 4: Table 3 with "Disease code" sorted in an ascending order.

ID	Zip code	Age	Salary	Disease code	Bucket
1	476**	[20; 29]	3k	211	1
2	476**	[20; 29]	5k	212	1
3	476**	[20; 29]	4k	213	1
4	479**	[40; 49]	10k	111	2
5	479**	[40; 49]	5k	112	2
6	479**	[40; 49]	9k	212	2
7	479**	[40; 49]	7k	222	2
8	472**	[30; 39]	9k	111	3
9	472**	[30; 39]	11k	112	3
10	472**	[30; 39]	10k	113	3
11	476**	[50; 59]	14k	121	4
12	476**	[50; 59]	13k	122	4

Table 5: The result table after applying permutation.

ID	Zip code	Age	Salary	Disease	Disease code	Bucket
1	476**	[20; 29]	3k	Concussion injury of brain	211	1
2	476**	[20; 29]	5k	Stroke	212	1
3	472**	[30; 39]	9k	Pulmonary emphysema	111	1
4	479**	[40; 49]	10k	Pulmonary emphysema	111	2
5	479**	[40; 49]	5k	Asthma	112	2
6	479**	[40; 49]	9k	Stroke	212	2
7	479**	[40; 49]	7k	Malignant tumor of colon	222	2
8	472**	[30; 39]	11k	Asthma	112	3
9	472**	[30; 39]	10k	Chronic obstructive bronchitis	113	3
10	476**	[50; 59]	14k	Non-small-cell carcinoma of lung TNM stage 4	121	3
11	476**	[20; 29]	4k	Alzheimer	213	4
12	476**	[50; 59]	13k	Primary small cell malignant neoplasm of lung TNM stage 4	122	4

Table 6: Information of the individual Bob.

Zip code	Age
47685	52

6.2.1. NCP on a Test Table. As mentioned in Figure 3, the diseases in the hierarchy refer to letters in order to facilitate the handling of diseases. Tables 8 and 9 represent the content of the original and the anonymized buckets, respectively.

Table 7: The final anonymized table.

ID	Zip code	Age	Salary	Disease	Disease code	Bucket
1	47***	[20; 39]	3k	Concussion injury of brain	211	1
2	47***	[20; 39]	5k	Stroke	212	1
3	47***	[20; 39]	9k	Pulmonary emphysema	111	1
4	479**	[40; 49]	10k	Pulmonary emphysema	111	2
5	479**	[40; 49]	5k	Asthma	112	2
6	479**	[40; 49]	9k	Stroke	212	2
7	479**	[40; 49]	7k	Malignant tumor of colon	222	2
8	47***	[30; 59]	11k	Asthma	112	3
9	47***	[30; 59]	10k	Chronic obstructive bronchitis	113	3
10	47***	[30; 59]	14k	Non-small-cell carcinoma of lung TNM stage 4	121	3
11	47***	[20; 59]	4k	Alzheimer	213	4
12	47***	[20; 59]	13k	Primary small cell malignant neoplasm of lung TNM stage 4	122	4

Table 8: The original buckets.

Disease	Bucket
{f, h, g}	1
{j, a, g, b}	2
{b, a, c}	3
{d, e}	4

Table 9: The anonymized buckets.

Disease	Bucket
{f, g, a}	1
{j, g, b, a}	2
{b, c, d}	3
{h, e}	4

Table 10: The generalized form of Table 8.

Disease	Bucket
i_1 = {f, h, g}	1
i_2 = {a, b, c, d, e, f, g, h, i, j}	2
i_3 = {b, a, c}	3
i_4 = {d, e}	4

Table 11: The generalized form of Table 9.

Disease	Bucket
i_1 = {a, b, c, d, e, f, g, h, i, j}	1
i_2 = {a, b, c, d, e, f, g, h, i, j}	2
i_3 = {a, b, c, d, e}	3
i_4 = {a, b, c, d, e, f, g, h, i, j}	4

Tables 10 and 11 are the generalized forms of Tables 8 and 9, respectively, based on the hierarchy in Figure 3.

The generalized form is made by replacing the existing values in each bucket by all the descendant values related to the minimum root, which encompasses the previous values before the generalization.

Table 10 represents the generalized form of Table 8; we recognize that only the values of bucket 2 are changed to their generalized form since the original values belong to three different categories of diseases that are "Respiratory infection," "lung diseases," and "Brain diseases." Thus, the bucket 2 of Table 10 will include all the descendant values belonging to the three categories of diseases mentioned before. Based on (1), we calculate the NCP for the four buckets in Table 10.

$$\begin{aligned} \mathrm{NCP}(i_1) &= \mathrm{NCP}(i_3) = \frac{3}{10}; \\ \mathrm{NCP}(i_2) &= \frac{10}{10}; \\ \mathrm{NCP}(i_4) &= \frac{2}{10}. \end{aligned} \tag{3}$$

NCP (i_1) = NCP (i_3) = 3/10 because the lowest common ancestors of i_1 and i_3 are "Brain diseases" and "Respiratory Gut-brain diseases," respectively, which have 3 leaves. The value of the denominator 10 refers to the number of leaves in the entire disease domain hierarchy. Moreover, NCP $(i_2) = 10/10 = 1$ because "Respiratory Gut-brain diseases" is the lowest common ancestor of i_2 (which has 10 leaves). Besides, NCP $(i_4) = 2/10$ since the lowest common ancestor of i_4 is "lung diseases" category which has 2 leaves. And based on equation (2), we calculate the NCP for the whole original generalized data set represented in Table 10. In this case, the NCP equals 0.37.

$$\mathrm{NCP}(D) = \frac{[(2 \times 3/10) + (1 \times 10/10) + (2 \times 3/10) + (2 \times 2/10)]}{(2 + 1 + 2 + 2)} = 0.37. \tag{4}$$

In our case, i_1, i_3, and i_4 are repeated twice; however, the set of elements i_2 is repeated only once. We repeat the same process to calculate NCP for Table 11 as done before for Table 10.

$$\begin{aligned}
&\mathrm{NCP}(i_1) = \mathrm{NCP}(i_2) = \mathrm{NCP}(i_4) = \frac{10}{10},\\
&\mathrm{NCP}(i_3) = \frac{5}{10},\\
&NCP(D) = \frac{[(3 \times 10/10) + (3 \times 10/10) + (4 \times 5/10) + (3 \times 10/10)]}{(3 + 3 + 4 + 3)} = 0.84.
\end{aligned} \tag{5}$$

The NCP is used to evaluate the information loss. Its value is between 0 and 1. In addition, the smaller the NCP is, the higher data utility is [18, 52]. If NCP equals 0, it means that there is no information loss; else if NCP equals 1, it means that there is a maximum information loss. In this case, we remark that the NCP of the anonymized and generalized test Table 11 equals 0.84 and the value of NCP belonging to the original generalized test Table 10 equals 0.37.

6.2.2. NCP on a Real Huge Data Set. It is interesting to show the performances of our algorithm on a real huge data set called "Careplans" related to the health sector. Therefore, we find 141 buckets including the combination of diseases (a, b, c) corresponding to "Respiratory infection" category and 105 buckets including the combination of diseases (g, h) corresponding to "Brain diseases" category as shown in Table 12. Thus, 246 buckets among 397 buckets must be anonymized in order to address the L-diversity limitation. According to the proposed algorithm, 210 (105×2) buckets will be permuted taking into consideration both "Respiratory infection" and "Brain diseases" categories. And the 36 remaining buckets will be permuted with other buckets even if they do not need to be anonymized. After that, 36 buckets including (a, b, c) will be permuted with 36 buckets including (g, h, j) among 37 buckets as shown in Table 13. Now, we will show the application of the NCP criterion on both the generalized form of original data set and the anonymized one. Tables 12 and 13 represent the original and the anonymized buckets, respectively. Tables 14 and 15 are the generalized forms of Tables 12 and 13, respectively, based on the hierarchy in Figure 3.

Based on (1), we calculate the NCP for the nine buckets in Table 14.

$$\begin{aligned}
&\mathrm{NCP}(i_1) = \mathrm{NCP}(i_9) = \frac{3}{10},\\
&\mathrm{NCP}(i_2) = \mathrm{NCP}(i_3) = \mathrm{NCP}(i_7) = \mathrm{NCP}(i_8) = \frac{5}{10},\\
&\mathrm{NCP}(i_4) = \mathrm{NCP}(i_5) = \mathrm{NCP}(i_6) = \frac{10}{10}.
\end{aligned} \tag{6}$$

And based on (2), we calculate the NCP for the whole original generalized data set represented in Table 14. In this case, the NCP equals 0.39.

$$\begin{aligned}
\mathrm{NCP}(D) &= \frac{[(6 \times (3/10) \times 141) + (5 \times (5/10) \times 18) + (5 \times (5/10) \times 28) + (3 \times (10/10) \times 13) + (3 \times (10/10) \times 5) + (3 \times (10/10) \times 24) + (5 \times (5/10) \times 26) + (5 \times (10/10) \times 37) + (6 \times (3/10) \times 105)]}{[(6 \times 141) + (5 \times 18) + (5 \times 28) + (3 \times 13) + (3 \times 5) + (3 \times 24) + (5 \times 26) + (5 \times 37) + (6 \times 105)]}\\
&= \frac{8413}{(10 \times 2147)}\\
&= 0.39.
\end{aligned} \tag{7}$$

We repeat the same process to calculate the NCP for Table 15 as done before for Table 14.

$$\begin{aligned}
&\mathrm{NCP}(i_1) = \mathrm{NCP}(i_4) = \mathrm{NCP}(i_5) = \mathrm{NCP}(i_6) = \mathrm{NCP}(i_8) = \mathrm{NCP}(i_{10}) = \frac{10}{10},\\
&\mathrm{NCP}(i_2) = \mathrm{NCP}(i_3) = \mathrm{NCP}(i_7) = \mathrm{NCP}(i_9) = \frac{5}{10},\\
&\mathrm{NCP}(D) = \frac{[(6 \times (10/10) \times 141) + (8 \times (5/10) \times 18) + (8 \times (5/10) \times 28) + (6 \times (10/10) \times 13) + (6 \times (10/10) \times 5) + (6 \times (10/10) \times 24) + (8 \times (5/10) \times 26) + (6 \times (10/10) \times 36) + (8 \times (5/10) \times 1) + (6 \times (10/10) \times 105)]}{[(6 \times 141) + (8 \times 18) + (8 \times 28) + (6 \times 13) + (6 \times 5) + (6 \times 24) + (8 \times 26) + (6 \times 36) + (8 \times 1) + (6 \times 105)]}\\
&= \frac{22360}{(10 \times 2528)}\\
&= 0.88.
\end{aligned} \tag{8}$$

Table 12: The original data set with buckets.

Number of occurrences of buckets	Buckets	Blocs
141	{a, b, c}	i_1
18	{b, c, d}	i_2
28	{b, d, e}	i_3
13	{b, d, f}	i_4
5	{d, f, g}	i_5
24	{d, g, h}	i_6
26	{g, h, i}	i_7
37	{g, h, j}	i_8
105	{g, h}	i_9

Table 13: The anonymized table of Table 12.

Number of occurrences of buckets	Buckets	Blocs
141	{g, b, c}	i_1
18	{b, c, d}	i_2
28	{b, d, e}	i_3
13	{b, d, f}	i_4
5	{d, f, g}	i_5
24	{d, g, h}	i_6
26	{g, h, i}	i_7
36	{a, h, j}	i_8
1	{g, h, j}	i_9
105	{a, h}	i_{10}

Table 14: The generalized form of Table 12.

Number of occurrences of buckets	Buckets	Blocs
141	{a, b, c}	i_1
18	{a, b, c, d, e}	i_2
28	{a, b, c, d, e}	i_3
13	{a, b, c, d, e, f, g, h, i, j}	i_4
5	{a, b, c, d, e, f, g, h, i, j}	i_5
24	{a, b, c, d, e, f, g, h, i, j}	i_6
26	{f, g, h, i, j}	i_7
37	{f, g, h, i, j}	i_8
105	{f, g, h}	i_9

Table 15: The generalized form of Table 13.

Number of occurrences of buckets	Buckets	Blocs
141	{a, b, c, d, e, f, g, h, i, j}	i_1
18	{a, b, c, d, e}	i_2
28	{a, b, c, d, e}	i_3
13	{a, b, c, d, e, f, g, h, i, j}	i_4
5	{a, b, c, d, e, f, g, h, i, j}	i_5
24	{a, b, c, d, e, f, g, h, i, j}	i_6
26	{f, g, h, i, j}	i_7
36	{a, b, c, d, e, f, g, h, i, j}	i_8
1	{f, g, h, i, j}	i_9
105	{a, b, c, d, e, f, g, h, i, j}	i_{10}

Figures 4 and 5 show the values of NCP criterion before and after the anonymization process, respectively.

Based on Figures 4 and 5, the amount of information loss had increased after applying the anonymization process; however, the amount of useful data had decreased.

The comparison between applying the NCP criterion on small data set and big one is presented in Table 16.

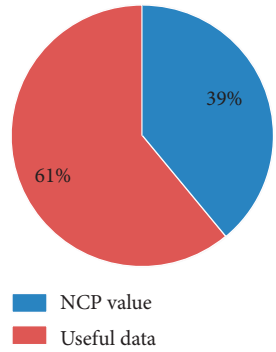

Figure 4: NCP before anonymization.

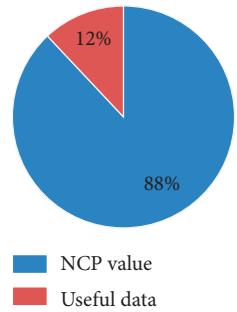

Figure 5: NCP after anonymization.

Table 16: Comparison between applying the NCP criterion on small data set and big one.

	NCP on a small data set	NCP on Careplans (1086 lines)
Before anonymization	0.37	0.39
After anonymization	0.84	0.88

The empirical analysis of our proposed algorithm is performed by doing the following steps. We start by a moderate input size of $N = 12$ (the size of our test table in the experimental part in the paper). Then, we measure and record the running time of the program and we repeat this processing every time we double the size N. Finally, we present the resulting plot of the empirical analysis of our proposed algorithm.

Table 17 contains the recorded values after applying the program on different data sets with different sizes.

Now, we are ready to present the plot of the previous results. So, we plot time versus the problem size and we get the following curve which represents the empirical analysis of our proposed algorithm (Figure 6).

We mention that the experiment is performed on Lenovo laptop with Intel Core i7 CPU at 2.10 GHZ, and 8 GB memory. In the next part of this section, we will discuss the results of the anonymization process. Then, we will show the resistance of the algorithm to the similarity attack. Finally, we will give some remarks about NCP.

6.3. Discussion. In Table 3, there are 3 buckets (1, 3, and 4) where each bucket contains values of Disease belonging to "Brain diseases," "Respiratory infection," and "lung diseases," respectively. Then, if an adversary accesses the table, he or she will know the family of diseases related to each bucket based on the hierarchy of diseases in Figure 3. Thus, our original Table 3 with L-diversity property cannot resist the similarity attack. After the application of our algorithm, the content of the buckets is no more corresponding to a specific category. For instance, bucket 1 belongs to two categories which are "Brain diseases" and "Respiratory infection." We have permuted between line 2 (Alzheimer) in bucket 1 and line 9 (Pulmonary emphysema) in bucket 3. Then, we applied the permutation process between the new line in bucket 3, which is already swapped containing Alzheimer disease, and line 11 (non-small-cell

Table 17: Time versus problem size.

Size N	Time (seconds)
12	0,149
24	0,321
48	0,602
96	1,288
192	2,136
384	4,249
768	9,742
1536	22,225

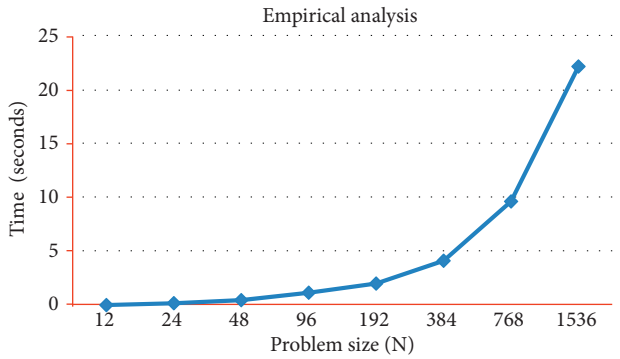

Figure 6: Empirical analysis.

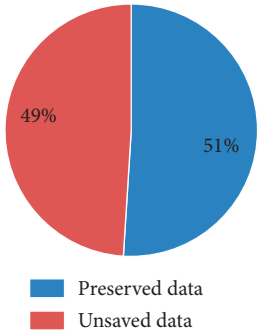

Figure 7: The preserved and the unsaved data after anonymization.

carcinoma of lung TNM stage 4) in bucket 4. We notice that the number of buckets needing permutation in Table 3 is odd. That is why bucket 3 has been involved in the permutation process twice. Now, even if an adversary has access to the final anonymized Table 7, he or she could not recognize the category of disease of a certain bucket because this bucket after the anonymization process contains at least two categories of diseases. In other terms, there is no correlation between tuples within the same bucket.

As shown in Figures 4 and 5, we remark that the value of NCP has increased after anonymization because the range of disease values within each bucket has increased. In other words, the value of NCP tends to value 1, which corresponds to the total amount of information loss.

We have calculated the NCP value before anonymization to know the amount of information loss caused by the application of K-anonymity and L-diversity techniques (Table 3). As mentioned in [53], the amount of NCP should equal the value 0 for the original data set, since we have not yet applied the anonymization process and consequently, we should not have information loss. Then, the exact amount of information loss for the anonymized data set should equal $(0.88 - 0.39 = 0.49)$. Figure 7 shows the exact amount of information loss.

We remark that the value of the unsaved data after anonymization (0.49) did not even reach 50% of the entire information in the data set. That means the result shows a good balance between privacy and data utility.

7. Conclusions

Ensuring privacy in big data is a genuine problem, which requires special attention. Researchers have suggested several anonymization algorithms to protect the individual's privacy. In this paper, we have focused on an anonymization technique that treats sensitive categorical attributes in order to ultimately treat both numerical and nonnumerical attributes. Our proposed algorithm addresses a particular limitation of L-diversity technique. In fact, even if L-diversity ensures that the data set will be divided horizontally into buckets where each bucket will contain only distinct values, these values may correspond to the same category. The proposed algorithm PM-HCA comes to present a solution to the similarity attack. Furthermore, the algorithm tests the degree of proximity between values within each bucket in the data set in order to identify the buckets that must be anonymized. The algorithm gives good results in terms of anonymization. Besides, we have measured the amount of information loss after the anonymization process through a well-known criterion called NCP. The obtained results are very interesting, which encourage us to define future trends of research. We plan to develop an algorithm dealing with sensitive categorical attributes while focusing this time on a threshold representing the distance between the nonnumerical values within each bucket. Thus, we would not be obliged to set the number of nodes to a maximum number of 9. Furthermore, we intend to validate our algorithm on various huge real data sets.

Abbreviations

NCP: Normalized certainty penalty
UL: Utility loss
EHRs: Electronic health records
PII: Personally identifying information
SA-MDAV: Semantic adaptive maximum distance average vector
NMBPA: Naive multisensitive bucketization permutation algorithm
CDMBPA: Closest distance multisensitive bucketization permutation algorithm
SDC: Statistical disclosure control
PSR: Personalized sensitivity rating
EMD: Earth mover's distance
AHP: Analytic hierarchy process.

Data Availability

The source code of our proposed algorithm during the current study is publicly available on GitHub: https://github.com/zakariae161/Proximity_algo.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Qu, S. Yu, L. Gao, and J. Niu, "Big data set privacy preserving through sensitive attribute-based grouping," in *Proceedings of the IEEE International Conference on Communications*, pp. 1–6, Paris, France, May 2017.

[2] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy," *Journal of Big Data*, vol. 5, no. 1, pp. 1–18, 2018.

[3] N. Victor, D. Lopez, and J. H. Abawajy, "Privacy models for big data: a survey," *International Journal of Big Data Intelligence*, vol. 3, no. 1, pp. 61–75, 2016.

[4] S. Kavitha, S. D. Raja Vadhana, and P. R. Vadhana, "An evaluation on big data generalization using k-anonymity algorithm on cloud," in *Proceedings of the IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–5, Coimbatore, India, January 2015.

[5] J. Zhang, B. Zhao, G. Song, L. Ni, and J. Yu, "Maximum delay anonymous clustering feature tree based privacy-preserving data publishing in social networks," *Procedia Computer Science*, vol. 147, pp. 643–646, 2019.

[6] T. Cui, J. Yang, N. Meng, and W. Xie, "A privacy protection algorithm based on hierarchical multiple sensitive attributes allowed by least mean square criterion," in *Proceedings of the Atlantis Press 2nd International Conference on Electronics, Network and Computer Engineering (ICENCE)*, pp. 599–605, Yinchuan, China, August 2016.

[7] I. Ozalp, M. E. Gursoy, M. E. Nergiz, and Y. Saygin, "Privacy-preserving publishing of hierarchical data," *ACM Transactions on Privacy and Security*, vol. 19, no. 3, pp. 1–29, 2016.

[8] X. Huang, "K-anonymity and L-diversity data anonymization in an in-memory database," Google Patents U.S. Patent Application No. 15/794,744, 2019, https://patentscope.wipo.int/search/en/detail.jsf?docId=US241664522&_fid=EP241674948.

[9] Z. El Ouazzani and H. El Bakkali, "Variable distinct L-diversity algorithm applied on highly sensitive correlated attributes," in *Proceedings of the 15th International Conference on Wireless and Mobile Communications (ICWMC)*, pp. 47–52, ThinkMind, Rome, Italy, July 2019.

[10] J. Wang, G. Cai, C. Liu, J. Wu, and X. Li, "A multi-level privacy-preserving approach to hierarchical data based on fuzzy set theory," *Symmetry*, vol. 10, no. 8, pp. 333–414, 2018.

[11] Z. El Ouazzani and H. El Bakkali, "New technique ensuring privacy in big data: variable t-closeness for sensitive numerical attributes," in *Proceedings of the IEEE 3rd International Conference on Cloud Computing and Technology Application (CloudTech'17)*, pp. 1–6, Rabat, Morocco, October 2017.

[12] X. Wang, J.-K. Chou, W. Chen et al., "A utility-aware visual approach for anonymizing multi-attribute tabular data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 351–360, 2018.

[13] D. Roy and S. Kumar Jena, "Determining t in t-closeness using multiple sensitive attributes," *International Journal of Computers and Applications*, vol. 70, no. 19, pp. 47–51, 2013.

[14] W. Zheng, Z. Wang, T. Lv, Y. Ma, and C. Jia, "K-anonymity algorithm based on improved clustering," in *Proceedings of the Springer International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, pp. 462–476, Melbourne, Australia, December 2018.

[15] D. Gunawan and M. Mambo, "Set-valued data anonymization maintaining data utility and data property," in *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, pp. 1–8, Langkawi Malaysia, January 2018.

[16] M. Orooji and G. M. Knapp, "Improving suppression to reduce disclosure risk and enhance data utility," in *Proceedings of the Institute of Industrial and Systems Engineers (IISE) Annual Conference*, pp. 1415–1420, Peachtree Corners, GA, USA, January 2019.

[17] D. Li, X. He, L. Cao, and H. Chen, "Permutation anonymization," *Journal of Intelligent Information Systems*, vol. 47, no. 3, pp. 427–445, 2016.

[18] Y. Ye, L. Wang, J. Han, S. Qiu, and F. Luo, "An anonymization method combining anatomy and permutation for protecting privacy in microdata with multiple sensitive attributes," in *Proceedings of the IEEE International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 404–411, Ningbo, China, July 2017.

[19] Z. El Ouazzani and H. El Bakkali, "A new technique ensuring privacy in big data: k-anonymity without prior value of the threshold k," *Procedia Computer Science*, vol. 127, pp. 52–59, 2018.

[20] F. Hassan, J. Domingo-Ferrer, and J. Soria-Comas, "Anonymization of unstructured data via named-entity recognition," in *Proceedings of the Springer International Conference on Modeling Decisions for Artificial Intelligence (MDAI)*, pp. 296–305, Mallorca, Spain, October 2018.

[21] L. El Haourani, A. A. Elkalam, and A. A. Ouahman, "Knowledge based access control a model for security and privacy in the big data," in *Proceedings of the ACM 3rd International Conference on Smart City Applications (SCA)*, pp. 1–8, Palma de Mallorce, Spain, October 2018.

[22] Y. Canbay, Y. Vural, and S. Sagiroglu, "Privacy preserving big data publishing," in *Proceedings of the IEEE International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp. 24–29, Ankara, Turkey, December 2018.

[23] S. Sangeetha and G. Sudha Sadasivam, "Privacy of big data: a review," in *Handbook of Big Data and IoT Security*, A. Dehghantanha and K. K. Choo, Eds., pp. 5–23, Springer, Berlin, Germany, 2019.

[24] C. Eyupoglu, M. Aydin, A. Zaim, and A. Sertbas, "An efficient big data anonymization algorithm based on chaos and perturbation techniques," *Entropy*, vol. 20, no. 5, pp. 373–418, 2018.

[25] Y. Sei, T. Takenouchi, and A. Ohsuga, "(l1,. . .lq)-diversity for anonymizing sensitive quasi-identifiers," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, pp. 596–603, Washington, DC, USA, August 2015.

[26] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[27] S. Yu, "Big privacy: challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.

[28] J. C.-W. Lin, Q. Liu, P. Fournier-Viger, Y. Djenouri, and J. Zhang, "Anonymization of multiple and personalized sensitive attributes," in *Proceedings of the Springer 20th International Conference on Big Data Analytics and Knowledge Discovery*, pp. 204–215, Regensburg, Germany, September 2018.

[29] P. S. Rao and S. Satyanarayana, "Privacy preserving data publishing based on sensitivity in context of big data using hive," *Journal of Big Data*, vol. 5, no. 1, p. 20, 2018.

[30] A. Anushree and G. L. D. Rio, "Big data anonymization in cloud using k-anonymity algorithm using map reduce framework," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 5, pp. 50–56, 2019.

[31] W. Iftikhar, Z. Mahmood, and M. Jayabalan, "A review of big data in the healthcare sector: evaluation and analysis of cervical cancer data," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 11, pp. 40–51, 2018.

[32] P. Ram Mohan Rao, S. Murali Krishna, and A. P. Siva Kumar, "Privacy preservation techniques in big data analytics: a survey," *Journal of Big Data*, vol. 5, no. 1, 2018.

[33] U. P. Rao, B. B. Mehta, and N. Kumar, "Scalable l-diversity: an extension to scalable k-anonymity for privacy preserving big data publishing," *International Journal of Information Technology and Web Engineering*, vol. 14, no. 2, pp. 27–40, 2019.

[34] A. Gebrehiwot and A. V. Pawar, "Research issue in data anonymization in electronic health service: a survey," in *Data Science and Big Data Analytics. Lecture Notes on Data Engineering and Communications Technologies, ACM-WIR*, D. Mishra, X. S. Yang, and A. Unal, Eds., vol. 16, pp. 139–148, Springer, Berlin, Germany, 2019.

[35] J. A. Shamsi and M. A. Khojaye, "Understanding privacy violations in big data systems," *IT Professional*, vol. 20, no. 3, pp. 73–81, 2018.

[36] K. Rajendran, M. Jayabalan, and M. Rana, "A study on k-anonymity, l-diversity, and t-closeness techniques focusing medical data," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 12, pp. 172–177, 2017.

[37] E. Poovammal and M. Ponnavaikko, "Preserving micro data release: categorical and numerical data," in *Proceedings of the 5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT)*, pp. 1–7, Hammamet, Tunisia, March 2009.

[38] K. Wang, W. Zhao, J. Cui, Y. Cui, and J. Hu, "A K-anonymous clustering algorithm based on the analytic hierarchy process," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 76–83, 2019.

[39] A. A. Mubark, E. Elabd, and H. Abdulkader, "Semantic anonymization in publishing categorical sensitive attributes," in *Proceedings of the IEEE 8th International Conference on Knowledge and Smart Technology (KST)*, pp. 89–95, Chiang Mai, Thailand, February 2016.

[40] J. Jia and L. Chen, "(l, m,d)-anonymity: A resisting similarity attack model for multiple sensitive attributes," in *Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 756–760, Chengdu, China, December 2017.

[41] R. Saeed and A. Rauf, "Anatomization through generalization (AG): a hybrid privacy-preserving approach to prevent membership, identity and semantic similarity disclosure attacks," in *Proceedings of the IEEE International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–7, Sukkur, Pakistan, March 2018.

[42] A. V. D. M. Kayem and C. Meinel, "Clustering heuristics for efficient T-closeness anonymisation," in *Proceedings of the Springer Database and Expert Systems Applications*, pp. 27–34, Regensburg, Germany, September 2017.

[43] S. Saraswathi and K. Thirukumar, "Enhancing utility and privacy using t-closeness for multiple sensitive attributes," *Advances in Natural and Applied Sciences*, vol. 10, no. 5, pp. 6–13, 2016.

[44] R. Wang, Y. Zhu, T.-S. Chen, and C.-C. Chang, "Privacy-preserving algorithms for multiple sensitive attributes satisfying t-closeness," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1231–1242, 2018.

[45] G. Hao and X. Ya-Bin, "Research on privacy preserving method based on T-closeness model," in *Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1455–1459, Chengdu, China, December 2017.

[46] A. V. D. M. Kayem, C. T. Vester, and C. Meinel, "Syntactic anonymisation of shared datasets in resource constrained environments," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXVIII. Lecture Notes in Computer Science*, A. Hameurlain, R. Wagner, S. Hartmann, and H. Ma, Eds., vol. 1125, pp. 27–60, Springer, Berlin, Germany, 2018.

[47] H. Wang, J. Han, J. Wang, and L. Wang, "(l, e)-diversity—a privacy preserving model to resist semantic similarity attack," *Journal of Computers*, vol. 9, no. 1, pp. 59–64, 2014.

[48] K. Oishi, Y. Tahara, Y. Sei, and A. Ohsuga, "Proposal of l-diversity algorithm considering distance between sensitive attribute values," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, vol. 94, pp. 1–8, Honolulu, HI, USA, November 2017.

[49] J. Walonoski, M. Kramer, J. Nichols et al., "Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 230–238, 2018.

[50] K. Murakami and T. Uno, "Optimization algorithm for k-anonymization of datasets with low information loss," *International Journal of Information Security*, vol. 17, no. 6, pp. 631–644, 2018.

[51] R. Li, S. An, D. Li et al., "K-anonymity model for privacy-preserving soccer fitness data publishing," in *Proceedings of the 2nd International Conference on Material Engineering and Advanced Manufacturing Technology (MEAMT)*, vol. 189, no. 4, pp. 1–6, Beijing, China, August 2018.

[52] Q. Gong, M. Yang, Z. Chen, W. Wu, and J. Luo, "A framework for utility enhanced incomplete microdata anonymization," *Cluster Computing*, vol. 20, no. 2, pp. 1749–1764, 2017.

[53] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "A framework for efficient data anonymization under privacy and accuracy constraints," *ACM Transactions on Database Systems*, vol. 34, no. 2, pp. 1–47, 2009.

[54] G. Loukides and A. Gkoulalas-Divanis, "Utility-preserving transaction data anonymization with low information loss," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9764–9777, 2012.

[55] Q. Gong, M. Yang, Z. Chen, and J. Luo, "Utility enhanced anonymization for incomplete microdata," in *Proceedings of the IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 74–79, Nanchang, China, May 2016.

[56] C. Hebert, D. Bernau, and A. Lahouel, "Anonymization techniques to protect data," United States Patent Application Publication, US 2018/0004978 A1, 2018, https://patents.google.com/patent/US20180004978A1/en.