Volume 4, No. 2, Jan-Feb 2013

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Modified KACTUS Algorithm Based Multi-Dimensional Suppression for K-Anonymity

A.S.Loganayaki*
Research Scholar, dept of Computer Science
Gobi Arts & Science College (Autonomous),
Gobichettipalayam. India
loganayakias@gmail.com

Dr.B.Srinivasan
Associate Professor, dept of Computer Science
Gobi Arts & Science College (Autonomous),
Gobichettipalayam. India
srinivasan_gasc@yahoo.com

Mr.P.Narendran
Head of the Department, dept of Computer Science
Gobi Arts & Science College (Autonomous),
Gobichettipalayam, India
narendranp@gmail.com

Abstract: Data mining is the process of extracting hidden information from database. The current trend in business collaboration shares the data and mined results to gain mutual benefit. The problem of privacy-preserving data mining has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information. Two common manipulation techniques used to achieve k-anonymity of a dataset are generalization and suppression. K-Anonymity of Classification Trees Using Suppression (kACTUS) is observed to provide good results in achieving k-anonymity. In KACTUS efficient multidimensional suppression is performed, that is values are suppressed only on certain records depending on other attribute values, without the need for manually-produced domain hierarchy trees. The k-anonymity models is extended by providing new definitions and use several anonymization techniques together in order to get better results in terms of accuracy than reported in the literature.

Keywords: KACTUS 2; k-anonymity; privacy preserving; decision tree; computational complexity

I. INTRODUCTION

Data mining and knowledge discovery in databases have been attracting a significant amount of research, industry, and media attention of late. Across a wide variety of fields, data are being collected and accumulated at a dramatic pace. There is an urgent need for a new generation of computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data.

Data mining is emerging as one of the key features of many business organizations. Privacy concerns over the ever-increasing gathering of personal information by various institutions led to the development of privacy preserving data mining. One way to enable effective data mining while preserving privacy is to anonymize the dataset that include private information about subjects before being released for data mining. One way to anonymize dataset is to manipulate its content so that the records adhere to k-anonymity**.**

These theories and tools are the subject of the emerging field of knowledge discovery in database (KDD). At an abstract level, the KDD field is concerned with the development of methods and techniques for making sense of data. The basic problem addressed by the KDD process is one of mapping low-level data (which are typically too voluminous to understand and digest easily) into other forms that might be more compact (for example, a short report), more abstract (for example, a descriptive approximation or model of the process that generated the data), or more useful (for example, a predictive model for estimating the value of future cases). At the core of the process is the application of specific data-mining methods for pattern discovery and extraction [1, 5].

Knowledge Discovery in Databases (KDD) is an automatic, exploratory analysis and modeling of large data repositories. KDD is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets [7].

A large degree of the current interest in KDD is the result of the media interest surrounding successful KDD applications, for example, the focus articles within the last two years in Business Week, Newsweek, Byte, PC Week, and other large-circulation periodicals. Unfortunately, it is not always easy to separate fact from media hype. Nonetheless, several well documented examples of successful systems can rightly be referred to as KDD applications and have been deployed in operational use on large-scale real-world problems in science and in business [3].

Data mining has emerged as a key tool for a wide variety of applications, ranging from national security to market analysis. Many of these applications involve mining data that include private and sensitive information about users. The private data are susceptible to theft by the hackers. To avoid such situations privacy regulations were promulgated in many countries (e.g., privacy regulation as part of HIPAA1 in the USA). The data owner is required to omit identifying data so that to assure, with high probability, that private information about individuals cannot be inferred from the dataset that are

© 2010, IJARCS All Rights Reserved

released for analysis or sent to another data owner. At the same time, omitting important fields from datasets, such as age in a medical domain, might reduce the accuracy of the model derived from the data by the DM process. The HIPAA privacy rules have affected significantly their ability to perform retrospective, chart-based research [8, 10].

Privacy-preserving data mining (PPDM) deals with the trade-off between the effectiveness of the mining process and privacy of the subjects, aiming at minimizing the privacy exposure with minimal effect on mining results. A dataset complies with k-anonymity protection if each individual's record stored in the released dataset cannot be distinguished from at least k-1 individuals whose data also appears in the dataset. This protection guarantees that the probability of identification an individual based on the released data in the dataset does not exceed 1/k. Generalization and suppression are the most common methods used for de-identification of the data in k-anonymity based algorithms.

In this paper kACTUS2 – Supervised Decision Tree-based K-Anonymity, a new algorithm that de-identifies (anonymizes) datasets so that to assure high degree of user's privacy when data-mining is applied, while having minimal impact on accuracy of data-mining results. The privacy of the users is measured by the compliant of the dataset to k-anonymity. KACTUS was specifically designed to support classification, but can be extended to support other data-mining methods.

II. LITERATURE SURVEY

This paper explores the possibility of using multiplicative random projection matrices for privacy preserving distributed data mining. It specifically considers the problem of computing statistical aggregates like the inner product matrix, correlation coefficient matrix, and Euclidean distance matrix from distributed privacy sensitive data possibly owned by multiple parties [6]. This class of problems is directly related to many other data-mining problems such as clustering, principal component analysis, and classification.

This paper makes primary contributions on two different grounds. First, it explores independent component analysis as a possible tool for breaching privacy in deterministic multiplicative perturbation-based models such as random orthogonal transformation and random rotation. Then, it proposes an approximate random projection-based technique to improve the level of privacy protection while still preserving certain statistical characteristics of the data. The paper presents extensive theoretical analysis and experimental results. Experiments demonstrate that the proposed technique is effective and can be successfully used for different types of privacy-preserving data mining applications.

Maintaining data mining accuracy on distorted datasets is an important issue in privacy preserving data mining. Using matrix approximation, then propose several efficient and flexible techniques to address this issue, and utilize unique characteristics of matrix factorization to maintain data pattern. Then support vector machine classification to compare accuracy maintenance after data distortion by different methods. With better performance than some classical data perturbation approaches, nonnegative matrix factorization and singular value decomposition are considered to be promising techniques for privacy preserving data mining [8]. Experimental results demonstrate that mining accuracy on the distorted data used these methods is almost as good as that on the original data, with added property of privacy preservation. It indicates that the matrix factorization-based data distortion schemes perturb only confidential attributes to meet privacy requirements while preserving general data pattern for knowledge extraction.

Privacy Preserving Data Mining (PPDM) addresses the problem of developing accurate models about aggregated data without access to precise information in individual data record. A widely studied perturbation-based PPDM approach introduces random perturbation to individual values to preserve privacy before data are published. Previous solutions of this approach are limited in their tacit assumption of single-level trust on data miners. The assumption and expand the scope of perturbation-based PPDM to Multilevel Trust (MLT-PPDM).

The more trusted a data miner is the less perturbed copy of the data it can access. Under this setting, a malicious data miner may have access to differently perturbed copies of the same data through various means, and may combine these diverse copies to jointly infer additional information about the original data that the data owner does not intend to release. Preventing such diversity attacks is the key challenge of providing MLT-PPDM services. To address this challenge by properly correlating perturbation across copies at different trust levels. Then prove that the solution is robust against diversity attacks with respect to privacy goal. That is, for data miners have access to an arbitrary collection of the perturbed copies, the solution prevent them from jointly reconstructing the original data more accurately than the best effort using any individual copy in the collection. The solution allows a data owner to generate perturbed copies of its data for arbitrary trust levels on-demand. This feature offers data owner's maximum flexibility.

III. PROBLEM FORMULATION

In this section several basic definitions to be used later in the paper are introduced, and the problem formulation is presented.

In a typical classification problem, a training set of labeled examples is given. The training set can be described in a variety of languages, most frequently, as a collection of records that may contain duplicates (also known as bag). A vector of attribute values describes each record. The notation A denotes the set of input attributes containing n attributes: $A = \{a1,...,ai,...,an\}$, and y represents the class variable or the target attribute. Attributes (sometimes referred to as features) are typically one of two types: categorical (values are members of a given set), or numeric (values are real numbers). When the attribute ai is categorical, it is useful to denote its domain values by $(dom)a_i$. Numeric attributes have infinite cardinalities. The instance space X (the set of all possible examples) is defined as a Cartesian product of all the input attribute domains: $X = dom(a_1) \times dom(a_2) \times dom(a_n)$

© 2010, IJARCS All Rights Reserved

The universal instance space (or the labeled instance space) *U* is defined as a Cartesian product of all input attribute domains and the target attribute domain, that is $U = X \times dom(y)$. The training set consists of a set of m records and is denoted as *S= (*<x1, *y1*>,…,<xm, *ym*>*)*. Usually, the training set records are distributed randomly and independently according to some fixed and unknown joint probability distribution *D* over *U*.

It is assumed that a given inducer *I* is used to build a classifier (also known as a classification model) by learning from *S*. The classifier can then be used for classifying unlabelled instances. The notation *I(S)* represents a classifier which was induced by training *(I)* with dataset *S*.

IV. METHODOLOGY

A. The k-anonymity protocol:

Given a population of entities *E*, an entity-specific table with input feature set $A = \{a_1, a_2,..., a_n\}$, *Q* is *quasi identifier* of *S*. The formulation defines a *quasi-identifier* as a set of features whose associated values may be useful for linking to re-identify the entity that is the subject of the data [10]. A dataset *S* and the quasi-identifier *Q* associated with it is said to satisfy *k-anonymity* if and only if each tuple in e (p *Q*(*S*))appears with at least k occurrences in p *Q*(*S*).

The bag *S* represents the Adult dataset from the UC Irvine Machine Learning Repository. This dataset contains census data and has become a commonly used benchmark for k-anonymity. The Adult dataset has 6 continuous attributes and 8 categorical attributes. The class attribute is income level, with two possible values, <=50K or >50K. In this dataset *Q=(age, workclass, fnlwgt, edu, edu-nun, marital-status, occupation, relationship, race, sex, native-country)* is a quasi-identifier since the values of these attributes can be linked to identify an individual. As in previous PPDM studies, the set of quasi-identifiers is provided by the user, and that there is only one set of quasi-identifiers. Examples of several records in the Adult dataset are presented below:

For example, assume k=2. The dataset described bellow does not satisfy k-anonymity requirements for Q= (age, workclass, fnlwgt, edu, edu-nun, marital-status, occupation, relationship, race, sex, native-country), the k-anonymity restriction because there are three records with the same values for the quasi-identifiers (k=2<3). However, the remaining records are unique, and thus do not comply with the k-anonymity restriction (k=2>1).

B. Supervised Decision Tree-based K-Anonymity:

The kACTUS algorithm is presented kACTUS consists of two main phases: In the first phase, a classification tree is induced from the original dataset, in the second the classification tree is used by a new algorithm developed in this study to k-anonymize the dataset.

C. Phase 1: Deriving the Classification Tree:

In this phase are employing a decision tree inducer (denoted by *CTI*) to generate a decision tree denoted by *CT*. The tree can be derived using various inducers. To concentrate on top-down univariate inducers which are considered the most popular decision tree inducers and include the well-known algorithms C4.5. Top down inducers are greedy by nature and construct the decision tree in a top-down recursive manner (also known as divide and conquer). Univariate means that the internal nodes are split according to the value of a single attribute. The decision tree is trained over the projection of the quasi-identifiers. The wrapped inducer *CTI* should be differentiated from the target inducer *I*. Inducer *I* is applied on the anonymous dataset, (that is after applying k-anonymization process). The aim of the *CTI* is to reveal which quasi-identifier is more relevant for predicting the class value.

Any internal node (non-leaf) with less than k instances cannot be used by itself for generating the anonymous dataset. Thus, even if such a node is provided in the classification tree it can be pruned in advance. In many decision trees inducers, such as C4.5, the user can control the tree growing process by setting the algorithm's parameters. Specifically the parameter MinObj ("minimum number of instances") indicates the number of instances that should be associated with a node in order it to be considered for splitting. By setting MinObj to *k*, one ensures that there are no non-complying internal-nodes that are needed to be pruned. Thus with this parameter setting can reduce the tree size without sacrificing the accuracy performance of the k-anonymous procedure. Still in Phase 2 described next no assumption regarding the internal nodes is made.

D. Phase 2: K-Anonymity Process:

In this phase to use the classification tree, that was created in the first phase to generate the anonymous dataset. To assume that the classification tree complies with the following properties:

a. The classification tree is univariate that is each internal node in the tree refers to exactly one attribute.
b. All internal nodes refer to a quasi-identifier attributes. This is true because the decision tree was trained over the projection of the quasi-identifier set.
c. Assuming a top-down inducer, the attributes are sorted (from left to right) according to their significance for predicting the class (where the right-most relates to the least significant attribute).
d. Complete Coverage: Each instance is associated with exactly one path from root to leaf. In the next phase utilize these properties for the k-anonymity process. Given a tree *CT* and node *v*, we define the following functions and procedures. Because these functions are straightforward they are used here without providing pseudo-code.
   (a). root(CT) – returns the root node of CT
   (b). parent(v) - returns the parent of v
   (c). height(v) – returns the height (length of longest path from that node to a leaf.) of v.
   (d). children(v)– returns the set of immediate descendants of v.
   (e). ant(v) – returns the antecedent associated with node v.
   (f). prune(CT, v) – prunes all descendants of v from the tree CT.

© 2010, IJARCS All Rights Reserved

(g). randomSelect(k, S) – return k randomly selected instances from relation S.

The supervised k-anonymity process is described in procedure *Anonymize*. The input to the *Anonymize* procedure includes the original dataset S, quasi-identifier set Q, classification tree *CT*, and the anonymity threshold *k*. The output of the algorithm is a k-anonymous dataset denoted by *S'*. For the sake of simplicity first assume that all quasi-identifiers are categorical.

Contrary to pruning procedures, the proposed algorithm may prune every path differently. Thus, instead of pruning all branches from a certain point to the leaves, some paths remain as-is (if they comply with k-anonymity) while other are pruned. Moreover instances that are associated with the same leaf can be pruned differently, because some may be randomly chosen to help their non-complying siblings [9].

Finally, The left with the root of the tree and examine whether the number of instances left with the root node comply with k. If such a situation occurs, then suppress them and then move them to the anonymous dataset. Note that if all attributes are quasi-identifiers (Q =A), to copy these instances to the anonymous dataset, but not before suppressing all input attributes. Thus these instances are left only with the target attribute. Still some induction algorithms can benefit from these empty instances, as it provides additional information regarding the class distribution. However, if the remaining instances do not comply with k, thus it is obvious that the new anonymous dataset may contain fewer instances than the original one, but experiments show that the number of removed instances is very small compared to the total number of instances [2].

E. kACTUS Properties:

Corollary 1*:* The kACTUS algorithm is correct

Proof: In order to prove the correctness of the algorithm, the dataset S' complies with k-anonymity. However this can be easily verified because just before calling the suppress procedure; we verify the k threshold.

Corollary 2*:* The computational complexity of kACTUS algorithm overhead is linearly related to the number of instances.

Proof: To find the computational overhead incurred by the kACTUS, in addition to the complexity of the decision tree inducer *CTI* (that is the complexity of the Anonymize procedure).

To assume that are given a tree structure such that each leaf holds the pointers to the instances complying with this leaf (that is complying with all the tests in the corresponding path from the root to the leaf). Note that all selection operations in kACTUS are performed only on the leaves. The number of iterations is bounded by the number of internal nodes, which cannot exceed the number of instances (m). Each iteration of the outer loop handles a single node. The number of children associated with the internal node is maximum dmax, which represent the largest attribute domain. Regarding the operations on the instances,

a. Summing up all instances suppression, we maximally manipulate m instances, each with $|Q|$ suppressions and $|A|$-$|Q|$values duplications. Thus the suppression operations end up with O (m$|A|$).
b. Each instance with $|A|$ attributes is added to an existing bag not more than $|Q|$ times. Note that the longest path is bounded by $|Q|$ because the tree is using only the quasi-identifiers. Moreover some instances may left the next tree level. This totally ends up with O (m$|A||Q|$) [4].

F. Proposed KACTUS 2 Algorithm:

The privacy preservation and data mining problems in terms of classification, to propose an algorithm for privacy preserving data mining that performs dataset anonymization using the k-anonymity model while taking into account its effect on classification results. To extend the k-anonymity model by providing new definitions and use several anonymization techniques together in order to get better results in terms of accuracy than reported in the literature.

K-anonymity is the method used for masking sensitive data which successfully solves the problem of re-linking of data with an external source and makes it difficult to re-identify the individual. Thus anonymity works on a set of quasi-identifiers (public sensitive attributes), whose possible availability and linking is anticipated from external dataset, and demands that the released dataset will contain at least k records for every possible quasi-identifier value.

Another aspect of k is its capability of maintaining the truthfulness of the released data (unlike other existing methods). This is achieved by generalization, a primary technique in k-anonymity. Generalization consists of generalizing attribute values and substituting them with semantically consistent but less precise values. When the substituted value doesn't preserve semantic validity the technique is called suppression which is a private case of generalization. Then present a hybrid approach called compensation which is based on suppression and swapping for achieving privacy. Since swapping decreases the truthfulness of attribute values there is a tradeoff between level of swapping (information truthfulness) and suppression (information loss) incorporated in our algorithm.

K-anonymity is exploring the issue of anonymity preservation. Since do not use generalization, and then do not need a priori knowledge of attribute semantics. Then investigate data anonymization in the context of classification and use tree properties to satisfy k-anonymization. Our work improves previous approaches by increasing classification accuracy.

KACTUS-2 receives a decision tree as input and works with its decision nodes. The algorithm does not use information about the value of the target node directly from the decision tree, thus when a term leaf node will be further encountered it will denote only that the node is a decision node which have zero child nodes.

Since the algorithm uses several helper functions which are quite straightforward, then don't provide their pseudo-code. However the general descriptions and explanations are presented below.

a. ***Root*** - returns the root node of CT
b. ***Height*** - returns the height (length of the longest path from the node to the leaf)
c. ***Parent*** - returns parent of the node

© 2010, IJARCS All Rights Reserved

- ***d.*** ***Count-Instances*** - counts instances in the dataset associated with a particular node
- ***e.*** ***Move-Complying-Node*** - moves instances associated with the complying node to the anonymized dataset with non-quasi-identifiers of the original instance if any.
- ***f.*** ***Remove-Leaf-Nodes*** - remove leaf nodes of a node
- ***g.*** ***Get-Total-Instance-Count*** - counts the total number of instances associated with all the nodes in the set
- ***h.*** ***Move-Root-Non-Complying-Nodes*** - moves the instances associated with root node first by suppressing all quasi-identifiers (the root node contains only one quasi-identifier) and keeps only the target class value of the original instance along with non-quasi-identifiers of the instance,
- ***i.*** ***Get-Non-Complying-Leafs*** - given a node, returns all leafs which don't comply with k-anonymity.
- ***j.*** ***Get-Complying-Leafs*** - given a node, returns all leafs which comply with k-anonymity.
- ***k.*** ***Move-Instances*** - like move-complying-node but takes set of nodes as an input.
- ***l.*** ***Calculate-final-Compliant-Entropy***-The explanation is given further in this section.
- ***m.*** ***Swap-From-Complying*** - performs swapping of the attribute value of the complying leaf node to the attribute value of the non-complying leaf node and swapping of the class value required by the non-complying node to keep its entropy at a low level.
- ***n.*** ***Move-Non-Complying-Instances***- like move-instances but before moving such instances, the attribute of the leaf node is removed from all the instances (suppressed).
- ***o.*** ***Compensate-From-Complying***– The algorithm requires the following input parameters:

a. k-anonymity threshold KT
b. swapping threshold ST
c. original dataset OD
d. classification tree CT
e. set of non-quasi-identifiers

In this research to presented a new method of using k-anonymity for preserving privacy in classification tasks. The proposed method requires no prior knowledge and can be used by any inducer.

V. EXPERIMENTAL RESULTS

The privacy preserving classification algorithms are usually evaluated only on the Adult dataset which has become a commonly used benchmark for k-anonymity. Fung also evaluated the TDR algorithm on the German credit dataset.

A. Performance Evaluation:

KACTUS 2 algorithm is proposed in this approach. The number of databases is given as 200. Then, the adjacency list is obtained.

Table 1: Accuracy Comparison of kACTUS and kACTUS 2 Algorithm

Dataset	Inducer	kACTUS	kACTUS 2
Japanese Credit	C4.5	85.08	86.64
	PART	83.19	84.82
Glass	C4.5	67.63	69.78
	PART	68.88	70.64
Ionosphere	C4.5	89.08	91.01
	PART	88.16	90.13

It is observed from the table that the proposed kACTUS 2 algorithm attains better accuracy when compared with kACTUS algorithm for all the three datasets taken for consideration.

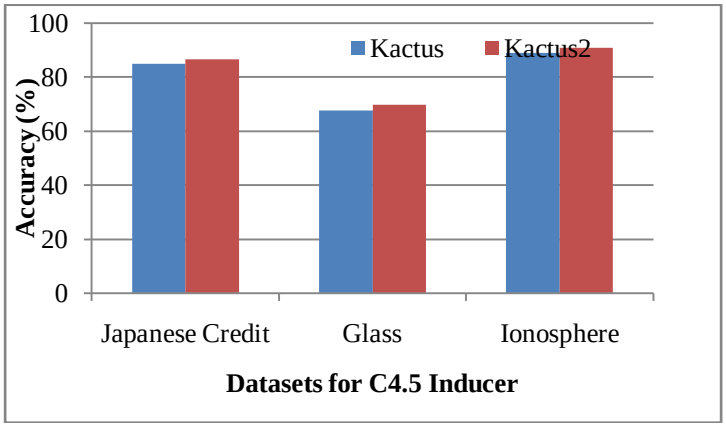

Figure 1: Accuracy Comparison of kACTUS and kACTUS 2 Algorithm for C4.5 Inducer

a. Japanese Credit Dataset:

When inducer C4.5 is considered, the accuracy of kACTUS algorithm is 85.08% where as for kACTUS2 algorithm it is 86.64%. Similarly for the PART inducer, the accuracy of kACTUS2 algorithm is better than the kACTUS algorithm.

b. Glass dataset:

When inducer C4.5 is considered, the accuracy of kACTUS algorithm is 67.63% where as for kACTUS2 algorithm it is 69.78%. Similarly for the PART inducer, the accuracy of kACTUS2 algorithm is 70.64% which is better than the kACTUS algorithm which attains the accuracy of 68.88%.

c. Ionosphere:

The accuracy of kACTUS algorithm for the C4.5 inducer is 89.08% where as for kACTUS2 algorithm it is 91.01%. Similarly for the PART inducer, the accuracy of kACTUS2 algorithm is 90.13% which is better than the kACTUS algorithm which attains the accuracy of 88.16%.

© 2010, IJARCS All Rights Reserved

Table 2: Comparing Accuracy with Generalization Methods

Case	Algorithm	Accuracy (%)
C4.5 Q1:8/14	TDS	83.01
	TDR	84.63
	KADET	82.80
	KACTUS	86.01
	KACTUS2	87.41
Logistics Q1:8/14	TDS	83.69
	TDR	84.83
	KACTUS	86.02
	KACTUS2	87.95
PART Q1:8/14	TDS	83.08
	TDR	84.68
	KACTUS	85.92
	KACTUS2	87.45

It is observed from that the algorithm, for the C4.5 Q1:8/14 case, the accuracy of kACTUS algorithm is 86.01% where as the accuracy of kACTUS2 is 87.41%.

Similarly, for Logistics Q1:8/14 case, the accuracy of 2kACTUS algorithm is 86.02% where as the accuracy of kACTUS algorithm is 87.95%.

Similarly for PART Q1:8/14, the accuracy of kACTUS algorithm is 85.92% where as the accuracy of kACTUS 2 is 87.45%. kACTUS 2 algorithm outperforms kACTUS algorithm in all the cases.

VI. CONCLUSION

In this paper a new method is presented for preserving the privacy in classification tasks using k-anonymity. The proposed method requires no prior knowledge regarding the domain hierarchy taxonomy and can be used by any inducer. The new method also shows a higher predictive performance when compared to existing state-of-the-art methods. Additional issues to be studied further include: Examining kACTUS with other decision trees inducers; revising kACTUS to overcome its existing drawbacks; extending the proposed method to other data mining tasks (such as clustering and association rules) and to other anonymity measures (such as l-diversity) which respond to different known attacks against k-anonymity, such as homogeneous attack and background attack.

VII. REFERENCES

[1] R. Agrawal., and Psaila, G. 1995. "Active Data Mining. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)", 3–8. Menlo Park, Calif.: American Association for Artificial Intelligence.

[2] A. Agrawal and R. Srikant, "Privacy Preserving Data Mining," ACM SIGMOD Record, vol. 29, no. 2, pp. 439-450, 2000. (ACM New York, NY, USA)

[3] Arabinda Nanda and Saroj Kumar Rout, "Data Mining & Knowledge Discovery in Databases: An AI Perspective", Proceedings of national Seminar on Future Trends in Data Mining (NSFTDM-2010):-10th may, 2010.

[4] B. Gilburd, A. Schuster and R. Wolff, "k-TTP: A New Privacy Model for Large-Scale Distributed Environments," Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 563–568, 2004. (ACM New York, NY, USA).

[5] M. Kantarcioglu, J. Jin and C. Clifton, "When Do Data Mining Results Violate Privacy" Proceedings of the 2004 International Conference on Knowledge Discovery and Data Mining, pp. 599-604, 2004. (Purdue University).

[6] Kun Liu, Hillol Kargupta, Jessica Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining", 2006.

[7] Oded Maimn and Lior Rokach, "Introduction to Knowledge Discovery in Databases".

[8] L. Rokach, R. Romano, O. Maimon, "Negation Recognition in Medical Narrative Reports, Information Retrieval", 11(6): 499-538, 2008 (Springer).

[9] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin and Y. Theodoridis, "State-of-the-Art in Privacy Preserving Data Mining", ACM SIGMOD Record, vol. 33, no.1, pp. 50-57, 2004. (ACM New York, NY, USA).

[10] M.S. Wolf and C.L. Bennett, "Local Perspective of the Impact of the HIPAA Privacy Rule on Research", Cancer-Philadelphia Then Hoboken, vol. 106, no. 2, pp. 474-479, 2006. (John Wiley & Sons).

© 2010, IJARCS All Rights Reserved