Privacy Preserving Association Rule Mining by Concept of Impact Factor using Item Lattice

B Janakiramaiah
DVR & Dr HS MIC College of Technology
Department of Computer Science and Engineering
Vijayawada, Andhra pradesh, India
bjanakiramaiah@gmail.com

Dr A.RamaMohan Reddy
S.V.University College of Engineering , S.V.University
Department of Computer Science and Engineering
Tirupathi, Andhra pradesh
India

Abstract: Association Rules revealed by association rule mining may contain some sensitive rules, which may cause potential threats towards privacy and protection. Association rule hiding is a competent solution that helps enterprises keeps away from the hazards caused by sensitive knowledge leakage when sharing the data in their collaborations. This study shows how to protect actionable knowledge for strategic decisions, but at the same time not losing the great benefit of association rule mining. A new algorithm has been proposed to eradicate sensitive knowledge from the released database based on the intersection lattice and impact factor of items in sensitive association rules. The proposed algorithm specifies the victim item such that the alteration of this item causes the slightest impact on the non sensitive frequent association rules. Experimental results demonstrate that our proposed algorithm is appropriate in real context and can achieve significant improvement over other approaches present in the literature.

Key–Words: Frequent itemset lattice , Sensitive itemset grouping , Privacy preserving , Hiding association rules.

1 Introduction

The significant advances in data collection and data storage technologies have provided economical storage of massive amounts of transactional data in data warehouses that reside in companies and public sector organizations. Apart from the benefit of using this data intrinsically (e.g., for keeping up to date profiles of the consumers and their procurements, retaining a list of the accessible products, their quantities and price, etc.), the mining of these datasets with the active data mining tools can disclose valuable knowledge that was undisclosed to the data holder beforehand. Furthermore, companies are often willing to cooperate with each other and with other entities who conduct similar business, towards the mutual benefit of their businesses. Significant knowledge patterns can be derived and shared among the partners during the collaborative mining of their datasets. At the same time, a massive repository of data contains confidential data and some sensitive knowledge, which may cause possible threats in the direction of privacy and protection.

Association rule mining extracts novel, concealed and useful patterns from huge repositories of data. These patterns are helpful for effective analysis, strategic planning and decision making in telecommunication networks, marketing, retail business, medical analysis, website linkages, financial transactions, advertising, and other applications. The sharing of association rules can bring lots of advantages in industry, research and business collaboration. At the same time, a huge repository of data contains private data and sensitive rules that must be protected before sharing [1].

An example scenario, taken from the work of Verykios et al. [2], motivates the need for applying an association rule hiding algorithms to defend sensitive association rules against confession. Let us assume that we are negotiating with the Dedtrees Paper Company, as purchasing directors of BigMart, a large supermarket chain. They offer their products at reduced prices, provided that we have agreed to give them access to our database of customer purchases. We accept the deal and Dedtrees starts mining our data. By using an association rule mining tool, they find that people who purchase skim milk also purchase Green Paper. Dedtrees now runs a coupon marketing campaign offering a 50 cents discount on skim milk with every purchase of a Dedtrees product. The campaign cuts heavily into the sales of Green Paper, which increases the prices to us, based on the low sales. During our next negotiation with Dedtrees, we find out that with reduced competition, they are unwilling to offer us a minimal price. Finally, we start losing business to our competitors, who were in a position to negotiate a better deal with Green Paper. In other words, the aforementioned scenario in-

dicates that BigMart should sanitize competitive information (and other significant corporate secrets of course) before delivering their database to Dedtrees, so that Dedtrees does not monopolize the paper market. Similar motivating examples for association rule hiding are discussed in the work of [3] [4].
On demand for diverse uneven requirements of knowledge discovery, data sharing, and privacy preserving, Privacy Preserving Data Mining (PPDM) has become a research hotspot in data mining [5] [6]. Association rule hiding is a sub-area of PPDM that aims to renovate an original database into a released database such that the sensitive association rules, which are used to formulate decisions, cannot be revealed, whereas the non-sensitive association rules can still be mined [7] [8] [9].

1.1 Association rule mining

Association rule mining is the process of discovering set of items (also known as itemsets) That regularly co-occur in a transaction database so as to produce significant association rules that hold on the data [10] [11]. Every association rule is defined as an implication of the form $X \Rightarrow Y$, where X, Y are frequently occurring itemsets in the transactional database, for which $X \cap Y = \Phi$ (i.e., X and Y are disjoint). The itemset $X \cup Y$ that leads to the generation of an association rule is called generating itemset. An association rule consists of two parts: the Left Hand Side (LHS) or antecedent, which is the part on the left of the arrow of the rule (here X), and the Right Hand Side (RHS) or consequent, which is the part on the right of the arrow of the rule (here Y). A set of metrics, known as support, confidence, lift, correlation, chi-squared, conviction and surprise are integrated with the task of association rule mining to drive the generation of association rules and expose only those rules that are expected to be of interest to the data owner. In particular, the measure of support eliminates rules that are not sufficiently supported by the transactions of the dataset and therefore expected to be uninteresting, i.e. occurring simply by chance. On the other hand, confidence measures the strength of the relation between the itemsets of the rule as it quantifies the reliability of the inference made by the rule. A low value of confidence in rule $X \Rightarrow Y$shows that it is quite rare for itemset Y to be present in transactions that contain itemset X. The process of association rule mining includes two main steps. The first step generates frequent itemsets that satisfy a minimum support threshold. The second step generates association rules that have confidence above a minimum confidence threshold from the frequent itemsets. Readily available in association rule mining algorithms, are Apriori, DHP, DIC, FP-Growth, Eclat, and ARMOR. The process of association rule mining as showed in Figure.1. The basic concepts of association rule mining [10],[11] are formally defined as follows: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of m literals. Each element of I is known as an item. X is an itemset if $X \subseteq I$. The transactional database $D = \{t_1, t_2, ..., t_n\}$ on I is a finite set of transactions, where each transaction $t_i \in D$ contains a set of items. Itemset $X \subseteq I$ is supported by a transaction t_i if $X \subseteq t_i$. The support of itemset X, denoted by $\alpha(X)$, is the number of transactions that contain X and is defined as $\alpha(X) =| \{t \in D \mid tsupportsX\} \mid$. An itemset X is called a frequent itemset if $\alpha(X) \geq \sigma$, where σ is the minimum support threshold given by users. An association rule is an implication $X \Rightarrow Y$, where $X, Y \subset I$ and $X \cap Y = \Phi$. The support of a rule $X \Rightarrow Y$ is specified by the support of itemized $X \cup Y$, i.e, $\alpha(X \Rightarrow Y) = \alpha(X \cup Y)$. The confidence of a rule $X \Rightarrow Y$ is $\beta(X \Rightarrow Y) = \alpha(X \cup Y)/\alpha(X)$. Let the minimum support threshold σ and the minimum confidence threshold δ be given by users or experts. The association rule $X \Rightarrow Y$ is called the strong association rule if $\alpha(X \Rightarrow Y) \geq \sigma$ and $\beta(X \Rightarrow Y) \geq \delta$.

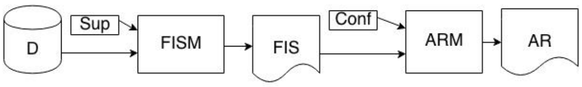

Figure 1: Process of association rule mining.

1.2 Frequent itemsets on the itemset lattice

A lattice structure can be used to enumerate the list of all possible itemsets. Figure.2 shows an itemized lattice for $I = \{a, b, c, d\}$. Itemsets that can be constructed from a set of items have a partial order with respect to the subset operator i.e. a set is more important than its proper subsets. This induces a lattice where nodes correspond to itemsets and arcs correspond to the subset relation. To illustrate the idea behind the Apriori principle, consider the itemset lattice shown in Figure.3. Suppose $\{b, c, d\}$ is a frequent itemset. Clearly, any transaction that contains b, c, d must also contain its subsets, $\{b, c\}, \{c, d\}, \{b, d\}, \{b\}, \{c\}, and\{d\}$. As a result, if $\{b, c, d\}$ is frequent, then all subsets of $\{b, c, d\}$ (i.e., the itemsets in the covered region in Fig.3 must also serve as frequent. Conversely, if an itemset such as $\{a, c\}$ is infrequent, then all of its supersets must be infrequent too. As illustrated in Figure.4, the entire sub-graph containing the supersets of $\{a, c\}$ can be pruned immediately once $\{a, c\}$ is considered to be infrequent. If we know that $\{a, c\}$ is infrequent, we never need to check any of the supersets. This fact is employed in support-based pruning. In contrasting the support measure, confidence measure has no

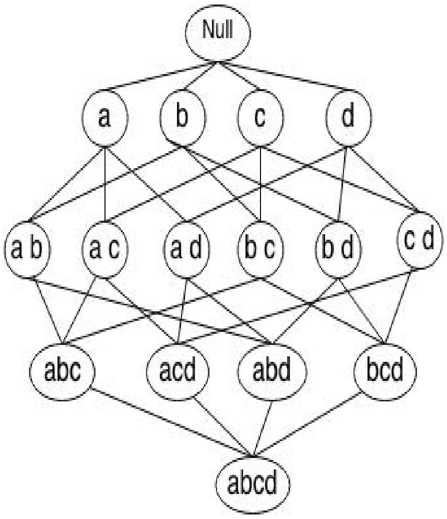

Figure 2: Itemset lattice.

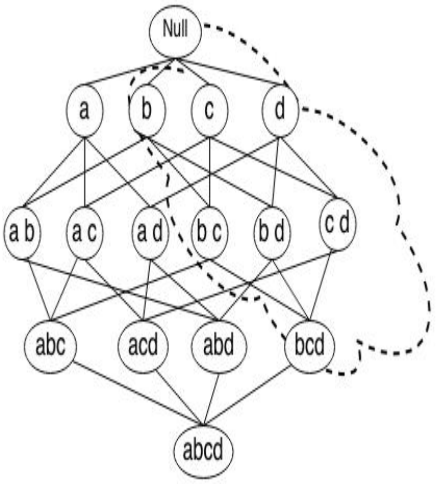

Figure 3: Apriori principle for frequent itemsets.

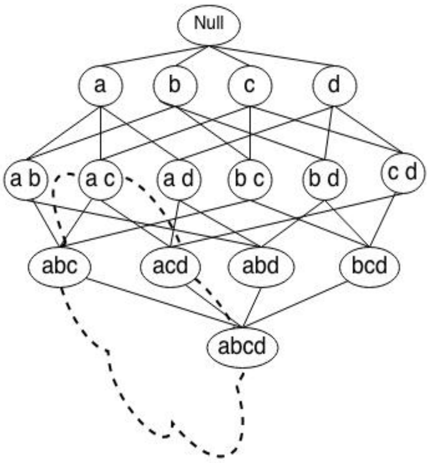

Figure 4: Apriori principle for infrequent itemsets.

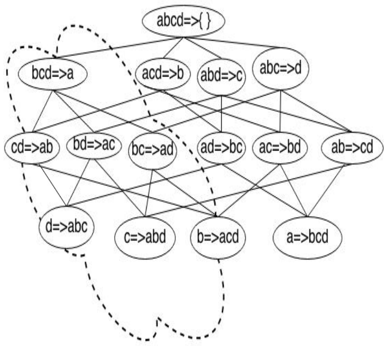

Figure 5: Apriori principle for confidence based pruning.

monotone property. Figure.5 shows a lattice structure for the association rules generation from the frequent itemset $\{a, b, c, d\}$. If any node in the lattice has low confidence, subsequently according to the complete sub-graph spanned by the node can be pruned straight away. Suppose the confidence for $\{bcd\} \Rightarrow \{a\}$ is low. All the rules containing item a in its consequent, including $\{cd\} \Rightarrow \{ab\}, \{bd\} \Rightarrow \{ac\}, \{bc\} \Rightarrow \{ad\}, and\{d\} \Rightarrow \{abc\}$ can be ruled out.

1.3 Sanitization

We focus on the problem of transforming a database into a new one that conceals some strategic patterns (restrictive association rules) and at the same time preserving the general patterns and trends from the original database. Data Sanitization is the process of making sensitive information in non-production databases safe from wider visibility. The process of transforming an original database into a sanitized one is called data sanitization [12]. The sanitization process acts on the data to eliminate or conceal a group of restricted association rules which contain sensitive information. It offers a suitable balance between a need for privacy and knowledge discovery. Figure.6 shows the process of sanitization.

1.4 Literature Review

Distortion based methods work by selecting precise things to incorporate into (or prohibit from) Preferred transactions of the original dataset so as to encourage the hiding of of the sensitive frequent itemsets. Two of the most habitually utilized techniques

Original Database
Sanitization process
Sanitized Database

Figure 6: Process of sanitization.

for information distortion includes the exchange of values among the transactions [13], [14], and the cancellation of exact items from the dataset.

Oliveira and Zaane [3] were the first to present multi rule hiding methodologies. The proposed methods are effective and require two scans of the dataset. In the first scan, an index is made to accelerate the methodology of recognizing the sensitive transactions. In the second scan, the algorithms sanitize the dataset by specifically removing the individual items that suit the covering up of the sensitive information. Three item restriction- based methods (known as MinFIA, MaxFIA, and IGA) are suggesting that specifically removes the items from transactions that are supported by the sensitive rules.

A more able methodology than that of [3] was introduced by Oliveira and Zaiane in [15]. The proposed method, called SWA, is a proficient, versatile, one-scan heuristic, which aims at giving a balance between the need of security knowledge discovery in association rule hiding. It accomplishes concealing multiple rules in one and only pass through the dataset, paying little mind to its size or the amount of sensitive rules that need to be restricted.

Amiri [16] proposes three effective, multiple association rules hiding heuristics that beat SWA by showing higher data utility and lower distortion. The first approach, called Aggregate, processes the transaction that supports the most sensitive and the minimum non-sensitive itemsets is chosen and prohibited from the dataset. Essentially, the Dis- Aggregate methodology goes for expelling individual items from transactions, instead of removing the whole transaction. The third approach, called Hybrid, is a unification of the two previous algorithms.

Wang and Jafari [17] propose two modification algorithms that go for the hiding of sensitive association rules which holds the sensitive items on their left-hand side . The principal procedure, called ISL, reduces the confidence of a sensitive rule by increasing the support of the itemset in its left-hand side. The second approach, called DSR, reduces the confidence of the rule by decreasing the support of the itemset in its right-hand side.

Simovici DA [18] Constructed a lattice like diagram of the database. At that point, sensitive itemsets hiding was attained by a greedy and iterative traversal of its prompt subset through the diagram, and distinguished the unified with the greatest support as the new candidate to be hidden. Additionally, by grouping the sensitive association rules focused around certain criteria, a group of sensitive rules might be hidden at once. Subsequently, less transactions are changed for concealing all the sensitive rules.

G.v. Moustakides [19] introduced two new algorithms which apply the thought of the maxmin condition, keeping in mind the end goal to minimize the effect of the concealing procedure to the altered positive boarder, which is structured by removing the sensitive itemsets and their super itemsets from the lattice of frequent itemsets.

Divanis AG [20] Proposed a strategy which does not reduce the support of the sensitive itemsets, however, added the new transactions to the database focused around minimizing the consequences for the non sensitive itemsets.

Shyue-Liang Wang [21] proposed an effective data mining algorithm MSI to keep up disinfected informative association rule sets. The proposed calculation incrementally disinfected the included dataset and united with the previously sanitized database with one database filtering using pattern-inversion trees.

Dai BR [22] Proposed a method which can hide sensitive frequent patterns in the incremental environment. At the point when the database is updated, the strategy utilizes a format based idea to control the support of sensitive patterns. A compact data structure SPITF was contrived to store all sensitive transactions such that we can choose perfect transactions from the entire database without losing any chance and can manage the incremental dataset effectively.

Hai Quoc Le [23] Presented a novel method to conceal a set of sensitive association rules in the context of imparting the data. The proposed methodology focused around an intersection lattice of the frequent itemsets to discover precisely items and transactions that might be adjusted to diminish the confidence of a sensitive association rule, yet less effect to alternate itemsets.

T.-P. Hong, et al. [24] proposed a novel greedy based methodology called Sensitive Items Frequency-Inverse Database Frequency (SIF-IDF) to evaluate the amount of transactions related with the given sensitive itemsets. It utilizes the idea of TF-IDF for decreasing frequency of sensitive itemsets in data sanitization.

Hai Quoc Le, et al. [1] anticipated a method to hide a set of sensitive association rules using the distortion method. The proposed method is focused around the Intersection lattice of frequent itemsets. By analyzing the characteristics of an intersection lattice of the frequent itemsets FI, itemsets in the generating set of FI (Gen(FI)) was indicated to be vulnerable against a decrease in the hiding methodology. To minimize the side effects, the HCSRIL method determine the victim item and least number of transactions such that the alteration of this thing causes the minimum measure of effect on itemsets in Gen(FI).

Janakiramaiah, et al. [25] Proposed a new data distortion method to hide sensitive association rules. The impact factor of the items in the rules will be calculated based on the number of non sensitive frequent items that are affected by the removal of that item. The item with affects the less non sensitive itemsets will be selected for alteration in order to improve the accuracy of the sanitized dataset.

2 Problem Formulation and Proposed Framework

We assume that we are provided with a database D, consisting of n transactions, minimum support (σ), confidence (δ) threshold set by the owner of the data. After performing association rule mining in D using thresholds σ and δ , we produce a set of association rules, denoted as R, among which a subset R_s of R contains rules which are considered to be sensitive from the owners perspective. Given the set of sensitive association rules R_s, the goal of association rule hiding methodologies is to construct a new, sanitized database D^1 from D, which achieves, to protect the sensitive association rulesR_s from disclosure, while minimally affecting the non-sensitive rules existing in R (i.e., those in $R - R_s$.) The hiding of a sensitive association rule corresponds to a lowering of its significance, illustrated in terms of support or confidence, in the resulting database. To hide a sensitive rule, the privacy preserving algorithm modifies the original database D in such a way that, when the sanitized database D^1 is mined at the same (or a higher) levels of confidence and support, the association rules that are discovered are all non-sensitive.

In the proposed framework, initially the association rules (R), will be mined from the database D by using an association rule mining algorithm (AR). Then the Data owner will specify the sensitive association rules(R_s), which need to be hidden from mining. By considering sensitive association rules and the original dataset as input to our proposed algorithm will release a sanitized dataset D^1. Then by applying any association rule mining algorithm on the sanitized dataset D^1. We can mine all association rules which are mined from original dataset D except the sensitive association rules ($R - R_s$) . The proposed framework is shown in Figure.7.

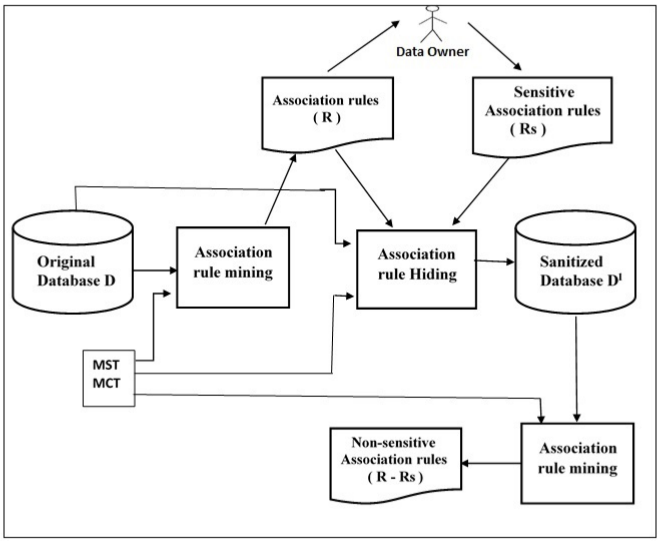

Figure 7: Proposed framework.

3 The Proposed Association Rule Hiding Algorithm

3.1 Association Rule Hiding Process

Let R_s be a group of sensitive association rules. Assume that the sensitive rule that needs to be hidden in each time is denoted by $X \Rightarrow Y$. Our method aims at hiding $X \Rightarrow Y$ by removing an item in X or Y from a number of transactions until $\alpha(X \Rightarrow Y) \leq \sigma$ or $\beta(X \Rightarrow Y) \leq \delta$. To reduce the side effects, we propose a heuristic association rule hiding algorithm based on four steps.

Step 1. Grouping the Rules: In this step we group the sensitive association rules in to the number of clusters in such a way that the rules in one cluster must share a common item set in Y. For each cluster a label will be assigned as the item which is having less support in the data set among the items that are shared by the rules in that cluster. Sort the clusters in decreasing order of their size. A rule may exist in more than one cluster if it shares the items with more than one rule in R_s. To eliminate the duplication, consider every pair of clusters and for every common rule in that couple of clusters apply the following process. If the size of the clusters in the pair is not equal then remove the common rule from smallest cluster. Otherwise, remove the rule from the cluster with a label which is having the smallest support in the data set. The process of grouping shown in Example 1.

Example 1. Let D be the transactional dataset present in Table 1.

Assume that the rules to be hidden are 1) $10 \Rightarrow 4, 6, 8$ 2) $8 \Rightarrow 4$ 3) $4 \Rightarrow 6, 8, 10$ 4) $2 \Rightarrow 6$ 5) $10 \Rightarrow 4, 6$.

The above rules can be classified into two groups. First Group consists of rules 1, 3, 4, 5 which are hav-

TID	Ids of Items Purchased
1	2,3,4,5,6,7,8,10
2	2,4,6,8,10
3	6,8,9,10
4	2,6,8,10
5	3,4,6,8,10
6	2,4,5,6,8
7	2,8,10
8	3,4,6,7,8,9,10
9	3,4,6,8,10
10	3,4,6,8,10
11	2,4,5,6,8,9,10
12	2,3,4,5,6,8,10
13	2,4,5,6,8,10
14	4,5,6,8
15	2,3,4,5,6,7,9
16	2,4,6,7,10
17	2,3,5,7,8
18	3,4,6,8,10
19	3,4,6,8,10
20	2,4,5,6,8,10

Table 1: Transactional Dataset.

ing the common item 6. So the label of the group is 6. Second Group consists of rules 1, 2, 5 which share item 4 as common hence label of the group two is 4. Rules 1, 5 are existing in both the groups. To eliminate the duplication consider Group one and Group two pair. The size of group one is 4 and the size of group two is 2. As these two sizes are not equal common rules will be deleted from smallest group i.e from group two. So finally after grouping

Group 1: 10 ⇒ 4, 6, 8; 4 ⇒ 6,8,10 ; 2 ⇒ 6 ; 10 ⇒ 4, 6 – with label 6

Group 2: 8 ⇒ 4 — with label 4.

Step 2. Transaction specification: This step aims to work out the minimum number of transactions that have to be modified in order to hide the sensitive rule. Let this number be denoted by n. Then, to hide the rule $X \Rightarrow Y$, we must have

$\alpha(XY) - n < \sigma$ or $\frac{\alpha(XY)-n}{\alpha(X)} < \delta$.

This Implies that $n > \alpha(XY) - \sigma$ or $n > \alpha(XY) - \lceil \alpha(X) * \delta \rceil$

Thus $n = min\{\alpha(XY) - \sigma + 1, \alpha(XY) - \lceil \alpha(X) * \delta \rceil + 1\}$

In addition to this, identifying the order of transactions for item modification is an important step towards reducing the side effects. Let T be the set of transactions in the dataset D. Thus, to attain a minimum impact on the non-sensitive association rules, T needs to be sorted in descending order of SIF (Sensitive Item Frequency) of each transaction. SIF of each transaction can be calculated as the degree of sensitivity divided by the length of that transaction(SIF=number of sensitive items in the transaction/length of that transaction). From the sorted order of the transactions select the first n transactions. Example 2 shows the calculation of n for a sensitive rule.

Example 2. Consider the transactional dataset in Table 1. Let the rule be $6, 10 \Rightarrow 8$ with support 14 and confidence 18.5.

Let $\sigma = 10$ and $\delta = 70$.

n = min $\{\alpha(6, 8, 10) - \sigma + 1, \alpha(6, 8, 10) - \lceil \alpha(6, 10) * 0.7 \rceil + 1\}$

$= min\{14 - 10 + 1, 14 - \lceil 15 * 0.7 \rceil + 1\}$

$= min(5, 5)$

$= 5$

The transactions which support $6, 10 \Rightarrow 8$ in the descending order of their SIF are shown in Table 2 . Because of the value of n is 5, first 5 transactions will be selected for modification to hide the rule $6, 10 \Rightarrow 8$.

Step 3. Victim item specification: The victim item is the item that has to be modified to hide a rule such that modifying this item minimizes the side effects. Example 3 shows how the victim item selection can reduce the side effects of the hiding process.

Tid	Items purchased	SIF
3	6,8,9,10	0.75
4	2,6,8,10	0.75
2	2,4,6,8,10	0.6
5	3,4,6,8,10	0.6
9	3,4,6,8,10	0.6
10	3,4,6,8,10	0.6
18	3,4,6,8,10	0.6
19	3,4,6,8,10	0.6
13	2,4,5,6,8,10	0.5
20	2,4,5,6,8,10	0.5
8	3,4,6,7,8,9,10	0.4285
11	2,4,5,6,8,9,10	0.4285
12	2,3,4,5,6,8,10	0.4285
1	2,3,4,5,6,7,8,10	0.375

Table 2: sensitive transactions in descending order of their SIF.

Example 3: Given transactional dataset D in Table 1 and minimum thresholds $\sigma = 10$ and $\delta = 70\%$. Assume that the sensitive rule that needs to be hidden is $10 \Rightarrow 6, 8$. To hide this rule, we need to remove either 10, 6 or 8 from transactions supporting $\{6, 8, 10\}$. We compare the impact of non-sensitive association rules when modifying 10 or 6 or 8 in n

transactions. We need to remove either 10, 6 or 8 in n transactions from the transactions that are supported by $\{6, 8, 10\}$, to hide the sensitive rule $10 \Rightarrow 6, 8$. The value of n can be evaluated with step 2 as follows. $n = min\{\alpha(6, 8, 10) - \sigma + 1, \alpha(6, 8, 10) - \lceil\alpha(10) * \delta\rceil + 1\}$
$= min\{14 - 10 + 1, 14 - \lceil 17.5 * 0.7\rceil + 1\}$
$= min(5, 3)$
$= 3$
The selected 3 transactions from the transactions that are supported by $\{6, 8, 10\}$ based on their SIF are shown in Table.3. Removing 6 from transactions supporting $\{6, 8, 10\}$ may affect the entire non-sensitive association rules which contain 6 as one of the item in the rule. Non-sensitive association rules which contain 6 as one of the item and effected by the removal of item 6 in three transactions along with their support are shown in Table 4. The results of step 3 of the algorithm are shown in table.4. Finally, if we remove 6, five rules will be hidden. So Impact-Factor (6) =5. In the similar manner Impact-Factor (8) = 7. So removal 6 will have less impact on the non-sensitive association rules. So the victim item will be chosen as item 6.

TID	Ids of Items
3	6, 8, 9, 10
4	2, 6, 8, 10
2	2, 4, 6, 8, 10

Table 3: Selected Transactions.

Step 4. Updating the Dataset and Sensitive rules:

LHS	RHS	Sup	Redu	Modif Sup	Modif sup of LHS	Modif Conf
2	6	10	2	8	12	0.66
10	4,6,8	12	1	11	16	0.68
4	6,8,10	12	1	11	16	0.68
10	6,8	14	3	11	16	0.68
8	6,10	14	3	11	18	0.61

Table 4: Result of step 3 for item 6.

The victim item is then removed from n transactions of D. Apply the association rule mining algorithm to identify the frequent association rules on modified dataset and update the set of sensitive association rules.

Example 4. Consider the Example 3. Removal of item 6 to hide $10 \Rightarrow 6, 8$ will also affect non-sensitive rules of the item set $\{4, 6, 8, 10\}$. If any one of the rules of the effected itemset $\{4, 6, 8, 10\}$ is there in sensitive rules R_s that will also be hidden due to the hiding process of $10 \Rightarrow 6, 8$.

3.2 The Proposed HHSRIF(Heuristic for Hiding Sensitive Rules using Impact Factor) Algorithm

The HHSRIF algorithm aims to hide the set of sensitive association rules mined from a given transaction dataset D, that satisfies given minimum thresholds σ and δ. The function Revise (victim, n, D) aims to remove victim item from n transactions supporting that rule. The function Revise-rules (R) and Revise-rules (R_s) aims to prune out of R and R_s, the rules that have support and confidence less than thresholds σ and δ respectively.

Algorithm HHSRIF

Input: D-Transactional Dataset; R- Set of association rules;
R_s-Set of sensitive association rules; σ and δ Minimum support and confidence thresholds

Output: Sanitized Dataset D^1 from which non-sensitive association rules can still be mined.

Method:

```
Step 1: Group the sensitive rules into a set of groups $G, \{G/\forall g \in G, \forall SR_i, SR_j \in g$ and $SR_i, SR_j$ share the same item set l in consequent of the rule $\}$
For each $g \in G$
    Assign label $\mu$ to g such that $\mu \in I$ and $\forall\lambda \in I, \alpha(\mu, D) \leq \alpha(\lambda, D)$.
End For
Sort(G). //in descending order of size
For every pair $g_i$ and $g_j \in G$
    For each $SR_k \in g_i \cap g_j$
    IF size ($g_i$) $\neq$ size ($g_j$)
        Remove $SR_k$ from smallest ($g_i$, $g_j$);
    ELSE
        Remove $SR_k$ from group with label $\mu$ such that $\alpha(\mu, D) \leq \alpha(\lambda, D)$ and $\mu, \lambda$ are labels of either $g_1$ or $g_2$.
    End IF
    End For
End For
Step 2: Repeat
For each transaction j in D
    SIF[j]=degree of sensitivity/length of j.
End For
    T=Sort(D) //in descending order sif
Select a rule $X \Rightarrow Y$ from $g_i \in G$ such that $X \Rightarrow Y$ is shortest rule in $g_i$
```

```
$n = min\{\alpha(XY) - \sigma + 1, \alpha(XY) - [\alpha(X) * \delta] + 1\}$
N []=First n transactions from T.
    IF ( $\alpha(XY) - \sigma + 1$) > $(\alpha(XY) - \lceil\alpha(X) * \delta\rceil + 1)$
 z=Y
   ELSE
   z=X ∪ Y
 END IF
Step 3:For each item $k \in z$
For each $r \in R$
IF $k \subseteq r$
Add r to TR;
End IF
End For
For each rule $r \in TR$
   $\alpha(r) = \alpha(r) - \alpha1(r)$ where $\alpha1(r)$= support of r with respect to n
End For
count=0;
  For each rule j in TR
IF $((\alpha(j) < \sigma)$ or $(\beta(j) < \delta))$
count=count+1.
End IF
End For
ImpactFactor(k)=count.
End For
Victim=min (ImpactFactor [])
Step 4: Revise (victim, n, D);
Revise-rules (R);
Revise-rules (G);
Until (G is Empty)
```

4 Running Example

Consider the dataset in Table.1 with minimum thresholds σ=10 and δ=70% . Let the set of sensitive association rules be $R_s = \{6, 10 \Rightarrow 4, 8; 4, 8 \Rightarrow 6; 4 \Rightarrow 6, 8; 6 \Rightarrow 4, 8; 6, 8 \Rightarrow 4, 10\}$ Next we apply the algorithm to hide R_s.

Step 1. Grouping R_s:

First the algorithm performs grouping (step 1). The groups after step 1 are shown in Table.5.

Groups	Rule
g1	6,10 ⇒ 4,8
	4 ⇒ 6,8
	6 ⇒ 4,8
g2	6,8 ⇒ 4,10
g3	4,8 ⇒ 6

Table 5: Result after grouping.

Step 2. Transaction specification:

The SIF of all transactions in sorted order are shown in Table.6. The rule that will be selected from g1 is $6 \Rightarrow 4, 8$. Calculate the number of transactions n for modification to hide the rule

$n = min\{14 - 10 + 1, 14 - \lceil 18 * 0.7\rceil + 1\}$

$= min\{5, 2\} = 2$

Next from Table.6 select two transactions (as n=2) shown in Table.7.

TID	Ids of Items Purchased	SIF
2	2,4,6,8,10	0.8
5	3,4,6,8,10	0.8
9	3,4,6,8,10	0.8
10	3,4,6,8,10	0.8
18	3,4,6,8,10	0.8
19	3,4,6,8,10	0.8
3	6,8,9,10	0.75
4	2,6,8,10	0.75
14	4 5 6 8	0.75
7	2,8,10	0.6666
13	2,4,5,6,8,10	0.6666
20	2,4,5,6,8,10	0.6666
6	2,4,5,6,8	0.6
16	2,4,6,7,10	0.6
8	3,4,6,7,8,9,10	0.5714
11	2,4,5,6,8,9,10	0.5714
12	2,3,4,5,6,8,10	0.5714
1	2,3,4,5,6,7,8,10	0.5
15	2,3,4,5,6,7,9	0.2857
17	2,3,5,7,8	02

Table 6: Transactions in descending order of SIF.

TID	Items
2	2,4,6,8,10
5	3,4,6,8,10

Table 7: Selected n transactions.

Step 3. Victim item selection: We have three items in both LHS and RHS i.e. 6, 4 and 8. This step aims at identifying the victim item whose removal affects non-sensitive rules to a less extent among 6, 4 and 8. For that we wish to calculate the impact factor of 6, 4 and 8. First, consider the item 4. To calculate the impact factor, consider the association rules which contain 4 either in LHS or RHS along with their support. Then identify the support of each rule with respect to n as reduction. Update the support of each rule as support-reduction and consider it as modified support when we remove 4 from n transactions. Then obtain the confidence of the rule based on the modified support. Then count the number of rules whose support is less than σ or confidence is less than δ. Store the count as impact-factor of 4. The calculations were presented in table9. From Table 9 Impact-Factor (4) = 12 and Impact-Factor (8) = 9. In the similar manner Impact-Factor (6) =9. The item which is having minimum impact-factor will be selected as victim i.e here item 8 is victim.

Step 4. Updating the Dataset and R_s groups: The selected victim item 8 in step 3 is now removed from selected two transactions i.e from TIDs 2 and 5. The modified dataset is shown in Table.10. Groups of R_s will also be updated. Here along with the selected rule $6 \Rightarrow 4, 8$, the other rules in that group will also be hidden i.e $4 \Rightarrow 6, 8$ and $6, 10 \Rightarrow 4, 8$. So group g1 becomes empty. Next the rule of g3 i.e $4 \Rightarrow 6, 8$ will be selected because it is having the shortest rule length. By implementing the above steps 1 to 4, Impact-Factor (4) =14, Impact-Factor (6) =12 and Impact-Factor (8) =15 i.e. item 6 will become the victim item and will be removed from one transaction i.e. TID 9. By updating the R_s groups along with group g3 rule, group g2 rules will also be hidden. So the set G is empty. All the sensitive rules will be hidden by extracting non-sensitive rules as much as possible from the sanitized dataset D^1 shown in Table.11.

5 Performance Measures

5.1 Hiding Failure:(HF)

When some sensitive association rules that cannot be hidden by the sanitization process, we call this problem as Hiding Failure, and are measured in terms of the percentage of sensitive association rules that is discovered from sanitized database D^1. The hiding failure is calculated as follows $HF = \frac{\sharp R_S(D^1)}{\sharp R_S(D)}$ where$\sharp R_S(D^1)$ denotes the number of sensitive association rules discovered from sanitized database D^1, and $\sharp R_S(D)$ denotes the number of sensitive association rules discovered from original database D.

5.2 Misses Cost/Lost rules:(MC)

Misses Cost is some non-sensitive association rules that can be discovered from the original database but cannot be mined from the sanitized database D^1. This happens when some non-sensitive association rules lose support or confidence below the minimum threshold values in the database due to the sanitization process. We call this problem as Misses Cost, and are measured in terms of the percentage of non-sensitive association rules that is not discovered from the sanitized database D^1. The misses cost is calculated as $MC = \frac{\sharp{\sim}R_S(D)-\sharp{\sim}R_S(D^1)}{\sharp{\sim}R_S(D)}$ where$\sharp \sim R_S(D)$ denotes the number of non-sensitive association rules discovered from original database D, and $\sharp \sim R_S(D^1)$ denotes the number of non-sensitive association rules discovered from sanitized databaseD^1.

LHS	RHS	Support	Modified Support	Modified Confi-dence
6	4	16	14	0.778
4	6	16	14	1
6,8	4	14	12	0.75
4,8	6	14	12	1
4,6	8	14	12	0.857
8	4,6	14	12	0.667
6	4,8	14	12	0.667
4	6,8	14	12	0.857
6,8,10	4	12	10	0.71
4,8,10	6	12	10	1.00
4,6,10	8	12	10	0.77
4,6,8	10	12	10	0.83
8,10	4,6	12	10	0.67
6,10	4,8	12	10	0.67
6,8	4,10	12	10	0.63
4,10	6,8	12	10	0.91
4,8	6,10	12	10	0.83
4,6	8,10	12	10	0.71
10	4,6,8	12	10	0.63
4	6,8,10	12	10	0.71
6,10	4	13	11	0.73
4,10	6	13	11	1.00
4,6	10	13	11	0.79
10	4,6	13	11	0.69
6	4,10	13	11	0.61
4	6,10	13	11	0.79
8	4	14	12	0.67
4	8	14	12	0.86
8,10	4	12	10	0.67
4,10	8	12	10	0.91
4,8	10	12	10	0.83
10	4,8	12	10	0.63
4	8,10	12	10	0.71
10	4	13	11	0.69
4	10	13	11	0.79

Table 8: Calculating the impact factors(Step 3) of 4 and 8. (Impact-Factor(4)=12)

5.3 Ghost rules/False rules/Artifactual Patterns:(GR)

Ghost rules occur when some artificial association rules are generated from D^1 as a product of the sanitization process. We call this problem as ghost

LHS	RHS	Support	Modified Support	Modified Confidence
2	8	10	8	0.67
6,8	4	14	12	0.86
4,8	6	14	12	1.00
4,6	8	14	12	0.75
8	4,6	14	12	0.75
6	4,8	14	12	0.67
4	6,8	14	12	0.75
6,8,10	4	12	10	0.83
4,8,10	6	12	10	1.00
4,6,10	8	12	10	0.77
4,6,8	10	12	10	0.83
8,10	4,6	12	10	0.77
6,10	4,8	12	10	0.67
6,8	4,10	12	10	0.71
4,10	6,8	12	10	0.77
4,8	6,10	12	10	0.83
4,6	8,10	12	10	0.63
10	4,6,8	12	10	0.63
4	6,8,10	12	10	0.63
8	4	14	12	0.75
4	8	14	12	0.75
8,10	4	12	10	0.77
4,10	8	12	10	0.77
4,8	10	12	10	0.83
10	4,8	12	10	0.63
4	8,10	12	10	0.63
8	6	16	14	0.88
6	8	16	14	0.78
8,10	6	14	12	0.92
6,10	8	14	12	0.80
6,8	10	14	12	0.86
10	6,8	14	12	0.75
8	6,10	14	12	0.75
6	8,10	14	12	0.67
10	8	15	13	0.81
8	10	15	13	0.81

Table 9: Calculating the impact factors(Step 3) of 4 and 8. (Impact-Factor(8)=9)

rules, and are measured in terms of percentage of the discovered association rules that are ghost rules. This is measured as $GR = \frac{|R^1|-|R\cap R^1|}{|R^1|}$ where $|R|$ and $|R^1|$ represent, respectively the set of association rules that can be generated from D and D^1.

TID	Ids of Items Purchased
1	2,3,4,5,6,7,8,10
2	2,4,6,10
3	6,8,9,10
4	2,6,8,10
5	3,4,6,10
6	2,4,5,6,8
7	2,8,10
8	3,4,6,7,8,9,10
9	3,4,6,8,10
10	3,4,6,8,10
11	2,4,5,6,8,9,10
12	2,3,4,5,6,8,10
13	2,4,5,6,8,10
14	4,5,6
15	2,3,4,5,6,7,9
16	2,4,6,7,10
17	2,3,5,7,8
18	3,4,6,8,10
19	3,4,6,8,10
20	2,4,5,6,8,10

Table 10: Modified dataset after step 4.

TID	Ids of Items Purchased
1	2,3,4,5,6,7,8,10
2	2,4,6,10
3	6,8,9,10
4	2,6,8,10
5	3,4,6,10
6	2,4,5,8
7	2,8,10
8	3,4,6,7,8,9,10
9	3,4,8,10
10	3,4,6,8,10
11	2,4,5,6,8,9,10
12	2,3,4,5,6,8,10
13	2,4,5,6,8,10
14	4,5,6
15	2,3,4,5,6,7,9
16	2,4,6,7,10
17	2,3,5,7,8
18	3,4,6,8,10
19	3,4,6,8,10
20	2,4,5,6,8,10

Table 11: Final Sanitized dataset D^1.

5.4 Difference between the original and sanitized datasets(Diff(D,D^1))

We could measure the dissimilarity between the original and sanitized database by simply comparing their histograms.

$Diff(D, D^1) = \frac{1}{\sum_{i=1}^{n} fD(i)} \sum_{i=1}^{n} [fD(i) - fD^1(i)]$

where fx(i) represents the frequency of the i^{th} item in the dataset x,and n is the number of distinct items in the original dataset.

6 Experiments and Evaluation

The data set for our evaluation have been placed in IEEE ICDM03 as the file name Retail.dat and has been available in Online at http://mi.ua.ac.be/data/. This dataset is provided to researchers in the area of data mining in order to support the analysis of their models. Retail dataset was contributed by Tom Brijs [26] and includes the retail market basket data from an unknown Belgian Retail Store. The dataset was gathered over three non-consecutive eras from the middle of December 1999 to the end of November 2000. The dataset comprises of 88,162 transactions and 16,469 product IDs. Table 12 shows the layout of the Retail.dat dataset, where the first column is the transaction identification (TID). Each transaction encloses the IDs (items) of products that were procured by a customer. The IDs are isolated by a space. For example, transaction with TID = 1 contains 30 products, which have the product IDs numbered 0, 1, 2, . . ., 29, which were procured by the first customer.

TID	Purchases
1	2 3 4 5 6 7 8 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
...	...
5815	39 48 89 1791
...	...
50000	39 3486 3827 4305
...	...
71000	11 48 279 301 424 1678
...	...
88162	32 39 205 242 1393

Table 12: Transactional Dataset of the Belgian retailer.

In this evaluation, we compared the HHSRIF algorithm with the HCSRIL algorithm presented in [1], SIF-IDF algorithm presented in [24] to assess the side effects and computational complexity. The HCSRIL algorithm uses intersection lattice of frequent item sets to reduce the side effects when compared with SIF-IDF algorithm. The SIF-IDF algorithm was a greedy based approach which assesses the quality of the transaction for reducing the frequency of sensitive patterns. The dataset was used for the testing is Retail.dat.

To observe the performance of the HHSRIF, HCSRIL and SIF-IDF algorithms, we considered K-fold validation method with K value at 5. The K-fold validation method randomly divides the entire set of association rules that can be mined from the given data set into a number of groups such that each group contains K rules. From the Retail.dat data set total of 236 association rules were mined with minimum support threshold MST=0.01 and minimum confidence threshold MCT=0.1. By applying the K-fold validation with K=5, the rules were randomly divided into 48 groups (47 groups, each contain 5 rules and 48th group contains only one rule). Then we applied the algorithms HHSRIF, HCSRIL, and SIF-IDF on 47 sets and results were discussed below.

We evaluate the performance of the algorithms based on four metrics, including Misses'cost, Artifacts or Ghost Rules, Hiding Failure and Accuracy of the Sanitized Dataset (difference between D and D^1). The efficacy of these algorithms with respect to the four metrics is shown below.

Misses cost means the percentage of the non sensitive data that is lost in the sanitization process. It can be measured in terms of frequent item sets and association rules. Figure 8 shows the efficiency of the proposed algorithm by minimizing the misses cost in terms of Frequent Item Sets (FIS) in the experiment conducted by K-fold validation method i.e. With 47 sets each contains five sensitive rules. In view of that, the HHSRIF algorithm attained improved results on any set among the 47 sets, in reducing the lost FIS (non-sensitive) compared with HCSRIL and SIF-IDF algorithms. Figure 9 shows the efficiency of the proposed algorithm by minimizing the misses cost in terms of Association Rules (AR). In the examination, the HHSRIF algorithm attained improved results on any set among the 47 sets, in reducing the lost AR (non-sensitive) compared with HCSRIL and SIF-IDF algorithms. Figure 11 shows the competence of the proposed algorithm by minimizing the misses cost in Frequent Item Sets (FIS) by considering the number of sensitive rules one to five. As the number of sensitive rules increases the percentage of lost FIS also increases in a large extent in the SIF-IDF and HCSRIL algorithms. In view of that, the HHSRIF algorithm attained improved results in reducing the lost FIS (non-sensitive) as the number of sensitive rules increased when compared to HCSRIL and SIF-IDF algorithms. Figure 12 shows the proficiency of the projected algorithm in reducing the misses cost in association rules

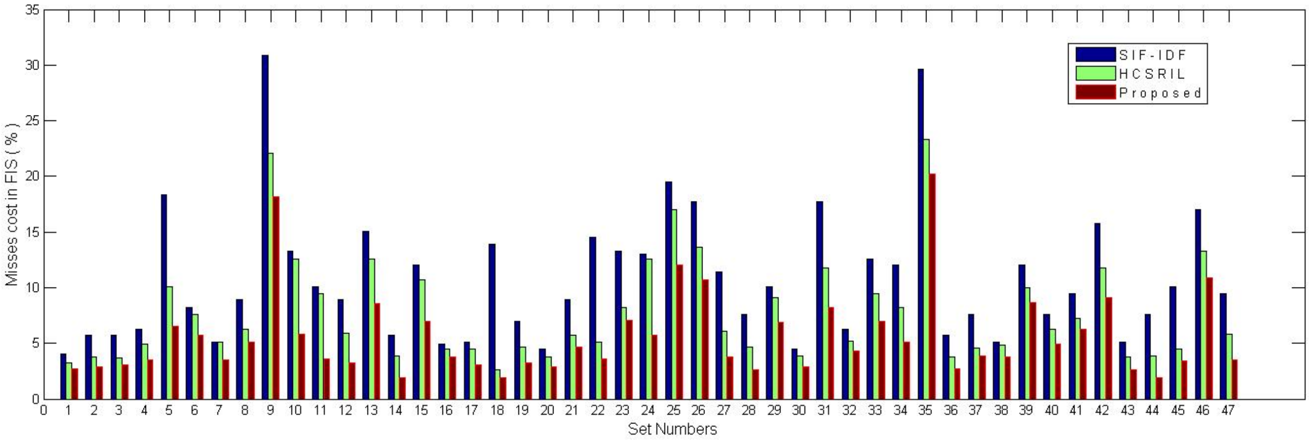

Figure 8: Comparison of Misses Cost in FIS based on individual sets of Sensitive Rules .

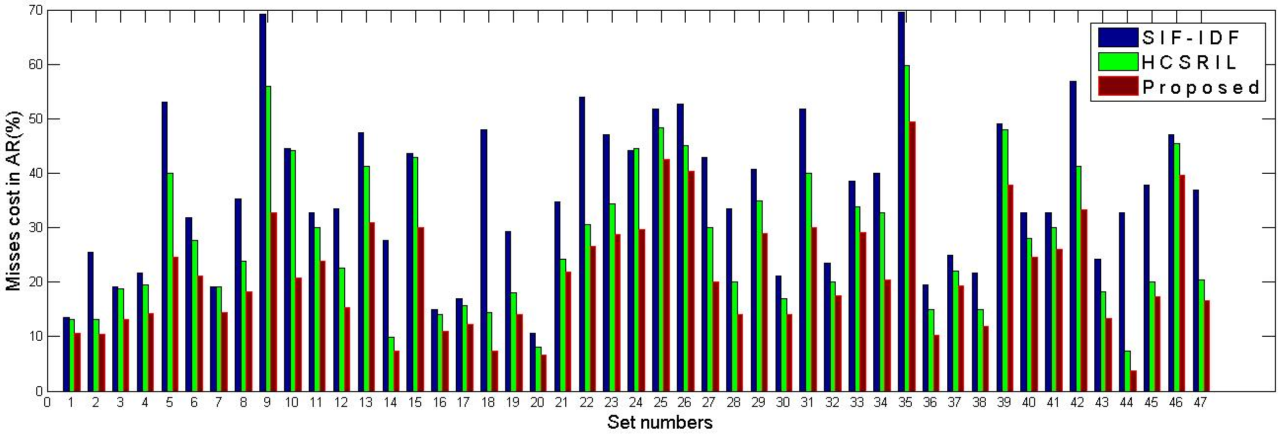

Figure 9: Comparison of Misses Cost in AR based on individual sets of Sensitive Rules.

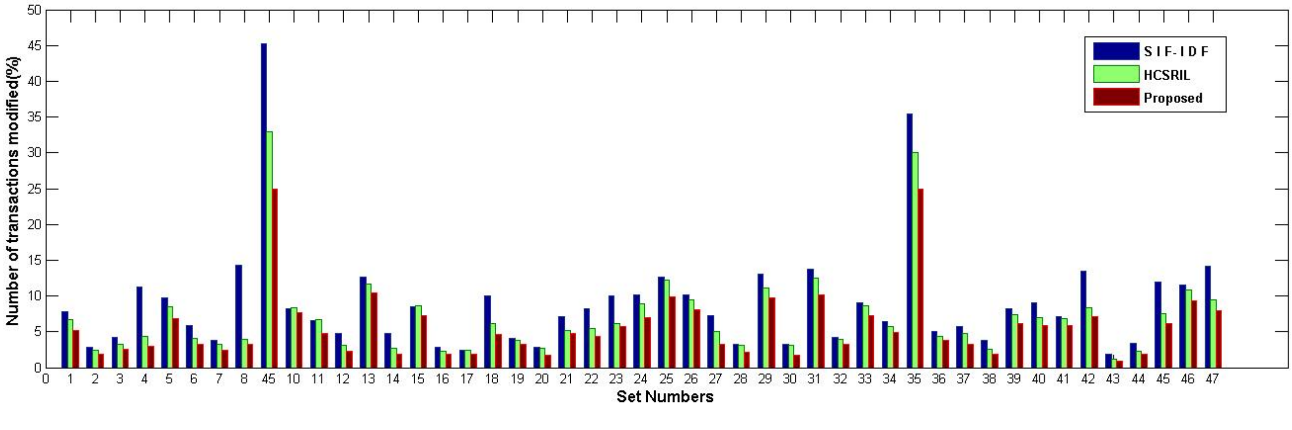

Figure 10: Comparison of Accuracy of Dataset based on individual sets of Sensitive Rules.

(AR) by considering the number of sensitive rules one to five. As the number of sensitive rules increases the percentage of lost AR also increases in a great extent of the SIF - IDF and HCSRIL algorithms. In view of that, the HHSRIF algorithm attained enhanced results in reducing the lost AR (non-sensitive) as the number of sensitive rules enlarged when compared to HCSRIL and SIF-IDF algorithms.

Accuracy or Difference (D, D^1) means the percentage of the number of transactions modified in the data set due to the sanitization process. The accuracy of the sanitized data set increases, as the number of transactions modified in the original data set decreases. Figure 10 shows the efficiency of the proposed algorithm in minimizing the Difference between D and D^1 in the experiment conducted by K-fold validation method i.e. in 47 sets each contains five sensitive rules. In view of that, the HHSRIF algorithm attained improved results on any set among the 47 sets, in reducing the difference between the D and D^1(number of transactions altered) when compared with HCSRIL and SIF-IDF algorithms. Figure 13 shows the proficiency of the projected algorithm in reducing the difference between D and D^1(number of transactions altered) by considering the number of sensitive rules one to five. As the number of sensitive rules increases the percentage of the number of transactions modified also increases to a great extent in the SIF-IDF and HCSRIL algorithms. In view of that, the HHSRIF algorithm achieved enhanced results in decreasing the difference between D and D^1 as the number of sensitive rules enlarged when compared to HCSRIL and SIF-IDF algorithms.

If any sensitive rules were disclosed when mine the sanitized data set, and then it is termed as hiding failure. The percentage of hiding failure for HHSRIF, HCSRIL and SIF-IDF algorithms, in 47 sets which are generated by K-fold validation method is 0Ghost rules (Artifactual patterns) mean new rules that are revealed from sanitized data set which are not mined from the original data set. With 47 sets of rules which are created with K-fold validation method the three algorithms HHSRIF, HCSRIL and SIF-IDF will not generate any new rules which are not disclosed when mine the original dataset. When mining the sanitized data set that was released by the three algorithms, no further rules will be disclosed.

In summary, the evaluation shows that the proposed algorithm HHSRIF yields excellent results when compared to HCSRIL and SIF-IDF algorithms in minimizing the side effects and data distortions.

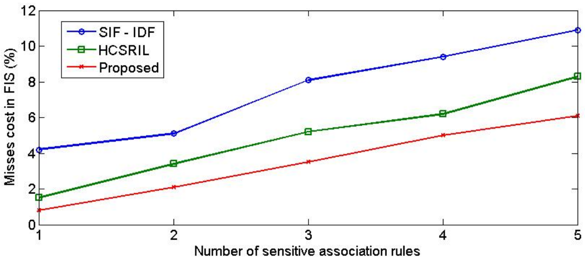

Figure 11: Comparison of Misses Cost in FIS based on Number of Sensitive Rules.

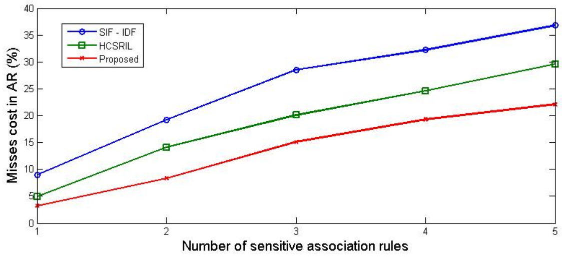

Figure 12: Comparison of Misses Cost in AR based on Number of Sensitive Rules.

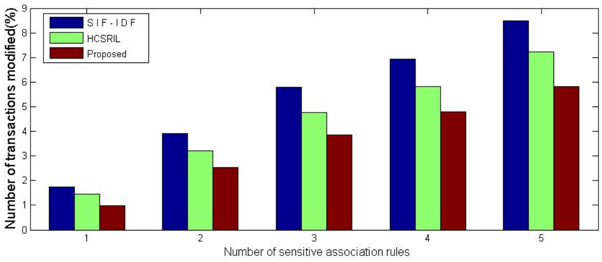

Figure 13: Comparison of Difference between the original and sanitized datasets based on Number of Sensitive Rules.

7 Conclusion

Association rule hiding is a significant concern in the risk management of enterprises when data are shared with others. Association rule hiding aims to smooth the progress of enterprises to stay away from the risks, which are caused by sensitive knowledge leakage by removing sensitive association rules from the database before sharing. A novel heuristic algorithm is proposed to hide from view a set of sensitive association rules using the distortion technique. The proposed algorithm is based on the item lattice of frequent association rules. By analysing the characteristics of the item lattice of frequent association rules, impact factors of the items in the sensitive rule will be estimated as number of non-sensitive rules that will be affected by removing that item. To reduce the side effects, the proposed algorithm precise the victim item and minimum number of transactions such that the modification of this item causes the slightest amount of impact on non sensitive association rules. The proposed algorithm was then applied in the risk avoidance of a retailer, when the retailer's data was shared. The results show that our approach outperforms earlier work and can be used in continuing and future enterprises. These contributions create more encouraging conditions for organizations planning to share their data with their partners, for mutual benefit and provide a power to the continued progress of their businesses. The future research direction on this topic includes expanding the tool-kit of privacy-preserving algorithms by developing primitives for the core data mining operations used today and make the algorithms and analyses applicable to a rapidly expanding variety of input data.

References:

[1] Le Hai Quoc; Arch int Somjit; Nguyen Huy Xuan; Arch-int Ngamnij. Association rule hiding in risk management for retail supply chain collaboration. *Computers in Industry*, 5 2013.

[2] Vassilios S. Verykios, Ahmed K. Elmagarmid, Elisa Bertino, Yucel Saygin, and Elena Dasseni. Association rule hiding. *IEEE Trans. on Knowl. and Data Eng.*, 16(4):434–447, April 2004.

[3] Oliveira Stanley R. M. and Osmar R. Privacy preserving frequent itemset mining. In *Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14*, pages 43–54, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.

[4] Xingzhi Sun and P.S. Yu. A border-based approach for hiding sensitive frequent itemsets. In *Data Mining, Fifth IEEE International Conference on*, pages 8 pp.–, Nov 2005.

[5] Bertino Elisa, Lin Dan, and Jiang Wei. A survey of quantification of privacy preserving data mining algorithms. In CharuC. Aggarwal and PhilipS. Yu, editors, *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, pages 183–205. Springer US, 2008.

[6] Lindell; Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15, 2008.

[7] Aris Gkoulalas-Divanis and Vassilios S. Verykios. Association rule hiding for data mining. In *Advances IN DATABASE SYSTEMS, Springer New York*, volume 5012, pages 99–103, 2010.

[8] Piotr Andruszkiewicz. Reduction relaxation in privacy preserving association rules mining. In Morzy Tadeusz, Hrder Theo, and Robert Wrembel, editors, *Advances in Databases and Information Systems*, volume 186 of *Advances in Intelligent Systems and Computing*, pages 1–8. Springer Berlin Heidelberg, 2013.

[9] Charu C Aggarwal and Philip S Yu. Privacy-preserving data mining : models and algorithms. In *Privacy-preserving data mining : models and algorithms*. New York Springer, 2008.

[10] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databeses. In *ACM SIGMOD Intl. Conf. Management of Data*, pages 207–216, 1993.

[11] R. Agrawal, T. Imielinski, and A. Swami. A performance perspective. In *IEEE Transactions on Knowledge and Data Engineering*, pages 914–925, 1993.

[12] Askari Mina, Safavi-Naini Reihaneh, and Barker Ken. An information theoretic privacy and utility measure for data sanitization mechanisms. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 283–294, New York, NY, USA, 2012. ACM.

[13] Atallah, Bertino E. M.J., Elmagarmid, A.K., Ibrahim, M. Verykios, and V.S. Disclosure limitation of sensitive rules. In *In: Proceedings of the IEEE Knowledge and Data Engineering Workshop*, page 4552, 1999.

[14] E. Dasseni, V.S.Verykios, A.K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *In Proceedings of the 4th international workshop on Information Hiding*, pages 369–383, 2001.

[15] S.R.M. Oliveira and O.R. Zaiane. Algorithms for balancing privacy and knowledge discovery in association rule mining. In *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International*, pages 54–63, 2003.

[16] Ali Amiri. Dare to share: Protecting sensitive knowledge with data sanitization. *Decision Support Systems*, 43, 2007.

[17] Shyue-Liang Wang and Jafari A. Using unknowns for hiding sensitive predictive association rules. In *Systems, Man and Cybernetics, 2005 IEEE International Conference*, volume 1, pages 164–169, 2005.

[18] D.A. Simovici and C. Djeraba. *Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics*. Advanced Information and Knowledge Processing. Springer, 2008.

[19] George V. Moustakides; Vassilios S. Verykios. A maxmin approach for hiding frequent itemsets. *Data and Knowledge Engineering*, 65, 2008.

[20] Aris Gkoulalas-Divanis and Vassilios S. Verykios. Exact knowledge hiding through database extension. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):699–713, 2009.

[21] Shyue-Liang Wang. Maintenance of sanitizing informative association rules. *Expert Systems with Applications*, 36(2, Part 2):4006 – 4012, 2009.

[22] L.-H. Chiang B.-R. Dai. Hiding frequent patterns in the updated database. In *Information Science and Applications (ICISA), 2010 International Conference*, pages 1–8, 2010.

[23] S. Arch-int H. Q. Le. A conceptual framework for privacy preserving of association rule mining in e-commerce. In *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference*, pages 1999–2003. ICIEA, 2012.

[24] Hong Tzung-Pei; Lin Chun-Wei; Yang Kuo-Tung; Wang Shyue-Liang. Using tf-idf to hide sensitive itemsets. *Applied Intelligence*, 38, 6 2013.

[25] Janakiramaiah Bonam. RamaMohan Reddy A. Kalyani G. Privacy preserving association rule mining based on the intersection lattice and impact factor of items. *IJCSI International Journal of Computer Science Issues*, 10, 11 2013.

[26] T. Brijs. Retail market basket data set. In *FIMI-2003*. Workshop on Frequent Itemset Mining Implementation FIMI 03, 2003.