Clustering Dissimilar Tuples: A Stronger Notion of Privacy

Srijayanthi Subramanian, Mohammed Fayaz A, Sandra Johnson, Sethukarasi Thirumaaran
1Department of Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, India, ssj.cse@rmkec.ac.in
2Department of Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, India, moha16224.cs@rmkec.ac.in
3Department of Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, India, sjn.cse@rmkec.ac.in
4Department of Computer Science and Engineering, R.M.K. Engineering College, Kavaraipettai, India, hod.cse@rmkec.ac.in

ABSTRACT

Identity disclosure and attribute disclosure have always been a major concern while publishing data. k-anonymity tries to solve identity disclosure but doesn’t prevent attribute disclosure which leads to homogeneity and background knowledge attack. Preserving privacy of an individual is becoming more challenging due to increasing number of homogeneity and background knowledge attacks. l-diversity model has been proposed to thwart these attacks but it doesn’t fulfill its obligations. Several authors found l-diversity model to be inadequate, hence they put forth another model called t-closeness. Over the years, many investigations and experimentations conducted by various researchers shows that t-closeness does not provide a clear relationship between the threshold value t and information gain and it also shows that Earth mover’s distance, a distance metric used by t-closeness model, becomes complex with multiple sensitive attributes. In view of this challenge, we propose a stronger notion of privacy called Clustering Dissimilar Tuples (CDT) to thwart homogeneity and background knowledge attack by formalizing the idea of processing the original dataset initially wherever these attacks possibly occur. The attacks are found to occur in the tuples of sensitive attributes. Hence CDT processes the tuples of sensitive attributes to form equivalence classes consisting of dissimilar tuples. Through experimental evaluations, we show that CDT is practical and can be implemented efficiently with minimum utility loss and maximum privacy gain.

Key words: anonymization, k-anonymity, l-diversity, t-closeness.

1. INTRODUCTION

Potential privacy breaches are enabled by the publication of large volumes of government and business data containing quasi-identifiers. Quasi-identifiers (personally identifiable information), when considered individually are not unique identifiers, but when combined becomes a digital weapon for information disclosure. This process is called Re-identification. De-identification is a process of preventing someone’s identity from being revealed. It is vitally important to ensure that data is de-identified to prevent information disclosure. There are two types of information disclosure as identified by D. Lambert [1]: identity disclosure and attribute disclosure. In an identity disclosure, a respondent is linked to an observation or a tuple in a published dataset. In an attribute disclosure, an intruder can get knowledge of a respondent with or without identification. To overcome these information disclosures, anonymization is used. In literature [2], [3] k-anonymity has been introduced.

k-anonymity states that each equivalence class should have at least k records. Though k-anonymity reduced identity disclosure, it failed to protect anonymized dataset from attribute disclosure. According to Machanavajjhala et al. [4], attribute disclosure leads to two major attacks: homogeneity attack and background knowledge attack.

For example, consider Table 1 containing original medical records and Table 2 containing anonymized medical records satisfying 3-anonymity. Attributes age, sex and place are quasi-identifiers and attribute disease is sensitive in nature. Suppose A knows B’s age is 12 and lives in Chennai and A also knows that B’s record is in the table. Then A can easily determine from Table 2 that A corresponds to the first equivalence class resulting in the identification of A’s disease.
Table 2: Anonymized medical records satisfying 3-anonymity

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Age</th>
<th>Sex</th>
<th>Place</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12-18</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>HIV</td>
</tr>
<tr>
<td>2</td>
<td>12-18</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>HIV</td>
</tr>
<tr>
<td>3</td>
<td>12-18</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>HIV</td>
</tr>
<tr>
<td>4</td>
<td>23-27</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>Lung cancer</td>
</tr>
<tr>
<td>5</td>
<td>23-27</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>Lung cancer</td>
</tr>
<tr>
<td>6</td>
<td>23-27</td>
<td>Male, Female</td>
<td>Chennai, Salem, Coimbatore</td>
<td>Heart disease</td>
</tr>
<tr>
<td>7</td>
<td>42-44</td>
<td>Male, Female</td>
<td>Madurai</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>42-44</td>
<td>Male, Female</td>
<td>Madurai</td>
<td>Heart disease</td>
</tr>
<tr>
<td>9</td>
<td>42-44</td>
<td>Male, Female</td>
<td>Madurai</td>
<td>Flu</td>
</tr>
</tbody>
</table>

According to Ninghui Li et al. [5], l-diversity is prone to various limitations: It may be strenuous and dispensable to achieve, it is inadequate to prevent attribute disclosure as it still endorses skewness and similarity attacks, and it doesn’t take into account the semantical closeness of the values of the sensitive attribute. To address these issues Ninghui Li et al. proposed t-closeness. Let c be an equivalence class in a table T and D be the distance between the distribution of a sensitive attribute in c and the distribution of the attribute in T. If an equivalence class c satisfies the following condition, then it is said to have t-closeness.

\[
D \leq t
\]

where t represents a threshold value. If all equivalence classes have t-closeness, then table T is said to have t-closeness. t-closeness uses Earth Mover’s Distance (EMD) [6], for calculating the distance between the distributions. Though it produces desired results for a single sensitive attribute, the mathematical relation struggles to find the distance between distributions for multiple sensitive attributes. Ninghui Li et al. clearly states aforementioned drawback as one of the limitations of t-closeness principle. Ninghui Li et al. also mentions that EMD does not provide any clarity in the relationship between the value t and information gain.

For example, let there be 4 distributions \(D_1\) (0.99, 0.01), \(D_2\) (0.89, 0.11), \(D_3\) (0.6, 0.4), and \(D_4\) (0.5, 0.5). The EMD for changing both \(D_1\) to \(D_2\) and \(D_1\) to \(D_3\) is 0.1 respectively. Though one can logically argue that the distance between the distributions \(D_1\) and \(D_2\) should be greater than that of \(D_3\) and \(D_4\), the results produced using EMD does not agree with this logical conclusion. This shows that it provides no guarantee to prevent homogeneity and background knowledge attack from taking place.

We propose a stronger notion of privacy called Clustering Dissimilar Tuples (CDT) that formalizes the idea of processing the original dataset initially where there are possible occurrences of a potential threat (homogeneity and background knowledge attack). Threats found to be occurring in the tuples of sensitive attributes. CDT processes the tuples of sensitive attributes to form equivalence classes consisting of dissimilar tuples. This effectively limits any chances for these attacks from taking place. Then for each equivalence class, the corresponding tuples of quasi-identifiers are processed and generalized before publishing the processed dataset.

2. LITERATURE REVIEW

In this digital age, it has become indispensable for Government, public and private institutions to have their data electronically available on the internet. According to D. Lambert [1], the availability of data publicly leads to two types of information disclosure: identity disclosure and attribute disclosure. To address identity disclosure k-anonymity has been introduced [2], [3]. It states that each equivalence class should have at least k records. In this way even if an intruder finds an equivalence class corresponding to
a respondent, the identity of the respondent would not be revealed as there would be $k$ records, i.e., each respondent has $\frac{1}{k}$ probability of getting disclosed.

Many investigations and experimentations introduced new $k$-anonymity models. Kai-Cheng Liu et al. [7] introduced optimized data de-identification using multidimensional $k$-anonymity and proved that it provides more reliable anonymous data and reduce the information loss rate. Widodo et al. [8] proposed an approach for distributing sensitive values in $k$-anonymity which outperformed systematic clustering when a high-sensitive value is distributed. Ping Zhao et al. [9] proposed a non-asymptotic bound on the performance of $k$-anonymity against information disclosure, taking into consideration intruder's background knowledge. Fan Fei et al. [10] applies $k$-anonymity to prevent Location-based Service (LBS) providers from stealing user location details. It uses a two-tier schema for the preservation of privacy based on $k$-anonymity. Jinbang Wang et al. [11] proposed a novel privacy notion called Client-based Personalized $k$-anonymity (CPKA). CPKA ensures that the query content of a user is protected from service providers in an autonomous vehicle. Yuanxiunan Gao et al. [12] proposed a novel algorithm, Principal Component Analysis-Grey Relational Analysis (PCA-GRA) $K$ anonymous algorithm, which significantly improved data utility in three aspects – information loss, feature maintenance, and classification evaluation performance.

Machanavajjhala et al. [4] agreed to the benefits of $k$-anonymity but sorted out that though it decreased the chances of identity disclosure, it paved a way to attribute disclosure. According to Machanavajjhala et al., attribute disclosure leads to two major attacks: homogeneity attack and background knowledge attack. To address these attacks Machanavajjhala et al. proposed $l$-diversity. It defines that there should be at least $l$ “well-represented” values for the sensitive attribute in an equivalence class. Several algorithms have been proposed for improving $l$-diversity by various researchers. Odsrue Temuujin et al. [13] designed an efficient $l$-diversity algorithm that uses anatomy and suppression for preserving privacy of dynamically changing published datasets. Keiichiro Oishi et al. [14] proposed $(l, d)$-semantic diversity which considers the similarity of sensitive attribute values with the help of addition of distances defined using categorization. Mohammed Atik Enam et al. [15] designed an $l$-diversity algorithm to improve the clustering quality of a point-set. Adeel Shah et al. [16] designed a novel security framework for Healthcare industry to provide strong patient anonymity level, anonymized data searching and successful correlation of PHR for medical research. Lin Yao et al. [17] introduced a scheme called Data Privacy Preservation with Perturbation (DPPP) to protect sensitive information on individual’s location trajectory. It also ensures DPPP satisfy $(l, \alpha, \beta)$-privacy. Hui Zhu et al. [18] developed $\tau$-Safe $(l, k)$-diversity privacy model to preserve privacy of individuals in sequential publication. This model is developed based on generalization and segmentation by individual anonymity satisfying $k$-anonymity and record anonymity satisfying $l$-diversity.

$l$-Diversity turns out to be strenuous and dispensable to achieve as it does not take into account the semantic closeness of the values of the sensitive attribute. To address these issues Ninghui Li et al. [5] proposed $t$-closeness which defines that the distance between the distribution of a sensitive attribute in an equivalence class and the distribution of the attribute in the whole table should be less than a threshold value $t$. $t$-closeness uses EMD [6] for calculating the distance between distributions. Many researchers came up with enhancing algorithms for $t$-closeness. Zakariae and Hanan [19] proposed variable $t$-closeness for sensitive numerical attributes, unlike fixed $t$ value this algorithm uses variable $t$ value. Guo Hao and Xu Ya-Bin [20] improved $t$-closeness model using parameter selection and adjustment method of the anonymous method. Yuchi Sei et al. [21] introduced two novel privacy models, namely, $(l_1, \ldots, l_q)$-diversity and $(t_1, \ldots, t_q)$-closeness by considering all the attributes to have both sensitive and quasi characteristics. Zhen Tu et al. [22] proposed a novel algorithm for protecting the trajectory of an individual against semantic and re-identification attack while retaining high data utility.

3. METHODOLOGY

3.1 Preliminary Definitions and its Algorithmic Implementations

3.1.1 Entropy of an Attribute

Let $X$ be an attribute and $i$ be an element present in $X$. Entropy is defined as a measurement of uncertainty or disorder and it is mathematically formulated as

$$\text{entropy}(X) = -\sum p(i) \log_{10}(p(i)) ; 0 \leq \text{entropy}(X)$$

where $p(i)$ represents the probability of occurrence of element $i$ in attribute $X$.

<table>
<thead>
<tr>
<th>Tuple no.</th>
<th>Age</th>
<th>Sex</th>
<th>Place</th>
<th>Race</th>
<th>Disease</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>m</td>
<td>Chennai</td>
<td>OC</td>
<td>HIV</td>
<td>100200</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>f</td>
<td>Salem</td>
<td>BC</td>
<td>cancer</td>
<td>13000</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>m</td>
<td>Coimbatore</td>
<td>OC</td>
<td>fever</td>
<td>56000</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>m</td>
<td>Salem</td>
<td>BC</td>
<td>cold</td>
<td>44500</td>
</tr>
<tr>
<td>5</td>
<td>57</td>
<td>m</td>
<td>Chennai</td>
<td>MBC</td>
<td>HIV</td>
<td>76000</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>f</td>
<td>Coimbatore</td>
<td>OBC</td>
<td>fever</td>
<td>10000</td>
</tr>
<tr>
<td>7</td>
<td>64</td>
<td>f</td>
<td>Madurai</td>
<td>SC</td>
<td>pneumonia</td>
<td>23000</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
<td>m</td>
<td>Madurai</td>
<td>ST</td>
<td>cancer</td>
<td>43000</td>
</tr>
<tr>
<td>9</td>
<td>64</td>
<td>f</td>
<td>Madurai</td>
<td>SC</td>
<td>cold</td>
<td>100200</td>
</tr>
<tr>
<td>10</td>
<td>34</td>
<td>f</td>
<td>Chennai</td>
<td>MBC</td>
<td>pneumonia</td>
<td>13000</td>
</tr>
</tbody>
</table>

Table 3: Medical dataset
Algorithm for entropy calculation
Input: Dataset D
Output: List of entropies, e, containing entropy of every attribute
(1) e ← empty vector
(2) for each attribute A in D do
   (a) s ← 0
   (b) for each unique element i in A do
      (i) n ← number of element i in A
         (ii) p ← \( \frac{n}{\text{number of elements in A}} \)
      (iii) s ← s + \( p \times \log_{10} \left( \frac{1}{p} \right) \)
   end for
   (c) s ← \( \frac{s}{\log_{10} 2} \)
   (d) e ← append(s)
end for

Algorithm for weight calculation
Input: e, representing list of entropies containing entropy of every attribute
Output: List of weights, w, containing weight of every attribute
no. of columns
(1) s ← \( \sum_{i=1}^{\text{no of columns}} e(i) \)
(2) w ← empty vector
(3) for each entropy i in e do
   (a) f ← \( 1 - \frac{i}{s} \)
   (b) w ← append(f)
end for

3.1.3 Gower’s Distance
Let \( t_i \) be \( i^{th} \) tuple, \( t_j \) be \( j^{th} \) tuple, and \( H_{ij} \) be the Gower’s distance between \( t_i \) and \( t_j \) in a dataset D. Gower’s distance defines how dissimilar \( t_i \) is to \( t_j \) or \( t_j \) is to \( t_i \). It is formulated as
\[
H_{ij} = \frac{\sum_{k=1}^{N} w_{ij} \times H_{ijk}}{\sum_{k=1}^{N} w_k}
\]
where \( w = \{w_1, w_2, w_3, \ldots, w_N\} \) represents a list of weights of N attributes, \( k \) value represents \( k^{th} \) attribute and \( H_{ijk} \) represents the distance between \( t_i \) and \( t_j \) in \( k^{th} \) attribute. Since \( H_{ijk} \) varies for categorical and numerical attributes, it is defined separately for each type of attribute. Let \( y_{ab} \) represents the element in \( a^{th} \) tuple and \( b^{th} \) attribute.

For a categorical attribute,
\[
H_{ijk} = 0 \text{ if } y_{ik} = y_{jk} \text{ and } H_{ijk} = 1 \text{ if } y_{ik} \neq y_{jk}
\]

For a numerical attribute,
\[
H_{ijk} = \frac{|y_{ik} - y_{jk}|}{z_k}
\]
\[
z_k = \text{maximum}(y_{ik}) - \text{minimum}(y_{ik})
\]

Out of all the distances in the world we chose Gower’s distance because it incorporates both categorical and numerical attributes while calculating the distance between the tuples by taking into consideration the weights of the attributes.

3.1.4 Gower’s Dissimilarity Matrix
Let \( t_i \) be \( i^{th} \) tuple and \( t_j \) be \( j^{th} \) tuple. It is a matrix \( g_{ij}((1, 2, 3, \ldots, i, \ldots, n-1) \ast (2, 3, 4, \ldots, j, \ldots, n)) \) consisting of Gower’s distance \( H_{ij} \) between \( t_i \) and \( t_j \) where \( i = 1, 2, 3, \ldots, n-1 \) and \( j = 2, 3, 4, \ldots, n \).
Table 4: Gower’s dissimilarity matrix for Table 3

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.49</td>
<td>0.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.63</td>
<td>0.47</td>
<td>0.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.34</td>
<td>0.83</td>
<td>0.6</td>
<td>0.66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.88</td>
<td>0.57</td>
<td>0.47</td>
<td>0.75</td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.98</td>
<td>0.58</td>
<td>0.83</td>
<td>0.85</td>
<td>0.8</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.69</td>
<td>0.58</td>
<td>0.54</td>
<td>0.56</td>
<td>0.6</td>
<td>0.8</td>
<td>0.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.84</td>
<td>0.71</td>
<td>0.85</td>
<td>0.74</td>
<td>0.75</td>
<td>0.78</td>
<td>0.3</td>
<td>0.68</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.73</td>
<td>0.53</td>
<td>0.77</td>
<td>0.78</td>
<td>0.53</td>
<td>0.54</td>
<td>0.44</td>
<td>0.76</td>
<td>0.74</td>
</tr>
</tbody>
</table>

For example, Table 4 represents Gower’s dissimilarity matrix for Table 3. The green shaded value 0.47 represents Gower’s distance between 2nd and 4th tuple. Hence Table 4 contains Gower’s distance between every tuple to every other tuple.

Algorithm for Gower’s dissimilarity matrix

Input: Dataset D, list of weights w containing weight of every attribute  
Output: Gower’s dissimilarity/distance matrix, g, containing distances between all the rows in D

1. \( v \leftarrow \text{empty vector} \)
2. for each combination of rows taken two at a time do
   a. let the two rows be \( i^{th} \) and \( j^{th} \) row
   b. \( f \leftarrow 0 \)
   c. for each attribute \( x \) in \( D \) do
      i. if \( x = \text{numerical} \) then
         1. \( s \leftarrow |x(i) - x(j)| \)
         2. \( r \leftarrow \text{maximum} - \text{minimum}(x) \)
         3. \( s \leftarrow \frac{s}{r} \)
      ii. else if \( x = \text{categorical} \) then
         1. if \( x(i) = x(j) \) then
            a. \( s \leftarrow 0 \)
         2. else
            a. \( s \leftarrow 1 \)
            end if
            end if
         iii. \( a \leftarrow w(x) \times s \)
         iv. \( f \leftarrow f + a \)
      end for
   d. \( a \leftarrow \text{sum of all weights in } w \)
   e. \( f \leftarrow \frac{f}{a} \)
   f. \( v \leftarrow \text{append}(f) \)
3. \( g \leftarrow \text{matrix data representation of } v \)

3.1.5 Gower’s Similarity Matrix

Let \( t_i \) be \( i^{th} \) tuple and \( t_j \) be \( j^{th} \) tuple. It is a matrix \( g_i((1, 2, 3, \ldots, i, \ldots, n-1) \times (2, 3, 4, \ldots, j, \ldots, n)) \) consisting of Gower’s closeness \( H'_{ij} \) between \( t_i \) and \( t_j \) where \( i = 1, 2, 3, \ldots, n-1 \) and \( j = 2, 3, 4, \ldots, n \).

Table 5: Gower’s similarity matrix for Table 3

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.76</td>
<td>0.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>0.78</td>
<td>0.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.88</td>
<td>0.31</td>
<td>0.64</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.68</td>
<td>0.78</td>
<td>0.44</td>
<td>0.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.04</td>
<td>0.67</td>
<td>0.31</td>
<td>0.28</td>
<td>0.36</td>
<td>0.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.52</td>
<td>0.66</td>
<td>0.71</td>
<td>0.69</td>
<td>\textbf{0.64}</td>
<td>0.36</td>
<td>0.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.29</td>
<td>0.49</td>
<td>0.28</td>
<td>0.45</td>
<td>0.44</td>
<td>0.39</td>
<td>0.91</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.46</td>
<td>0.71</td>
<td>0.41</td>
<td>0.4</td>
<td>0.72</td>
<td>0.71</td>
<td>0.8</td>
<td>0.42</td>
<td>0.45</td>
</tr>
</tbody>
</table>

For example, Table 5 represents Gower’s similarity matrix for Table 3. The green shaded value 0.64 represents the Gower’s closeness between 5th and 8th tuple. Hence Table 5 contains Gower’s closeness between every tuple to every other tuple.

Algorithm for Gower’s similarity matrix

Input: Gower’s dissimilarity matrix, \( g \)
Output: Gower’s similarity matrix, \( g_s \)

1. \( i \leftarrow 1 \)
2. \( g_s \leftarrow \text{empty matrix} \)
3. while \( (i \leq \text{number of rows}(g_d)) \) do
   a. \( j \leftarrow 1 \)
   b. while \( (j \leq \text{number of columns}(g_d)) \) do
      i. \( g_s(i, j) \leftarrow 1 - g_s(i, j)^2 \)
      end while
   end while
   c. \( i \leftarrow i + 1 \)
end while

3.1.6 k-Medoids

It is a partitioning technique which clusters \( n \) tuples into \( k \) clusters using k-medoids algorithm. A medoid \( m \) is a tuple in an equivalence class \( e \) with minimum dissimilarity among the dissimilarities with all other tuples in \( e \).

Its time complexity is \( O(k \times (n - k)^2) \). If the dataset has large number of records and small \( k \) value, it results in increasing the time complexity of this algorithm.

Gower’s distance can be incorporated into two clustering algorithms and they are \( k \)-medoids algorithm and hierarchical clustering algorithm. The hierarchical clustering takes a huge amount of time in clustering large datasets. Hence, we chose \( k \)-medoids for clustering as we are dealing with huge data.
Algorithm for k-medoids
Input: Dataset D, distance matrix g, k representing number of clusters to be formed
Output: Clusters formed using k-medoids
(1) Select k observations from dataset D as medoids.
(2) Associate each observation to the closest medoid using g.
(3) obj_function ← sum (all the distances of observations to their respective medoids)
(4) while obj_function decreases do
   (a) for each medoid a and a non-medoid b do
      (i) Swap a and b.
      (ii) Associate each observation to the closest medoid.
      (iii) obj_function ← sum (all the distances of observations to their respective medoids)
      (iv) if newly computed obj_function is more than that in the previous step then
         (1) undo the swap
         end if
   end for
end while

3.1.7 Silhouette Width

For every tuple f, the Silhouette width SWf is defined as the ratio of the difference between cohesion c_f and separation s_f (to the nearest neighbouring cluster) to the maximum of c_f and s_f. Cohesion c_f is defined as the average distance between tuple f and all other tuples of the equivalence class to which f belongs. Let e be an equivalence class which does not contain f. For every e, separation s_e is calculated as the average distance between f and all other tuples of e. Only a minimum of all the separations s_e is considered. Hence, SW_f is formulated as

\[
SW_f = \frac{s_f - c_f}{\max(s_f, c_f)}; -1 \leq SW_f \leq 1
\] (8)

There are many cluster-quality measures such as the Silhouette width, the Davies - Bouldin index, the Calinski - Harabasz index, the Dunn index and many more. Out of all the measures we found Silhouette width to be providing more optimum k value for clusters than any other measures when used in our algorithm. Hence, we chose the Silhouette width for measuring optimum k value for the given dataset D.

3.1.8 Utility Loss

Utility loss of a tuple in an anonymized dataset is defined as the root of the sum of the squared mean of utility loss of all the quasi attributes. Let UL(t_i) represent utility loss of i_th tuple, n represent the total number of records, and m represent the total number of quasi attributes. UL(t_i) is formulated as

\[
UL(t_i) = \sqrt{\frac{\sum_{j=1}^{m} UL(A_j)^2}{m}}; 0 \leq UL(t_i) \leq 1
\] (9)

Since quasi attributes can be represented as either categorical or numerical, UL(A_j) is defined separately for categorical and numerical attributes.

For categorical attribute,

\[
UL(A_j) = \frac{\text{no. of elements in the cell}_{ij} - 1}{\text{no. of unique elements in } A_j}
\] (10)

\[0 \leq UL(A_j) \leq 1\]

For numerical attribute,

\[
UL(A_j) = \frac{r_{\text{max}} - r_{\text{min}}}{\max(A_j) - \min(A_j)}; 0 \leq UL(A_j) \leq 1
\] (11)

where r_{max} and r_{min} represent the maximum and minimum
values of the numerical range for an anonymized numerical data. \( \max(A_i) \) and \( \min(A_i) \) represent maximum and minimum values of the attribute \( A_i \). Utility loss for an entire anonymized dataset, \( UL(D') \), is defined as the weighted average of \( UL(t_i) \) where \( i = 1, 2, 3, \ldots, n \). It is formulated as

\[
UL(D') = \frac{\sum_{i=1}^{k} N_i \ast UL_i}{n}; 0 \leq UL(D') \leq 1
\]  

(12)

where \( k \) represents the number of equivalence classes, \( N_i \) represents the number of records in \( i^{th} \) equivalence class and \( UL_i \) represents utility loss of any one of the tuples in \( i^{th} \) equivalence class since the utility loss of every tuple in an equivalence class is the same.

### 3.1.9 Privacy Provided by Generalized Quasi-Identifiers

Privacy of quasi-identifiers is defined as the ratio of the difference between the entropy of ideal quasi attributes and anonymized quasi attributes to the entropy of ideal quasi attributes. Ideal quasi attributes represent the ideal state where all the tuples of the quasi attributes are unique. Let \( e \) denotes entropy, \( P \) denotes privacy and \( Q \) represents quasi-identifiers. Privacy provided by quasi-identifiers is formulated as

\[
P(Q) = \frac{e(Q_i) - e(Q')}{e(Q_i)}; 0 \leq P(Q) \leq 1
\]  

(13)

where \( Q_i \) represents tuples of ideal quasi-identifiers and \( Q' \) represents tuples of anonymized quasi-identifiers. It ranges from 0 to 1.

### 3.1.10 Privacy Provided by Sensitive Attributes (Prevention of Homogeneity Attack on Sensitive Attributes)

Let there be \( l \) sensitive attributes and \( k \) equivalence classes. \( P(S) \) represents privacy provided by sensitive attributes which is defined and formulated as

\[
P(S) = \sum_{i=1}^{l} \sum_{j=1}^{k} \frac{PHA_{ij}^2}{k}; 0 \leq P(S) \leq 1
\]  

(14)

where \( PHA_{ij} \) represents prevention of homogeneity attack in \( i^{th} \) sensitive attribute and \( j^{th} \) equivalence class. It is formulated as

\[
PHA_{ij} = \frac{e(i^{th} \text{ sensitive attribute and } j^{th} \text{ equivalence class})}{e(Ideal_{ij})}
\]

\[0 \leq PHA_{ij} \leq 1
\]  

(15)

where \( Ideal_{ij} \) represents that all the elements/tuples, in \( i^{th} \) sensitive attribute and \( j^{th} \) equivalence class, are unique.

### 3.1.11 Privacy Provided by Anonymized Dataset \( D' \)

It is defined as the root of the mean of the sum of squared values of privacy provided by quasi-identifiers and privacy provided by sensitive attributes. It is represented as \( P(D') \) and formulated as

\[
P(D') = \sqrt{\left(\frac{P(Q)}{2}\right)^2 + \left(\frac{P(S)}{2}\right)^2}; 0 \leq P(D') \leq 1
\]  

(16)

### 3.2 CDT algorithm

#### Outline of Algorithm

**Input:** Dataset \( D \), containing \( n \) records, consisting of both numerical and categorical attributes.

**Output:** Generalized dataset \( D' \) with minimum possibility of utility loss and homogeneity and background knowledge attack.

1. Split dataset \( D \) into two sets of attributes \( D_1 \) and \( D_2 \) where \( D_1 \) contains tuples of \( m \) sensitive attributes and \( D_2 \) contains tuples of \( l \) quasi attributes.
2. Form equivalence classes of dissimilar tuples from \( D_1 \).
3. Cluster similar tuples from \( D_2 \) corresponding to each equivalence class formed.
4. Form \( D' \) by merging \( D_1 \) and \( D_2 \).
5. Generalize quasi-identifiers in the merged dataset \( D' \).

#### 3.2.1 Clustering of Dissimilar Tuples of Sensitive Attributes

1. Read input dataset \( D_1 = \{A_1, A_2, A_3, A_4, \ldots, A_m\} \) containing \( m \) sensitive attributes and \( n \) records where \( A \) can be either categorical or numerical.
2. Calculate entropy \( e = \{e(A_1), e(A_2), e(A_3), e(A_4), \ldots, e(A_m)\} \) for all the sensitive attributes.
3. Using entropy calculate weight \( w = \{w(A_1), w(A_2), w(A_3), \ldots, w(A_m)\} \).
4. With the help of calculated weights from the previous step, compute Gower’s dissimilarity/distance matrix \( g_{ij}((1, 2, 3, \ldots, i, \ldots, n-1) \times (2, 3, 4, \ldots, j, \ldots, n)) \) where \( i \) and \( j \) represent \( i^{th} \) and \( j^{th} \) tuples in \( D_1 \) respectively.
5. Calculate Gower’s similarity matrix, \( g_s((1, 2, 3, \ldots, i, \ldots, \ldots, n-1) \times (2, 3, 4, \ldots, j, \ldots, n)) \) where \( i \) and \( j \) represent \( i^{th} \) and \( j^{th} \) tuples in \( D_1 \) respectively, using Gower’s dissimilarity matrix \( g_d \).
6. Compute average Silhouette width \( SW \) for clusters from 2 to \( n-1 \) where \( n \) is the number of records in \( D_1 \).
7. Assign \( k \) ← no. of clusters corresponding to minimum average Silhouette width, i.e., number of clusters corresponding to min (SW). We choose minimum average silhouette width because we try to form clusters of dissimilar tuples.
8. Perform \( k \)-medoids algorithm on the dataset \( D_1 \), using Gower’s similarity matrix \( g_s \).
9. It forms equivalence classes \( S \) each containing dissimilar sensitive tuples.
3.2.2 Clustering of Similar Tuples of Quasi Attributes Corresponding to Each Cluster in S

(1) \( D_2 = \{ A_1, A_2, A_3, A_4, \ldots, A_l \} \) containing \( l \) quasi attributes and \( n \) records where \( A \) can be either categorical or numerical.

(2) for each equivalence class in \( S \) do
(a) Corresponding tuples of \( D_2 \), be \( T \), are considered
(b) Calculate entropy \( e = \{ e(A_1), e(A_2), e(A_3), e(A_4), \ldots, e(A_l) \} \) for all the attributes of \( T \).
(c) Using entropy calculate weight \( w = \{ w(A_1), w(A_2), w(A_3), w(A_4), \ldots, w(A_l) \} \).
(d) With the help of calculated weights from the previous step, compute Gower’s dissimilarity/distance matrix \( g_{d}((1, 2, 3, \ldots, i, \ldots, n-1) * (2, 3, 4, \ldots, j, \ldots, n)) \) where \( i \) and \( j \) represent \( i^{th} \) and \( j^{th} \) tuples in \( T \) respectively.
(e) Compute average Silhouette width \( SW \) for clusters from 2 to \( n-1 \) where \( n \) is the number of records in \( T \).
(f) Assign \( k \leftarrow \) no. of clusters corresponding to maximum average Silhouette width, i.e., number of clusters corresponding to max (\( SW \)).
(g) Perform \( k \)-medoids algorithm on the dataset \( T \), using Gower’s dissimilarity matrix \( g_{d} \).
(h) It forms clusters \( Q \) of similar quasi tuples.

end for

3.2.3 Merging of Sensitive and Quasi Attributes

(1) \( D' = S + Q \) (i.e.) \( S \) and \( Q \) have merged in the order of tuples present in the clusters of \( Q \) accordingly.

3.2.4 Generalization of Merged Dataset

(1) For every cluster in \( D' \) generalize tuples of quasi attributes according to the data present within that cluster, i.e., make all the data of quasi attributes in a cluster same as each other.

(2) This dataset could be published to anyone as it would protect privacy of an individual as the dataset is anonymized.

Table 6: Tuples of sensitive attributes of Table 3

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Tuple no.</th>
<th>Race</th>
<th>Disease</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>OC</td>
<td>HIV</td>
<td>100200</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>BC</td>
<td>cancer</td>
<td>13000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>OC</td>
<td>fever</td>
<td>56000</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>BC</td>
<td>cold</td>
<td>44500</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>MBC</td>
<td>HIV</td>
<td>76000</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>OBC</td>
<td>fever</td>
<td>10000</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>SC</td>
<td>pneumonia</td>
<td>23000</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>ST</td>
<td>cancer</td>
<td>43000</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>SC</td>
<td>cold</td>
<td>100200</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>MBC</td>
<td>pneumonia</td>
<td>13000</td>
</tr>
</tbody>
</table>

For example, let the dataset represented by Table 3 be \( D \). \( D \) has been split into two datasets each containing sensitive and quasi attributes respectively. Let the dataset containing sensitive attributes be represented as \( D_1 \) and the dataset containing quasi attributes be represented as \( D_2 \). \( D_1 \) contains race, disease, and salary as these are considered to be sensitive attributes and \( D_2 \) contains age, sex, and place as these are considered to be quasi attributes. CDT algorithm begins by clustering dissimilar tuples of \( D_1 \). Table 6 represents \( D_1 \), \( k \)-medoid algorithm is used for clustering of dissimilar tuples of \( D_1 \) using Gower’s similarity matrix. Minimum average Silhouette width for \( D_1 \) is found to be 8, as shown in Figure 1, but we found 2 to be providing desired results compared to the results formed when \( k \) was 8. Hence, two Clusters are formed as shown in Figure 2.

Table 7: Tuples of corresponding quasi attributes of \( C_1 \)

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Tuple no.</th>
<th>Age</th>
<th>Sex</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12</td>
<td>m</td>
<td>Chennai</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>23</td>
<td>m</td>
<td>Salem</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>24</td>
<td>f</td>
<td>Coimbatore</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>64</td>
<td>f</td>
<td>Madurai</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>42</td>
<td>m</td>
<td>Madurai</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>34</td>
<td>f</td>
<td>Chennai</td>
</tr>
</tbody>
</table>
Let $1^{st}$ cluster be $C_1$ and it consists of tuples 1, 4, 6, 7, 8, 10 and $2^{nd}$ cluster be $C_2$ and it consists of tuples 2, 3, 5, 9. Table 7 consists of tuples of quasi attributes corresponding to $C_1$ and it is represented as $D_{21}$ and Table 8 consists of tuples of quasi attributes corresponding to $C_2$ and it is represented as $D_{22}$. Similar tuples of $D_{21}$ are clustered using $k$-medoid algorithm with the help of Gower’s dissimilarity matrix of $D_{21}$. Figure 3 shows the evaluation of the optimum $k$ value for clustering dataset $D_{21}$. The optimum $k$ value is found to be 2 as it corresponds to maximum average Silhouette width. Tuples of $D_{21}$ have been clustered into two sets $C_{11}$ and $C_{12}$. $C_{11}$ consists of tuples 1, 4, 8 and $C_{12}$ consists of tuples 6, 7, 10. Figure 4 clearly shows the clustering of the tuples along with their Silhouette width. Similar tuples of $D_{22}$ are clustered using $k$-medoid algorithm with the help of Gower’s dissimilarity matrix of $D_{22}$. Figure 5 shows the evaluation of the optimum $k$ value for clustering it dataset $D_{22}$. The optimum $k$ value is found to be 2 as it corresponds to maximum average Silhouette width for $D_{22}$. It has been clustered into 2 sets $C_{21}$ and $C_{22}$. $C_{21}$ consists of tuples 2, 9 and $C_{22}$ consists of tuples 3, 5. Figure 6 clearly shows the clustering of the tuples along with their Silhouette width.

Tuples of $C_1$ are merged with tuples of $C_{11}$ and $C_{12}$ in the order of the tuples present in $C_{11}$ and $C_{12}$ respectively and tuples of $C_2$ are merged with tuples of $C_{21}$ and $C_{22}$ in the order of the tuples present in $C_{21}$ and $C_{22}$ respectively. At last the tuples of quasi attributes of the merged dataset $D'$ are generalized according to the clusters $C_{11}$, $C_{12}$, $C_{21}$ and $C_{22}$ respectively. Table 9 clearly shows the anonymized dataset of $D$ with each equivalence class represented with a unique cluster number.

Table 8: Tuples of corresponding quasi attributes of $C_2$

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Tuple no.</th>
<th>Age</th>
<th>Sex</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>45</td>
<td>f</td>
<td>Salem</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>36</td>
<td>m</td>
<td>Coimbatore</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>57</td>
<td>m</td>
<td>Chennai</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>64</td>
<td>f</td>
<td>Madurai</td>
</tr>
</tbody>
</table>

Every equivalence class of Table 9 has different sensitive values because every cluster has been formed using equivalence classes consisting of dissimilar tuples of sensitive attributes. Thereby reducing the homogeneity attack and background knowledge attack maximum possible. Also, quasi attributes are generalized with minimum utility loss possible. Generalized anonymization helps to prevent an intruder from accessing data from the published dataset. Hence, increasing the privacy of an individual.
Table 9: Anonymized Dataset of Table 3

<table>
<thead>
<tr>
<th>Cluster no.</th>
<th>Tuple no.</th>
<th>Age</th>
<th>Sex</th>
<th>Place</th>
<th>Race</th>
<th>Disease</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>12-42</td>
<td>m</td>
<td>Chennai, Madurai, Salem</td>
<td>OC</td>
<td>HIV</td>
<td>100200</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>12-42</td>
<td>m</td>
<td>Chennai, Madurai, Salem</td>
<td>BC</td>
<td>cold</td>
<td>44500</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>12-42</td>
<td>m</td>
<td>Chennai, Madurai, Salem</td>
<td>ST</td>
<td>cancer</td>
<td>43000</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>24-64</td>
<td>f</td>
<td>Chennai, Coimbatore, Madurai</td>
<td>OBC</td>
<td>fever</td>
<td>10000</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>24-64</td>
<td>f</td>
<td>Chennai, Coimbatore, Madurai</td>
<td>SC</td>
<td>pneumonia</td>
<td>23000</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>24-64</td>
<td>f</td>
<td>Chennai, Coimbatore, Madurai</td>
<td>MBC</td>
<td>pneumonia</td>
<td>13000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>45-64</td>
<td>f</td>
<td>Madurai, Salem</td>
<td>BC</td>
<td>cancer</td>
<td>13000</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>45-64</td>
<td>f</td>
<td>Madurai, Salem</td>
<td>SC</td>
<td>cold</td>
<td>100200</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>36-57</td>
<td>m</td>
<td>Chennai, Coimbatore</td>
<td>OC</td>
<td>fever</td>
<td>56000</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>36-57</td>
<td>m</td>
<td>Chennai, Coimbatore</td>
<td>MBC</td>
<td>HIV</td>
<td>76000</td>
</tr>
</tbody>
</table>

4. RESULT AND DISCUSSION

The main goal is to investigate the performance implications of the CDT approach in terms of utility loss, privacy gain and prevention of homogeneity and background knowledge attack. Since background knowledge attack is unpredictable as it depends on the intruder, only homogeneity attack has been evaluated.

Table 10: Technical specifications of the system used

<table>
<thead>
<tr>
<th>Properties</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Windows 10</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel Core i7-8700k</td>
</tr>
<tr>
<td>Processor base frequency</td>
<td>3.70 GHz</td>
</tr>
<tr>
<td>Installed memory (RAM)</td>
<td>16 GB</td>
</tr>
<tr>
<td>System type</td>
<td>x64 based processor</td>
</tr>
</tbody>
</table>

Adult dataset, taken from UCI machine learning repository, has been used. Randomly 1000 records are chosen and processed. Age (numerical), Sex (categorical) and Place (categorical) are used as quasi attributes and Occupation (categorical), Education (Categorical) and FNLWGT (Categorical) are used as sensitive attributes. Algorithms are implemented in R language version 3.6.2 using RStudio version 1.2.5033. Table 10 denotes the configuration of the system used for the implementation of algorithms.

We split the dataset $D$ into five groups $G_1$, $G_2$, $G_3$, $G_4$, and $G_5$ each containing 100, 200, 300, 400, and 500 tuples respectively for which utility loss is evaluated as depicted in Figure 7. Figure 7 represents utility loss, in percentage, in the Y-axis and number of tuples in the X-axis. $G_1$, after processing, gives 42.51% utility loss and similarly, $G_2$, $G_3$, $G_4$, and $G_5$ shows 46.05%, 44.80%, 50.20% and 51.89% utility loss respectively. From Figure 7, it can be derived that utility loss of any dataset is approximately 47%.

Figure 8 depicts a plot representing privacy provided by generalized quasi attributes, prevention of homogeneity attack in sensitive attributes and privacy gain of the anonymized dataset against no. of tuples. Privacy gain is calculated by measuring the difference between the privacy provided by the dataset before and after processing it. The contribution of generalized quasi attributes to overall privacy for $G_1$, $G_2$, $G_3$, $G_4$, and $G_5$ are 58.60%, 62.04%, 67.29%, 73.87%, and 75.27% respectively. The contribution of prevention of homogeneity attack to overall privacy for $G_1$, $G_2$, $G_3$, $G_4$, and

![Figure 7: Utility loss vs No. of tuples](image1)

![Figure 8: Privacy provided by generalized quasi attributes, Prevention of homogeneity attack in sensitive attributes and Privacy gain of anonymized dataset against No. of tuples](image2)
G5 are 71.28%, 63.73%, 64.30%, 76.65%, and 58.81% respectively. Similarly, the privacy gain of the anonymized dataset for G1, G2, G3, G4, and G5 are 52.27%, 46.27%, 46.37%, 52.89%, and 42.50% respectively. When noticed carefully even upon increasing the number of tuples there aren’t any significant changes shown in the privacy gain of the datasets and it is between 40-60% range as depicted in the Figure 8.

We processed the entire dataset D containing 1000 records and evaluated utility loss, privacy provided by generalized quasi attributes and prevention of homogeneity attack in sensitive attributes as depicted in Figure 9. Utility loss is found to be 62.28% and similarly, privacy provided by processed quasi and sensitive attributes are found to be 79.84% and 69.41% respectively.

Figure 10 depicts No. of tuples vs Elapsed time, in seconds for G1, G2, G3, G4, and G5. Y-axis represents elapsed time and X-axis represents number of tuples as depicted in Figure 10. The elapsed time for G1, G2, G3, G4, and G5 are found to be 3.61 sec, 6.68 sec, 28.33 sec, 67.51 sec, and 157.69 sec respectively. The plot shows an increasing trend in elapsed time as the number of tuples increases.

This project’s main objective is to minimize utility loss and maximize the privacy gain by preventing homogeneity attack from happening in sensitive attributes. Based on the results obtained we can conclude that the utility loss has been minimized and prevention of homogeneity attack has been maximized.

5. CONCLUSION AND FUTURE WORK

In this study, we have shown the incompetencies of l-diversity and t-closeness in thwarting homogeneity and background knowledge attack and proposed a stronger privacy notion to thwart afore-mentioned attacks. From the results, our algorithm has proven to be providing maximum privacy gain, minimum privacy loss, and maximum thwart to homogeneity and background knowledge attacks.

When all the records are similar there is a chance that our algorithm, forming clusters of dissimilar tuples, forms as many clusters as close to the number of records which makes anonymization a difficult task. Considering this as an avenue for future work, we are trying to form a clustering algorithm incorporating similarity of all the records in a dataset in such a way that it does not form as many clusters as close to the total number of records. Basically, we are preparing a dynamic clustering algorithm which adapts itself to the input data in order to provide better results with the optimum number of clusters no matter how similar or dissimilar the records are.

REFERENCES


