GRAPH BASED LOCAL RECODING FOR DATA ANONYMIZATION

K. Venkata Ramana and V.Valli Kumari

Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India
{kvramana.auce, vallikumari}@gmail.com

ABSTRACT

Releasing person specific data could potentially reveal the sensitive information of an individual. k-anonymity is an approach for protecting the individual privacy where the data is formed into set of equivalence classes in which each class share the same values. Among several methods, local recoding based generalization is an effective method to accomplish k-anonymization. In this paper, we proposed a minimum spanning tree partitioning based approach to achieve local recoding. We achieve it in two phases. During the first phase, MST is constructed using concept hierarchical and the distances among data points are considered as the weights of MST and in the next phase we generate the equivalence classes adhering to the anonymity requirement. Experiments show that our proposed local recoding framework produces better quality in published tables than existing Mondrian global recoding and k-member clustering approaches.

KEYWORDS

Anonymity, Local recoding, Minimum Spanning tree Partition, Data Privacy, Priority Queue

1. INTRODUCTION

Huge volumes of operational data and information are being collected by various vendors and organizations. This data is analysed by different business and government organizations for the purpose of decision making and social benefits such as statistical analysis, medical research, crime reduction and other purposes. However, analysing such data causes new privacy threats to individuals [4]. Traditional approach is to de-identify the microdata by removing identifying attributes like social security number, name and address [17]. Even though, these de-identified attributes are removed the possibility of revealing an individual still exists through linking attack [17, 18]. *k-anonymity* is one such model to avoid the linking attack, in which the domain of each quasi identifier attribute is divided into equivalence classes and each equivalence class contains at least *k* identical elements[3, 17, 25]. Samarati and Sweeney formulated *k*-anonymization mechanism using generalization and suppression. In generalization we replace more specific value with less specific value [18, 11]. For example, the value of the age 24 is replaced by the range [20-25] using attribute domain hierarchy of age. Suppression is another form of generalization in which the least significant digit for continuous attributes are replaced with symbols like '*'. For example, the attribute zip-code value "535280" is suppressed by "2352**". Global recoding [12, 13, 20] and local recoding [16, 21] are two such approaches to achieve *k*-anonymization through generalization and suppression.

DOI : 10.5121/ijdms.2013.5401

1.1 Local Recoding Versus Global Recoding

In global recoding, the domain of the quasi identifier values are mapped to generalized values for achieving *k*-anonymity [12, 13, 21]. The limitation of the global recoding is; the domain values are over generalized resulting in utility loss where as in local recoding, the individual tuple is mapped to a generalized tuple [16, 21]. The information loss of the global recoding is more than the local recoding approach. We show how these two techniques differ with an example.We followed the scheme as presented in [21] for clear understanding of local and global anonymization schemes. Let us consider the 2-Dimensional data region shown in Figure.1 (a) with an anonymity constraint of *k*=3. Let the 2-D attribute values are (x_1, x_2, x_3), (y_1, y_2, y_3) and are partitioned into 9 regions as shown in Figure.1 (a). Here the count value of the region (x_2,y_2) is less than three. Therefore, we need to merge this region to another region to meet the anonymity requirement. In the global recoding generalization scheme, a merged region stretches over the range of other attributes. For example, the merged region in Fig. 1(b) covers all values of attribute 1 since all occurrences of y_1 and y_2 in attribute2 have to be generalized.From the table point of view, domain (y_1, y_2, y_3) is mapped to domain ([y_1, y_2], y_3). The global recoding generalization causes some unnecessary merges, for example., regions (x_1, [y_1-y_2]) and (x_3, [y_1-y_2]). This is the overgeneralization problem of global recoding generalization.

In local recoding generalization scheme, any two or more regions can be merged as long as the aggregated attribute value such as [y_1-y_2] satisfies the anonymity requirement. For example, regions (x_2, y_1) and (x_2, y_2) are merged into (x_2, [y_1-y_2]) and regions (x_1, y_1), (x_1, y_2) (x_3, y_2) keep their original areas. In Figure. 1(c) a table view of all the tuples of the region (x_2, y_1) and (x_2, y_2) are mapped to (x_2, [y_1-y_2]), but tuples of the regions (x_1, y_1), (x_1, y_2) and (x_3, y_2) remain unchanged. This clearly shows that this scheme is much better when compared to global generalization scheme.

	y_1	y_2	y_3
x_1	3	8	0
x_2	7	2	6
x_3	0	8	4

	$[y_1y_2]$	y_3
x_1	11	0
x_2	9	6
x_3	8	4

	y_1	y_2	y_3
x_1	3	8	0
x_2	9		6
x_3	0	8	4

Attr1	Attr2	Count
x_1	y_1	3
x_1	y_2	8
x_2	y_1	7
x_2	y_2	2
x_2	y_3	6
x_3	y_2	8
x_3	y_3	4

(a)

Attr1	Attr2	Count
x_1	$[y_1$-$y_2]$	11
x_2	$[y_1$-$y_2]$	9
x_2	y_3	6
x_3	$[y_1$-$y_2]$	8
x_3	y_3	4

(b)

Attr1	Attr2	Count
x_1	y_1	3
x_1	y_2	8
x_2	$[y_1$-$y_2]$	9
x_2	y_3	6
x_3	y_2	8
x_3	y_3	4

(c)

Figure. 1 (a) Original Data (b) Generalization by a Global recoding approach (c) Generalization by a local recoding approach

In this paper, we present local recoding generalization approach based on minimum spanning tree partitioning. This paper is organized as follows: Related work is given in Section 2. Section 3 presents the basic definitions and terminologies that were used throughout the paper. We present our proposed MST based local recoding model in section 4. Section 5 contains essential quality measures necessary for assessing our method. Algorithm and the complexity measures of our

approach were discussed in Section 6. We present our experimental evaluation in Section 7 and we finally conclude along with future work in Section 8.

2. RELATED WORK

Several global and local recoding generalization algorithms were proposed to accomplish *k*-anonymity requirement. In multidimensional global recoding, the entire domain is partitioned into set of non-overlapping regions and each region contains at least *k*-data points. These data point in each region are generalized so that all the points in the region share the same quasi identifier value. However this method may cause high data distortion due to over generalization of the domain [12, 13].

Local recoding method can improve the quality of anonymization by reducing the amount of generalization. Most of the local recoding generalization algorithms follow clustering based approach where each cluster should satisfy anonymity requirement [1, 2, 6, 10, 14, 19, 28]. [2] Proposed condensation based approach where the data is condensed into multiple groups having pre-defined size. In each group they maintain statistical information like mean and correlation among different records. The anonymized data which is obtained by this approach preserves high privacy based on the in distinguishability level defined. However, the main limitation of this approach is, it produces high information loss because large numbers of records were merged into a single group. Gagan Aggrawal et al. proposed *r-gather clustering* for anonymity where the data records are partitioned into clusters and release the cluster centres, along with their size, radius, and a set of associated sensitive values [14]. Grigorious et al. addressed sampling based clustering for balancing the data utility and privacy protection. In this approach the tuples are grouped based on the median of the data[28]. These approaches mainly deal with only numerical attributes, but this approach is not quite effective for the categorical attributes.

Hua Zhu et al. proposed density based clustering approach to achieve *k*-anonymity [19]. The key idea of this algorithm is to generate the equivalence classes based on density and is measured by *k*-nearest neighbour distance. Ji-Won Byun et al. formulated greedy approach in which *k*-anonymity problem is transformed into *k*-member clustering to attaining the privacy protection of the data [6]. A frame work called KACA to accomplish the *k-anonymity*,in which grouping of the tuples is done by attribute hierarchical structures [21]. [19, 6, 21] can handle both numerical and categorical attributes but fail in determining exact boundaries for the equivalence classes resulting into inappropriate generalization. This may lead to less utility while deriving desired patterns.

On the other hand, privacy preserving is achieved through cryptographic based techniques [1, 10, 27, 29]. Here, privacy is protected when multiple parties try to share their sensitive data. This sharing of data is protected by applying secure cryptographic protocols. These approaches partition the data either horizontally or vertically and then distributed among the parties. Since data mining techniques involve in handling millions of records it may seriously result for the cryptographic protocols to increase their communication cost leading to an impractical state. Also these methods hide data from unauthorized users during data exchange.

3. PRELIMINARIES

Let T be the microdata to be published. The table contains m attributes $A = \{A_1, A_2,..., A_m\}$ and their domains $\{D[A_1], D[A_2],... D[A_m]\}$ respectively. The concept hierarchies of domains are $\{H_1, H_2, ...,H_n\}$. A tuple $t \in T$ is represented as $t = (t[A_1], t[A_2], ...,t[A_m])$, where $t[A_i]$ is i^{th} attribute value of tuple t.

Definition 1(Tuple Partitioning and Local recoding generalization): Let T be the table contains n tuples and is partitioned into m subsets $\{S_1, S_2, S_3,...,S_m\}$, such that each tuple belongs to exactly one subset. $\cup_{i=1}^{m} S_i = T$and for any $1 \leq i \neq j \leq m$, $S_i \cap S_j = \emptyset$. The local recoding generalization function *f** is a function that maps each tuple of S_i to some recoded tuple t^1, where t^1 is obtained replacing for all tuples of S_i with *f**(t).

For example, the tuples of in table 1(a) partitioned into three subsets {{1, 2, 3}, {4, 5, 6, 7}, {8, 9, 10}}. These subset quasi-identifier values {{<Male, 21, 535280>, <Male, 24, 535280>, <Male, 25, 535280>}, {<Female, 26, 535280>, <Female, 26, 535285>, <Female, 32, 535288>, <Female, 32, 535292>}, {<Male, 36, 535292>,<Male, 36, 535296>, <Male, 38, 535296>}}are recoded to {{<Male, [20-25], 535280>},{<Female, [20-40], 5352**>}, {<Male, [36-40], 53529*>}}using concept hierarchy as shown in Figure. 2. Hence, the table generates local recoded equivalence classes.

Table 1. (a) Original table(b) 3-anonymity view by global recoding (c) 3-anonymity view by local recoding

ID	Gender	Age	Zip-code	Disease
1	Male	21	535280	Flu
2	Male	24	535280	HIV
3	Male	25	535280	Heart Disease
4	Female	26	535280	Heart Disease
5	Female	26	535285	Cancer
6	Female	32	535288	Flu
7	Female	32	535292	Flu
8	Male	36	535292	HIV
9	Male	36	535296	Cancer
10	Male	38	535296	Obesity

ID	Gender	Age	Zip-code	Disease
1	Male	[20-40]	5352**	Flu
2	Male	[20-40]	5352**	HIV
3	Male	[20-40]	5352**	Heart Disease
4	Female	[20-40]	5352**	Heart Disease
5	Female	[20-40]	5352**	Cancer
6	Female	[20-40]	5352**	Flu
7	Female	[20-40]	5352**	Flu
8	Male	[20-40]	5352**	HIV
9	Male	[20-40]	5352**	Cancer
10	Male	[20-40]	5352**	Obesity

ID	Gender	Age	Zip-code	Disease
1	Male	[20-25]	535280	Flu
2	Male	[20-25]	535280	HIV
3	Male	[20-25]	535280	Heart Disease
4	Female	[20-40]	5352**	Heart Disease
5	Female	[20-40]	5352**	Cancer
6	Female	[20-40]	5352**	Flu
7	Female	[20-40]	5352**	Flu
8	Male	[36-40]	53529*	HIV
9	Male	[36-40]	53529*	Cancer
10	Male	[36-40]	53529*	Obesity

Definition 2 (Equivalence class): The equivalence class of tuple *t*in table T,is the set of tuples in T with identical quasi-identifiers to t. For example in table 1(b), the equivalence class of tuples 1, 2, 3 is <Male, [20-25], 535280>.

Definition 3 (*k*-Anonymity property): A table T is said to be *k*-anonymous with respect to the quasi-identifier attribute if the size of the each equivalence class is at least *k*.
For example, the 3-anonymous view of the Table 1(a) as shown in Table 1(b) and Table 1(c) in which the size of each equivalence class is at least 3.

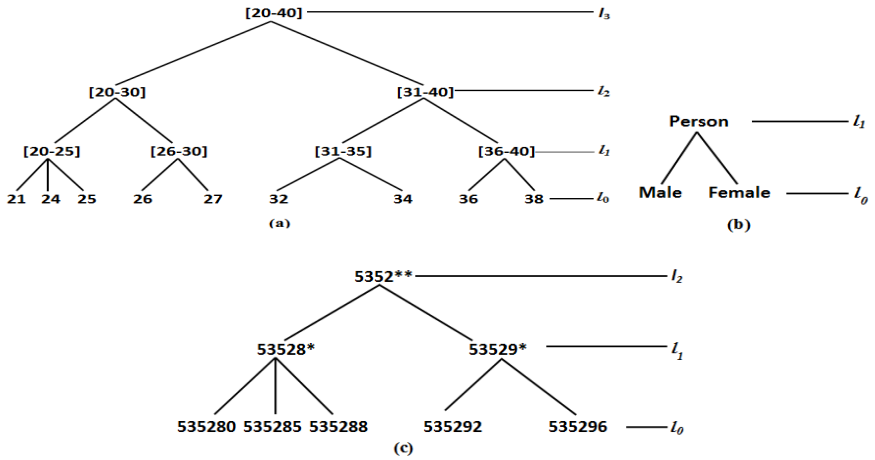

Figure 2. Concept Hierarchies for attributes: (a) Age (b) Gender (c) Zip-code

4. LOCAL RECODING ANONYMIZATION MODEL

The framework MST based local recoding for data anonymity is shown in Figure. 3 This frame work consist the following steps.

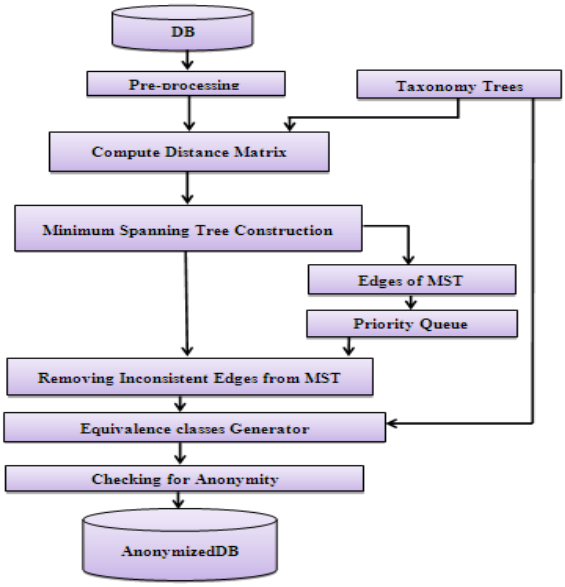

Figure 3. Framework of the MST based local recoding

Step 1: Data preprocessing and distance matrix computation.
Step 2: Construct minimum spanning tree based on the distance matrix.
Step 3: Remove the longest edges and form the initial clusters.
Step 4: Generate equivalence classes using concept hierarchical structures (Taxonomy trees) for each attributes and check for anonymity level.

4.1 Data preprocessing and compute distance matrix

In this step the quasi identifiers attributes which are to be anonymized and its concept hierarchies are selected. For example Gender, Age and Zip-code attributes are quasi identifier attributes in

the Table. 1 (a) and its concept hierarchies are shown in Figure 2. We compute the distances among all tuples based on the following definitions

Definition 4 (Concept Hierarchical Distance (CHD)): Let v and v^lbe the two nodes of concept hierarchy and H be the height of the tree. The concept hierarchical distance between the two node are defined as

$$\text{CHD}\ (v,\ v^l) = \frac{|v^1 - v|}{H} \tag{1}$$

Here $|v^l - v|$ is the difference between the two levels of v and v^l and $v^l > v$.
For example the values of the attribute Age in concept hierarchy are {24, [20- 25], [20-30], [20-40]} as shown in Figure 1(a). The distance between the value 24 and [20-30] is 2/3. The CHD is zero if both values are at same level or at the same leaf nodes and CHD is one if the value lies at the root.

Definition 5 (Conceptual Hierarchical Effort of Record (CHE)): Letr and r^1be the tuple and generalized tuple respectively. The Conceptual hierarchical effort of a record is defined as the amount of effort needed to change the attribute values of record one (low) level to another (generalized) level in concept hierarchies of attributes of a tuple. i.e.

$$\text{CHE}\ (r,\ r^1) = \sum_{i=1}^{m} \text{CHD}(a_i, a_i^1) \tag{2}$$

Here, a_i, a_i^1 are the original attributes values and generalized attribute values of the record r respectively. For example, consider record r_2 {M, 24, 535280} in the Table 1(a) and its generalized values r_2^1 {M, [20-40], 535**} in Table 1(b). The CHD (M, M) = 0, CHD (24, [20 - 40]) = 1 and CHD (535280, 5352**) = 1, therefore CHE (r_2,r_2^1) = 2.

Definition 6 (Hierarchical Distance between two Records (HDist)): Let r_1 and r_2 be the two records and their closest common ancestor be the r_{12}.The hierarchical distance between two tuples defined as

$$\text{HDist}(r_1, r_2) = \text{CHE}(r_1, r_{12}) + \text{CHE}(r_2, r_{12}) \tag{3}$$

Forexample, in Table 1(a)the common ancestor of records r_2 and r_3 is {M, [20- 25], 535280}. The conceptual hierarchical effort, CHE (r_2, r_{23}) = 0.333 and CHE (r_3, r_{23}) = 0.333. Therefore the hierarchical distance between records r2 and r3 is 0.333 + 0.333 = 0.666.

Definition 7 (Distance Matrix): Given micro table T with n records {r_1, r_2, r_3, …,r_n}and each record contains m quasi identifiers. The distance matrix is defined as

$$D_T = \left[\text{Hdist}(r_i, r_j)\right]_{n \times n} \forall i, j \in n \tag{4}$$

4.2 Minimum Spanning Tree Construction

Our Method relies on Kruskals algorithm for constructing Minimum spanning tree[22, 23, 26]. The nodes of the MST are the data points (records) of the micro table and the weight of edges are the concept hierarchical distance between two data points.The hierarchical distances among the quasi identifier of Table 1(a) as shown in Figure. 2. Construct MST for the Table 1(a) as shown in Figure 5(a). The edges of the MST are stored in priority queue according to their weights in decreasing order.

$$\begin{bmatrix} 0 & 0.66 & 0.66 & 3.34 & 4.32 & 5.0 & 6.0 & 4.0 & 4.0 & 4.0 \\ - & 0 & 0.66 & 3.34 & 4.32 & 5.0 & 6.0 & 4.0 & 4.0 & 4.0 \\ - & - & 0 & 3.34 & 4.32 & 5.0 & 6.0 & 4.0 & 4.0 & 4.0 \\ - & - & - & 0 & 1.0 & 3.0 & 4.0 & 6.0 & 6.0 & 6.0 \\ - & - & - & - & 0 & 3.0 & 4.0 & 6.0 & 6.0 & 6.0 \\ - & - & - & - & - & 0 & 2.0 & 5.34 & 5.34 & 5.34 \\ - & - & - & - & - & - & 0 & 3.34 & 4.32 & 4.32 \\ - & - & - & - & - & - & - & 0 & 1.0 & 2.32 \\ - & - & - & - & - & - & - & - & 0 & 0.66 \\ - & - & - & - & - & - & - & - & - & 0 \end{bmatrix}$$

Figure. 4. Concept-tree hierarchical distance matrix

4.3 Remove the longest edges and forming the initial clusters:

For forming the initial clusters we remove the longest edges from the MST forest. Hence the MST forest is split into set of sub trees. Each of the sub-tree is the initial cluster and the node of the subtree is called the member of a cluster. The Maximum number of edges removed from MST is $\left\lceil {}^{n}/_{k} \right\rceil - 1$, here n is the number of data point in MST and *k* is the anonymity constraint.

Table 2. Priority queue edges and its weights

Priority Queue	**Edge**	1-4	7-8	4-6	6-7	4-5	8-9	1-2	2-3	9-10
	Weight	3.34	3.34	3.0	2.0	1.0	1.0	0.66	0.66	0.66

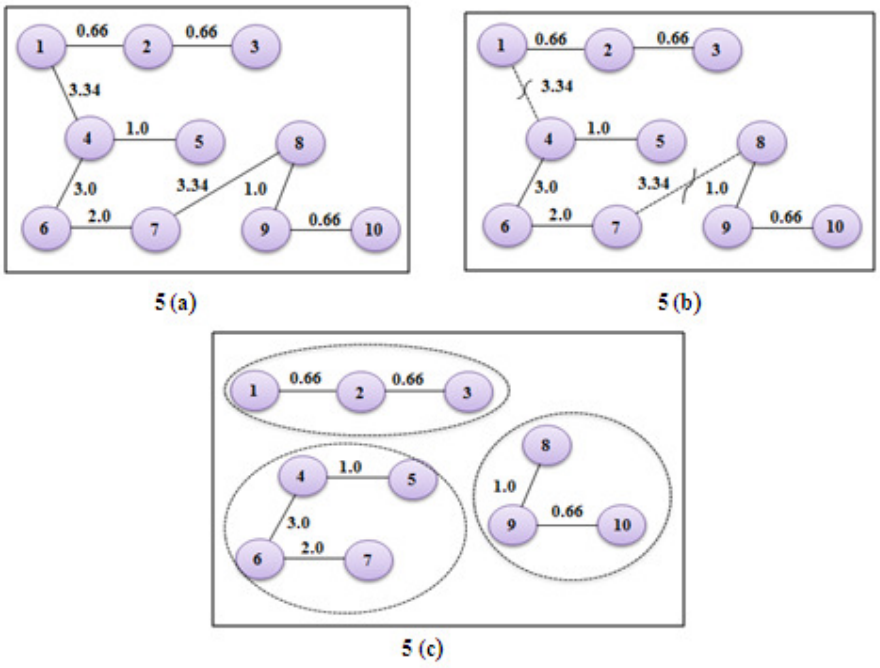

Figure5. (a) MST Construction (b) Longest Edge Removal (c) Equivalences Class Generation

For example if the anonymity requirement is 3 for the MST forest as shown in Figure 5(a), then the number of edges to be removed from MST is 2. The first two longest edges are 1-4 and 7-8 as

shown in table 2 and then these two edges are removed from MST. Hence the forest is divided into three initial clusters as shown in Figure. 5(c).

4.4 Generating the Equivalence Classes:

After forming the initial clusters, by using the concept hierarchies we transform the clusters into equivalence classes. For example the equivalence classes of clusters of the Figure. 5(c) are <Male, [20-25], 535280>, <Female, [20-40], 5352**> and <Male, [36-40], 53529*>.

5. QUALITY MEASURES OF *k*-ANONYMIZATION

The quality of anonymization can be expressed in the form of information loss and utility. Several measure were proposed in the literature, such as Classification metric [9],Normalized certainty penalty (NCP) [15], Global certainty penalty (GCP) [15], Entropy [7, 8], Model accuracy [12], Discernability penalty [5], Normalized equivalence class size metric [21], Query quality [16]. In this paper, we adopt NCP, GCP, Discernability penalty, Normalized equivalence class size (CAVG) for representing the quality of anonymity.

The NCP of the equivalence class E, for the numerical and categorical attributes as follows

$$NCP\ (A^{E}_{numeric}) = \frac{A^{E}_{\max} - A^{E}_{min}}{A_{max} - A_{min}} \tag{5}$$

Here, A^{E}_{max}, A^{E}_{min} are the maximum and minimum values of the equivalence class E and A_{max}, A_{min} are the maximum and minimum values of entire attribute domain.

$$NCP^{E}_{categorical} = \begin{cases} 0, & cardinality(l) = 1 \\ cardinality(l)/|A_{categorical}|, & otherwise \end{cases} \tag{6}$$

Where, l is number of leaf nodes rooted at current node, cardinality (l) represents the number of leaves in the sub-tree l and $|A_{categorical}|$ is total number of distinct categorical values of attribute A. NCP determines only the information loss of single equivalence class. The GCP represents the information loss of entire table. The GCP of the anonymized table T^1 defined as follows.

$$GCP(T') = \frac{\sum_{E_i \in S} |E_i|.NCP(E_i)}{d.N} \tag{7}$$

Here, S is the set of equivalence classes of the anonymized table T^1, E_i is the size of the equivalence class, d is the dimensionality of the quasi identifier attributes and N is number of tuples of the table T. The value of GCP lies between the 0 and 1 where 1 signifies only one equivalence class covering all the tuples in the table and 0 indicates the no information loss i.e no generalization is performed.

The discernibility penalty is another quality metric in which the penalty is assigned to each tuple based on how many tuples in the transformed dataset are indistinguishable from it. The better the anonymization the discernibility penalty cost will reduce.

$$DM = \sum_{Equivalence\ classes\ E} |E|^2 \tag{8}$$

Where the |E| is the size of the equivalence class
Normalized equivalence class size (CAVG) metric measures how well the partitioning approaches the best case where each tuple is generalized in a group of *k* indistinguishable tuples and is formally give as

$$CAVG = \frac{total\ number\ of\ records}{total\ number\ of\ equivalence\ classes.k} \quad (9)$$

An objective is to reduce the normalized average equivalence class size.

6. ALGORITHM

In this section, we discuss how to achieve *k*-anonymization by MST based local recoding. Our approach is viewed as graph based clustering problem, in which minimum spanning tree is used as data structure to generate clusters. Initially, we find the distance matrix among the all QI tuples using attribute hierarchies. We adopted Kruskal algorithm for the constructing the MST. The data points (QI tuples) are the nodes and the distances among the data points are the weights of the MST edges.

MST based Local Recoding Algorithm

Input: Quasi Identifier Data points (QID), anonymity constraint *k*, Distance matrix $M_{n \times n}$among all the data points(tuples)
Output: anonymized table T^l
Method:

1. **Begin**
2. T ← GenerateMST (QID)
3. E ←Φ
4. C← Φ
5. **For** all e ∈ Edges (T) **do**
6. E ← E ∪ { e }
7. **End for**
8. **For** i← 1 to $\left\lfloor n/k \right\rfloor$ – 1 **do**
9. PQ ← PriorityQueue (E)
10. **End for**
11. ST ← PartMST (T , PQ)
12. **For** each sub-tree t∈ ST **do**
13. C← C ∪ GenEquiClass(t)
14. **End for**
15. **While** there exists some equivalence class C such that |C| < *k* do
16. **For** each Class C such that |C| <*k* do
17. scan all other equivalence classes once to find group C^1
18. such that NCP(C∪ C^1) is minimized
19. merge the equivalence classes C and C^1
20. **End For**
21. **End While**
22. **End Begin**

In Kruskal algorithm each vertex is in its own tree in a forest. Then, algorithm considers each edge in turn, ordered by increasing weight. If an edge (*u*, *v*) connects two different trees, then (*u*, *v*) is added to the set of edges of the MST, and two trees connected by an edge (*u*, *v*) are merged into a single tree. On the other hand, if an edge (*u*, *v*) connects two vertices in the same tree, then edge (*u*, *v*) is pruned. After constructing the MST, the edges are stored in the priority queue in decreasing order. Step 8 to step 11 shows removing the $\left\lceil n/k \right\rceil$ - 1 longest edges from MST forest

and the splitting the MST forest into $\lceil n/k \rceil$ sub-trees. Each sub-tree of the forest is nothing but an equivalence class and the datapoints of each sub-tree are generalized to same value based on concept hierarchies. After forming the equivalence classes we check for the anonymity constraint k in each class. Those classes, which do not satisfy k-anonymity such classes, are merged into other classes based on the NCP of the class. The time complexity for computing distance matrix $O(n^2)$. The complexity of MST construction is $O(|E| \log_2|E|) = O(n \log_2 n)$ where $|E|$ denotes number of edges inthe graph. The time complexity for forming the initial cluster is $O(n)$. Hence the overall complexity of the this algorithm is $O(n^2) + O(n \log_2 n) + O(n) \approx O(n^2)$

7. EXPERIMENTATION

We conducted several experiments to show the efficacy of our algorithm. All the experiments were conducted on a bench mark dataset, adult available at UCI machine learning datasets repository [24]. We analysed the dataset and removed the missing values from it and the final dataset holds 30,162 records. The dataset consists of several numerical and categorical attributes out of which we considered age as numerical and other attributes {Work-class, Gender, Education and Occupation} as categorical. The number of distinct values and the height of the concept hierarchy tree are detailed in the Table 3. We implemented our algorithm in java on Core2duo machine @2.90GHz with 4GB RAM and Windows 7 operating system. We compared our proposed algorithm with to state-of-art methods Mondrian multidimensional global recoding [12] and k-member clustering methods [6] in terms of quality metrics.

Table 3. Adult Dataset Description

	Attribute	Type	Distinct values	Tree height
1	Age	Numeric	74	5
2.	Work class	Categorical	7	3
3	Gender	Categorical	2	1
4	Education	Categorical	16	4
5	Occupation	Categorical	14	2

7. 1 Quality of Anonymized Table:

In this section, we present our experimental results for the data quality metrics. The total information loss of the anonymized table is measured using GCP. The GCP of three algorithms for increasing values of k (k = 3, 6, …, 21) and for different QI values are measured and is shown in Figure 7. (a), (b), (c), (d), (e). We observed that our approach produced less information loss when compared with Mondrian and k-member clustering.

The second data quality measure of anonymized table is measured using Discrenability[21].It measures the data quality based on the size of each equivalence class. Intuitively, data quality diminishes as more records become distinguishable with respect to each other, and DM effectively captures this effect of the k-anonymization process. The DM for three algorithms for increasing values of k (k= 3, 6, ..., 21) is shown in Figure.8. MST based partitioning algorithm gives better DM when compared with the remaining two algorithms. We also measure the DM with respect to the different QI sizes as shown in Figure. 8. (a), (b), (c), (d), (e).

Figure. 9.shows the experimental results with respect to CAVG metric described in section 6. This metric measures how well the partitioning approaches determine the best case where each tuple is generalized into a group of k indistinguishable tuples. The CAVG of three algorithms for

increasing values of k (k= 3, 6, ..., 21) are shown in Figure. 9 (a),(b),(c),(d),(e). MST based partitioning algorithm achieves better CAVG when compare with [12][6].

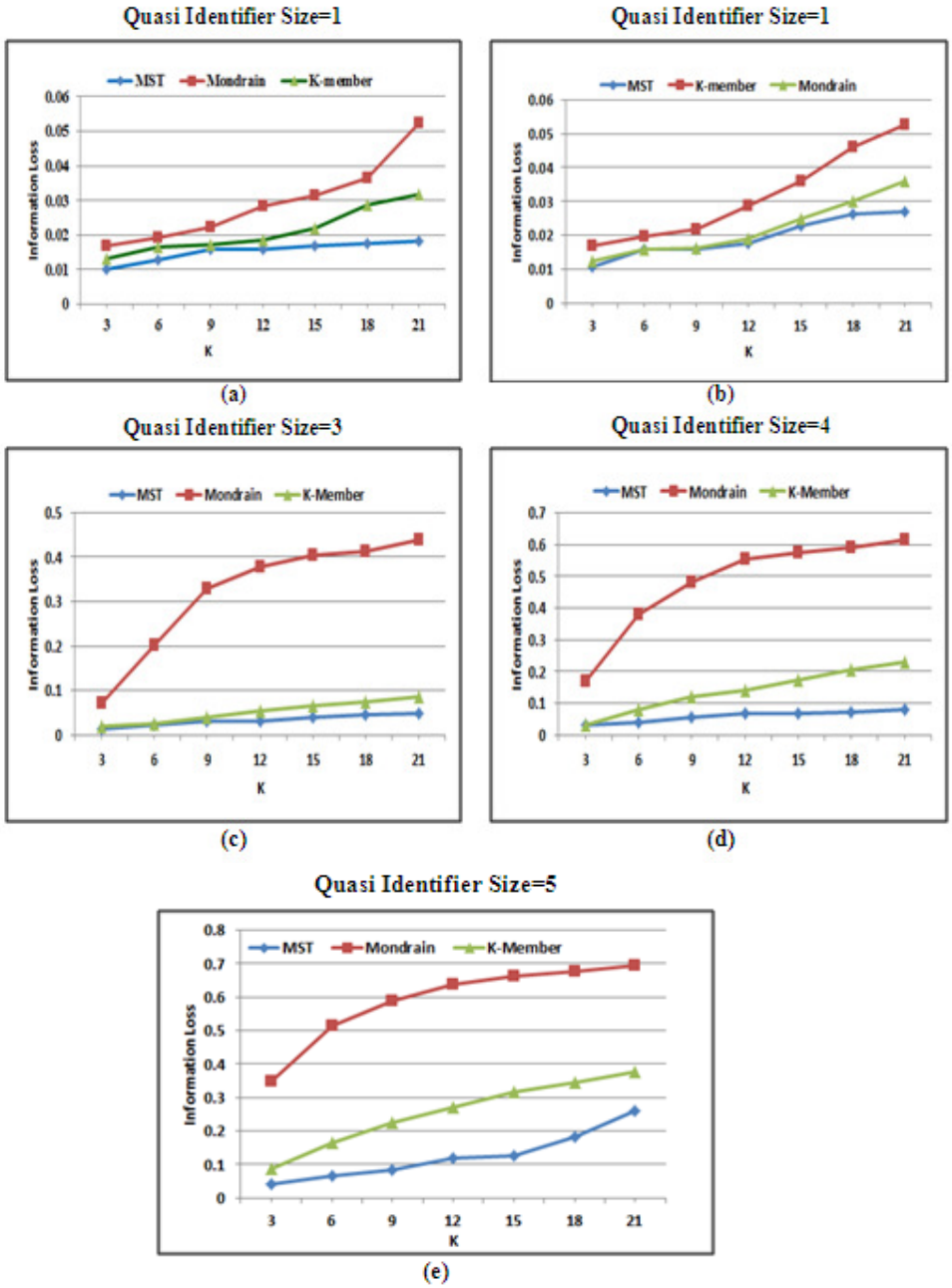

Figure. 7. Information loss of different methods with variant of k (a) Quasi identifier size= 1 (b) Quasi identifier Size = 2 (c) Quasi identifier Size = 3 (d) Quasi identifier Size = 4 (e) Quasi identifier Size=5

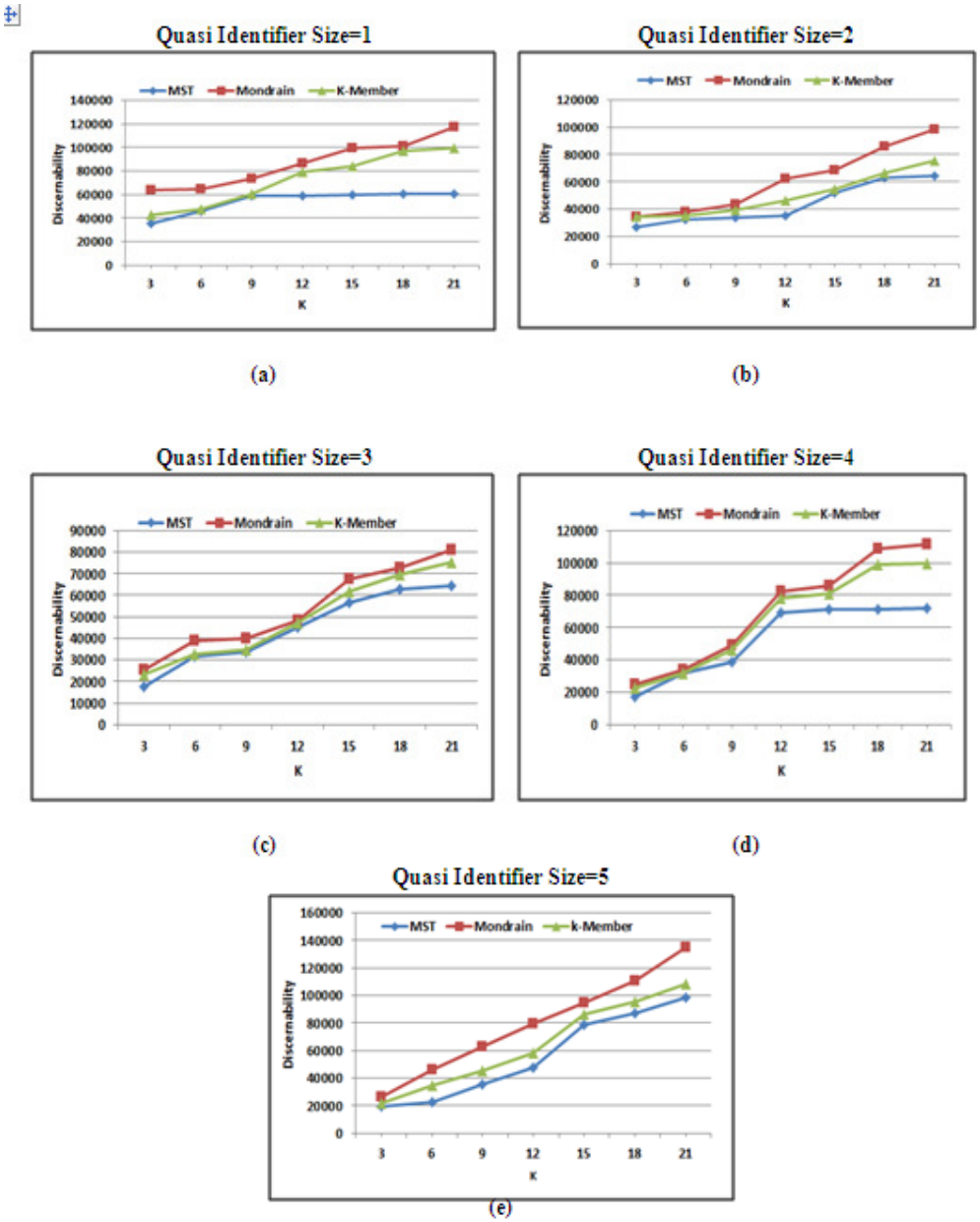

Figure. 8. Discrenability of different methods with variant of *k* (a) Quasi identifier size= 1 (b) Quasi identifier Size = 2 (c) Quasi identifier Size = 3 (d) Quasi identifier Size = 4 (e) Quasi identifier Size = 5

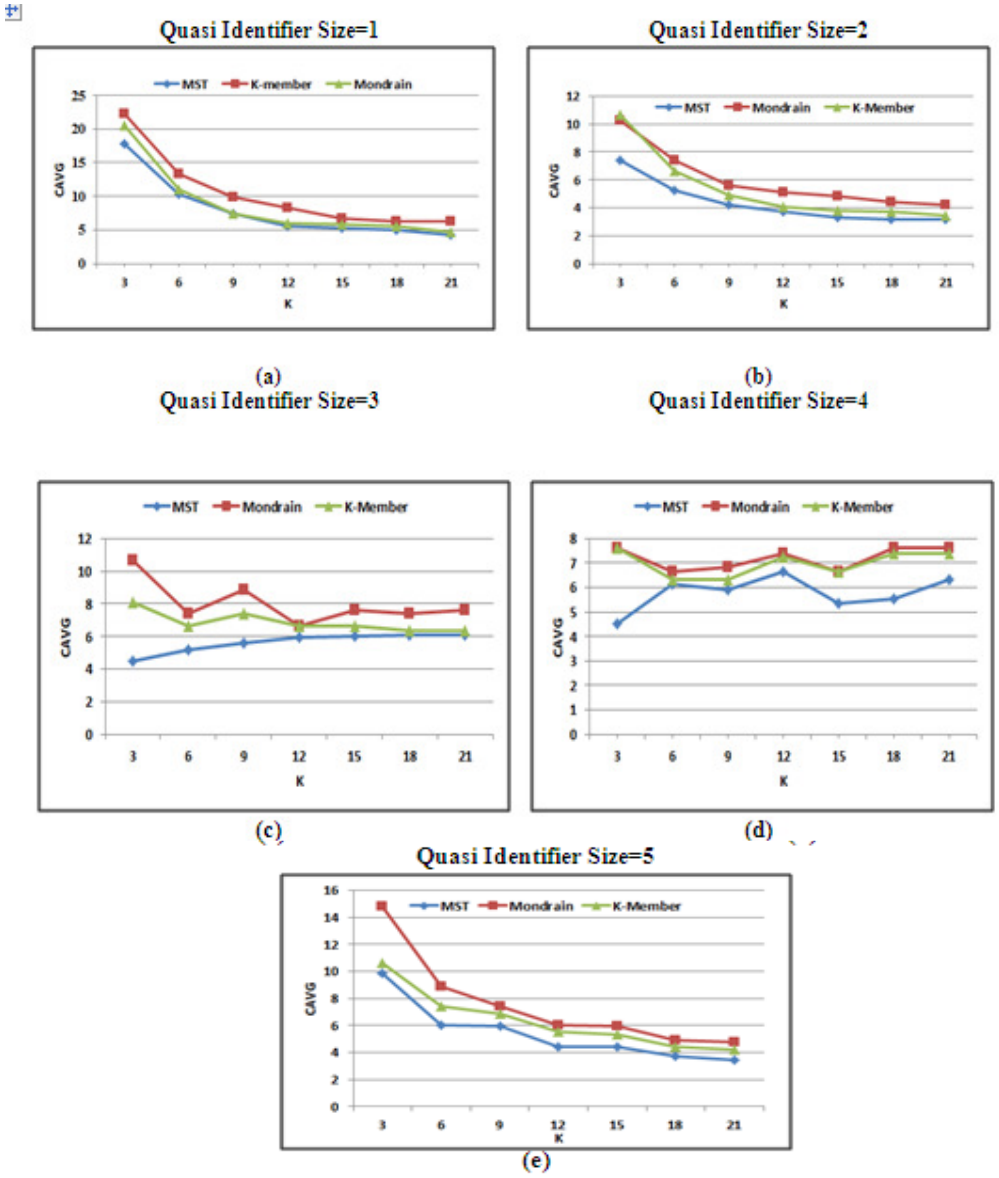

Figure 9. Normalized Average Equivalence Class (CAVG) of different methods with variant of *k*
(a) Quasi identifier size= 1 (b) Quasi identifier Size = 2 (c) Quasi identifier Size = 3
(d) Quasi identifier Size = 4 (e) Quasi identifier Size = 5

8. CONCLUSION AND FUTURE WORK

In this paper, we studied local recoding for *k*-anonymity as clustering problem and proposed Minimum Spanning Tree based partitioning approach to achieve *k*-anonymity. We defined concept hierarchical distances, which are used to form equivalence classes and also different metrics like information loss, discernability and normalized equivalence class (CAVG) are adopted for measuring the quality of the anonymized dataset. Our experiments show that our method results significantly with less information loss, less discernability and better CAVG than *k*-member clustering and Mondrian global recoding algorithms. Our approach has scalability

limitation during MST construction. In Future, we focus on improving the scalability by applying some parallel algorithm for constructing MST.

REFERENCES

[1] J.Liu, J.Luo, J. Z. Huang & L.Xiong (2012)"Privacy preserving DBSCAN clustering",In Proceedings of the ACM conference PAIS'12, Germany, pp.177-185.

[2] C. Aggarwal&P.Yu (2004) "A Condensation Approach to Privacy Preserving Data Mining", In Proceedings of the 9th International Conference on Extending Database Technology (EDBT'04), Crete, Greece, pp. 183-199.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Ranigrahy&D. Thomas (2005) "Anonymizing Tables", In Proceedings of the 10th International Conference Database Theory (ICDT'05), Edinburgh, UK, pp. 246-258.

[4] R. Aggrawal& R. Srikant(2000)"Privacy-preserving data mining", In Proceedings of the ACM SIGMOD, Vol.29 (2), pp. 439–450.

[5] R. Bayardo, &R. Agrawal(2005) "Data privacy through optimal k-anonymization", In Proceedings of the 21st International Conference on Data Engineering (ICDE'05), pp. 217–228.

[6] J. Byun, A. Kamra,E. Bertino&N. Li (2007) "Efficient k-anonymization using clustering techniques",In Proceedings of the 12th International conference on Database Systems for Advanced Applications (DASFAA-2007), pp.188-200.

[7] B. C. M. Fung, K. Wang &P.Yu(2005)"Top-down specialization for information and privacy preservation", In Proceedings of the 21stInternational Conference on Data Engineering (ICDE'05), pp. 205–216.

[8] B. C. M. Fung, K.Wang&P.Yu(2007)"Anonymizing Classification Data for Privacy Preservation", IEEE Transactions on Knowledge and Data Engineering, Vol.19 (5), pp.711- 725.

[9] V. Iyengar (2002)"Transforming data to satisfy privacy constraints", In Proceedings of 8ththe International conference on Knowledge Discovery and data mining (ACM SIGKDD'02), pp. 279-288.

[10] J. Vidya(2004) Privacy Preserving Data Mining over Vertically Partitioned Data, PhD thesis, Department of Computer Science, Purdue University.

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

[12] K. LeFevre, D. DeWitt&R. Ramakrishnan (2006)"Mondrian: multidimensional k-anonymity", In Proceedings of the 22ndInternational Conference on Data Engineering (ICDE-2006), Atlanta, GA, USA, pp. 25--.

[13] K. LeFevre, D. DeWitt& R. Ramakrishnan, R. (2005)"Incognito: Efficient full-domain k-anonymity", In Proceedings of the 31stInternational Conference on Management of Data (SIGMOD 2005), Baltimore, MA, USA, pp. 49-60.

[14] G.Aggarwal, T. Feder, K.Kenthapadi, S. Khuller, R.Panigrahy, D.Thomas&A. Zhu (2006)"Achieving Anonymity via clustering". In Proceedings of PODS-2006, pp. 153-166.

[15] G. Ghinita, P. Karras, P. Kalnis& N. Mamoulis (2007) "Fast data anonymization with low information loss", In Proceedings of the 33rdInternational conference on Very large data bases (VLDB-07), pp. 758-769.

[16] J. Xu, W. Wang, J. Pei,X.Wang, B. Shi& A. W. C. Fu (2006) "Utility- based anonymization using local recoding", In Proceedings of the 12thinternational conference on Knowledge discovery and data mining (KDD-06), pp. 785-790.

[17] L.Sweeney(2002) "k-anonymity: a model for protecting privacy",International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol.10 (5), pp.557-570.

[18] L.Sweeney(2002) "Achieving k-anonymity privacy protection using generalization and Suppression", International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10(5), pp. 571–588.

[19] H. Zhu& X. Ye (2007) "Achieving k-anonymity via a Density-Based Clustering Method", In Proceedingsof the 8thInternational conference on Web-age information management conference (WAIM-07), pp.745 -752.

[20] T. Manolis, M. Nikos, &P.Kalnis(2010) "Local and Global recoding methods for anonymizing set-valued data",VLDB Journal, Vol. 20, pp.83–106.

[21] J. Li, R. Chi-Wing Wong, A.Wai-Chee Fu& J. Pei (2008)"Anonymization by Local Recoding in Data with Attribute Hierarchical Taxonomies", IEEE Transactions on Knowledge and Data Engineering, Vol. 20,pp.1181-1194.
[22] J. Kruskal(1956) "On the shortest spanning sub tree and the travelling sales problem", In Proceedings of the American Mathematical Society, pp. 48-50.
[23] R. Prim (1957)"Shortest connection networks and some generalization", In Bell systems technical journal, pp.1389-1401.
[24] A. Frank&A. Asuncion (1998) "UCI Machine Learning Repository", [http://archive.ics.uci.edu/ml]. Irvine, CA:University of California, School of Information and Computer Science.
[25] S. C.D.Vimercati, S.Foresti, & G.Livraga.(2011)"Privacy in Data Publishing", In Proceedings of the DPM 2010 and SETOP 2010, LNCS 6514, pp. 8–21.
[26] M. Laszlo & S. Mukherjee (2005)"Minimum Spanning Tree Partitioning Algorithm for Micro-aggregation", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 7, pp.902-911.
[27] A.Amirbekyan&V. EstivillCastro(2006)"Privacy Preserving DBSCAN for Vertically Partitioned Data", In Proceedings of the international conference on intelligence and informatics, pp. 141–153.
[28] G.Loukides&Jian-Hua Shao(2008) "An Efficient Clustering Algorithm for k-Anonymisation", Journal of Computer Science and Technology, Vol. 23(2), pp. 188-202.
[29] A.Inan, V. K. Selim, Y.Saygin, E.Savas, A.AzginHintoglu& L. Albert (2006) "Privacy Preserving Clustering on Horizontally Partitioned Data", In Proceedings of the 22nd International Conference on Data Engineering Workshops(ICDEW-06), pp. 646-666.

Authors

Mr. K VenkataRamana holds an M.Tech degree in Computer Science and Technology from Andhra University, Visakhapatnam and is presently working as Assistant Professor in the department of Computer Science and Systems Engineering. His research interests include privacy issues in Data Engineering and Web Technologies.

Prof. V. Valli Kumari holds a PhD degree in Computer Science and Systems Engineering from Andhra University Engineering College, Visakhapatnam and is presently working as Professor in the same department. Her research interests include Security and privacy issues in Data Engineering, Network Security and E-Commerce. She is a member of IEEE and ACM.