AN ASSOCIATION RULE HIDING METHOD BASED ON GROUPING AND WEIGHTING

Mohammad Reza KEYVANPOUR[1], Mehrnoush BARANI SHIRZAD[2], Elnaz MAHDI KHOSHOUEI[3]

This study proposes a method for association rule hiding to deal with the risk of disclosing sensitive information. Our model called GW algorithm utilizes weighting, in which removing some items and reducing support or confidence of sensitive association rule improves association rule hiding. For selecting items to remove a sensitive rule, a grouping strategy is introduced that selects items whose removal helps hide other sensitive rules. To reduce the side effects, non-sensitive rules are applied as well. The experimental results indicate that GW algorithm outperforms baselines in terms of required time, misses costs and artificial patterns with fewer side effects.

Keywords: Data Mining, Privacy Preserving Data Mining, Association Rule Hiding

1. Introduction

Association rule mining methods work towards discovering frequent items and associative rules in large databases. Since in association rule mining disclosing the relationship between items can reveal sensitive information and knowledge, techniques of privacy preserving data mining (PPDM) to protect the sensitive association rules are applied. Association rule hiding is a PPDM strategy has been proposed with the aim of removing sensitive association rules before releasing the database.

Sensitive association rules in large data repositories must be protected [1] Limiting the access to confidential data is considered as a way of protecting them [2] but this method does not guarantee the complete protection of sensitive data. Security preserving recently has drawn a considerable attention [3]. Association rule hiding methods are expected to remove sensitive association rules in a way that bring the fewest side effects [4]. In [5] authors defined the problem as follows: for a transactional database with minimum support, minimum confidence and a set of sensitive rules, the database should change in a way that with the

[1] Prof., Dept. of Computer Engineering, University of Alzahra, Tehran, Iran, E-mail: Keyvanpour@alzahra.ac.ir

[2] Data Mining Laboratory, Dept. of Computer Engineering, University of Alzahra, Tehran, Iran

[3] Dept. of Computer Engineering, University of Alzahra, Tehran, Iran

same minimum support and confidence, all strong rules in the database satisfy the following criteria: 1) not having sensitive rules 2) not hiding non-sensitive rules 3) not having spurious rule.

Association rule hiding can be categorized to different approaches [6] among them heuristic based approach which aims to find a set of transactions and items to be modified has high scalability and high speed [7]. This method is computationally optimum [7]. Considering high scalability and high speed, we apply this approach in this study. Authors in [8] provided a conceptual framework for association rule hiding composed of three components, including 1) Data Pre-processing; 2) Association rule mining and 3) Sensitive association rule hiding. This proposal focuses on the third component.

In this paper, a new algorithm called GW algorithm is proposed that introduces a novel rule grouping strategy to select items for modification. The goal of this algorithm is to select items and remove them to have the maximum effect on sensitive rules and the least effect on non-sensitive rules. Also a weighting method is proposed in order to reduce the side effects on mined knowledge form sensitive and non-sensitive rules.

2. Related Work

Authors in [7] provided a categorization of related studies of association rule hiding. They categorized current algorithms to following approaches: 1) heuristic distortion, 2) heuristic blocking, 3) fast heuristic, 4) border-based, 5) probabilistic distortion, 6) exact distortion, 7) exact database extension, 8) database reconstruction, 9) anonymity-based hiding, and 10) inverse frequent itemset mining approaches.

The first heuristic method and the first association rule hiding algorithm was proposed in [9]. In [10] authors proposed five algorithms for hiding the sensitive knowledge with minimum impact on the database. Another algorithm was proposed [11] that searches the transactional database only once and reduces the support sensitive transactions. In [5] users can apply their limitations such as lack of pattern, no need for hiding non-sensitive rules or both of them. Study of [12] categorized sensitive transactions according to their degree of conflict and victim-items for modification are selected in a way that do not have an impact on other non-sensitive rules. Methods which remove items from or insert items into the database belong to distortion-based groups, in which they do not have disclosure of information, but they contain spurious knowledge [13]. Authors in [14] used unknown values rather than reduce support and confidence of rules, to preventing the mining of spurious rules. This algorithm is a block-based algorithm. In these databases support and confidence are defined in a range whose lower and upper bound is computable. For a rule (X→Y) support is defined as equation (1):

$$Sup(X \to Y):[\min Sup(X \to Y), \max Sup(\mathrm{X} \to \mathrm{Y})] \quad (1)$$

$$minSup(X \to Y) = \frac{a}{|D|}$$

$$maxSup(X \to Y) = \frac{a+b+c+d}{|D|}$$

where $a = |(X=1) \cup (Y=1)$, b=|(X=1)∪(Y=?)|, c=|(X=?)∪(Y=1)| and d=|(X=?)∪(Y=?)| and $\min Sup(X \to Y)$ shows the percentage of the transactions which contain 1s for all the items in $X \to Y$ and $\max Sup(X \to Y)$ is the percentage of the transactions that contain either 1 or ? for all the items in $X \to Y$. Similarly, confidence is defined in equation 2 as follows:

$$Conf(X \to Y):[minConf(X \to Y), maxConf(X \to Y)] \quad (2)$$

$$minConf(X \to Y) = \frac{a}{|(X=1)|+e+d}$$

$$maxConf(X \to Y) = \frac{a+b+c+d}{|(X=1)|+e+d}$$

where e=|(X=?)∪(Y=0)|.

Study of [15] mentioned the problem with the blocking approach in which itemsets with unknowns are minable with all users. They proposed to provide rules that are not available in the database and apply unknown values in their itemsets. Also, in [16] authors considered community detection as an important information source for information security services to spot user’s misbehaving an alternative method has been developed. The border-based approach within the sanitization of the database instead of focusing on all non-sensitive items only focuses on frequent non-sensitive items. Border-based approach has been investigated as well [17,18]. Constraint-satisfaction based approach transforms association rule hiding into an optimization problem [19,20]. In reconstruction-based approach the aim is to prevent disclosure of sensitive knowledge rather than sensitive data. In [21] authors presented a framework for this approach including.

3. Definitions

Let I={i_1, ..., i_n} be a set of items and D be a set of transactional databases. Itemset X is a subset of I. Each transaction t∈ D has identity number indicated by TID. The transaction t supports X itemset if X⊂t. It is assumed that the items of an itemset or a transaction are saved in alphabetic order. Table 1 shows a database with 3 items and 6 transactions. AB is a set of items supported by T1. Itemset X has s support if s percent of transactions support it. Support X is indicated with Sup(X). All possible itemsets and their supports are shown in Table 2.

Table 1

Item and transactions Transactions, their possible itemsets and supports

TID	Items
T_1	ABC
T_2	ABC
T_3	ABC
T_4	AB
T_5	A
T_6	AC

Table 2

Possible itemsets and supports for table 1 items

Itemset	Support
A	100%
B	66%
C	66%
AB	66%
AC	66%
BC	50%
ABC	50%

Itemset X is frequent if the number of its repetitions in D is at least equal to a threshold of minsup. An association rule is in the form of X→Y (where X⊂I, Y⊂I and X∩Y= ∅) in which the transactional database supports this rule with a value larger than the minsup value and confidence value for this rule is at least equal to minconf. It is assumed that X and Y are two itemsets. $X \Rightarrow Y$ is called an association rule with a minsup and a minconf if it has the following conditions:

Support value for itemset $X \cup Y$ is at least equal to minsup

Confidence value for $X \Rightarrow Y$ is at least equal to minconf

The first criterion guarantees that the number of transactions that are related to the rule are enough. The second criterion guarantees that the rule is strong enough based on conditional probability.

Rule X→Y have support of s if $S \leq Sup$(X→Y), equation (3) shows Sup(X→Y):

$$Sup(X \to Y) = \frac{|X \cup Y| * 100}{|N|} \tag{3}$$

Where N is the number of transactions of D) also X→Y in database D has confidence, if $C \leq Conf$(X→Y). Equation (4) below shows the $Conf$(X→Y) :

$$Conf(X \rightarrow Y) = \frac{|X \cup Y| * 100}{|X|} \tag{4}$$

Where |A| is the number of occurrences of itemset A in a set of transactions in D and A in transaction t occurs if $A \subset t$. Support is a metric of evaluating frequency of a rule while confidence is a measure of assessing the power of relation between itemsets.

An association rule mining method finds all frequents items. Association rules are created from these items. In order to preserve privacy, association rules which are strong and considered interesting should be hidden [10]. These rules are referred as sensitive association rules.

4. Proposed Method: GW Algorithm

Herein, the transactional databases are considered in the binary format, where the existence or non-existence of attributes are indicated by 1 and 0. The goal in each stage is to hide one rule and method aims to hide all sensitive rules. Here sensitive items are first selected independent of transaction and then sensitive transactions are preferred for modification by the proposed weighting algorithm. Three main stages are taken (Fig. 1). Steps including association rule mining, sensitive association rule selection and association rule hiding. First, association rules are mined according to minsup and minconf. Then sensitive rules are selected by the user and delivered to the association rule hiding algorithm. The sensitive association rule in the sanitized database is not extractable.

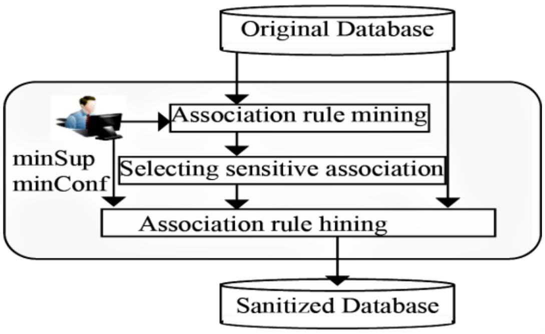

Fig.1. Association rule hiding framework

4.1. Association Rule Mining

This section consists of 1) data pre-processing, 2) generation of a set of frequent items, 3) production of association rules (Fig. 2). In Pre-processing phase, some processes are performed on database in order to transform the data to a proper form as the input of the algorithm used to find frequent items. These processes include operations such as collection, selection, and cleaning. In Finding Frequent Items step FP-Growth which is a divide and conquer based algorithm is used for generating a set of frequent items. The FP-Growth is an efficient and scalable algorithm which first constructs the FP-Tree and then mines it to find a set of frequent patterns [22]. In Finding Association Rules step, for Y a frequent itemset where Y=I_1, I_2, ...,I_k, k≥2 all rules which use these items are generated. The left side of this rule is a set of X of Y that has at least one item of y. The right side is defined in the same way but these two sides do not share any item. In this way from a set of items with k members, m rule can be mined which is computed with following formula (equation (5)):

$$m = C\binom{k}{1}(2^{k-1}-1)+C\binom{k}{2}(2^{k-2}-1)+\ldots+C\binom{k}{k-2}(2^{k-1}-1)+c\binom{k}{k-1} \qquad (5)$$

After mining rules, the rules whose confidential values are higher than the threshold are set as candidates.

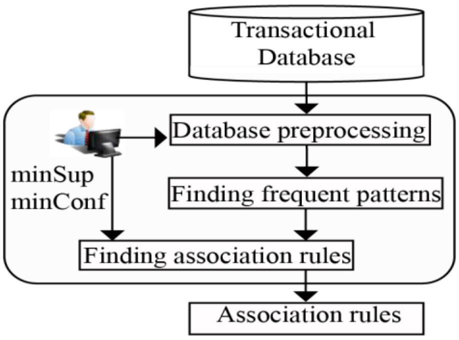

Fig.2. Association rule mining

4.2. Sensitive Association Rule Selection

In this part users select sensitive association rules according to their preferences, policy enforcement or business benefit conflict [23]. Automatic labelling is considered as a task [24]. Association rules are shown to users in a sorted format according to their length in an ascending order [25]. In each iteration, after selecting one sensitive association rule all related extra rules are automatically considered as sensitive association rules and are added to the sensitive association rule set.

4.3. Association Rule Hiding

An association rule hiding strategy includes [23]: a) Modifying strategy selection: in which removing the sensitive right-side items reduces support or confidence. b) Selection of sensitive items: here an item is selected such that by removing that item a larger number of sensitive items can be hidden. A sensitive rule grouping strategy is proposed in follows. c) Selection of transactions: after determining sensitive items of all rules, the algorithm starts from the strongest sensitive association rule and finds the transactions. Then sensitive items are removed from related transactions and in each iteration the hiding rule is evaluated. If a rule is hidden by reduced support or confidence, the removing process stops and the algorithm switches to the next rule. Fig. 3 illustrates the process of rule hiding.

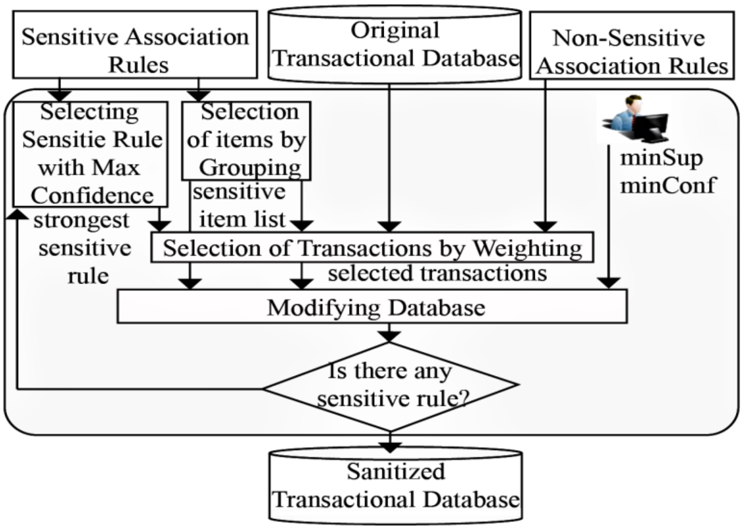
Sensitive Association Rules
Original Transactional Database
Non-Sensitive Association Rules
Selecting Sensitie Rule with Max Confidence
Selection of items by Grouping
minSup
minConf
sensitive item list
strongest sensitive rule
Selection of Transactions by Weighting
selected transactions
Modifying Database
Is there any sensitive rule?
Sanitized Transactional Database

Fig.3. Association rule hiding

4.3.1. Selection of Sensitive Items by Grouping

Selecting proper items to be modified is one essential element of the association rule hiding technique. Reducing the number of modifications performed on the database is a peripheral goal in association rule hiding. The selection of sensitive items and transactions is done in a way that it has the fewest side effects on other rules and databases. Here we present an algorithm for grouping sensitive rules according to their common items to select an item which simultaneously helps hide multiple sensitive rules.

The main idea of this algorithm is constructing groups of similar rules which have a higher level of similarity in lower levels. For every sensitive rule several levels are considered which in each level, one sensitive item is introduced. Sensitive items that are introduced in a higher level are more sensitive since they include a larger number of rules and in lower levels a larger number of rules are joint and have fewer sensitive items.

```
Grouping Algorithm ( $R_S$, Threshold, d, $d_{max}$ )
while ($|R_S|$>Threshold and d<$d_{max}$) do
find items of sensitive rules
find frequency of pairs of right and left in rules
select item pair (X<Y) with maximum frequency
compute cover set C= X.Y
$\forall r_j \in C_j : r_j . i_s[d] = Y$
```

```
7 if $|C|$ > Threshold C= C - (X and Y) Sensitive Rule Grouping Algorithm ($R_s$, Threshold, d+1, $d_{max}$)
8 else $R_s = R_s – C$, go to step 1
9 $\forall r_j \in R_s : r_j . i_s [d]$ = a random item of the right side
Output: lists of sensitive rules & sensitive items
```

Input parameters of algorithm are Rs: a set of sensitive rule, threshold: maximum number of rules of a group where $1 \leq \text{Threshold} \leq |R_s|$, d: level in each iteration where $0 \leq d \leq d_{max}$ and d_{max}: maximum level that is determined by the user and $0 \leq d_{max} \leq |R_s|$. The process repeats line 2 to 8 while $|R_S| > \text{Threshold}$ and $d < d_{max}$ otherwise jumps to line 9. In line 2, to find items of the sensitive rule, for all sensitive items the matrix of sensitive items is constructed (rows show left side items and columns indicate right side items). In line 3 the number of iterations in each pair of items in the sensitive rule calculated. In line 4 pairs of (X,Y) with maximum frequency are selected. In line 5 the cover sets of C(X,Y) are computed which include sensitive rules that have X on the left side and Y on the right side. In line 6 for each member of the cover set of C(X,Y), the sensitive rule in the current level is defined by item Y. In line 7 if the size of cover set C(X,Y) is bigger than the threshold, X,Y are removed from rules of cover sets C(X,Y) and the Grouping Algorithm is recursively recalled with value of R_s, Threshold, d+1, d_{max}. In line 8 the algorithm removes the rules of the cover sets of C(X,Y) from and starts from step 1 with the new sensitive rule set. In line 9 one of the right-side items of a sensitive rule is considered randomly as a sensitive item of the rule in that level. Output includes the lists of sensitive rules and sensitive items in each level.

4.3.2. Selection of Transactions by Weighting

This part consists of three steps: a) finding transactions that support the sensitive rule, b) computing the number of transaction that must be removed, and c) computing the weight of transactions. We propose a new method for step c. In the first step, transactions that completely support the selected rule are selected. If transactions that just support the left side of the selected rule is picked, removing supportive items reduce the left side of the rule and confidence of the rule will increase which is against the main goal of the problem. Considering removing the transactions that support the right side of the rule, will bring no effect on support and confidence of the sensitive rule. So transactions with one or more sensitive rules are considered. In the second step, assuming that $\sum X \cup Y$ is set of transactions that support $r_1 : X \rightarrow Y$, to decrease the confidence of the rule compared to the threshold, the minimum number of transactions that must be modified is computed by equation (6) as follows:

$$n = [Sup(X \cup Y) - Sup(X) * minConf|D|] \qquad (6)$$

Since the strategy simultaneously reduces support and confidence of the rule, in each iteration of the modification, the support of the rule is compared with the threshold support. If before modifying all needed transactions, the rule is hidden by reduced support, the process of transaction modification will stop. As for the final step, a metric to compare and rank candidate transactions to select a subset with length of N is needed. This metric will have the least side effect and is named weight. To select transactions for removal ranking strategies have been proposed [10,23].

Our proposed weight method focused on side effects of modifications on other rules including both sensitive and non-sensitive ones. Table 3 illustrates the effect of removing x sensitive items from transaction Ti on other rules in different situations.

Table 3

Weighting Rules

Row	Rule	Side	Support of rule by T_i	Effect on supp	Effect on Conf	Probable side effects
1	Sen	Left	Complete	Reduce	Reduce	Desirable
2	Sen	Left	Left	-	Increase	Strengthen sensitive
3	Sen	Left	Right	-	-	-
4	Sen	Right	Complete	Reduce	Reduce	Desirable
5	Sen	Right	Left	-	-	-
6	Sen	Right	Right	-	-	-
7	Non	Left	Complete	Reduce	Reduce	Missed rule, weaken
8	Non	Left	Left	-	Increase	Strengthen non-
9	Non	Left	Right	-	-	-
10	Non	Right	Complete	Reduce	Reduce	Missed rule, weaken
11	Non	Right	Left	-	-	-
12	Non	Right	Right	-	-	-

Integrating the rules in the table 3, the proposed weight for ranking the selected transactions is formulated as equation (7):

$$W(T_i) = \frac{1 + 2N_{sf}}{1 + 2N_{nsf} + N_{sp} + 1} \tag{7}$$

Where $W(T_i)$ is the weight for candidate Ti , for all selected items from grouping. N_{sf} is the number of sensitive rules containing sensitive items which Ti supports completely (rows 1 and 4). N_{nsf} shows the number of non-sensitive rules containing sensitive items that Ti supports completely (rows 7 and 10). N_{sp} shows the number of sensitive rules containing sensitive items on the left side that Ti supports (row 2) and N_{nsp} is the number of non-sensitive rules containing sensitive items on the left side that Ti supports (row 4). For states in which both support and confidence are affected, 2 is considered as factor; and for situation in which only one of the support or confidence are influenced, 1 is set as factor.

In the worst situation, transactions do not support any other sensitive rule except the current rule and partially support all sensitive rules and all non-sensitive rules completely. In the best situation, transactions support all sensitive rules completely and do not support any non-sensitive rules (both completely and partially) and do not support any sensitive rule partially. Equation (8) formulates this situation:

$$\frac{1}{1+2N_{nsf}+N_{sp}} \leq W(T_i) \leq 2 \tag{8}$$

4.3.3. Modifying Database

In this stage, after hiding each rule, the selected transactions are determined and after modifying each transaction, support of the sensitive rule is compared against the support's threshold value; if it is less than the threshold, the process of hiding that rule will stop.

5. Experiments

5.1 Experimental Settings

To evaluate the proposed methods two datasets including Mushroom and T10I4D100k are mined. Mushroom is provided from UCI and T10I4D100k is a spars artificial dataset designed by IBM researchers to mine frequent patterns. In this study transactions were selected by stratified sampling method. Table 4 lists the dataset's properties.

Table 4

Dataset's Properties

Dataset	No. transactions	No. items	Ave. length of transactions
Mushroom	8124	119	23
T10I40D700	700	1000	10

Evaluation metrics of association rule hiding algorithms are of two types: internal measure which their value change by modifying the hidden strategy and external measures relate to the algorithm's behaviour in interaction with big data or computational speed [7]. Internal measures are categorized into data sharing based and pattern sharing based [26]. Here, a pattern sharing metric called Misses Cost [26,27] and a data sharing measure called Artificial Pattern [27] are applied. Artificial Pattern indicates the generated artificial rules due to data modification, and is defined as equation (9):

$$A_P = \frac{|P'| - |P \cap P'|}{|P'|} \tag{9}$$

where P is the mined itemset from the database and P' is the mined itemset from the sanitized database D'.

Misses cost evaluates the number of non-sensitive rules that are hidden because of the hiding process whose support or confidence decrease less than the threshold and are not available in the sanitized database. Equation (10) shows the misses cost:

$$M_C = \frac{S_R^{'}(D) - S_R^{'}(D^{'})}{S_R^{'}(D)} \tag{10}$$

Where $SR'(D)$ is the number of non-sensitive rules in the original database and $SR'(D')$ shows the number of non-sensitive rules in the sanitized database.

Since the number of sensitive rules in comparison with other rules is small for each dataset, different percentages of rules are considered as sensitive rules, and the algorithms are tested by them. Each test is done 10 times and the average of results is reported. The number of sensitive association rules is small in comparison with non-sensitive association rules, we examine different percentage of sensitive association rules.

Table 5

Datasets for test

Title	Dataset	No. sensitive rule	Percentage of sensitive rule
D_1	Mushroom	5	0.47
D_2	Mushroom	31	2.97
D_3	Mushroom	95	9.11
D_4	T10I40D700	2	0.037
D_5	T10I40D700	57	1.07
D_6	T10I40D700	259	4.9

Authors in [28] applied self-training as an algorithm to classify unlabeled data with small amount of labelled data. The parameter tuning is listed in table 6. Several values for minsup and minconf have been set and we found execution time and the memory requirement increase when these parameters are set to higher values. For grouping, the threshold is set to 1 to have all levels of grouping.

Table 6

Parameter tuning

Dataset	minsup	minconf	No. mined association rules
Mushroom	0.5	0.5	1042
T10I40D700	0.005	0.95	5283

Measures include number of modifications, performance time, Misses Cost and Artificial Patterns. The baseline includes RS1 Algorithm, RS2 Algorithm and 1.b Algorithm [10] and a combined method. All algorithms apply removing items of the right side of the transaction as hiding strategy and all are aimed at not having a sensitive rule. The most significant differences of these methods are in strategies of sensitive item selection and transaction selection for modification, summarized in Table 7.

Table 7

Algorithms

Method	Item selection strategy	Sensitive transaction selection strategy
GW	Proposed Grouping	Proposed Weighting
Rs2	Most frequent	$W(T)=\frac{1}{1+NUM_{non-sen}(t)}$
RS1	Most frequent	$W(T)=\frac{NUM_{sen(t)}}{[1+NUM_{non-sen(t)}]*NUM_{sen(t)}}$
1.b	Most frequent	Minimum transaction length
Combined	Most frequent	Proposed weighting

5.1 Experimental Results

Measures include number of modifications, performance time, Misses Cost and Artificial Patterns. Having more modifications indicates the more difference between initial database and sanitized database. Less number of modifications is desirable [26]. We examined three variants of our method including applying both grouping and weighting (G-W), only weighting (nG-W), just grouping (G-nW), and without grouping and weighting (nG-nW). In absence of grouping and weighting selection is done randomly.

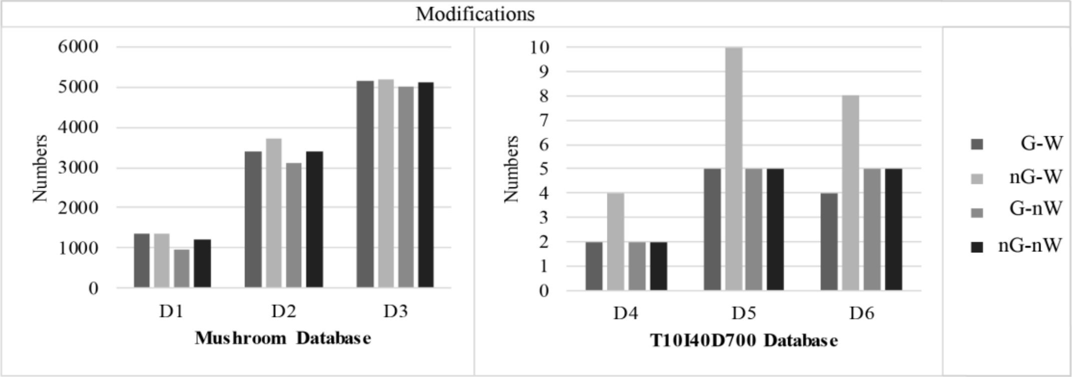

Fig. 4. Number of modifications for each dataset part

As it can be seen in Fig. 4, for the sparse dataset T10I40D700, GW algorithm gains the best results between all variants, according to number of modifications. For Mushroom database GW algorithm provides competitive results in comparison with others. G-nW outperforms other models. This can be as a result of the fact that weighting reduces the attempting of grouping for modifying the database. Due to the number of modifications on database, the GW algorithm has been selected as our proposed method to compare with baselines.

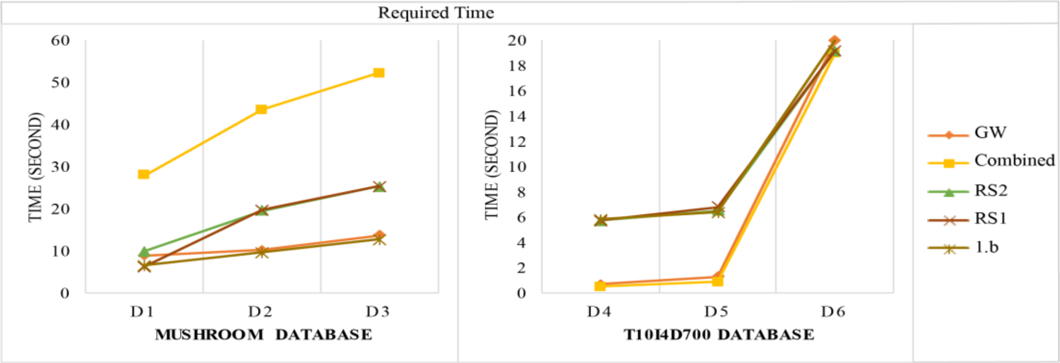

Fig.5. Required Time

Fig. 5 indicates the performance time in seconds for GW and other algorithms. Clearly, GW algorithm shows best results on Mushroom dataset. This is quite similar to the result of b.1 algorithm. GW algorithm consumes less time in item selection and more time on transaction selection via weighting. In contrast b.1 algorithm spends more time on item selection and less time for selecting transactions. Since grouping is done for all rules just once at the beginning of the algorithm, item selection is quick while in RS1 and RS2 this process is repeated for each rule and thus these algorithms spend considerable time on this phase. In the combined algorithm both item selection and transaction selection are time consuming. Time performance for dataset T10I40D700 dataset is similar to Mushroom results. Since this dataset is not as dense as Mushroom the time for weighting is less and the whole time is shorter. The GW algorithm worked better than RS1, RS2 and 1.b. and is compatible with the combined method.

Fig. 6 shows the misses cost of all algorithms. RS2 algorithm gained the best results in comparison with others on Mushroom dataset. GW algorithm has the worst performance according to misses cost. One reason for this weakness is the fact that grouping was done just once. Also, in such a compressed database like Mushroom after hiding one rule, number of modifications is large which there are significant differences between sanitized database and its original version. Performance of 1.b proves this. For T10I40D700 dataset comparisons between GW algorithm with others indicate that GW algorithm has better results by grouping items and focusing on decreasing the side effects. In spars dataset like T10I40D700 item frequency is not a proper metric for selecting items and weighting measure in spars dataset is the most suitable. Due to selecting items without considering side effects and sensitive and non-sensitive rules, 1.b algorithm has more misses rules, in comparison with other methods.

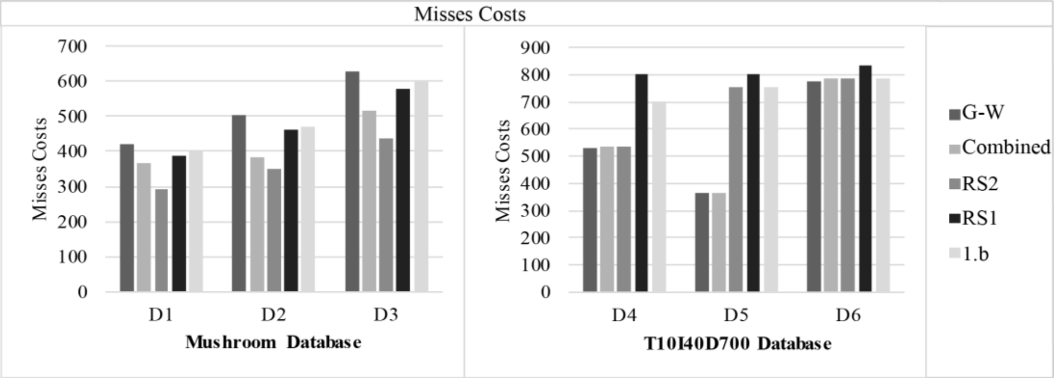

Fig. 6. Misses Costs

Fig. 7 presents information about artificial patterns. In Mushroom dataset which is a dense dataset in which misses cost is high, less artificial patterns are visible. In Algorithm 1.b the number of generated artificial patterns is minimum.

GW algorithm is almost the second minimum. All weighting algorithms do not consider the side effects of producing artificial patterns. In T10I40D700 dataset RS2, RS1 and 1.b gained zero for D5 and all values for D6 equal to zero. Artificial patterns are generated when the proposed weighting method was applied in this sparse dataset while other algorithms did better. This may be rooted in the fact that these patterns exist in the database and the weighting method cannot distinguish among them before modifying the database.

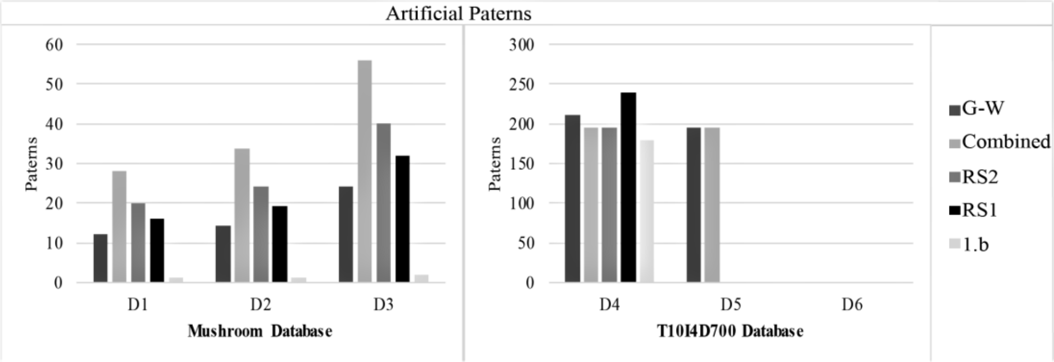

Fig. 7. Artificial patterns

6. Conclusions

We proposed GW algorithm a heuristic algorithm based on grouping and weighting. This strategy consists of a method for item selection for modification based on grouping sensitive rules in order to select items whose removal leads to hide sensitive rules and have less impact on non-sensitive rules. For transaction selection, a new heuristic method based on weighting was presented to consider information of sensitive, non-sensitive and itemset in order to reduce the side effects on mined knowledge. In order to clarify our model novelty, we have compared our model with several baselines under the same conditions. All models have been conducted on the same part of database and all utilized technique of removing items of the right side of the transaction as their hiding strategy with the aim of elimination of a sensitive rule. Comparison between GW algorithm and baselines proved its merit based on several criteria. By GW strategy, many sensitive rules are hidden while other rules are visible. Thus, the time of hiding is reduced therefore required time for this model has been decreased. Regarding misses costs GW algorithm almost outperformed the baselines. Reduction in artificial patterns is considered as a plan. Future challenges include questions as to whether parallel processing or cloud processing can help to reduce the time of association rule hiding.

REFERENCES

[1]. *S. Keer, A. Singh,* "Hiding Sensitive Association Rule Using Clusters of Sensitive Association Rule", in International Journal of Computer Science and Network, **vol.3**, 2012.

[2]. *C. Clifton, D. Mark*s, "Security and Privacy Implications of Data Mining", ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery 5-19, 1996.

[3]. *M. Moradi, M. Keyvanpour,* "CAPTCHA and its Alternatives: A Review". in Security and Communication Networks 8:12 2135-2156, 2015.

[4]. *H.Q. Le, S.Arch-int, N. Arch-int,* "Association Rule Hiding Based on Intersection Lattice", in Mathematical Problems in Engineering, 2013.

[5]. *Y.H. Wu, Ch. Chiang, A. Chen,* "Hiding Sensitive Association Rule with Limited Side Effects", IEEE Transactions on Knowledge and Data Engineering 19:1, 2007

[6]. *A. Gkoulalas-Divanis, J. Haritsa, M. Kantarcioglu,* "Privacy Issues in Association Rule Mining", in Frequent Pattern Mining, 2014, pp. 369-401

[7]. *S. Verykios Vassilios*, "Association rule hiding methods" in Data Mining Knowl Discov **vol.3**, 2013, pp. 28–36

[8]. *H. Q. Le, S. A. Arch-int,* "Conceptual Framework for Privacy Preserving of Association Rule Mining in E-Commerce", in IEEE Conference on Industrial Electronics and Applications 1999-2003, 2012.

[9]. *M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios,* "Disclosure Limitation of Sensitive Rules" in Proceedings of IEEE Workshop on Knowledge and Data Engineering Exchange 45, 1999.

[10]. *S.V. Verykios, A.K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni,* "Association Rule Hiding" in IEEE Transactions on knowledge and data engineering, **vol.16**, no.4, 2004.

[11]. *S.R. M. Oliveira, O.R. Zaiane,* "Protecting Sensitive Knowledge by Data Sanitization", in Proceedings of IEEE International Conference on Data Mining, 2003, pp.19-2

[12]. *X. Li, Z. Liu, C. Zuo,* "Hiding Association Rules Based on Relative-non-Sensitive Frequent Item sets", Proceedings of IEEE International Conference on Cognitive Informatics, 2009, pp. 384-389

[13]. *A. Gkoulalas-Divanis, V.S. Verykios,* "Association Rule Hiding for Data Mining". US: Springer, 2010.

[14]. *Y.Saygin, V.S. Verykios, C. Clifton,* " Using Unknowns to Prevent Discovery of Association Rules", 2001.

[15]. *E.D. Pontikakis, Y. Theodoridis, A.A Tsitsonis, L. Chang, And V.S. Verykios,* “Quantitative and Qualitative Analysis of Blocking in Association Rule Hiding” in Proceedings of the ACM Workshop on Privacy in the Electronic Society, 2004, pp.29-30

[16]. M.*Pourkazemi, M. Keyvanpour,* “A survey on community detection methods based on the nature of social networks”, in 3rd International Conference on Computer and Knowledge Engineering, IEEE, 2013, pp.114–120

[17]. *X.Sun, P.S Yu,* “A Border-based Approach for Hiding Sensitive Frequent Item sets”. Proceedings of IEEE International Conference on Data Mining, 2005, pp.27-30

[18]. *A. Farea, A. Karci,* “Towards Association Rule Hiding Heuristics vs Borderbased Approaches”, IEEE International Conference on Electrical and Electronics Engineering, 2015, pp.28-29

[19]. *S.Menon, S. Sarkar, S. Mukherjree,* “Maximizing Accuracy of Shared Databases When Concealing Sensitive Patterns”. Information System Research **vol.16**, no.3, 2005, pp.256-270

[20]. *A.Gkoulalas-Divanis, V.S. Verykios,* “Extract knowledge hiding through database extension”, IEEE Transactions on Knowledge and Data Engineering, **vol.21**, no.5, 2009, pp.699-713

[21]. K. Pathak, N.S. Chaudhari, A. Tiwari, “Privacy-Preserving Data Sharing Using Data Reconstruction Based Approach”, IJCA Special Issue on Communication Security comnetcs, **vol.1**, 2012, pp.64-68

[22]. J. *Han, J., Pei, Y. Yin,* “Mining Frequent Patterns without Candidate Generation”. ACM SIGMOD international conference on Management of data, NY, USA, 2000.

[23]. *P. Cheng, J.F. Roddick, S.C. Chu, and C.W. Lin,* “Privacy Preservation through a Greedy, Distortion-based Rule Hiding Method”. Applied Intelligence. **Vol.44**, no.2, 2016, pp. 295-306

[24]. *H. Hassanzadeh, M. Keyvanpour,* “A two-phase hybrid of semi-supervised and active learning approach for sequence labeling”. Journal Intelligent Data Analysis. **vol.17**, no.2, 2013.

[25]. *M. Kryszkiewicz,* “Representative Association Rules”. Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining 19988, pp.198-209

[26]. S. O. Oliveira, A. Zaïane, “Unified Framework for Protecting Sensitive Association Rules in Business Collaboration”. International Journal of Business Intelligence and Data Mining **vol.3**, no.1, 2006.

[27]. *S.R.M. Oliveira, O.R. Zaiane, “*Privacy Preserving Frequent Item set Mining”. Proceedingsof IEEE International Conference on Privacy, Security and Data Mining, 2002, pp. 43-54

[28]. *M. Keyvanpour, M.B. Imani,* “Semi-supervised text categorization: Exploiting unlabeled data using ensemble learning algorithms”. Intelligent Data Analysis, **vol17**, no.3, 2013, pp.367-38