Privacy Preserving Association Rule Mining based on the Intersection Lattice and Impact Factor of Items

Janakiramaiah Bonam[1], Dr RamaMohan Reddy A[2] and Kalyani G[3]

**[1]Department of Computer Science and Engineering, DVR & Dr HS MIC College of Technology
Vijayawada, Andhra Pradesh, India**

[2] Department of Computer Science and Engineering, S.V.University, Tirupati, India

**[3]Department of Computer Science and Engineering, DVR & Dr HS MIC College of Technology
Vijayawada, Andhra Pradesh, India**

Abstract

Association Rules revealed by association rule mining may contain some sensitive rules, which may cause prospective threats towards privacy and protection. A number of researchers in this area have recently made efforts to preserve privacy for sensitive association rules in transactional databases. In this paper, we put forward a heuristic based association rule hiding algorithm to get rid of the sensitive knowledge from the released database based on the intersection lattice of an item. The projected algorithm specifies the victim item based on the concept of impact factor of an item in the sensitive rule on the non sensitive frequent item sets. The impact factor of an item in the sensitive association rule is equal to the number of non sensitive frequent item sets that are affected by removing that item from the required number of transactions. Lower the impact factor of an item, lower is its effect on the non sensitive frequent item sets. Proposed algorithm exhibits the concept of intersection lattice and impact factor to conceal several rules by modifying less significant number transactions. As modifications are fewer, data excellence is very less exaggerated.

***Keywords**: Association Rules, Privacy Preserving, Intersection lattice, Sanitization, Data distortion.*

1. Introduction

Association rule mining extracts novel, hidden and useful patterns from huge repositories of data. These patterns are useful for effective analysis and decision making in telecommunication network, marketing, business, medical analysis, website linkages, financial transactions, advertising and other applications. The sharing of frequent rules can bring lot 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. On demand to various mismatched requirements of data sharing, privacy preserving and knowledge discovery, Privacy Preserving Data Mining (PPDM) has become a research hotspot in data mining. Simply, the association rule hiding problem is to hide secret, sensitive rules contained in data from being discovered, while without losing non-sensitive at the same time. The problem of frequent association rules hiding motivated many authors [5], [7], [10], [14], and proposed different approaches. The majority of the proposed approaches can be classified along two principal research directions: (i) Data hiding approaches and (ii) Knowledge hiding approaches.

1.1 Data hiding approaches

Data hiding methods [3], [11] collect methodologies that explore how the privacy of raw data, or information, can be maintained before the course of mining the data. The approaches of this category aim at the removal of confidential or private information from the original data prior to its discloser and operate by applying techniques such as transformation, generalization, perturbation and sampling, etc.

1.2. Knowledge hiding approaches

These approaches involve methodologies that aim to protect the sensitive data mining results rather than the raw data itself, which were produced by the application of data mining tools on the original database. These can be further classified into two subcategories: Data Distortion techniques and Data Blocking techniques. Data Distortion [4],[5],[8],[13] is implemented by deleting or adding items to reduce the support of the sensitive rule, while data blocking [9],[14] is implemented by replacing certain items with a question mark (?) to make the support of the sensitive rule uncertain.

2. Related Works

Distortion based approaches operate by selecting specific items to include to (or exclude from) selected transactions of the original database in order to facilitate the hiding of the sensitive frequent itemsets. Two of the most commonly employed strategies for data distortion involve the swapping of values between transactions [5][13], as well as the deletion of specific items from the database [14].

Atallah [13] were the first to propose an algorithm for the hiding of sensitive association rules through the reduction in the support of their generating itemsets.

Dasseni [5] generalize the hiding problem in the sense that they consider the hiding of both sensitive frequent itemsets and sensitive association rules. The authors propose three single rule heuristic hiding algorithms that are based on the reduction of either the support or the confidence of the sensitive rules, but not both. In all three approaches, the goal is to hide the sensitive rules while minimally affecting the support of the non-sensitive itemsets.

Verykios [17] extend the previous work of [5] by improving and evaluating the association rule hiding algorithms of [5] for their performance under different sizes of input datasets and different sets of sensitive rules.

Oliveira [14] were the first to introduce multiple rule hiding approaches. The proposed algorithms are efficient and require two scans of the database, regardless of the number of sensitive itemsets to hide. During the first scan, an index file is created to speed up the process of finding the sensitive transactions and to allow for an efficient retrieval of the data. In the second scan, the algorithms sanitize the database by selectively removing the least amount of individual items that accommodate the hiding of the sensitive knowledge. Three item restriction-based algorithms (known as MinFIA, MaxFIA, and IGA) are proposed that selectively remove items from transactions that support the sensitive rules.

A more efficient approach than that of [14] and the work of [5] [19] [20] was introduced by [15]. The proposed algorithm, called SWA, is an efficient, scalable, one-scan heuristic which aims at providing a balance between the needs for privacy and knowledge discovery in association rule hiding. It achieves to hide multiple rules in only one pass through the dataset, regardless of its size or the number of sensitive rules that need to be protected.

Amiri [1] proposes three effective, multiple association rule hiding heuristics that outperform SWA by offering higher data utility and lower distortion, at the expense of increased computational speed. Although similar in philosophy to the previous approaches, the three proposed methodologies do a better job in modeling the overall objective of a rule hiding algorithm. The first approach, called Aggregate, computes the union of the supporting transactions for all sensitive itemsets. Among them, the transaction that supports the most sensitive and the least non-sensitive itemsets is selected and expelled from the database. The same process is repeated until all the sensitive itemsets are hidden. Similarly to this approach, the Disaggregate approach aims at removing individual items from transactions, rather than removing the entire transaction. It achieves that by computing the union of all transactions supporting sensitive itemsets and then, for each transaction and supporting item, by calculating the number of sensitive and non-sensitive itemsets that will be affected if this item is removed from the transaction. Finally, it selects to remove the item from the transaction that will affect the higher number of sensitive and the least number of non-sensitive itemsets. The third approach, called Hybrid, is a combination of the two previous algorithms.

Wu [18] propose a sophisticated methodology that removes the assumption of [5] regarding the disjoint relation among the items of the various sensitive rules. Using set theory, the authors formalize a set of constraints related to the possible side-effects of the hiding process and allow item modifications to enforce these constraints.

Pontikakis [4] propose two distortion-based heuristics to selectively hide the sensitive association rules. The proposed schemes use efficient data structures for the representation of the association rules and effectively prioritize the selection of transactions for sanitization. However, in both algorithms the proposed hiding process may introduce a number of side effects, either by generating rules which were previously unknown, or by eliminating existing non-sensitive rules. The first algorithm, called Priority-based Distortion Algorithm (PDA), reduces the confidence of a sensitive association rule by reversing 1's to 0's in items belonging in the rule's consequent. The second algorithm, called Weight-based Sorting Distortion Algorithm (WDA), concentrates on the optimization of the hiding process in an attempt to achieve the least side-effects and the minimum complexity. This is achieved through the use of priority values assigned to transactions based on weights.

Wang [16] [12] propose two data modification algorithms that aim at the hiding of predictive association rules, i.e. rules containing the sensitive items on their left hand side (rule antecedent). Both algorithms rely on the distortion of a portion of the database transactions to lower the confidence of the sensitive association rules. The first strategy, called ISL, decreases 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 (rule consequent).

Lee, [6] introduce a data distortion approach that operates by first constructing a sanitization matrix from the original data and then multiplying the original database (represented as a transactions-by-items matrix) with the sanitization matrix in order to obtain the sanitized database. The applied matrix multiplication strategy follows a new definition that aims to enforce the suppression of selected items from transactions of the original database thus reduce the support of the sensitive itemsets. Along these lines, the authors develop three sanitization algorithms: Hidden-First (HF), Non-Hidden-First (NHF) and HPCME (Hiding sensitive Patterns Completely with Minimum side Effect on non-sensitive patterns).

3. Problem Definition

We focus on the knowledge hiding thread of PPDM and study on specific class of approaches which are collectively known as association rule hiding approaches. In the context of privacy preserving association rule mining, we do not concentrate on privacy of individuals; rather, we concentrate on the problem of protecting sensitive knowledge mined from databases. The sensitive knowledge is represented by a special group of association rules called sensitive association rules. These rules are most important for strategic decision and must remain private (i.e., the frequent rules are private to the owner of the data). The problem of protecting sensitive knowledge in transactional databases draw the assumption that Data owners have to know in advance some knowledge (frequent item sets and/or rules) that they want to protect. Such rules are fundamental in decision making, so they must not be discovered. The problem of protecting sensitive knowledge in association rule mining can be stated as, given a data set D to be released, a set of association rules R mined from D, and a set of sensitive item sets or rules, $R_S \subseteq R$ to be hidden. How can we get a new data set D^1, such that the rules in R_S cannot be mined from D^1, while the rules in R- R_S can still be mined as many as possible. In this case, D^1 becomes the released database.

4. Proposed Framework

In the proposed framework, initially the association rules, R will be mined from the database D by using any association rule mining algorithm (AR). Then the user will specify the sensitive rules, R_S which need to be hidden from mining. By considering sensitive rules and original dataset as input our proposed algorithm HRSIF 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 rules. The proposed framework is shown in figure 1.

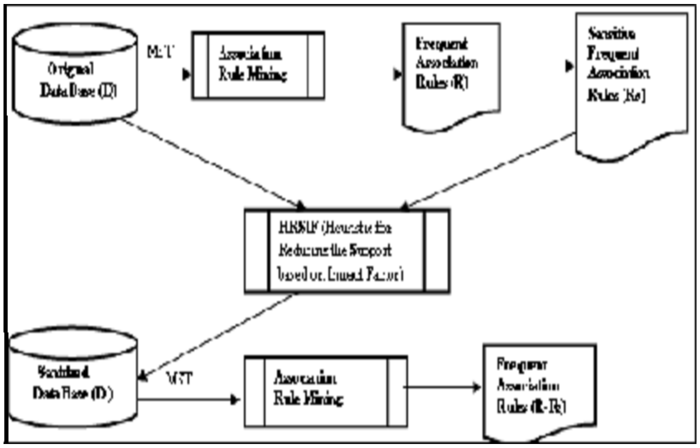
Original Data Base (D)
MST
Association Rule Mining
Frequent Association Rules (R)
Sensitive Frequent Association Rules (Rs)
HRSIF (Heuristic for Reducing the Support based on Impact Factor)
Sanitized Data Base (D')
MST
Association Rule Mining
Frequent Association Rules (R-Rs)

Figure 1: Proposed Framework for Association rules hiding

5. The Proposed Algorithm

The algorithm uses intersection lattice and impact factor of items in the sensitive association rules to decide the victim item to hide the sensitive rule.

5.1 Intersection lattice of items

We adopt lattice theory that is presented in [21].Let I be a finite nonempty item set. It is obvious that the power set of I, denoted by Poset(I) , is an ordered set under the relation ⊆ .It can be verified that (Poset(I); ⊆) forms a lattice, where sup(a, b)=a∪ b and inf(a,b)=a∩b.

If M⊆ I and (M; ⊆) is a lattice satisfying the properties that sup (a,b)=a∪ b and inf(a,b)=a∩b, for all a and b, then(M; ⊆) is called a set lattice. Similarly if (M; ⊆) is a semilattice satisfying inf (a, b) = a∩b, for all a and b, then (M; ⊆) is said to be intersection lattice. It is obvious that intersection of elements in an intersection lattice (M; ⊆) belongs to M. In other words, an intersection lattice (M; ⊆) is closed under the intersection operator.

Let FIS be a set of frequent item sets. By the Apriori property, if A, B ∈ FIS, then $A \cap B \in FIS$.It can be inferred that FIS is an intersection lattice.

Example 1: Consider the database shown in Table 1 and its corresponding intersection lattice shown in Fig 2.

Table 1: Database

Tid	List of Items	Tid	List of items
1	A,B,C,J	11	E,F,I,J
2	B,F,G,I,J	12	E,G,H,I
3	A,C,G,J	13	A,B,C,D,G
4	A,E,F,G,H,I,J	14	B,C,F,J
5	A,B,G,I,J	15	A,B,F,G
6	A,C,D,J	16	F,G,I,J
7	A,B,C,F,G,I,J	17	C,F,G,J
8	B,C,E,H,I	18	A,B,E,J
9	B.C.F.J	19	B,C,E,I
10	A.D.F.G.I.J	20	A,B,G,I

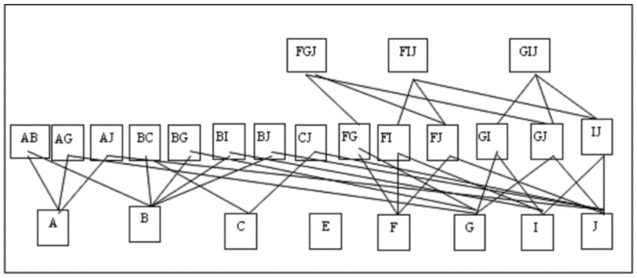

Figure 2: Intersection Lattice of FIS

5.2 HRSIF (Heuristic for Reducing the Support based on Impact Factor) Algorithm

Let R_S be the set of sensitive association rules. Presume that the sensitive rule that needs to be hidden each time is denoted by X →Y. Our method aims at hiding X →Y by removing an item in X∪Y from a number of transactions until Support (X →Y) < minimum support threshold (MST) or Confidence(X →Y) < minimum confidence threshold (MCT).

The algorithm specifies the victim item based on the concept of impact factor of an item in the sensitive rule. The impact factor of an item in the sensitive association rule is equal to the number of non sensitive frequent item sets that are affected by removing that item from the required number of transactions. Lower the impact factor of an item, lower is its effect on the non sensitive frequent item sets.

Step 1: Identifying the number of transactions

This step aims to compute the minimum number of transactions that need to be modified in order to hide the sensitive rule. Let this number be denoted by T_n. Then to hide the rule X →Y, we must have

$$\text{Support (XY)} - T_n < \text{MST or (Support (XY)} - T_n) / \text{Support(X)} < \text{MCT}$$

$$\Longrightarrow \quad T_n > \text{Support(XY)} - \text{MST} \;\text{ or }\; T_n > \text{Support(XY)} - [\text{Support(X)} * \text{MCT}]$$

Thus T_n= min{Support (XY) - MST +1, Support (XY) – [Support (X) * MCT] + 1}.

Furthermore identifying the order of transactions for item modification is an important step in reducing the side effects. Let T_{XY} be a set of transactions that support the rule X →Y. The transactions have a smaller size and contain fewer item sets and association rules. Thus to achieve the minimum impact on the non sensitive association rules, T_{XY} needs to be sorted in ascending order of size of each transaction. If transactions are having the same size then sort them in ascending order of number of items of the transaction presented in X∪Y.

Step 2: Victim Item Selection

The victim item is the item that needs to be removed to hide a rule such that removing this item minimizes the effect on non sensitive items. Example 2 shows how the victim item selection can reduce the side effects of the hiding process.

Example 2: Consider the transactional data set D and MST=2 & MCT=30% as in Example 1. Assume that the sensitive rule that need to be hidden is F →G, J. To hide this rule, we need to remove F or G or J from some transactions supporting FGJ. Next we compare the impact on the intersection lattice of FIS when modifying F or G or J.

Removing F or G or J from the transactions supporting FGJ directly affects the FGJ. Thus we consider the impact on the superset of that item. If we remove G, Support (GIJ) < MST. GIJ was also be hidden along with FGJ. So the impact factor of G can be considered as 1. If we remove J, Support (FIJ) and Support (GIJ) is less than MST. That is two more items are also be hidden with FGJ, so the impact factor of J is 2. If we remove F, Support (FI) and S (FIJ) is less than MST. That is two more items are also be hidden with FGJ so the impact factor of F is also 2. G will have less impact factor when compared to F and J. So G will be selected as victim item.

Step 3: Updating the transactions and updating the support counts of FIS

The victim is removed from T_n transactions which are supporting X → Y. After modifying the database, update the support counts of FIS.

The complete algorithm is as follows.

```
Algorithm HRSIF( )

Input:  The data set D
        Minimum support threshold, MST
        Minimum confidence threshold, MCT
        Frequent Item Sets, FIS
        Set of association rules to be hidden, $R_S$

Output: Sanitized Data Set $D^1$

Method:

  For each rule $X \rightarrow Y \in R_S$

        Step 1:  Compute $T_{XY}$;
                 Sort ( $T_{XY}$);
                 $T_n$= min{Support (XY) - MST +1,
                        Support (XY) – [Support (X) * MCT] + 1};

        Step 2:  For each item $I_i \in X \cup Y$
                    $IF_i$ = Get impact factor ($I_i$);
                 End for;
                 Victim item = min{ $IF_1$, $IF_2$, …., $IF_n$}
                       where n is number of items in $X \cup Y$;

        Step3: Remove victim item from $T_n$ transactions;
               Update support (FIS);

  End for;
```

6. Illustrative Example

Consider the data set shown in Table 2. The minimum support threshold, MST=8 and minimum confidence threshold, MCT=60%. Let the set sensitive association rules to be hidden R_S= {D →BH, E →F}.

We apply HRSIF algorithm to hide R_S. First HRSIF considers rule D →BH for hiding.

Table 2: Database

Tid	List of Items	Tid	List of items
1	C,F,H	11	B,C,E,F,J
2	A,B,C,D,H,J	12	B,D,E,H
3	B,D,E,H	13	D,E,F,H,J
4	B,E,F,G	14	B,C,D,E,F,G,H,J
5	C,D,J	15	D,J
6	B,C,D,E,F,G,H	16	G,I
7	B,C,E,F,G,J	17	B,C,D,E,F,H
8	B,C,D,H,J	18	B,C,D,H,J
9	B,C,E,F,G,J	19	A,C,D,E,F,H,J
10	E,F,J	20	A,D,E,F,H

Step 1:

T_{DBH} = {2,3,6,8,12,14,17,18}

Sorted T_{DBH} = {3,12,8,18,17,6,14}

T_n = min{Support (DBH) - MST +1, Support (DBH) – [Support (D) * MCT] + 1};

= min {8-8+1, 8-[13 *60%]+1}

=min{1,1}

=1

Step 2: FIS={B:12,C:12,D:13,E:13,F:12,H:12,J:12,BC:9, BD:8,BE:9,BH:8,CD:8,CF:8,CH:8,CI:9,DE:8, DH:11,DJ:8,EF:11,EH:8,BDH:8,DEH:8}

Ii={B,D,H}

Impact Factor of B:

Superset or power set of B in FIS={B,BC,BD,BE,BH,BDH}

To hide D →BH, we need to modify 1(Tn) transaction i.e 3rd transaction.

Super set of B that are also be supported by 3rd transaction along with their support is {B:12,BD:8,BE:9,BH:8,BDH:8}.

So if we remove B from 3rd transaction BDH will be hidden and at the same time the non sensitive itemsets {BD:8,BH:8} will also be hidden

⇒ **IF(B)=2.**

Impact Factor of D:

Superset or power set of D in FIS={D,BD,CD,DE,DH,DJ,BDH,DEH}

To hide D →BH, we need to modify 1(T_n) transaction i.e 3rd transaction.

Super set of D that are also be supported by 3rd transaction along with their support is {D:13,BD:8,DE:8,DH:11,BDH:8,DEH:8}.

So if we remove D from 3rd transaction BDH will be hidden and at the same time the non sensitive itemsets {BD:8,DE:8,DEH:8} will also be hidden.

⇒ **IF(D)=3.**

Impact Factor of H:

Superset or power set of B in FIS={H,BH,CH,DH,EH,BDH,DEH}

To hide D →BH, we need to modify 1(T_n) transaction i.e 3rd transaction.

Superset of B that are also be supported by 3rd transaction along with their support is {H:12,BH:8,DH:11,EH:8,BDH:8,DEH:8}.

So if we remove H from 3rd transaction BDH will be hidden and at the same time the non sensitive itemsets {BH:8,EH:8,DEH:8} will also be hidden.

⇒ **IF(H)=3.**

Victim item= min{IF(B),IF(D),IF(H)}
= min{ 2,3,3}
= 2

So the item B will be selected as Victim item.

Step 3:

Remove victim item B in one transaction from sorted T_{BDH} i.e 3rd transaction.

Update the support of FIS

After update

FIS={B:11,C:12,D:13,E:13,F:12,H:12,J:12,BC:8,BE:8,CD:8, CF:8,CH:8,CI:9, DE:8, DH:11,DJ:8,EF:11,EH:8,DEH:8}

The rule D →BH will be hidden. Next the algorithm considers E →F for hiding.

Step 1:

T_{EF} = {4,6,7,9,10,11,13,14,17,19,20}

Sorted T_{EF} = {10,4,11,13,20,7,9,17,6,19,14}

T_n = min{Support (EF) - MST +1, Support (EF) – [Support (E) * MCT] + 1};

= min {11-8+1, 11-[13 *60%]+1}

=min{4,4}

=4

Step 2:

FIS={B:11,C:12,D:13,E:13,F:12,H:12,J:12,BC:8,BE:8,CD:8, CF:8,CH:8,CI:9, DE:8, DH:11,DJ:8,EF:11,EH:8,DEH:8}

Ii={E,F}

Impact Factor of E:

Superset or power set of E in FIS={E,BE,DE,EF,EH,DEH}

To hide E →F, we need to modify 4 (T_n) transactions i.e 10th, 4th, 11th and 13th transactions.

Super set of E that are also be supported by 10,4,11 and 13 transactions along with their support is {E:13,BE:8,DE:8,EF:11,EH:8,DEH:8}.

So if we remove E from 10,4,11and 13 transactions EF will be hidden and at the same time the non sensitive itemsets {BE:8,DE:8,EH:8.DEH:8} will also be hidden

⇒ **IF(E)=4.**

Impact Factor of F:

Superset or power set of F in FIS={F,CF,EF}

To hide E →F, we need to modify 4(T_n) transactions i.e 10th, 4th, 11th and 13th transactions.

Super set of F that are also be supported by 10,4,11 and 13 transactions along with their support is {F: 12, EF: 11}.

So if we remove F from 10,4,11 and 13 transactions EF will be hidden and no non sensitive itemsets will get affected.

⇒ **IF(F)=0**

Victim item= min{IF(E),IF(F)}
= min{ 4,0}
= 0

So the item F will be selected as Victim item.

Step 3:

Remove victim item F in 4 transactions from sorted T_{EF} i.e 10th, 4th, 11th and 13th transactions.

Update the support of FIS

After update

FIS={B:11,C:12,D:13,E:13,F:8,H:12,J:12,BC:8,BE:8,CD:8, CF:8,CH:8,CI:9, DE:8, DH:11,DJ:8,EH:8,DEH:8}

Now all the Sensitive rules will be hidden i.e R_S is empty. The sanitized data set is as shown in Table 3.

Table 3: Sanitized Database (D^1).

Tid	List of Items	Tid	List of items
1	C,F,H	11	B,C,E,J
2	A,B,C,D,H,J	12	B,D,E,H
3	D,E,H	13	D,E,H,J
4	B,E,G	14	B,C,D,E,F,G,H,J
5	C,D,J	15	D,J
6	B,C,D,E,F,G,H	16	G,I
7	B,C,E,F,G,J	17	B,C,D,E,F,H
8	B,C,D,H,J	18	B,C,D,H,J
9	B,C,E,F,G,J	19	A,C,D,E,F,H,J
10	E,J	20	A,D,E,F,H

7. Performance Measures

7.1 Hiding Failure :(HF)

When some sensitive rules are discovered from D^1, we call this problem as Hiding Failure, and it is measured in terms of the percentage of sensitive rules that are discovered from D^1. The hiding failure is measured by $HF = \frac{\#R_S(D^1)}{\#R_S(D)}$ where $\#R_S(D^1)$ denotes the number of sensitive rules discovered from sanitized database(D^1), and $R_S(D)$ denotes the number of sensitive rules discovered from original database(D).

7.2 Misses Cost / Lost Rules :(MC)

Some non-sensitive rules can be hidden by mining algorithms accidentally. This happens when some non-sensitive rules lose support in the database due to the sanitization process. We call this problem as Misses Cost, and it is measured in terms of the percentage of legitimate patterns that are not discovered from D^1. The misses cost is calculated as follows:

$MC = \frac{\#\sim R_S(D) - \#\sim R_S(D^1)}{\#\sim R_S(D)}$ where #~ $R_S(D)$ denotes the number of non-sensitive patterns discovered from original database D, and #~ R_S (D^1) denotes the number of non-sensitive rules discovered from sanitized database D^1.

7.3 Artifactual Rules/ Ghost Rules :(AR)

Some artificial rules are going to be generated from D^1 as a product of the sanitization process. We call this problem as Artifactual rules, and it is measured in terms of the percentage of the discovered rules that are artifacts.

7.4 dif/ Accuracy (D,D^1)

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

8. Experimental Results

All the experiments were conducted on PC, Intel i5 CPU @ 2.50 GHz and 4 GB of RAM running on windows 7, 64-bit operating system. To measure the effectiveness of the algorithm, we used a dataset generated by the IBM synthetic data generator and FIMI Repository [2].

In this study, we compared the HRSIF algorithm with theMaxMin2 algorithm presented in [22] to evaluate the side effects and computational complexity. The MaxMin2 algorithm is based on border approach and gained efficiency in minimizing the side effects compared with the previous heuristic approach [22]. The dataset was used for the experiment is Retail.dat. To examine the performance of the HRSIF and MaxMin2 algorithms, we varied the number of sensitive association rules from one to five rules for each experiment, as presented in Table 6. We compare the performance of these algorithms based on five metrics, including lost rule, ghost rule, false rule, and accuracy.

Fig. 3 shows the efficiency of the proposed algorithms in the lost rules minimization. Accordingly, the HRSIF algorithm achieved better results in reducing lost rules compared with MaxMin2 algorithm. The trends indicate that when the number of sensitive association rules is increased, HRSIF caused fewer lost rules than MaxMin2. In particular, Maxmin2 caused a very high percentage of lost rules, two times that of HRSIF, when dealing with five sensitive association rules.

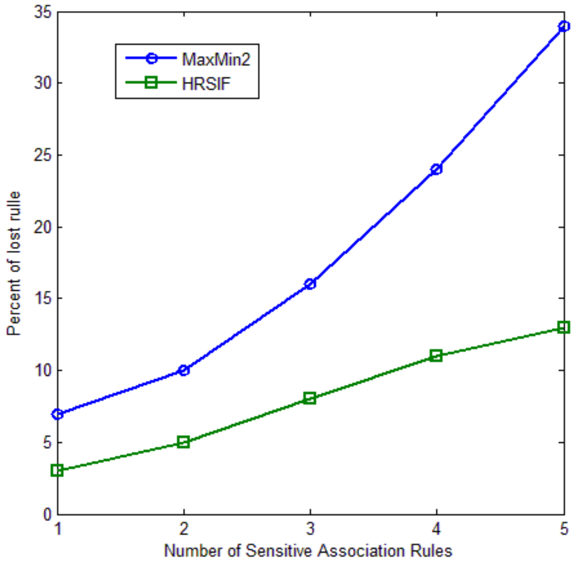

Figure 3: Percentage of lost rules produced by HRSIF and Maxmin2.

Fig. 4 shows that only a few ghost rules were produced by the HRSIF and MaxMin2 algorithms. Although the MaxMin2 algorithm introduced nearly 0.5% ghost rules while HRSIF did not produce any ghost rules when hiding one or two sensitive rules, the percentage of ghost rules produced by these algorithms are very low. In general, the ghost rules produced by these algorithms are quite similar.

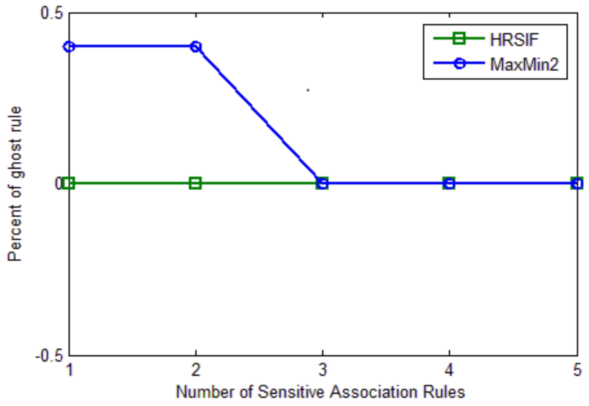

Fig 4 Percentage of ghost rules produced by HRSIF **and Maxmin2.**

Fig. 4 shows efficiency of the proposed algorithm in the Hiding Failure. Accordingly, the HRSIF and MaxMin2 algorithm will not produce any sensitive rules form D^1, when hiding any number sensitive rules.

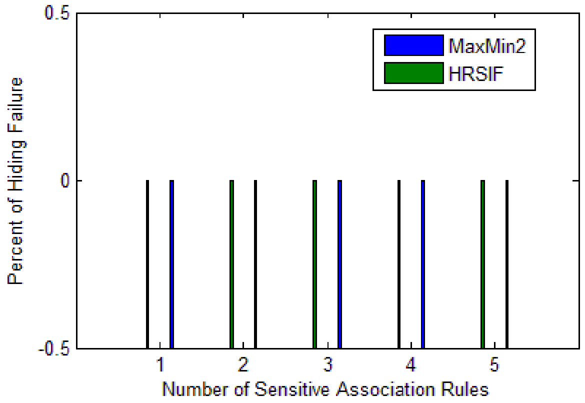

Fig 5 Percentage of Hiding Failure produced by HRSIF **and Maxmin2.**

Fig 6. Shows that the HRSIF algorithm needed fewer distortions than MaxMin2. High accuracy (>98%) when handling the selected sensitive association rules means the released database was slightly distorted. Thus, although HRSIF achieved a slight higher accuracy than MaxMin2, both of them attained very high accuracy when dealing with five sensitive association rules, which guarantees the capability of these algorithms in the real application.

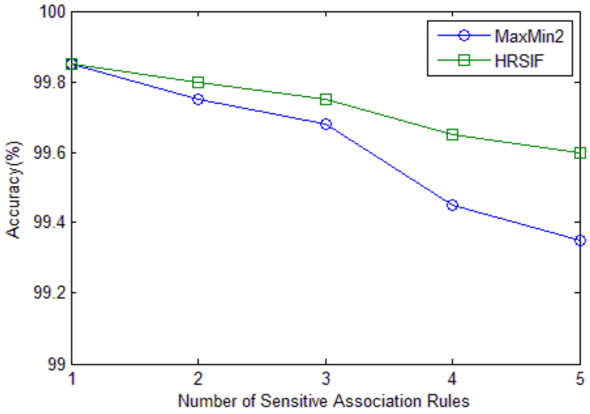

Fig.6.Accuracyof dataset caused by HRSIF and Maxmin2.

9. Conclusion

A heuristic algorithm was proposed to hide a set of sensitive association rules using the intersection lattice of frequent item sets for privacy preserving Association rule mining. We have implementation of **H**euristic for **R**educing the **S**upport based on **I**mpact **F**actor (HRSIF) for hiding sensitive rules from transactions and generating a sanitized database D^1. To minimize the side effects(HF,MC,AR), the HRSIF algorithm specified the victim item and minimum number of transactions such that the modification of this item causes the least amount of impact on non sensitive item sets. The proposed algorithm, HRSIF, will not produce any Hiding Failure, Artifactual rules and also fewer Misses cost from D^1. Our further research will focus on finding optimal sensitive transactions to further minimize the Misses cost.

References

[1] A. Amiri, 2007, Dare to share: Protecting sensitive knowledge with data sanitization. *Decision Support Systems*, pages 181–191.

[2] B.Goethals,2003, The fime repository. *FIME 2003*.

[3] J. Domingo-Ferrer, 2009, Non-perturbative masking. *In Encyclopedia of database systems US: Springer*, page 1912.

[4] A. A. Tsitsonis E. D. Pontikakis and V. S. Verykios, 2004, An experimental study of distortion based techniques for association rule hiding. *In Proceedings of the 18th Conference on Database Security (DBSEC)*, pages 325–339.

[5] A. K. Elmagarmid E. Dasseni, V. S. Verykios and E. Bertino, 2001, Hiding association rules by using confidence and support. *In Proceedings of the 4th International Workshop on Information Hiding*, page 369.

[6] C. Y. Chang G. Lee and A. L. P. Chen, 2004, Hiding sensitive patterns in association rules mining. *In Proceedings of the 28th International Computer Software and Applications Conference(COMPSAC)*, pages 424–429.

[7] S.V Gkoulalas-Divanis, A. Verykios, 2010, Association rule hiding for data mining. *Springer*, (ISBN:9781441965691).

[8] B. Janakiramaiah, A. Ramamohan Reddy, and G.Kalyani, 2012, Data distortion approaches for privacy preserving in association rule mining. *International Journal of Advances in Computing and Information technology*, (ISSN 22779140):579–594.

[9] B. Janakiramaiah, A. Ramamohan Reddy, and G.Kalyani, 2013, An approach for privacy preserving in association rule mining using data restriction. *International Journal of Engineering Science Invention*, (ISSN 2319 6734):27–34.

[10] B. Janakiramaiah, A. Ramamohan Reddy, and M.K Kumari, 2009, Parallel privacy preserving association rule mining on pc clusters. *IEEE International Advance Computing Conference (IACC)*, pages 1538–1542.

[11] D. Ramakrishnan R. LeFevre, K. DeWitt, 2005, Efficient full-domain k-anonymity. *In SIGMOD*, pages 49–60.

[12] S. L.Wang and A. Jafari, 2005, Using unknowns for hiding sensitive predictive association rules. *In Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 223–228.

[13] A. Elmagarmid M. Ibrahim M. Atallah, E. Bertino and V. S. Verykios, 1999, Disclosure limitation of sensitive rules. *In Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, pages 45–52.

[14] S. R. M. Oliveira and O. R. Zaane, 2002 Privacy preserving frequent itemset mining. *In Proceedings of the 2002 IEEE International Conference on Privacy, Security and Data Mining(CRPITS)*, page 43-54.

[15] S. R. M. Oliveira and O. R. Zaane, 2003, Protecting sensitive knowledge by data sanitization. *In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 211–218.

[16] B. Parikh S.-L.Wang and A. Jafari, 2007, Hiding informative association rule sets. *Expert Systems with Applications*, pages 316–323.

[17] E. Bertino-Y. Saygin V. S. Verykios, A. K. Emagarmid and E. Dasseni, 2004, Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering*, pages 434–447.

[18] C. M. Chiang Y. H. Wu and A. L. P. Chen, 2007, Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering*, pages 29–42.

[19] V. S. Verykios Y. Saygin and C. W. Clifton, 2001, Using unknowns to prevent discovery of association rules. *ACM SIGMOD Record*, pages 45–54.

[20] V. S. Verykios Y. Saygin and A. K. Elmagarmid, 2002, Privacy preserving association rule mining. In Proceedings of the 2002 International Workshop on Research Issues in Data Engineering: Engineering ECommerce/EBusiness Systems (RIDE), pages 151–163.

[21] G.Gratzer, 2011, Lattice Theory: Foundation, 2010 Mathematics Subject Classification, Springer, Basel, AG.

[22] G. Tuncel, G. Alpan, Risk assessment and management for supply chain networks: a casestudy, ComputersinIndustry 61(2010)250–259.

B Janakiramaiah – He was born in 1979. He received his bachelor’s degree in Computer Science and Engineering from Nagpur University, Masters in Computer Science and Engineering from Jawaharlal Nehru Technological University. He is currently working as Associate Professor in DVR & Dr HS MIC College of Technology, Kanchikacherla, India. He is now research scholar in JNTUH, Hyderabad, India. His interests are Privacy preserving data mining, Machine Learning, Soft Computing.

Dr A Rama Mohan Reddy – He was born in 1958. He received his Masters in Computer Science and Engineering from NIT, Warangal, Ph.D in Computer Science and Engineering from Sri Venkateswara University, Tirupati. He is currently working as a Professor and Head of Computer Science and Engineering department in SVU College of Engineering, Tirupati, India. His interests are Data Mining, Software Engineering and Software Architectures.

G Kalyani – She was born in 1979. She received her bachelor’s degree in Computer Science and Engineering from Acharya Nagarjuna University, Masters in Computer Science and Engineering from Jawaharlal Nehru Technological University. She is currently working as Associate Professor in DVR & Dr HS MIC College of Technology, Kanchikacherla, India. Her interests are Privacy preserving data mining, Machine Learning, Operating Systems, Data Base Management Systems.