2013 International Conference on Advances in ICT for Emerging Regions (ICTer): 198 - 206

Hybrid Framework for Privacy Preserving Data Sharing

Dr.Ruvan Kumara Abeysekara and Prof. Weishi Zhang
Dalian Maritime University
contactruvan@gmail.com, teesiv@dlmu.edu.cn

Abstract

Privacy preserving data mining has become increasingly popular and continuously evolving field of study. It allows sharing of privacy sensitive data for analysis purposes. The recent advancement in data mining technology to analyze vast amount of data has played an important role in several areas of Business processing. Data mining also opens new threats to privacy and information security if not done or used properly. Therefore this research elaborates and introduces new Hybrid Algorithm for Privacy Preserving Data Sharing. It opens the gates to touch finer points of Hybrid methodologies in privacy preserving data mining. Experiments based on the discussions of literature, mainly about data sanitization done to prove the set of hypothesis mentioned on this paper.

Keywords— Privacy preserving, Data mining, Data sanitization, Data mining Algorithms

1. INTRODUCTION

High quality and useful knowledge is to be found in the integrated data from various sources. Concern about data privacy, however is a major obstacle of data sharing [1].Traditional concerns are mainly on the disclosure of identity information in data, referred to as sensitive facts in this research [2]. With the increasing power of data mining techniques, more current concerns are on the disclosure of inferable relationships in data, referred to as private knowledge in this paper [3]. The problem is not data mining itself, but the way data mining is done. "Data mining results rarely violate privacy, as they generally reveal high level knowledge rather than disclosing instances of data" [4].

The notion of privacy itself is difficult to formalize and quantify, and it can take different flavors depending on the context. The definition of the privacy is vague; nevertheless a dictionary definition of privacy that is relevant to data mining is "freedom from unauthorized intrusion" [5]. Most privacy laws in legislation as an example, European Community privacy guidelines [6] or the United States healthcare laws [7]) only apply to "individually identifiable data". Privacy preserving data mining technique must ensure that any information disclosed. Legal constraints which are in some jurisdictions (as an example, throughout the European Community) make data sharing impossible.[8] This is almost certainly the main constraint on the sharing of data between most large databases, and the topic has been the subject of considerable research and application development, particularly funded by the EC[9].

Vast amount of operational data and information are stored at different organizations and enterprises. Most of the stored data is useful only when it is shared and analyzed with other related data. For example, data from hospitals, pharmacies, health administrations and insurance companies is very useful for monitoring adverse reactions of drugs and detecting fraud when the data is shared and analyzed. However, the data normally contains certain sensitive facts, such as patients' information, and private knowledge, such as associations between the customer retaining rate and a promotion scheme of a pharmacy chain [10].

This research work investigates the feasibility of achieving Privacy Preserving Data Sharing using Hybrid Algorithm on Data Mining Environments. The major hypothesis statement of this research is Privacy preservation in data sharing, by amalgamation of existing group of algorithms is possible.

This research demonstrates empirically and theoretically the practicality and feasibility of achieving PPDS. In particular, it shows that a tradeoff between the strength of privacy preserving and the utility of data can be accomplished.

2. THE KDD PROCESS

Knowledge discovery in databases, also called the KDD process, is a non trivial process of discovering useful knowledge from data [11, 12]. This process consists of several steps, as can be seen in Figure 1. In this process, knowledge simply refers to information that is relevant and actionable represented by patterns or models. A pattern describes relationships among the facts in a subset of the given data, while a model is a characterization of the global dataset.

In this context, discovering knowledge means finding patterns, fitting a model to data, or even any general high level description of a set of data.

The major steps in the KDD process can briefly be defined as follows:

978-1-4799-1276-6/13/$31.00 ©2013 IEEE

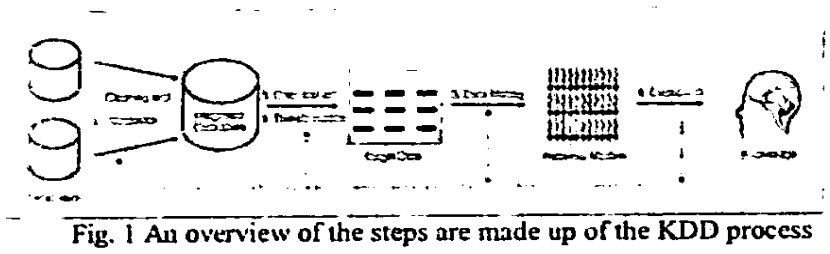

Fig. 1 An overview of the steps are made up of the KDD process

1. Data Cleaning: In the real world, data are often noisy or incomplete, and unless this is understood and corrected, it is likely that many interesting patterns will be missed and the reliability of detected patterns will be low. Data cleaning routines act on the data by filling in missing values, smoothing noisy data, identifying and removing outliers, and resolving inconsistencies.

2. Data Integration: The way that data are merged from multiple data stores may need to be transformed into forms appropriate for mining. In this step, data from multiple sources (with heterogeneous data) are combined into a coherent data store, as in data warehousing. These sources include multiple DBMSs, data cubes, or even flat files.

3. Data Selection: In this step, the goal is to identify the data that are relevant to the analysis task and retrieve them from the database. Only the selected data are subject to the mining process.

4. Data Transformation: In this step, the data are transformed or consolidated into forms appropriate for mining. This is achieved by performing summary, aggregation, generalization, or normalization operations.

5. Data Mining: This is the central activity in the KDD process. It is concerned with the extracting of implicit, previously unknown, and potentially useful patterns from the data. To do so, computational techniques are applied to produce a particular enumeration of patterns (or models) from the data.

6. Pattern Evaluation: In this step, interestingness (These functions are used to separate uninteresting patterns from knowledge. Interestingness measures for associations' rules include support and confidence.) Measures are applied with the purpose of searching for valuable patterns. This may be accomplished by using interestingness thresholds to filter out discovered patterns. Also, correlation analyses may be applied to evaluate the importance of the discovered information.

After the pattern evaluation phase, visualization and knowledge representation techniques can be used to present the mined knowledge to the users. However, as can be seen in Figure 1, the KDD process may iterate many times over previous steps and the process usually requires a great deal of experimentation.

One of the most studied problems in data mining is the process of discovering association rules from large databases.

3. The Basics of Association Rule Mining

3.1 The Support Confidence Framework

Most of the existing algorithms for association rules mining rely on the support confidence framework introduced in [13, 14].

Formally, association rules are defined as follows: Let $I=\{i_1,\ldots,i_n\}$ be a set of literals, the called items. Let D be a database of transactions, where each transaction t is an item set such that $t \subseteq I$. A unique identifier, called TID, is associated with each transaction. A transaction t supports X, a set of items in I, if $X \subset t$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I, Y \subset I$ and $X \cap Y = \phi$. Thus, say that a rule $X \Rightarrow Y$ holds in the database D with confidence φ if $|X \cup Y|/|X| \geq \varphi$, where $|A|$ is the number of occurrences of the set of items A in the set of transactions D. Similarly, say that a rule $X \Rightarrow Y$ holds in the database D with support σ if $|X \cup Y|/N \geq \sigma$, where N is the number of transactions in D.

While the support is a measure of the frequency of a rule, the confidence is a measure of the strength of the relation between sets of items. A survey of algorithms for association rules can be found in [15].

3.2 Interestingness Measures

Interesting rules are defined as rules describing surprising uncommon situations [16]. Support and confidence, reviewed in the previous section, are the most basic measures of rule interestingness. However, in many cases support and confidence are not sufficient. Many other measures have been proposed in the literature [17] for ranking association patterns according to their degree of interestingness. Some of those measures are described as follows:

Lift: also known as interest [18] and strength [16], lift is defined as

$$lift(X \rightarrow Y) = support(X \cup Y)/(support(X) \times support(Y))$$

Lift only measures co-occurrence not implication, in that it is completely symmetric. This is based on statistical independence; if lift is equal to 1 then the condition X and the conclusion Y are independent. If lift is greater than 1 then the condition is associated with the conclusion. If lift of a rule is between 0 and 1 then the condition is negatively associated with the conclusion.

Coverage: describes the importance of dependency [19]. This measure is defined as $coverage(X \rightarrow Y) = support(X \cup Y)/support(Y)$. A rule has coverage c if c% of all transactions that contain Y also contains X. Its values belong to the interval [0, 1].Note that the formulas of confidence and coverage are very similar.

Conviction: this measure was derived from the implication definition [20]. Logically, $X \rightarrow Y$ can be

rewritten as ~(A ^ ~B). This measure shows the level of dependence between A and B. After some transformation, conviction is defined as conviction(X → Y) = support(X)×(1-support(Y))/(support(X)-support(X∪Y)).Conviction values are in the interval[0,+∞]. If conviction is equal to 1, the antecedent and the consequent are independent. Conviction is different from confidence because it does not suffer from the same problem of producing misleading rules.

Piatetsky-Shapiro: This measure was introduced in [20], and it is defined as PS(X → Y) = support(X [∪Y) - support(X) ×support(Y). Absolute value of this measure shows dependence between antecedent and consequent. Statistically independence occurs at PS = 0. The values for this measure fall in the interval [-0.25, 0.25].

Other Measures: other measures to determine the interestingness of association patterns can be found in [21]. The work presents a comparative study of interestingness measures for ranking association patterns. The key finding of this study was that there is no measure that is consistently better than the others in all cases. However, there are situations in which many of these measures are highly correlated with each other (e.g., when support based pruning is used). For example, measures such as Laplace, Jaccard, Piatetsky-Shapiro, overage, confidence, and Cosine IS behave similarly in the region of low support values, which typically occurs in large databases.

3.3 Sensitive Rules and Sensitive Transactions

Protecting sensitive knowledge in transactional databases is the task of hiding a group of association rules which contains sensitive knowledge. These rules are referred to as sensitive association rules and define them as follows:

Definition 1 (Sensitive Association Rules) Let D be a transactional database, R be a set of all association rules that can be mined from D based on a minimum support σ, and Rule SH be a set of decision support rules that need to be hidden according to some security policies. A set of association rules, denoted by SR, is said to be sensitive i

"SR ⊂ R" and SR would derive the set Rule SH. ~SR is the set of non sensitive association rules such that ~S_R R ∪ S_R = R. A group of sensitive association rules is mined from a database D based on a special group of transactions. We refer to these transactions as sensitive transactions and define them as follows:

Definition 2 (Sensitive Transactions) Let T be a set of all transactions in a transactional database D and SR be a set of sensitive association rules mined from D. A set of transactions is said to be sensitive, denoted ST, if S_T T ⊂ T and ∀ t ∈ S_T ,∃ sr ∈ S_R such that items(sr)⊆t.

4. The Framework for Privacy Preserving Data Mining in a Hybrid Environment

The framework to address hybrid methodology for privacy preservation in association rule mining is introduced in this section. First step is convert database in to k-anonymizd database using best k-anonymaization techniques such that, generalization and suppression as depicted in Figure 2

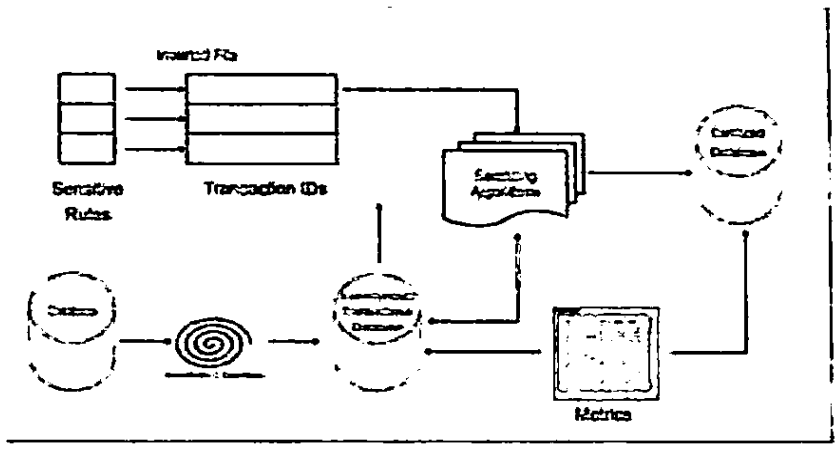

Fig. 2 The sketch of the framework for hybrid method for privacy preserving association rule mining.

The framework encompasses an inverted file to speed up the sanitization process, a library of sanitizing algorithms used for hiding sensitive association rules from the database, and a set of metrics to quantify not only how much private information is disclosed, but also the impact of the sanitizing algorithms on the transformed database and on valid mining results.

4.1 K-Anonymization

K-anonymization is a method based on k-anonymity model that answers the question has been discussed "How can a data holder release a version of its private data with scientific guarantees that the individuals who are the subjects of the data cannot be re identified, while that data remain practically useful? "[20]. Such as, a medical institution may want to release a table of medical records. Even though the names of the individuals can be replaced with dummy identifiers, some set of attributes (so called the quasi-identifier) can leak confidential information. For instance, the birth date, zip code and the gender attributes in the disclosed table can uniquely determine an individual. Joining such a table with some other publicly available information source, like a voter's list table, which consists of records containing the attributes that make up the quasi-identifier as well as the identities of individuals, the medical information, can be easily linked to individuals. k-anonymity prevents such a privacy breach by ensuring that each individual record can only be released if there is at least k − 1 other (distinct) individuals whose associated records are indistinguishable from the former in terms of their quasi-identifier values.

4.2 Inverted File

Sanitizing a transactional database consists of identifying the sensitive transactions and adjusting them. To speed up this process, a transactional database is scanned only once and, at the same time, retrieval facility is build (inverted file) [14]. The inverted file's vocabulary is composed of all the sensitive rules to be hidden, and for each sensitive rule there is a corresponding list of transaction IDs in which the rule is present.

4.3 Library of Sanitizing Algorithms

Specified framework, the sanitizing algorithms modify some transactions to hide sensitive rules based on a disclosure threshold ψ controlled by the database owner. This threshold not directly controls the balance between knowledge disclosure and knowledge protection by controlling the proportion of transactions to be sanitized. For instance, if $\psi = 50\%$ then half of the sensitive transactions will be sanitized, when $\psi = 0\%$ all the sensitive transaction will be sanitized, and when $\psi = 100\%$ no sensitive transaction will be sanitized. In other words, represents the ratio of sensitive transactions that should be left untouched. Advantage of this threshold is that it enables a compromise between hiding association rules while missing non sensitive ones, and finding all non sensitive association rules but uncovering sensitive ones.

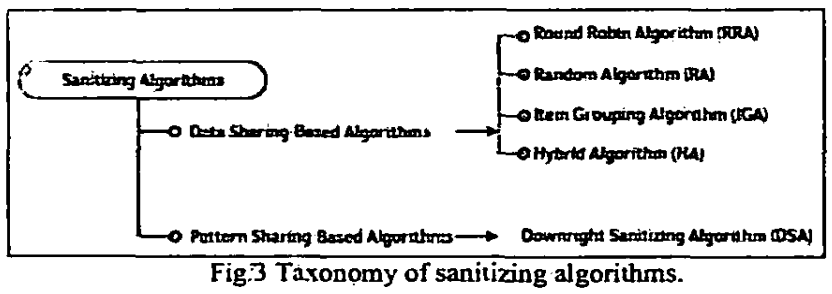

Fig.3 Taxonomy of sanitizing algorithms.

Prior introductions, the sanitization process acts on the data to remove or hide the group of sensitive association rules representing the sensitive knowledge. To accomplish this, a small number of transactions that participate in the generation of the sensitive rules have to be modified by deleting one or more items from them. In doing so, the algorithms hide sensitive rules by reducing either their support or confidence below a privacy threshold (disclosure threshold). In the latter, the sanitizing algorithm acts on the rules mined from a database, instead of the data itself. Moreover hybrid approach on this regard gives tremendous outcome to achieve good privacy protection. The algorithm removes all sensitive rules before the sharing process.

4.4 Set of Metrics

There are two major groups of metrics as Data sharing based metrics and Pattern sharing based metrics.

Data sharing based metrics are related to the problems as follows,

Problem 1 - occurs when some sensitive association rules are discovered in the sanitized database. We call this problem Hiding Failure (HF), and it is measured in terms of the percentage of sensitive association rules that are discovered from D'. Ideally, the hiding failure should be 0%.

Problem 2 - occurs when some legitimate association rules are hidden as a side effect of the sanitization process. This happens when some non sensitive association rules lose support in the database due to the sanitization process. We call this problem Misses Cost (MC), and it is measured in terms of the percentage of legitimate association rules that are not discovered from D'. In the best case, this should also be 0%.

Problem 3 - occurs when some artificial association rules are generated from D' as a product of the sanitization process. It is called, problem Artifactual Patterns (AP), and it is measured in terms of the percentage of the discovered association rules that are artifacts, that is, rules that are not present in the original database. Artifacts that are generated when new items are added to some transactions to alter (decrease) the confidence of sensitive rules. For instance, in a rule $X \rightarrow Y$, if the items are added to the antecedent part X of this rule in transactions that support X and not Y, then the confidence of such a rule is decreased.

Dissimilarity between the original and sanitized databases could be measured by computing the difference between their sizes in bytes. However, it is believed that this dissimilarity should be measured by comparing their contents instead of their sizes. Comparing contents of those is more intuitive and gauges more accurately the modifications made to the transactions in the database.

To measure the dissimilarity between the original and the sanitized datasets, it could simply compare the difference in their histograms. Herein case, the horizontal axis of a histogram contains all items in the dataset, as the vertical axis corresponds to their frequencies. Sum of the frequencies of all items gives the total of the histogram.

Pattern sharing based metrics are related to the problems as follows,

Problem 1 - conveys the non sensitive rules that are removed as a side effect of the sanitization process. This problem is referred as Side Effect. It is related to the misses cost problem in data sanitization (Data sharing based metrics).

Problem 2 - occurs when using some non sensitive rules, an adversary may recover some sensitive ones by inference channels. Such a problem is referred as Recovery Factor.

Side Effect Factor (SEF) measures the number of non sensitive association rules that are removed as a side effect of the sanitization process.

Recovery Factor (RF) expresses the possibility of an adversary recovering a sensitive rule based on non sensitive ones. The recovery factor of one pattern takes into account the existence of its subsets. The rationale behind the idea is that all nonempty subsets of a frequent item set must be frequent. Thus, if all subsets are recovered of a sensitive item set (rule), say that the recovery factor for such an item set is possible, and thus it assigned the value 1. However, the recovery factor is never certain, that is, an adversary may not learn an item set even with its subsets. On the other hand, when not all subsets of an item set are present, the recovery of the item set is improbable, thus value 0 is assigned to the recovery factor.

4.5 Algorithms

Heuristic to hide sensitive rules in transactional databases is described in here. Data sharing based algorithms are then introduced that rely on these heuristic.

Heuristic 1: Sanitization Based on the Degree of Sensitive Transactions

To alleviate the complexity of the optimal sanitization, we could use some heuristics. A heuristic does not guarantee the optimal solution, but usually finds a solution close to the best one in a faster response time [22].

First heuristic for data sanitization is based on the fact that, in many cases, a sensitive transaction participates in the generation of one or more sensitive association rule to be hidden. The number of sensitive rules is referred to support by a sensitive transaction as the degree of a sensitive transaction defined as:

Definition 1- (Degree of a Sensitive Transaction) Let D be a transactional database and S_T a set of all sensitive transactions within D. Degree of The sensitive transaction t, denoted by degree (t), such that $t \in S_T$, is defined as the number of sensitive association rules that can be found in t.

5. HYBRID ALGORITHM

Hybrid algorithm, presented in the next sections, act on the original database taking into account the degree of sensitive transactions. For instance, given the number of sensitive transactions to alter, based on, algorithms select for each sensitive rule the sensitive transactions whose degree is sorted in descending order. The rationale is that by k-anonymization and sanitizing the sensitive transactions that share a common item with more than one sensitive rule, the hiding strategy of such rules is optimized and, consequently, the impact of the k-anonymization and sanitization on the discovery of the legitimate association rules is minimized.

Hybrid algorithms, which rely on Heuristic 1, have essentially five major steps,

Step 1: Create k-anonymized database using k-Anonymous Decision Tree method

Step 2: Scan a database and identify the sensitive transactions for each sensitive association rule. This step is accomplished when the inverted file is built;

Step 3: Based on the disclosure threshold, calculate for each sensitive association rule the number of sensitive transactions that should be sanitized and mark them. Most importantly, the sensitive transactions are selected based on their degree (descending order);

Step 4: For each sensitive association rule, identify a candidate item that should be eliminated from the sensitive transactions. This kind of candidate item is called the victim item;

Step 5: Scan the database again, identify the sensitive transactions marked to be sanitized and remove the victim items from them.

In general, the inputs for this algorithm is a transactional database D, list of attributes A, k-anonymity parameter, a set of sensitive association rules S_R, and a disclosure threshold controlled by the database owner, while the output is the sanitized database D'. Most of the parts of this algorithm are described in previous paragraphs separately.

To illustrate how the presented algorithms work, consider the sample transactional database in Figure 5.2(a). Suppose that have a set of sensitive association rules SR = {A,B→D; A,C→D}. This example yields the following results,

Step 1: The algorithms scan the database to identify the sensitive transactions. For this example, the sensitive transactions S_T containing the sensitive association rules are {T1,T3,T4}. The degrees of the transactions T1, T3 and T4 are 2, 1 and 1 respectively. In particular, the rule A,B→D can be mined from the transactions T1 and T3 and the rule A,C→D can be mined from T1 and T4.

Step 2: Suppose that we set the disclosure threshold ψ to 50%. Then the algorithms sort the sensitive transactions in descending order of degree. The algorithms sanitize half of the sensitive transactions for each sensitive rule. In this case, only the transaction T1 will be sanitized.

Step 3: In this step, the victim items are selected. Note that the three algorithms employ different strategies for this selection. The Round Robin algorithm selects the victim items for each rule taking turns. The item A is selected for both rules minimizing the impact on the database. The Random algorithm selects one item for each rule randomly. Let us assume that the item A was selected for the first rule and the item C was selected for the second rule. The Item Grouping Algorithm clusters sensitive rules that share a common item.

```
Input :D, A, k, SR, Psi
output: D'
1 begin algorithm
      1 procedure Kanonymization(D,A,k)
      2 //D – dataset, A – list of attributes, k – anonymity parameter
      3 r  implies that  root node
      4 candList  implies that ((a, r) ; a element of A)
      5 while candList contains candidates with positive gain do
      6 bestCand  implies that  candidate from candList with highest gain
      7 if bestCand maintains k-anonymity then
      8 Apply the split and generate new nodes N
      9 Remove candidates with the split node from candList
      10 candList implies that  candList union ((a, n) ; a element of  A, n element of N)
      11 else
      12 remove bestCand from candList
      13 end if
      14 end while
      15 return kanonymized D.
      16 end procedure

      2: procedure Sanitization(D,SR, Psi )
      input : D, SR, Psi
      output: D'
      1 begin
      2 // Step 1: Identifying sensitive transactions and building index T
      3 foreach transaction t element of D do
      4 for k = 1 to size(t) do
      5 Sup(itemk, D) implies that Sup(itemk,D)+1; //Update support of each itemk in t;
      6 Sort the items in t in alphabetic order;
      7 foreach sensitive association rule sri  element of  SR do
      8 if items(sri) subset of or equal to t then
      9 T [sri].tid list  implies that T [sri].tid list ( TID_of(t);
      10 end
      11 end
      12 end
      13 // Step 2: Selecting the number of sensitive transactions
      14 foreach sensitive association rule sri  element of  SR do
      15 Sort the vector T [sri].tid_list in descending order of degree;
      16  NumbTranssri implies that |T [sri]| x (1 -  Psi );
      17 // |T [sri]| is the number of sensitive transactions for sri
      18 end
      19 // Step 3: Identifying victim items for each sensitive transaction
      20 3.1 Group sensitive rules in a set of groups GP such that For all G element of
      GP,
      21 For all sri; srj  element of  G, sri and srj share the same itemset I. Give the
      class label
      22 Alfa to G such that Alfa element of I and For all Beta element of I, sup(Alfa, D)
      less than or equal to sup(Beta, D);
      23 3.2 Order the groups in GP by size in terms of number of sensitive rules
      24 in the group;
      25 // Compare groups pairwise Gi and Gj starting with the largest
      26 3.3 forall srk element of Gi  intersection Gj do
      27 if size(Gi) 6 not equal to size(Gj ) then
      28 remove srk from smallest(Gi;Gj);
      29 else
      30 remove srk from group with class label Alfa such that sup(Alfa, D) greater
      than or equal to sup  (Beta, D)
      and Alfa, Beta are class labels of either Gi or Gj ;
      31 end
      32 end
      33 3.4 foreach sensitive association rule sri  element of  SR do
      34 for j = 1 to NumbTranssri  do
      35ChosenItem  implies that Alfa such that Alfa is the class label of G and sri
      element of  G;
      36Victims[T [sri, j]].item_list  implies that Victims[T [sri, j]].item_list union
      ChosenItem;
      37 end
      38 end
      39 // Step 4: D' implies that  D
      40 Sort the vector Victims in ascending order of tID;
      41 j  implies that 1;
      42 foreach transaction t element of D do
      43 if tID == V ictims[j].tID then
      44 t  implies that (t -  V ictims[j].item_list);
      45 j  implies that j + 1;
      46 end
      47 end
      48 end
      48 end procedure

2 Kanonymization(DB,a,K)

3 Sanitization(d,sr, sl )

4 end algorithm
```

Both rules share the items A and D. In this case, only one item is selected, say the item D. By removing the item D from T1 the sensitive rules will be hidden from T1 in one step and the disclosure threshold will be satisfied.

Step 4: The algorithms perform the sanitization taking into account the victim items selected in the previous step.

EVALUATION OF ALGORITHM

The effectiveness of the sanitizing algorithms is measured in terms of the number of sensitive association rules effectively hidden, as well as the proportion of non sensitive rules accidentally hidden due to the sanitization process.

The effectiveness of the sanitizing algorithms was studied based on three major conditions, as follows

C1: the disclosure threshold (ψ) was set to 0% and fixed the minimum support threshold σ, the minimum confidence threshold φ, and the number of sensitive rules to hide.

C2: the parameters were fixed as done in condition C1 but varied the number of sensitive rules to hide.

C3: the disclosure threshold (ψ) was set to 0%, fixed the minimum confidence threshold φ and the number of sensitive rules to hide, and varied the minimum support threshold σ for each dataset.

Note that in all the three conditions above, the disclosure threshold (ψ) is purposely set to 0%. In this particular case, no sensitive rule is allowed to be mined from the sanitized dataset. Later (in special cases section), will show that a database owner could also slide the disclosure threshold (ψ > 0) to allow a balance between knowledge discovery and privacy protection in the sanitized database.

Table 6.3 shows a summary of the best sanitizing algorithms, in terms of misses cost, under condition C1. The algorithm HA yielded the best results in almost all the cases. The exceptions are the scenarios S2, S3, and S4 of the dataset Retail that contains sensitive rules with high support items. In this case, the algorithm RA benefit from the selection of the victim items, a choice which varies in each sensitive transaction, is alleviating the impact on the sanitized dataset. As a result, the values for misses cost are slightly minimized.

Table 1 Summary of the best algorithms in terms of misses cost under condition C1

Dataset	ψ= 0%, 6 sensitive rules			
	S1	S2	S3	S4
Kosarak	HA	HA	HA	HA
Retail	HA	HA	HA	HA
Adults	HA	HA	HA	HA
Webdocs	HA	HA	HA	HA

The same behavior of the sanitizing algorithms was also observed when analyzing misses cost under conditions C2 and C3. Table 2 provides a summary of the best sanitizing algorithms in terms of misses cost under condition C2. The results confirm the same finding we reported previously for the algorithms RA for the

dataset Retail. The only exceptions here are the scenario S2 for the datasets Retail and Adults. In the dataset Retail, HA yielded the best results of misses cost with up to 2 sensitive rules being sanitized. As the number of sensitive rules increased, the HA yielded the best values of misses cost. The same behavior can be observed in the dataset Adults. When the number of rules to be sanitized increases, HA yields the best results for misses cost.

Table 2 Summary of the best algorithms for misses cost under condition C2

Dataset	ψ= 0%, 6 varying the no. of rules			
	S1	S2	S3	S4
Kosarak	HA	HA	HA	HA
Retail	HA	HA	RA	RA
Adults	HA	HA	HA	HA
Webdocs	HA	HA	HA	HA

Table 3 shows the results of misses cost under condition C3. As can be seen, the same trends were observed, except in scenario S4 for the dataset Retail in which the algorithm RRA yielded better results than those in the RA. However, the results for misses cost in the RRA are slightly better.

Table 3 Summary of the best algorithms for misses cost under condition C3

Dataset	ψ= 0%, 6 varying the values of σ			
	S1	S2	S3	S4
Kosarak	HA	HA	HA	HA
Retail	HA	HA	RA	RRA
Adults	HA	HA	HA	HA
Webdocs	HA	HA	HA	HA

Under condition C2, when the number of rules to be sanitized was increased, the algorithms RA, RRA, yielded the best results for the differential between the original and the sanitized datasets, that is, dif(D,D′).

Based on the results for dif(D,D′), a natural question arises: how can RA, and RRA get the best results for dif(D,D′) and not for misses cost? The main reason is that the victim items in these three algorithms are dynamic, that is, a new victim item is selected for each sensitive transaction to be sanitized. This approach reduces support of every item in a sensitive rule (one item is selected for each sensitive transaction) regardless of whether an item has high or low support. Reducing items with high support would prune the candidate generation of discovered rules in the sanitized dataset, compromising the values of misses cost. On the contrary, the victim item selected by the HA, for a sensitive rule, is fixed for all sensitive transactions. Moreover, the HA always selects the item with lower support for each rule, which greatly improves the values of misses cost.

Regarding the third performance measure, artifactual patterns, and one may claim that when decrease the frequencies of some items, the relative frequencies in the database may be modified by the sanitization process, and new rules may emerge. However, in these experiments, the problem artifactual pattern AP was always 0% with all algorithms regardless of the values of ψ . Sanitization, indeed, does not remove any transaction. On the other hand, some of the sanitizing algorithms introduced in [23] present the case in which artifactual patterns appear (i.e., AP > 0), since the sensitive rules are hidden by reducing their confidence below a privacy threshold. To do so, some items are added to transactions that participate in the generation of the antecedent part X, but not the consequent part Y of a rule, where the rule is the form X→ Y. Adding items to some transactions results in the generation of new association rules that are not supposed to exist in the original database.

CPU Time for the Sanitization Process

The scalability of the sanitization algorithms was tested with the size of the database as well as the number of rules to hide. To do so, the Kosarak dataset was selected since it is the sophisticated one used in the experiments.

We varied the size of the original database D from 150K transactions to 900K transactions, while fixing the disclosure threshold ψ = 0% and keeping the set of sensitive rules constant (6 original sensitive rules that are mutually exclusive). Fig. 4 shows that the algorithms scale well with the database size. The algorithms IGA, RRA, RA and HA yielded lower CPU time. The algorithms IGA, RRA, RA and HA require only two scans.

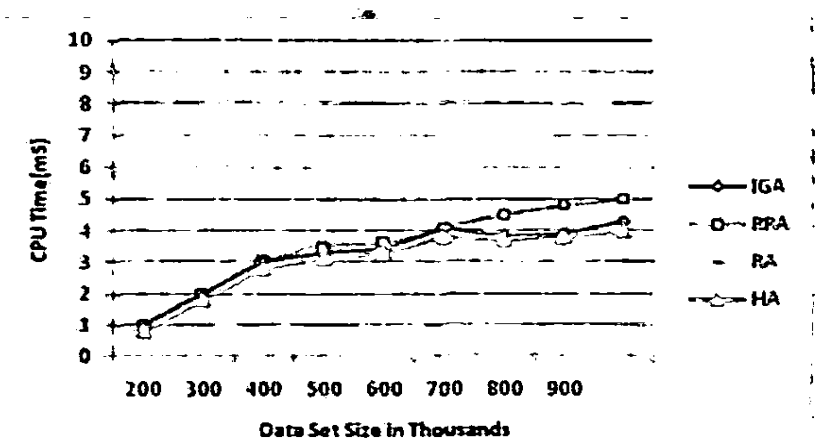

Fig.4 (a) Results of CPU time for the sanitization process

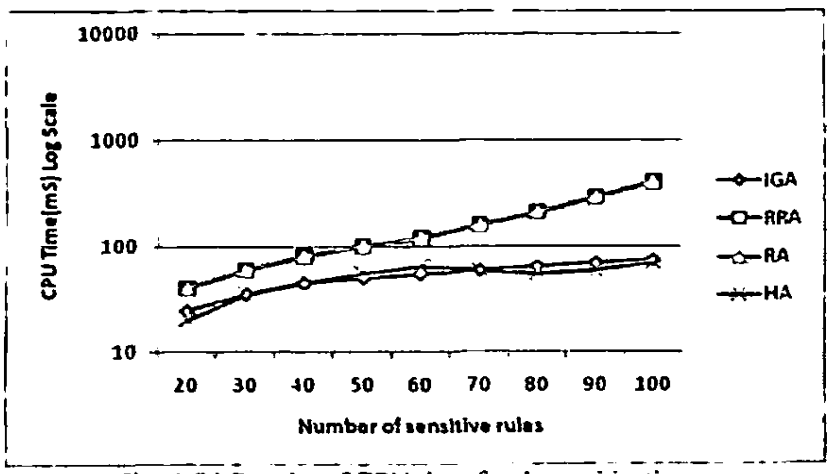

Fig. 4 (b) Results of CPU time for the sanitization process

As can be observed, the algorithms IGA, RRA, RA and HA increase CPU linearly, even though their complexity in main memory is not linear. If we increase the number of sensitive rules or even if we select a group of sensitive rules with very high support, these algorithms may not scale linearly. However, there is no compelling need for sanitization to be a fast operation since it can be done offline.

The I/O time (scans over the dataset) is also considered in these figures. This demonstrates good scalability with the cardinality of the transactional database.

The number of sensitive rules to hide were also varied from approximately 20 to 100 selected randomly, while fixing the size of the dataset Kosarak and fixing the support and disclosure thresholds to $\psi = 0\%$. Fig. 6.4 (b) shows that the algorithms scale well with the number of rules to hide.

The values are plotted in logarithmic scale. Although HA requires 2 scans, it was fastest. The main reason is that the HA performs a number of operations in main memory to fully sanitize a database. The HA requires one scan to build an inverted index where the vocabulary contains the sensitive rules and the occurrences contain the transaction IDs. In the second scan, HA sanitizes only the transactions marked in the occurrences. Note that when the number of sensitive rules increases, the intersection of items among the rules tends to increase as well.

The scalability of the sanitizing algorithms is mainly due to the inverted files use in the approaches for indexing the sensitive transaction IDs per sensitive rule. There is no need to scan the database again whenever want to access a transaction for sanitization purposes. The inverted file gives direct access with pointers to the relevant transactions.

CONCLUSION

In the context of HMPPDT, It was noticed that sanitization is a challenging task. It can render a released database almost useless when not done properly. To alleviate the difficulties of sanitizing a database, different conditions were investigated under which a data owner can tune the parameters of the sanitizing algorithms to get the most of the sanitization process.

Data sharing based and pattern sharing based algorithms were evaluated by performing a broad set of experiments against real datasets. This evaluation was carried out to suggest guidance on which algorithms perform best under different conditions. In particular, It was observed that the algorithm HA (Hybrid Algorithm) presented an outstanding performance in our experiments. In almost all the cases, HA yielded the best results in terms of misses cost and hiding failure. Another interesting point observed from the experiments was the fact that the best results of misses cost and hiding failure were obtained as we increased the size of the datasets. The inability to generalize the results for classes of categories of data mining algorithms may be a tentative threat for disclosing information. Developing hybrid methodologies and techniques for data mining provide new opportunities to achieve the task early and leads the researches in to new profiles.

REFERENCES

[1] Y. Lindell and B. Pinkas. Privacy preserving data mining. Journal of Cryptology, 15(3):177– 206, 2002.

[2] L. Sweeney. k-anonymity: a model for protecting privacy. International journal on uncertainty, Fuzziness and knowldege based systems, 10(5):557 – 570, 2002.

[3] M. Kantarciolu, J. Jin, and C. Clifton. When do data mining results violate privacy? In KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 599–604, New York, NY, USA, 2004. ACM Press.

[4] Clifton, C., Kantarcioğlu, M., & Vaidya, J., Defining privacy for data mining. In Proceedings of the National Science Foundation Workshop on Next Generation Data Mining (pages. 126-133), Baltimore, MD, USA, 2003.

[5] Merriam-webster online dictionary.

[6] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Communities. No I.(281):31–50, Oct. 24 1995.

[7] Standard for privacy of individually identifiable health information. Federal Register, 67(157):53181–53273, Aug 14 2002.

[8] Mukkamala, R.; Ashok, V.G., Fuzzy-based Methods for Privacy-Preserving Data Mining, Information Technology: New Generations (ITNG), 2011 Eighth International Conference on, 2011.

[9] Tran Khanh Dang; Küng, J.; Phuong, H.V.Q., Protecting Privacy While Discovering and Maintaining Association Rules, New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on 2011.

[10] Blanton, M., Achieving Full Security in Privacy-Preserving Data Mining, Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom), 2011.

[11] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In Advances in Knowledge Discovery and Data Mining. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy (eds.), pages 1-34. MIT Press, Cambridge, MA, 1996.

[12] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA., 2001.

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

[14] L. Sweeney. k-anonymity: a model for protecting privacy. International journal on uncertainty, Fuzziness and knowldege based systems, 10(5):557 – 570, 2002.

[15] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In Advances in

Knowledge Discovery and Data Mining. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy (eds.), pages 1-34. MIT Press, Cambridge, MA, 1996.

[16] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, CA., 2001.

[17] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measurefor Association Patterns. In Proc. of the 8th ACM SIGKDD Intl. Conf. on KnowlegdeDiscovery and Data Mining, pages 32-41, Edmonton, AB, Canada, July 2002.

[18] J. A. Hartigan "Clustering Algorithms". Wiley., 1975.

[19] Lindell Y. and Pinkas B. Privacy preserving data mining. J. Cryptol., 15(3):177–206, 2002.

[20] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In Proceedings of 24th ACM-SIGMOD Symposium on Principles of Database Systems (PODS), 2005.

[21] Du W. and Zhan Z. Building decision tree classifier on private data. In C. Clifton and V. Estivill-Castro (eds.). IEEE Int. Conf. on Data Mining Workshop on Privacy, Security, and Data Mining, 2002, pages. 1–8

[22] Klösgen, W.. KDD: Public and private concerns.IEEE EXPERT, 1995.10(2), 55-57

[23] Rebecca Wright, "Progress on the PORTIA Project in Privacy Preserving Data Mining," A data surveillance and privacy protection workshop held on 3rd June 2008.