International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 1 || January. 2013 || PP.27-34

An Approach for Privacy Preserving in Association Rule Mining Using Data Restriction

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

[1] *Research Scholar of JNTUH, Assoc. Professor,*
Department of computer science and Engineering,
DVR & Dr HS MIC College of Technology, Vijayawada, A.P, India
[2] *Professor & Head, Department of Computer Science and Engineering,*
S.V.University, Tirupathi, India.
[3] *Assoc.Professor, Department of computer science and Engineering,*
DVR & Dr HS MIC College of Technology, Vijayawada, A.P, India

ABSTRACT: *The sharing of data or mined association rules can bring a lot of advantages for research, marketing, medical analysis and business partnerships; however, large repositories of data contain private data and sensitive rules that must be protected before released. The challenge is protection private data and sensitive rules contained in the source database, while non-sensitive rules can still be mined normally. To address this challenging problem, different sanitization methods were projected in literature. We discuss different data restriction methods from sanitization process. We introduce the taxonomy of sanitization algorithms and validate all data restriction algorithms against real and synthetic data sets. We also considered a set of metrics to evaluate the effectiveness of the algorithms by perform the experimental studies on different data restriction algorithms.*

Keywords— *Privacy Preserving, Sanitization, Association Rule mining, Data Restriction.*

I. INTRODUCTION

Data mining extracted novel, hidden and useful knowledge from vast repositories of data and has become an effective analysis and decision means in corporation. Association rule mining is one of the most important and fundamental problems in data mining. The sharing of data for data mining can bring a lot of advantages for research, marketing, medical analysis and business collaboration; however, huge repositories of data contain private data and sensitive rules that must be protected before published. The challenge is on protecting data and actionable knowledge for strategic decisions, but at the same time not losing the great benefit of association rule mining.

The problem of association rule hiding motivated by many authors [2, 6, 9], and different approaches were proposed. Roughly, they can fall into two groups: data sanitization [1] by data modification approaches and knowledge sanitization by data reconstruction approaches. For our comparison study, we selected the data sanitizing algorithms by Data Restriction Techniques in the literature: (1) SWA, (2) IGA.

II. BACKGROUND

In this section, we briefly review the basics of association rules and provide the definitions of sensitive rules. Subsequently, we describe the process of protecting sensitive knowledge in transactional databases.

2.1 The Basics of Association Rules

One of the most studied problems in data mining is the process of discovering association rules from large databases. Association rule mining is the process of discovering sets of Items that frequently co-occur in a transactional database to produce significant association rules that hold for the data. Most of the existing algorithms for association rules rely on the support-confidence framework.

Formally, association rules are defined as follows: Let $I = \{ i_1, i_2 \ldots i_m \}$ be a set of items. Let D the task relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form A =>B, where $A \subset I$, $B \subset I$ and $A \cap B = \Phi$. The rule A => B holds in the transaction set D with support s, where s is the percentage of transaction in D that contains $A \cup B$. The rule A => B has confidence c in the transaction set D if c is the percentage of

transactions in D containing A which also contain B. 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.

Support(s) of an association rule is defined as the percentage/fraction of records that contain (A ∪ B) to the total number of records in the database.

$$\text{Support (A=>B)} = \frac{Support\ \ count\ \ of\ (A \cup B)}{Total\ \ number\ \ of\ transaction\ \ in\ D}$$

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain (A ∪ B) to the total number of records that contain A.

$$\text{Confidence (A=>B)} = \frac{Support\ \ count\ \ of\ (A \cup B)}{Support\ \ count\ \ of\ (A)}$$

2.2 Sensitive Rules

Protecting sensitive knowledge in transactional databases is the task of hiding a group of association rules, which contain sensitive knowledge. We refer to these rules as sensitive association rules and define them as follows:

Definition :- (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 Rules R_S be a set of decision support rules that need to be hidden according to some security policies. A set of association rules, denoted by R_S , is said to be sensitive if $R_S \subset R$. $\sim R_S$ (R- R_S) is the set of non-sensitive association rules such that $\sim R_S \cup R_S = R$.

1. Problem Definition

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 rules are private in the company or organization owning the data).

The problem of protecting sensitive knowledge in transactional databases draw the assumption that Data owners have to know in advance some knowledge (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 rules R mined from D, and a set of sensitive rules $R_S \subset R$ to be hided, 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.

2. Sanitizing Algorithms

In our framework, the sanitizing algorithms modify some transactions to hide sensitive rules based on a disclosure threshold μ controlled by the database owner. This threshold indirectly controls the balance between knowledge disclosure and knowledge protection by controlling the proportion of transactions to be sanitized.

4.1 Sanitizing Algorithms: Major Steps

1. Find sensitive transactions for each restrictive pattern;
2. For each restrictive pattern, identify a candidate item that should be eliminated (victim item);
3. Based on the disclosure threshold ψ, compute the number of sensitive transactions to be sanitized;
4. Based on the number found in 3, remove the victim items from the sensitive transactions.

We classify our algorithms into two major groups: Data Modification algorithms and Data Reconstruction algorithms, as can be seen in Figure 1.

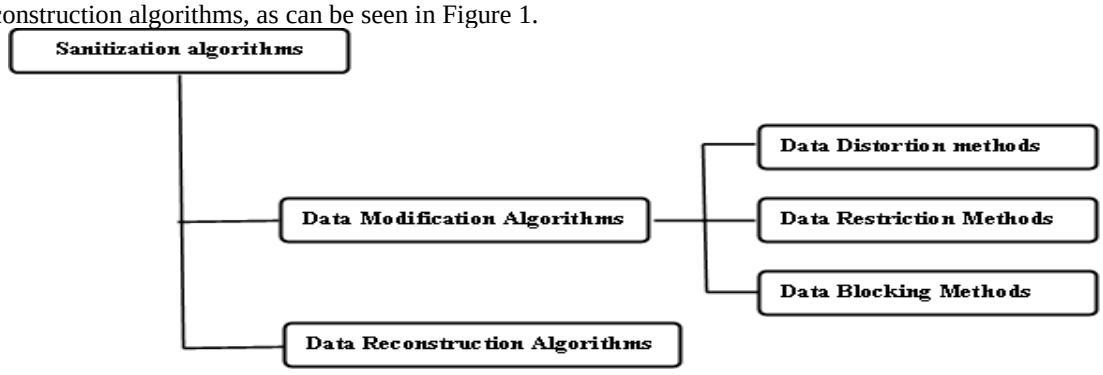


Figure 1: A taxonomy of sanitizing algorithms.

4.2 Data Modification Algorithms

Data modification methods hide sensitive association rules by directly modifying original data. Most of the early methods belong to this track. According to different modification means, it can be further classified into the three subcategories: Data Distortion methods, Data Restriction methods and Data Blocking methods.

4.2.1 Data Distortion is based on data perturbation or data alteration, and in particular. The procedure is to alter a selected set of 1 (true) value to 0(false) values (delete items) or 0 values to 1 values (add items) if we consider the transaction database as a two-dimensional matrix. Its aim is to decrease the support or confidence of the sensitive rules below the user predefined threshold value.

4.2.2 Data Restriction is an approach of deleting of items from the transactions of transactional databases, which are present in sensitive association rules. Two different approaches were existed in restriction. Those are i) delete all the sensitive items from all the transactions which are supporting those sensitive rules.

ii) Delete few items, which are sensitive from some of the transactions, which are supporting those sensitive rules until support less than min support threshold or confidence less than the minimum confidence threshold.

4.2.3 Data-Blocking is another data modification approach for association rule hiding. Instead of making data distorted, blocking approach is implemented by replacing certain data items with a question mark "?"(Unknown) [7]. the introduction of this special unknown value brings uncertainty to the data, making the support and confidence of an association rule become two uncertain intervals respectively.

4.3 Data Reconstruction Algorithms

Data reconstruction methods put the original data aside and start from sanitizing the so-called "knowledge base". The new released data is then reconstructed from the sanitized knowledge base [3, 4].

SANITIZATION ALGORITHMS	**DATA MODIFICATION ALGORITHMS**	**DATA DISTORSION METHODS**	**Algo 1a,Algo 1b, Algo 2a,Algo 2b,Algo 2c Naïve MinFIA,MaxFIA**
		DATA RESTRICTION METHODS	**IGA,SWA**
		DATA BLOCKING METHODS	**GIS**
	DATA RECONSTRUCTION ALGORITHMS		**CIILM**

Table 1: Classification of different algorithms.

The above Table 1 contains various algorithms related to each category of sanitization algorithms.

III. ALGORITHMS

We now present in detail two Data Restriction algorithms.

5.1 SLIDING WINDOW ALGORITHM (SWA):

The main thought behind the Sliding Window Algorithm [8], denoted by SWA, is to sanitize the sensitive transactions with the shortest sizes. The underlying principle is that by removing items from shortest transactions we would reduce the impact on the sanitized database since the shortest transactions have smaller number combinations of association rules.

The sketch of the Item Algorithm is given as follows:

Algorithm: Sliding Window Algorithm (SWA)
Input: Data Base D, Restrictive Rules Rr , Window Size K.
Output: Sanitized Data Base D^1

Begin

Step 1:
For each K transactions in D do
For each restrictive rule r € Rr do
1. T[r] ← Find Sensitive Transactions(r, D);

Step 2:
1. Compute the frequencies of all items in the restricted rules w.r.t T[r]
2. For each restrictive rule r € Rr do
4.1. Victimr ← item I with maximum frequency I € r

Step 3:
For each restrictive rule r € Rr do
1. NumTransr ← (T[r] *(1-μ)
2. Sort the transactions of T[r] in ascending order of size.

Step 4: D^1 ← D
For each restrictive rule r € Rr do
1. TransToSanitize ← Select first *NumTransr* transactions from T[r]
2. In D^1 for each transaction t € TransToSanitize do
3.1. t ← (t - Victimr)

In Step 1 the Sliding window algorithm builds an index for all sensitive transactions in D. In step 2 the algorithm selects the victim item for every restricted association rule. First it computes the frequencies of all the items in restricted association rules w.r.t sensitive transactions T[r].The item with the maximum frequency will be selected as victim for that restricted association rule. Based on the disclosure threshold μ step 3 identifies the number of transactions to be sanitized and transactions of in ascending order of their sizes. Step 4 first copies the D into D^1 .For each restrictive rule based on number of transactions to sanitize, the victim item will be removed from the transactions.

To illustrate how the SWA algorithm works consider the transactional database in Table 2(a).Suppose we have the restricted association rules as {(I1,I2 → I4) and (I3 → I4)}.Table 2(b) shows the sanitized database.
Step 1: By scanning the data base identify the sensitive transactions as {T1, T3, T5, T6}.
Step 2: Compute the frequencies. The frequency of I1, I2, I3, and I4 are 2, 3, 3 and 4 respectively.
For the rule I1, I2 → I4, I4 will be selected as victim.
For the rule I3 → I4 also I4 will be selected as victim.

Step 3: We set the disclosure threshold μ as 25%. We sanitize half of the sensitive transactions for each restricted rule. In this case transactions T3, T5 and T6 will be sanitized.

Step 4: We perform sanitization by considering the victim item selected in step 2 i.e I4.Remove I4 from T3, T5 and T6.

TID	ITEM SETS
T1	I1,I2,I3,I4
2	I1,I2
T3	I2,I3,I4
T4	I2,I3
T5	I1,I2,I4
6	I3,I4
T7	I2,I4

(a)

TID	ITEMSETS
T1	I1,I2,I3, I4
T2	I1,I2
T3	I2,I3
T4	I2,I3
T5	I1,I2
T6	I3
T7	I2,I4

(b)

Table 2: (a) Sample transactional database **(b) Sanitized database with SWA algorithm.**

5.2 ITEM GROUPING ALGORITHM (IGA):

The main thought behind the Item Grouping Algorithm [5], denoted by IGA, is to group restricted rules in groups of rules sharing the similar itemsets. If two restrictive rules overlap, by sanitizing the sensitive transactions containing both restrictive rules, one would take care of hiding these two restrictive rules at once and thus reduce the impact on the sanitized database.

In Step 1 the item grouping algorithm builds an index for all sensitive transactions in D. In step 2 the algorithms groups the restricted rules based on similar items in the rules and then sort the transactions associated with them in descending order and for each restricted rule it identifies the victim item.Based on the disclosure threshold μ step 3 identifies the number of transactions to be sanitized. Step 4 first copies the D into D^1 .For each restrictive rule based on number of transactions to sanitize, the victim item will be removed from the transactions.

To illustrate how the IGA algorithm works consider the transactional database in table 3(a).Suppose we have the restricted association rules as {(I1,I2 → I4) and (I3 → I4)}.Table 3(b) shows the sanitized database.

Step1: By scanning the data base identify the sensitive transactions as {T1, T3, T5, T6}.The degree of the transactions are 2, 1, 1 and 1 respectively.

Step 2: The two rules can grouped in together because they have a common item I4.The victim item is also be selected as I4.

Step3: We set the disclosure threshold μ as 50%. We sanitize half of the sensitive transactions for each restricted rule. In this case transactions T1 and T3 will be sanitized.

Step 4: We perform sanitization by considering the victim item selected in step 2 i.e I4.Remove I4 from both T1 and T3.

The sketch of the Item Algorithm is given as follows:

Algorithm: Item Grouping Algorithm (IGA)
Input: Data Base D, restrictive rules Rr,
Output: sanitized Data Base D^1
Begin
Step 1:
For each restrictive rule r € Rr do
1. T[r] ← Find Sensitive Transactions(r, D);
Step 2:
1. Group restrictive rules in a set of groups GP
2. Order the groups in GP by size in terms of number of restrictive rules in the group.
3. Compare group's pair wise Gi and Gj starting with the largest.
For all r € Gi ∩ Gj do
3.1. If size (Gi)! = size (Gj) then remove r from smallest(Gi;Gj)
3.2. Else remove r from group
4. For each restrictive rule r € Rr do
4.1. V ictimr ← item of G and r € G
Step 3:
For each restrictive rule r € Rr do
1. NumTransr ← (T[r] *(1-μ)
Step 4:
D^1 ← D
For each restrictive rule r € Rr do
1. Sort Transactions (T[r])
2. TransToSanitize ← Select first *NumTransr* transactions from T[r]
3. In D^1 foreach transaction t € TransToSanitize do
3.1. t ← (t - Victimr)
End

TID	ITEM SETS
T1	I1,I2,I3,I4
T2	I1,I2
T3	I2,I3,I4
T4	I2,I3
T5	I1,I2,I4
T6	I3,I4
T7	I2,I4

TID	ITEMSETS
T1	I1,I2,I3
T2	I1,I2
T3	I2,I3
T4	I2,I3
T5	I1,I2,I4
T6	I3,I4
T7	I2,I4

Table 3: (a) Sample transactional database **(b) Sanitized database with IGA algorithm**

IV. PERFORMANCE EVALUATION

All the experiments were conducted on PC, Intel Pentium dual core with 2.80GHz and 2 GB of RAM running on a windows operating system. To measure the effectiveness of the algorithm, we used a dataset generated by the IBM synthetic data generator. The performance of the algorithms has been measured according to following criteria.

6.1 Performance Measures

Hiding Failure:

When some restrictive patterns are discovered from D^1, we call this problem as Hiding Failure, and it is measured in terms of the percentage of restrictive patterns that are discovered from D^1. The hiding failure is measured by

$$HF = \frac{\#R_s(D^1)}{\#R_s(D)}$$

where #Rs(D^1) denotes the number of restrictive patterns discovered from sanitized database(D^1), and #Rs(D) denotes the number of restrictive patterns discovered from orginal database(D).

Misses Cost:

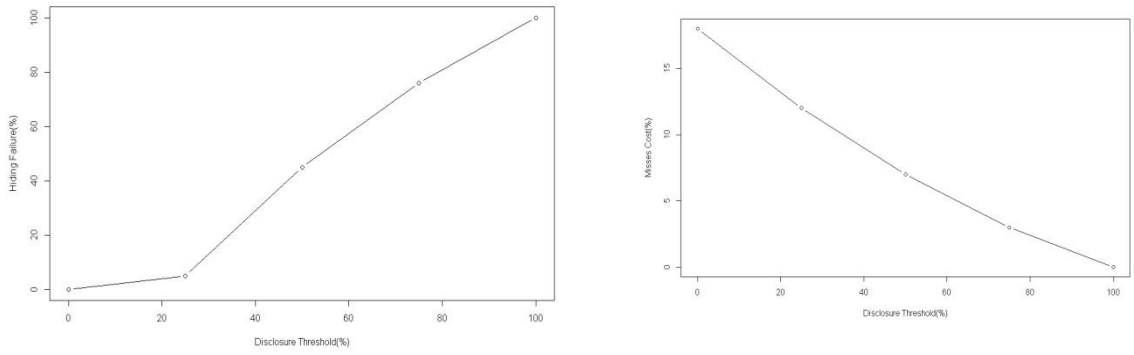
Some non restrictive patterns can be hidden by mining algorithms accidentally. This happens when some non-restrictive patterns lose support in the database due to the sanitization process. We call this problem as Misses Cost, and it is measured in terms of the percentage of legitimate patterns that are 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 #~ Rs(D) denotes the number of non-restrictive patterns discovered from original database D, and #~ Rs(D^1) denotes the number of non-restrictive patterns discovered from sanitized database D^1.

6.2 Performance Evaluation of Algorithm SWA

The effectiveness is measured in terms of the Hiding failure, as well as the Misses Cost. We selected for our experiments a set of ten sensitive association rules from the dataset. To do so, we ran the Fp-growth algorithm select such association rules. Figure 2a shows the effect of the disclosure threshold on the hiding failure and Figure 2b shows the effect of the disclosure threshold on the Misses cost. In case SWA, having different disclosure thresholds reduces the values of misses cost. Similarly, sliding the disclosure threshold improves the values of misses cost. On the other hand, the values of hiding failure increase since misses cost and hiding Failure is typically contradictory measures, i.e., improving one usually incurs a cost in the other.

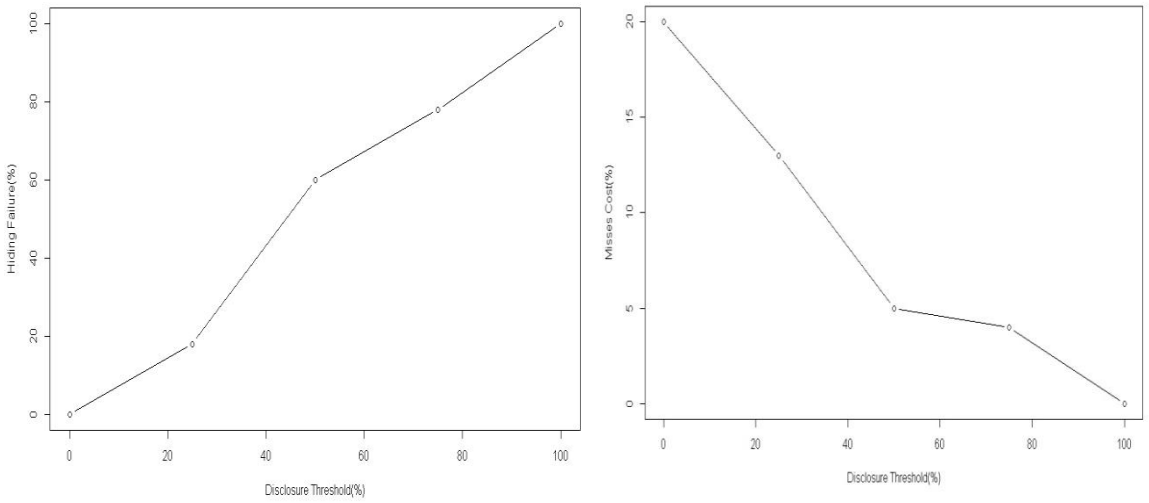


(a) **(b)**

Figure 2: Effect of Disclosure Threshold on hiding failure and Misses cost

6.3 Performance Evaluation of Algorithm IGA

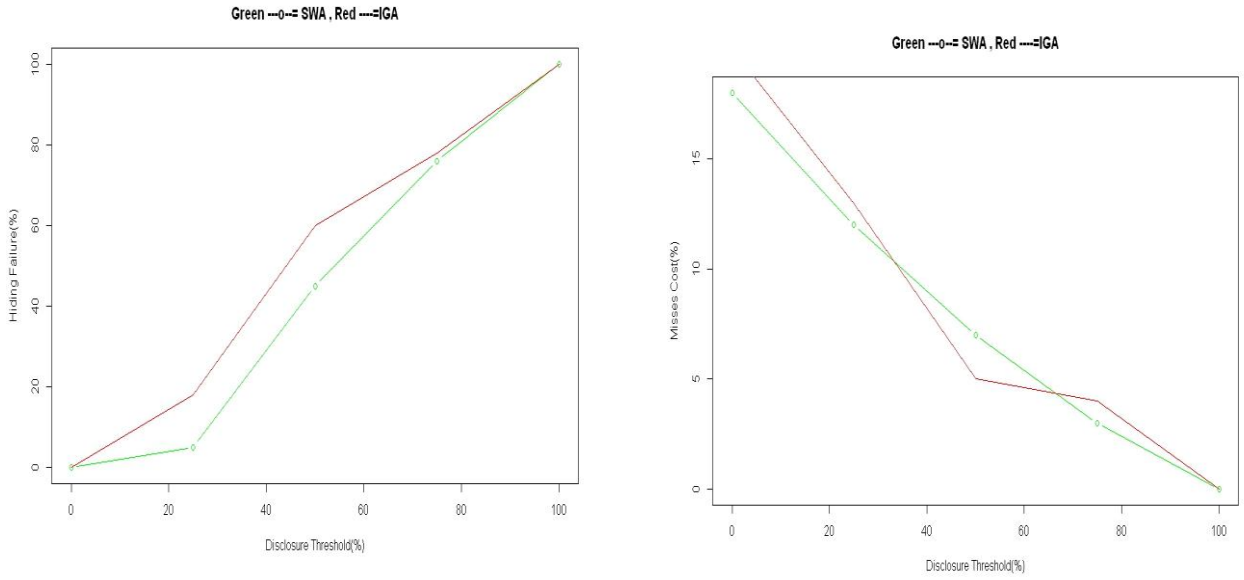
Figure 3a shows the effect of the disclosure threshold on the hiding failure and Figure 3b shows the effect of the disclosure threshold on the Misses cost. If disclosure threshold is set 0% then all sensitive rules hidden but misses cost is 20%. Disclosure threshold is set 50% then only 40% of sensitive rules hidden and Misses cost is only 8%.



(a) **(b)**

Figure 3: Effect of Disclosure Threshold on hiding failure and Misses cost

6.4 Comparative Evaluation of the Algorithms



(a) **(b)**

Figure 4: Effect of Disclosure Threshold on hiding failure and Misses cost

The effect of the disclosure threshold on the hiding and disclosure threshold on the Misses cost of the algorithm are shown in Figure 4a and Figure 4b. In both cases SWA algorithm has an advantage over the IGA algorithm. SWA also has an advantage over the IGA algorithm. The advantage is that SWA allows a database owner to set a specific disclosure threshold for each restrictive rule.

V. CONCLUSION

In this paper we presented two basic approaches in order to protect sensitive rules from disclosure. The first approach scans one group of transactions at a time and sanitizes the sensitive rules presented in such

transactions based on a set of disclosure threshold defined by a database owner. There is a disclosure threshold that can be assigned to each sensitive association rule. The second approach groups sensitive association rules in clusters of rules sharing the same itemsets. If two or more sensitive rules intersect, by sanitizing the shared item of these sensitive rules, one would take care of hiding such sensitive rules in one step. We also measured the performance of the algorithms according to two criteria: 1) the effect of the disclosure threshold on the hiding failure and 2) the effect of the disclosure threshold on the Misses cost. We concluded that SWA algorithm has an advantage over the IGA algorithm. The advantage is that SWA allows a database owner to set a specific disclosure threshold for each sensitive rule.

REFERENCES

[1]. Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., and Verykios, V.S. Disclosure limitation of sensitive rules. In: Scheuermann P, ed. Proc. of the IEEE Knowledge and Data Exchange Workshop (KDEX'99). IEEE Computer Society, 1999. 45-52.

[2]. R. Agrawal and R. Srikant. Privacy Preserving Data Mining. Proceedings of ACM SIGMOD Conference, 2000.

[3]. Chen, X., Orlowska, M., and Li, X. A new framework for privacy preserving data sharing. In: Proc. of the 4th IEEE ICDM Workshop: Privacy and Security Aspects of Data Mining. IEEE Computer Society, 2004. 47-56.

[4]. Dasseni, E., Verykios, V.S., Elmagarmid, A., and Bertino, E. Hiding association rules by using confidence and support. In: Proc. of the 4th Int'l Information Hiding Workshop (IHW'01). Springer-Verlag, 2001. 369 383.

[5]. Oliveira, S.R.M. and Zaïane, O.R. Privacy preserving frequent itemset mining. In: Proc. of the 2nd IEEE ICDM Workshop on Privacy, Security and Data Mining. Australian Computer Society, 2002. 43-54.

[6]. Oliveira, S.R.M. and Zaïane, O.R. A unified framework for protecting sensitive association rules in business collaboration. Int'l Journal of Business Intelligence and Data Mining, 2006, 1(3):247-287.

[7]. Saygin, Y., Verykios, V.S., and Clifton, C. Using unknowns to prevent discovery of association rules. SIGMOD Record, 2001, 30(4):45-54.

[8]. S. Oliveira, O. Zaiane, Protecting sensitive knowledge by data sanitization, Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03) 99–106, Melbourne, FL, November 2003.

[9]. Wang, E.T., Lee, G., and Lin, Y.T. A novel method for protecting sensitive knowledge in association rules mining. In: Proc. of the 29th Annual Int'l Computer Software and Applications Conf.. IEEE Computer Society, 2005. 511-516.