Privacy Preserving Data Publishing Based on *k*-Anonymity by Categorization of Sensitive Values

Manish Sharma, Atul Chaudhary, Manish Mathuria, Shalini Chaudhary
Department of Computer Science & Information Technology, Govt. Engineering College, Ajmer

Abstract - In many organizations large amount of personal data are collected and analyzed by the data miner for the research purpose. However, the data collected may contain sensitive information which should be kept confidential. The study of Privacy-preserving data publishing (PPDP) is focus on removing privacy threats while, at the same time, preserving useful information in the released data for data mining. The number of privacy preserving data publishing techniques is proposed to protect sensitive data from the outside world. *K*-anonymity is one of the best method which is easy and efficient to achieve privacy in many data publishing applications. It has some weaknesses like data utility reduction and more information loss which need to be focus and optimize. Therefore, the main challenge of research is to minimize the information loss during anonymization process. This paper introduces a new approach for privacy preserving method which is based on categorization of sensitive attribute values. The sensitive attribute value is categorized into high sensitive class and low sensitive class. Anonymization is performed only on those tuples which belong to high sensitive class, whereas tuples belong to low sensitive class published as it is. An experimental result shows that our proposed method is efficient compare to traditional *k*-anonymity, in terms of data utility and information loss.

Index Terms - Privacy Preserving; *k*-Anonymity; Quasi Identifier; Data Utility, Sensitive Classes.

1 INTRODUCTION

Many organizations like credit card companies, real estate companies, search engines and hospitals collect and hold large volumes of data. They would like to publish the data for the purposes of data mining. When these organizations publish data it also contains a lot of sensitive information, so they would like to preserve the privacy of the individuals [1]. To protect the privacy of the individual data, data provider removes the identified key attributes like name, address SSN No, ID etc. However, inspite of removing the key attributes there is no guarantee of the anonymity. The information which is released often contains other data called as Quasi-identifiers such as, birth date, sex, and ZIP code [2], which can be linked to publicly available information to re-identify the individual, thus leaking information that was not intended for disclosure. These types of linking attack create a serious issue. There can be different ways to achieve the goal of privacy in which the releasing some limited data instead of pre-computed heuristics is an increased flexibility and availability for the users. Privacy Preserving Data Publishing [1] looks for methods to transform the original data such that heuristics determined from the transformed data is close to original heuristics and the privacy of users is not dying out. *K*-anonymity is a way to achieve this. It requires each tuple in the published table to be indistinguishable from at least *k*-1 other tuples. There are three kind of attribute in *k*- anonymity. First is key attribute which is generally the name, ID etc. and it is removed at time of releasing, second class is quasi identifier which are generally linked with publicly available database to re-identify the individual, these class contains attributes such as gender, age, post code. Third class is sensitive attribute which is used by researcher and generally published directly. The classification of attributes is shown in Table 1.

TABLE 1
CLASSIFICATION OF ATTRIBUTE IN *K*-ANONYMITY

Key Attribute	Quasi Identifier			Sensitive Attribute
Name	**Gender**	**Age**	**Zip code**	**Disease**
Aruna	Female	25	423101	Flu
Mohit	Male	27	423508	HIV+

How quasi identifiers can be used to re-identify individual using linking attack is given in example below. The two tables are given, Table 2 contains Medical data set and Table 3 contains Voter data set which is also available publically. By linking Zip code, Age and Sex of Medical table with Voter list table attacker can identify that Arjun is suffering from cancer and in this way the privacy of individual is violated. This is happened because the combination of quasi identifiers value is unique in medical data set, if release data in such a way that there is no unique combination for quasi identifiers then this type of re-identification by attacker is not possible. This can be done using anonymizing tables.

TABLE 2
DIAGNOSIS DATA SET

ID	Zip Code	Age	Gender	Disease
1	13051	21	M	Flu
2	13053	26	F	Cancer
3	13063	33	M	Viral
4	13068	37	F	HIV+
5	14851	45	M	Cancer
6	14856	48	F	Headache
7	14867	53	M	HIV+
8	14869	59	F	Viral

TABLE 3
VOTER DATA SET

Name	Zip Code	Age	Gender
Amit	13245	49	M
Rohini	13067	35	F
Santosh	14245	28	M
Sangeeta	13156	41	F
Arjun	14851	45	M
Manish	14458	33	M
Anil	16634	52	M

Traditional approach to avoiding the identification of the individual records in the published data is to removing the identifying attributes such as name, SSN No, address, ID etc.

In order to avoid linking attacks using quasi-identifiers, Sweeney [2] proposed the *k*-anonymity model, where some of the quasi-identifier fields are suppressed or generalized. A table is called *k*-anonymous, if each tuple in the published table is indistinguishable from at least *k*-1 other tuples with respect to the every set of quasi identifier attribute. Hence, there are at least *k* records that share the same combination of values of quasi identifier attribute. It's ensuring that individuals cannot be uniquely identified by linking attacks. Table 4 shows a 2-anonymous view corresponding to Table 2. The sensitive attributes (Disease) is retained without change in this example.

TABLE 4
2-ANONYMOUS VIEW OF TABLE 2

ID	Zip Code	Age	Gender	Disease
1	130**	>20	M	Flu
2	130**	>20	M	Cancer
3	130**	3*	F	Viral
4	130**	3*	F	HIV+
5	148**	>40	*	Cancer
6	148**	>40	*	Headache
7	148**	5*	Person	HIV+
8	148**	5*	Person	Viral

2 LITERATURE REVIEW

In past, many algorithms have been proposed for implementing *k*-anonymity to provide privacy preservation via generalization and suppression. Sweeney and Samarati [2] introduce *k*- anonymity in which each quasi identifier attribute domain is partitioned into set of intervals by replacing the attribute value with corresponding intervals. Here the tuples in the published table is indistinguishable from at least *k*-1 other tuples with respect to their quasi identifier. Various Models like global recording, local recording, multidimensional recording, micro aggregation and clustering were proposed to achieve *k*-anonymity principle [3], [4], [5], [6]. Model such as l-diversity proposed in 2006 by A. Machanavajjhala [7] to solve *k*-anonymity problem. It tries to put constraints on minimum number of distinct values seen within a equivalence class for any sensitive attribute, S. Venkatasubramanian in 2007 [8] present a model called *t*-closeness was introduced to overcome attacks possible on *l*-diversity like similarity attack. An enhanced *k*-anonymity model [9] was proposed by J.Li and K. Wang to protect both relationship and identification to sensitive information in order to deal with the problem of *k*- anonymity. Bayardo and Agrawal [10] proposed an optiol algorithm which focus on fully generalization of table and specialized the dataset in a minimal *k*-anonymous table. Fung et al. [11] present a top-down approach to make a table satisfied *k*-anonymous. LeFevre et al. [12] use bottom up technique in their algorithm. Pei [13] discussed the approaches for multiple constraints and incremental updates in *k*-anonymity.

However the traditional *k*-anonymity models take consider that the all values of the sensitive attributes are sensitive and need to be protected. In fact, the values which will breach individual's privacy are in the minority of the whole sensitive attribute dataset. The previous models lead to excessively generalize and more information loss in publishing data. The work presented in this paper mainly focus on generalization and anonymity of the tuples which are really sensitive and need to be preserve the privacy of individuals.

3 PROBLEM DEFINITION

Traditional *k*-anonymity model is used generalization and suppression to provide privacy in table. It takes all tuples as sensitive tuples in publishing table, but this method suffers from more information loss because all tuples are generalized or suppressed.

In our proposed method we presented an algorithm which protects individual privacy as well as only the highly sensitive values tuples should be generalized with a satisfied parameter *k*. Other tuples should not be generalized and published directly. Our objectives of this research work are:

- **Data Utility:** Data utility is an important part because if data utility is minimum than it also affects the accuracy of data mining tasks. Our goal is to eliminate the privacy leak and also increase the data utility. This is only achieved by generalizing only those tuples having most sensitive attribute values.
- **Privacy:** The research work results achieve *k* –anonymity which provides the privacy by using generalization in such a way that re-identification is not possible.
- **Minimum Information Loss:** Information loss is minimized by giving sensitive level for sensitive attribute values. Those tuples which belong to the high sensitive levels are only generalized and rest of the tuples is published as it is.

4 BASIC NOTATION

Let T {K1, K2,……,Kj, Q1, Q2,……,Qp, S} be a table. Q1, Q2,,……,Qj denote the quasi identifier specified by the administrator (application). S denotes the sensitive attribute. It contain those values which must be kept secret from people. K1, K2,….,Kj denotes the identifying/key attributes of table T, which is to be removed before releasing a table. Let t[X] denotes the value of attribute X for tuple t. |T| denote the number of records present in table T.

Let T be the initial table and T* be the released micro table. It is a *k*-anonymous table. The attribute for *k*-anonymity table are classified into three categories: Quasi Identifier, Key Attributes and Sensitive Attributes.

Definition 1 (Quasi Identifier): A set of non-sensitive attribute {Q1,…….Qp} of a table T is called quasi identifier. It is generally linked with publicly available database to re-identify the individual.

Definition 2 (Key Attribute): Key attribute consist values which is the most unique values for to identify the individuals from set S. Key attributes contain name, ID, SSN No. etc.

Definition 3 (Sensitive Value Set): A set H consist of values which is user selected as most sensitive values (high sensitive) from set S. It is denoted by H. A set L consist of values which is user selected as not so much sensitive values (low sensitive) from set S. It is denoted by L.

Definition 4 (Sensitive Tuple): Let t ∊ T, if t[S] ∊ H we called t is a sensitive tuple.

Definition 5 (*K*-Anonymity): A table T satisfy *k*-anonymity if for every tuple t of T there exist (*k*-1) other tuples ti1, ti2,…tik-1 ∊ T such that t[F] = ti1[F]= ti2[F] =……..= tik-1[F] for all F ∊ QI (Quasi Identifier).

5 SYSTEM ARCHITECTURE

Fig. 1 presents the system architecture of our proposed work. The architecture is divided into various modules as follows:

5.1 Identification of Attributes

The attributes of tables are classified in to three classes. Identification of an attribute is an important task. One need to identify the proper quasi identifier, sensitive attribute and key attributes. Unique attribute such as name, ID, SSN No. are treated as key attributes and it is removed from the published data. Quasi identifiers are the attribute which is basically used by the attacker for the linking attack. The selection of sensitive attribute is important because there is a need to anonymized only the most sensitive data to avoid the overhead and to increase the data utility. In our method key attribute is removed and quasi identifier and sensitive attribute are usually kept in the released and initial data set.

5.2 Define Sensitive Class

After identification of the sensitive attributes, it is require to define the sensitive class for different sensitive attribute values. Those sensitive attribute values which are highly sensitive and need to be anonymized is classify into high sensitive class (H) and low sensitive values are classify into low sensitive class (L). Those tuples which belong to H move into Table T1 and those table which belong to L move into table T2. After partitioning according to classes, the statistics of quasi identifier present in table T1 is measured.

It helps into generalization of table T1 and making it anonymized.

5.3 Anonymized only Most Sensitive Class Tuples

The anonymization is performed only on the most sensitive attribute values which is belong to class H. Generalization is used to perform the anonymization. Suppression is also used for anonymity but it leads to more information loss. The data is generalized by constructing the Domain Generalization Hierarchy (DGH) for the corresponding quasi identifiers. For example the DGH of age, gender and zip code are shown in Fig.2 (a), Fig.2 (b) and Fig.2 (c).

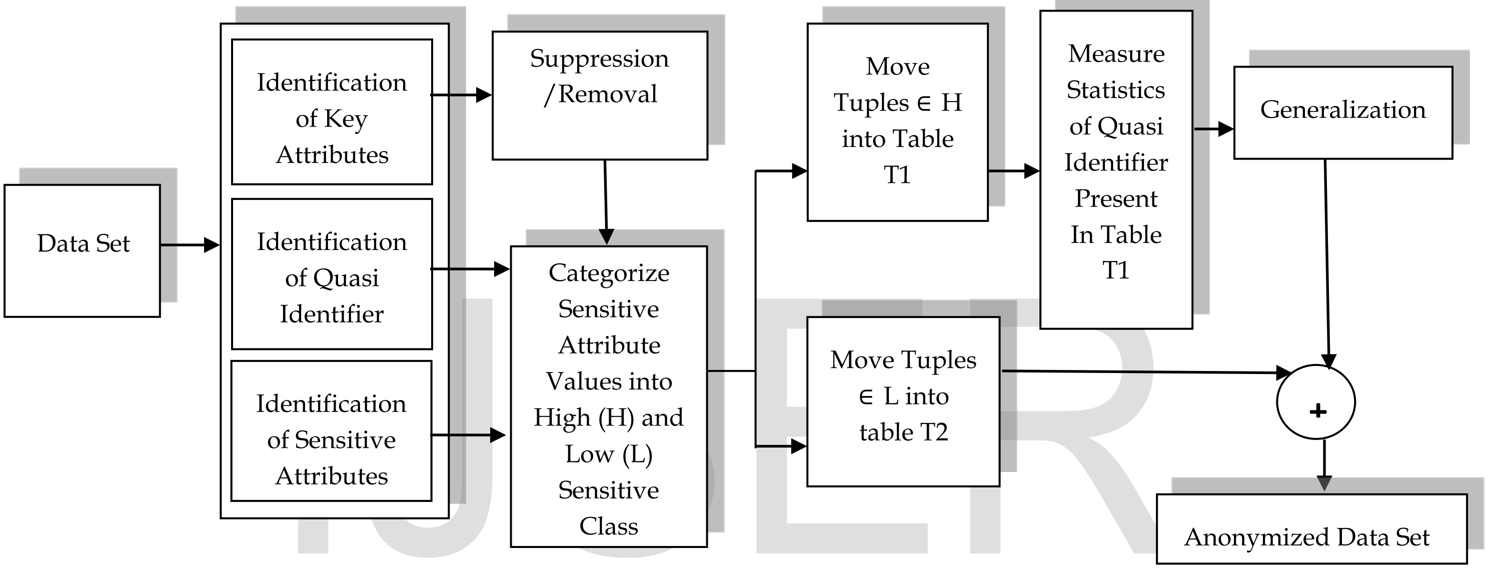
Data Set
Identification of Key Attributes
Identification of Quasi Identifier
Identification of Sensitive Attributes
Suppression /Removal
Categorize Sensitive Attribute Values into High (H) and Low (L) Sensitive Class
Move Tuples ∈ H into Table T1
Move Tuples ∈ L into table T2
Measure Statistics of Quasi Identifier Present In Table T1
Generalization
+
Anonymized Data Set

Fig. 1 System Architecture

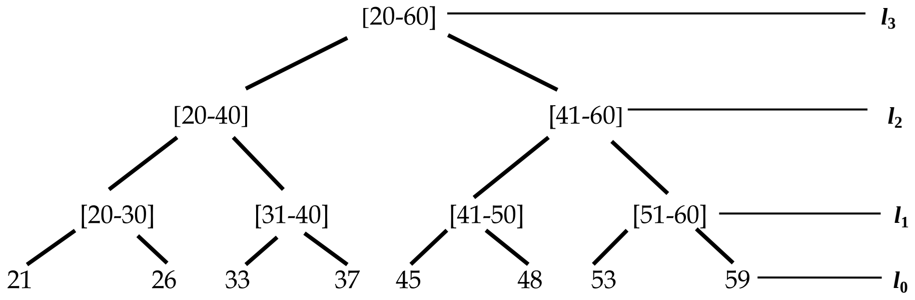

Fig 2(a) Domain Generalization Hierarchy of Age

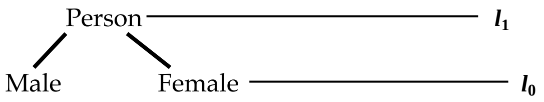

Fig.2 (b) Domain Generalization Hierarchy of Gender

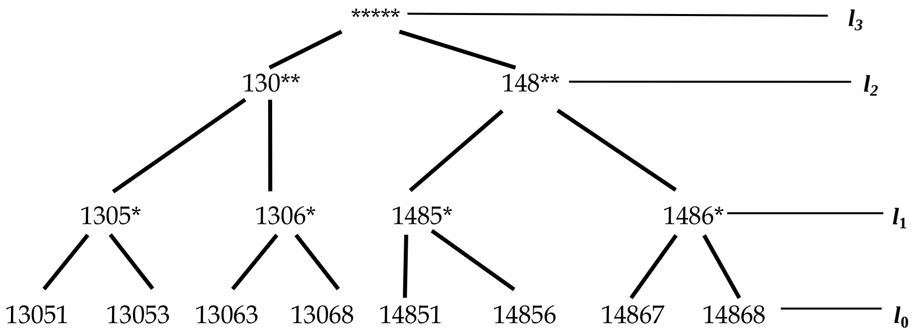

Fig. 2(c) Domain Generalization Hierarchy of Zip Code

After generalization both the high sensitive and low sensitive attribute values tables are join to get anonymized dataset. In the anonymized data set, there are multiple *k*-anonymous records where the attacker finds it difficult to find exact data of a person.

6 PROPOSED ALGORITHM

A review of the broad areas of privacy preserving data publishing and the underlying algorithm has been done. *K*-anonymity methods consider all tuples as equally sensitive and so all the tuples get anonymized which leads to more information loss. So for achieving privacy with minimum information loss and maximum data utility we need to develop a method. For this, we proposed an algorithm that overcomes above problem. The core concept of our solution is to categorize the sensitive attribute values in two classes:

- **High sensitive class:** A set of sensitive attribute values H= {s1, s2,…., sn} that are highly sensitive like HIV+ and Cancer.

- **Low sensitive class:** A set of sensitive attribute values L= {s1, s2,…., sk} that are low sensitive like Flu, Headache and Viral.

Algorithm:

Input – Table T, set of Quasi Identifier Q, Sensitive Attribute S, Key Attribute A, Anonymized parameter *k*.

Output – Anonymized table T*.

Step I: Select input table T.

Step II: Select quasi identifier (Q), Sensitive attribute (S), Key attribute (A) from the table T.

Step III: Remove/ suppress the key attribute.

Step IV: Categorize the sensitive attribute values into two classes H and L.

Step V: For each tuple whose sensitive value belongs to class H i.e. if t[S] ∈ H:

Move these tuples into table T1 and apply generalization on quasi attributes to anonymized it.

Step VI: For each tuple whose sensitive value belongs to class L i.e. if t[S] ∈ L:

Move these tuples into table T2 and do not apply generalization on it.

Step VII: Append rows of table T1 and T2.

T*= T1+T2.

Table 5 is the outcome table after applying traditional *k*-anonymity and table 6 is the table T* after applying our proposed algorithm on table II. Anonymity parameter value is 2. Sensitive values like HIV+ and Cancer are selected as high sensitive values and tuples belonging to those values are moved to table T1 and generalization is applied on quasi attribute zip code, age and gender to anonymize those tuples. Sensitive values like Flu, Viral and Headache are selected as low sensitive values and they released as it is.

By comparing these two outputs we can easily see that in traditional *k*-anonymity, information loss is more than compared to our algorithm.

Table 5
Traditional *k*-anonymity applied on Table 2
(when *k* = 2)

Zip Code	Age	Gender	Disease
130**	[20-30]	*	Flu
130**	[20-30]	*	Cancer
130**	[31-40]	Person	Viral
130**	[31-40]	Person	HIV+
148**	[41-50]	*	Cancer
148**	[41-50]	*	Headache
148**	[51-60]	Person	HIV+
148**	[51-60]	Person	Viral

Table 6
Proposed Algorithm Applied on Table 2
(when *k* = 2)

Zip Code	Age	Gender	Disease
130**	[20-40]	Person	Cancer
130**	[20-40]	Person	HIV+
13051	21	M	Flu
13063	33	M	Viral
14856	48	F	Headache
14869	59	F	Viral
148**	[41-60]	Person	Cancer
148**	[41-60]	Person	HIV+

7 QUALITY MEASUREMENT OF ANONYMIZATION

Protecting privacy of the data is achieved through anonymization. The other aspect of privacy is to produce the anonymized data that must be useful for deriving useful patterns for statistical analysis, i.e., published data should remain for practical use. There are various types of categories of information metrics for measuring the data utility. Data metric measures the data quality of the entire anonymous table with respect to the data quality in the original table. Quality metric is basically used to identify an anonymous table with maximum utility or minimum information loss. Several quality measure such as minimal distortion [2, 14], Information loss [15], Discernibility metric [10, 17], classification metric [16] and normalized equivalence class metric [17] were widely used.

This paper used Information Loss and Discernibility Metric for measuring the quality of anonymized table.

7.1 Information Loss

Information loss caused by the anonymization can be measured by how well the generalized tuple approximate the original one. After generalization some attribute values of a tuple are generalized to an interval. To measure the utility of attributes in the anonymization, weighted normalized certainty penalty (NCP) is proposed [15]. For a numerical attribute the NCP measures it's normalize interval size after generalization and for a categorical value, NCP measures its normalized number of descendants in the hierarchy tree after generalization. Some weight is assigned to each attribute to reflect its usefulness in the analysis on the anonymized data. The weighted normalized certainty penalty should be minimized.

For the numerical attribute, consider a table T with quasi identifier (A1……….An). Suppose a tuple t = (x1……….xn) is generalized to tuple t' = ([y1,z1],………[yn,zn]) such that yi ≤ xi ≤ zi (1≤ i≤n). Then we define normalized certainty penalty (NCP) of tuple t on an attribute Ai as:

$$NCP_{A_i}(\mathrm{t}) = \frac{z_i - y_i}{|A_i|} \qquad (1)$$

$$where|A_i| = \max_{t \in T} t.A_i - \min_{t \in T} t.A_i$$

Hierarchical tree are used for the generalization in categorical attribute. Attribute values of different granularity are specified by the hierarchical tree. Suppose a tuple t has value v on categorical attribute Ai, than it is generalized to a set of values v1,…….vm. Common ancestor of v1,…..,vm denoted by ancestor (v1,……,vm) in the hierarchy tree is found, and use the size of ancestor (v1,…..,vm), that is, the number of leaf nodes that are descendants of ancestor (v1,……,vm), to measure the generalization in categorical attribute. It is defined as:

$$NCP_{A_i}(t) = \frac{|ancestor\ (v_1 \ldots\ldots v_m)|}{|A_i|} \qquad (2)$$

where |Ai| is denoted the number of distinct value on Ai in the most specific level.

Consider both the numerical and categorical attributes, we define the weighted normalized certainly penalty of a tuple t as

$$\mathrm{NCP(t)} = \sum_{i=1}^{n}(w_i . \mathrm{NCP}_{A_i}(t)) \qquad (3)$$

$$\text{Where } \sum_{i=1}^{n} w_i = 1.$$

7.2 Discernibility

Discernibility metric [10, 17] measures the number of tuples that are not different from each other. Each tuple in an equivalence E incurs a cost |E|. It is equal to the sum of the squares of the sizes of the equivalence classes.

$$\mathrm{DM} = \sum_{\text{Equivalance classes E}} E^2 \qquad (4)$$

The objective of anonymization is to minimizing discernibility cost.

8 EXPERIMENTAL RESULTS

The experiment is implemented in JSP and MySQL 5.5 and run on 2.3 Ghz Intel core i3 processor with 2 GB RAM. The window 8.1 is used as an operating system. An Adult dataset from the UCI Machine Learning Repository was used [18]. The adult dataset is publically available dataset which is standard dataset for checking the performance of *k*-anonymity algorithm. Adult dataset contain 32561 tuples from US census data. After preprocessing and removing the missing values tuples 30162 tuples are selected. Initially we consider only 1000 tuples for the experimental purpose. This dataset contain 11 attributes and we retain only 5 attributes such as Age, Race, Marital, Gender and Occupation.

TABLE 7
DELINEATION OF ADULT DATA SET

Attribute	Type	Distinct	Generalization	Tree Height
Age	Numeric	66	10-, 20-, 30-	4
Gender	Categorical	2	Person	1
Race	Categorical	5	Taxonomy Tree	2
Marital	Categorical	6	Taxonomy Tree	3
Occupation	Categorical	14	Sensitive Attribute	

Age, Race, Marital and Gender are consider as quasi-identifier and occupation is consider as sensitive attribute. Among all the sensitive attribute values, “Tech-support” and “Sales” are consider as most sensitive values which need to be protected. These values include in high sensitive class and other values included in low sensitive class. The corresponding height of the chosen attribute and their type and number of distinct value are shown in Table 7.

Proposed method is compared with the traditional *k*-anonymity method. Comparison is done on the basis of information loss and discernibility metric showed in equation (3) and (4).

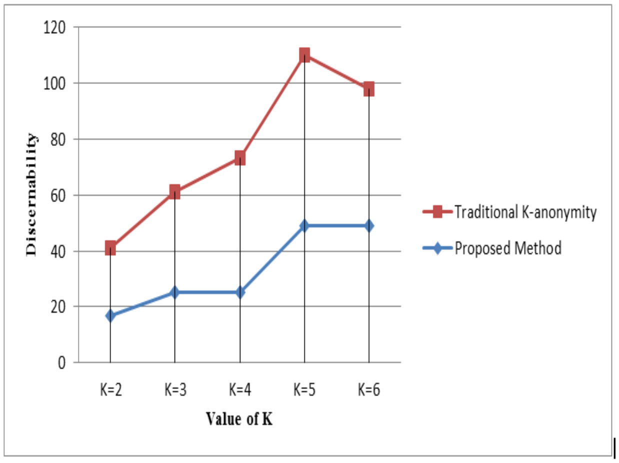

Fig. 3 (a) shown above presents how discernibility metric (DM) differs for both traditional *k* anonymity and the proposed method in this paper. As it is known that discernibility should be minimized, so it is found that proposed method achieves better discernibility for different values of *k*.

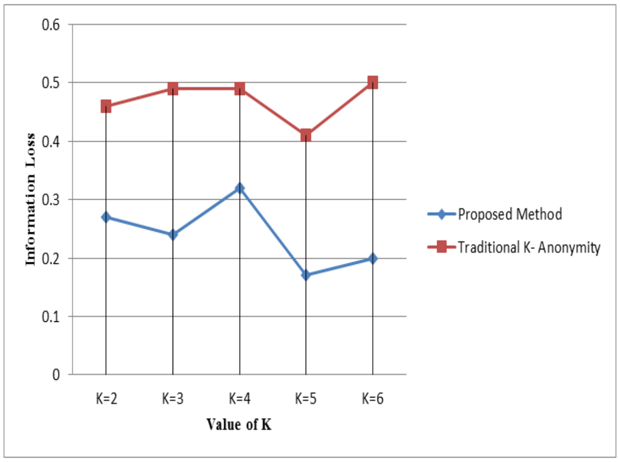

Fig. 3 (b) shown above presents the information loss with the increase in the value of k in both traditional *k*-anonymity and the proposed approach. The proposed method shows a significant improvement by reducing the information loss when compared to existing approach.

9 CONCLUSION

Privacy preserving is growing field of research. The publication of data (micro data) of any individual without revealing their private or sensitive information is a very important problem. Many organizations publish their data for the mining purpose, so they would like to preserve the sensitive data before sending to mining operations.

K-anonymity privacy preserving technique has been proposed for protecting privacy in data publishing, but it consider all sensitive attribute at same level and apply generalization on all. This leads to some issue like:

- Information Loss
- Data Utility
- Privacy Measure

So further modification with existing scheme of *k*-anonymity is done which provide the privacy with minimum information loss and maximum data utility. This paper presents a new approach based on categorization of sensitive attribute values in different classes which results information loss is reduced as only most sensitive tuples are anonymized, data utility is increased and privacy of individual is also preserved.

10 REFERENCES

[1] Yan Zhao, Ming Du, Jiajin Le, Yongcheng Luo, "A Survey on Privacy Preserving Approaches in Data Publishing", First International Workshop on Database Technology and Applications, IEEE 2009.

[2] L. Sweeney, "k-anonymity: A model for protecting privacy", International Journal on Uncertainty Fuzziness Knowledge Based System, 10 (5), PP. 557-570, 2002.

[3] Xu, J. Wang, W. Pei, X. Shi, A.W.C, 2006. "Utility based anonymization using local recoding". In proceeding of the SIGKDD'06, PP. 785-790.

[4] LFevre, K; David, J; Dewitt and Ramakrishnan; R. "Multidimensional k-anonymity", In proceeding of the 22nd International Conference on Data Engineering (ICDE' 06), Washington, USA, 2005.

[5] Agarwal G., Feder T., Krishnaram K., Samir K., Rina Panigrahy and Zhu. "Achieving Anonymity via Clustering", PODS'06, Chicago, Illinois, USA.

[6] Michael L. and Mukharjee S. "Minimum Spanning Tree Partitioning Algorithm For Micro Aggregation", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 7, PP. 902-911, 2005.

[7] A. Machanavajjhala, J. Gehrke, D. Kifer and M. Venkatasubramaniam, "l-diversity: Privacy beyond k-anonymity" In proceeding of the IEEE ICDE 2006.

[8] N. Li, T. Li, S. Venkatasubramaniam. "t-closeness: Privacy beyond k-anonymity and l-diversity", ICDE 2007: 106-115.

[9] R.Wang, J. Li, A. Fu, K. Wang. "(α,k) anonymity: An enhanced k-anonymity model for privacy preserving data publishing", KDD 2006: 754-759.

[10] R. Bayerdo and R. Agrawal. "Data privacy through optimal k-anonymity". In proceeding of the 21st International

Conference on Data Engineering (ICDE), PP: 217-228, Tokyo, Japan, 2008.

[11] B. Fung, K. Wang, P. Yu. "Top Down Specialization" For Information Conference on Data Engineering (ICDE'05), PP: 205-216.

[12] LeFevere, Dewitt et. al. "Incognito: Efficient Full Domain k-anonymity". Proc. ACM, SIGMOD, International Conference on Management of Data.

[13] J. Pei, J. Xu, W. Wang, K. Wang. "Maintaining k-anonymity against incremental updates". Proceeding of the 19th International Conference on Scientific and Statistical Database.

[14] Samarati, P. 2001. "Protecting respondent‟ s identities in microdata Release", In IEEE Transactions on Knowledge and Data Engineering, Vol.13 (6). pp. 1010-1027.

[15] Gabriel, G., Panagiotis, K., Panos, K., and Nikos, M. 2009. "A Framework for Efficient Data Anonymization under Privacy and Accuracy Constraints", ACM Transactions on Database Systems, Vol. 34, No. 2, Article 9, Publication date: June 2009.

[16] Iyengar, V. 2002. "Transforming data to satisfy privacy constraints", In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), Alberta, Canada.

[17] Jiuyong, Li., Wong, R.C.W., Fu, A.W.C., and Jian Pei. 2008. "Anonymization by Local Recoding in Data with Attribute Hierarchical Taxonomies", IEEE Transactions On Knowledge and Data Engineering, Vol. 20, pp.1181-1194.

[18] Newman, D.J., Hettich, S., Blake, C.L., and Merz, C.J. 1998. UCI Repository of Machine Learningdatabases,http://www.ics.uci.edu/~mlearn/MLRepository.html.