Journal of Computer Science and Cybernetics, V.35, N.4 (2019), 337–354
DOI 10.15625/1813-9663/35/4/14131

DISTORTION-BASED HEURISTIC METHOD FOR SENSITIVE ASSOCIATION RULE HIDING

BAC LE[1,*], LIEN KIEU[2], DAT TRAN[3]

[1]*Faculty of Information Technology, University of Science, VNU-HCM, Vietnam*
[2]*Deparment of Mathematics-Informatics, People's Security University, HCMC, Vietnam*
[3]*Faculty of Information Sciences and Engineering, University of Canberra, Australia*
[*]*lhbac@ fit.hcmus.edu.vn*

Abstract. In the past few years, privacy issues in data mining have received considerable attention in the data mining literature. However, the problem of data security cannot simply be solved by restricting data collection or against unauthorized access, it should be dealt with by providing solutions that not only protect sensitive information, but also not affect to the accuracy of the results in data mining and not violate the sensitive knowledge related with individual privacy or competitive advantage in businesses. Sensitive association rule hiding is an important issue in privacy preserving data mining. The aim of association rule hiding is to minimize the side effects on the sanitized database, which means to reduce the number of missing non-sensitive rules and the number of generated ghost rules. Current methods for hiding sensitive rules cause side effects and data loss. In this paper, we introduce a new distortion-based method to hide sensitive rules. This method proposes the determination of critical transactions based on the number of non-sensitive maximal frequent itemsets that contain at least one item to the consequent of the sensitive rule, they can be directly affected by the modified transactions. Using this set, the number of non-sensitive itemsets that need to be considered is reduced dramatically. We compute the smallest number of transactions for modification in advance to minimize the damage to the database. Comparative experimental results on real datasets showed that the proposed method can achieve better results than other methods with fewer side effects and data loss.

Keywords. Privacy Preserving Data Ming; Association Rule Hiding; Side Effects; Distortion-Based Method.

1. INTRODUCTION

In today's competitive environment, collaboration between organizations and businesses is a requirement for their development. Successful collaboration can bring products to market faster, reduce production and logistics costs, drive market share and increase sales. Therefore, data sharing becomes important in the development of every member and partnership involved in collaboration. Data mining becomes a useful tool for extracting knowledge from shared data sources between parties. However, there is an increase of risks of disclosing the sensitive knowledge when the database is released to other parties or provided for data mining centers. For example, if X is an itemset of Honda motorcycle brands, Y is the itemset

of motorbike accidents, the announcement of the correlation between X and Y will cause disadvantages for the Honda motorcycle business, and provide a significant advantage for Hondas competitors. Therefore, data providers want to hide sensitive association rules so that they cannot be discovered by data mining algorithms. To address this issue, the original database can be modified by adding new items or removing existing items to reduce the support or the confidence of sensitive rules below specified thresholds set by the data owner. This research direction is essential when we want to protect privacy in data mining.

Association rule hiding is an emerging area of data mining known as data sanitization that aims to transform a transaction database into a sanitized version in order to protect sensitive knowledge and patterns, with sensitive rules set by the data owner. The studies in association rule hiding are mainly focused on proposing optimal algorithms with the least significant side effects to the database, so that any association rule mining algorithms that may be applied to the sanitized version will be incapable of uncovering the sensitive rules under certain parameter settings and will be able to mine the non-sensitive rules only. However, the problem arises in balancing the confidentiality of the disclosed data with the legitimate mining needs of the data users. Many different sanitization algorithm have been proposed to minimize the side effects on the sanitized database. Acording to [18], there are fifty-four scientific algorithms primarily spanning the period 2001 - 2017. Privacy-preserving data mining in association rules has been studied in the following main approaches: Heuristic, border-based, exact and evolutionary.

Heuristic approach includes efficient and fast algorithms that select a set of transactions using predefined criteria. Because of its high efficiency and scalability, heuristic methods recently attract a lot of attention from researchers. However, these algorithms produce undesirable side effects that lead to the identification of approximate solutions because of the fact that the heuristic-based algorithms always aim at taking locally best decisions with respect to the hiding of sensitive knowledge, but not globally best. Therefore, heuristic approaches fail to create optimal solutions. Some heuristic-based algorithms for hiding sensitive knowledge are as follows. [3] first proposed a protection algorithm for data sanitization to avoid the inference of association rules. [5] proposed three single rule heuristic hiding algorithms 1.a, 1.b and 2.a that are based on the reduction of either the support or the confidence of the sensitive rules, but not both. [15] was the first to introduce multiple rules hiding approach such as *minFIA, maxFIA,* and *IGA, Nave.* [2] proposed three effective, multiple association rule hiding heuristics such as *Aggregate, Disaggregate, Hybrid.* The Relevance_Sorting algorithm was introduced by [4] that formulates a heuristic for determining transactions for sanitization. In order to reduce the distortion ratio, this algorithm computes the minimum number of transactions that need to be modified to conceal a sensitive rule.

Border approach focuses on reduction of the side effects of sanitization on the non-sensitive itemsets. This approach considers the association rule hiding through the modification of the borders in the lattice of the frequent and infrequent itemsets of the original database. [16] first introduced the process of the modification of the borders to hide sensitive patterns while maintaining the non-sensitive itemsets with low support. The border-based algorithms sanitize the transactions with minimum impact on the results of the released database. The BBA [16], max-min1, and max-min2 [14] algorithms use the border theory to hide the frequent itemsets.

Exact approach tries to conceal the sensitive patterns by causing minimum distortion to

the sanitized database. It considers the problem of frequent itemset hiding as a constraint satisfaction problem (CSP), and formulates the CSP as an integer program to minimize the number of sanitized transactions or items [13]. Exact hiding methodologies achieve to model the hiding problem in a way that allows them to simultaneously hide all the sensitive knowledge. On the negative side, the exact algorithms required more computational time and more memory to solve optimization problems [6].

In the past few years, evolutionary algorithms have been studied for hiding sensitive itemsets. Other approaches include two selection strategies for selecting victim items and transactions. Besides, the genetic algorithm (GA)-based framework contains only transactions selection strategy to be deleted from the database or to be inserted into the database. The cpGA2DT [11], sGA2DT, and pGA2DT [9] are deletion-based algorithms that designed a fitness function to evaluate the goodness of chromosome that determines data sanitization side effects. Each chromosome encodes a solution consisting of a set of transactions to be deleted. In addition, in [10] a GA-based approach has been presented to sanitize the data in order to hide sensitive high utility itemsets through transaction insertion.

Recently, there are some new techniques for privacy preserving data mining have been proposed. A new and efficient approach has been introduced which benefits from the cuckoo optimization algorithm for the sensitive association rules hiding (COA4ARH). The algorithm in [1] is presented for calculating the minimum number of sensitive items which should be removed to hide sensitive rules, as well as limit the loss of non-sensitive rules. Privacy preserving ultility mining has become an important topic in privacy preserving data mining, Lin in [12] has proposed the algorithm which is designed to efficiently delete sensitive high profit in transaction databases or decrease their utilities using the concepts of maximum and minimum utility. In 2018, an electromagnetic field optimization algorithm (EFO4ARH) [17] has proven to have the best results for hiding sensitive rules. The algorithm shows a reduction in the side effects and better preservation of data quality.

The limitation of the above-mentioned approaches is that they cause side effects and data loss. In this paper, we introduce a new approach to hiding sensitive rules based on heuristic method. The algorithm hides rules by removing some items in the identified transactions that fully support for them, so that sensitive rules cannot be discovered in sanitized databased at some specified threshold, while the side effects and the number of removed items are minimized. The modified transactions are evaluated and selected by a weight value based on information about non-sensitive maximal itemsets that contain at least one item to the consequent of the sensitive rule which they support. The transactions that contain less non-sensitive ones are modified first. The database damage could be minimized by computing the smallest number of transactions to be modified in advance for hiding a given sensitive rule. We then determine one item to the consequent of the rule to remove from the modified transaction so that the side effects are minimized. We conduct experiments to compare the proposed method with the Relevance_Sorting algorithm [4], and our experimental results show that the proposed method in some cases achieves satisfactory results with fewer side effects and data loss.

The rest of the paper is organized as follows: Section 2 presents the problem statement and the notations used in this study. Section 3 describes the proposed method. Section 4 presents the experimental results and discussions, and Section 5 concludes the paper.

2. THE PROBLEM STATEMENT

Let $I = i_1, i_2, ..., i_m$ be a set of items available. An itemset X is a subset of I. A transaction database D is a relation consisting of a set of transactions. $D = t_1, t_2, ..., t_m$, where each transaction t_i is a set of items in I, such that $t_i \subseteq I$. Each transaction has a unique transaction identifier number denoted as TID.

Table 1. A transaction database D

TID	Items
1	a, b, c, g
2	b, d, e, f
3	a, b, c, h
4	a, d, e, f
5	a, b, c, d, e, f
7	b, e, g
7	a, b, c, d, e, f
8	c, d, f, h

The support of the itemset X is the number of transactions in D that contain X. Likewise, the relative support of X is the fraction (or percentage) of the transactions in database which contain itemset X, denoted as Supp(X)

$$\text{Supp}(X) = \frac{|T \in D : X \subset T|}{|D|}. \tag{1}$$

An itemset X is called frequent if Supp(X) is at least equal to a minimum relative support threshold (denoted as MST) specified by user.

If X is frequent itemset (Supp(X) $\geq MST$) and no superset of X is frequent (i.e., it does not exist a frequent $|X'| > |X|$), X is a maximal frequent itemset. Let MFI denote all the maximal frequent itemsets.

Let R be the association rules that are extracted from D. Each association rule is defined as an implication of the form $X \to Y$, where X is the antecedent part of the rule and Y is the consequent of the rule, such that $X \in I$, $Y \in I$ and $X \cap Y = \emptyset$. It means the transaction contains both X and Y. The support of the rule $X \to Y$ is the fraction of the number of transactions that include both itemsets $X \cup Y$ and the number of transactions in D, denoted as Supp($X \to Y$)

$$\text{Supp}(X \to Y) = \frac{|T \in D : X \cup Y \subset T|}{|D|}. \tag{2}$$

The confidence of the rule $X \to Y$ is the percentage of the number of transactions that include both itemsets $X \cup Y$ and the number of transactions that include itemset X in D, denoted as Conf($X \to Y$)

$$\text{Conf}(X \to Y) = \frac{|X \cup Y|}{|X|}. \tag{3}$$

For each association rule, a minimum support threshold and a minimum confidence threshold (MCT) are determined by the data owner. The following conditions need to be satisfied for a strong rule $X \to Y$

- $\text{Conf}(X \cup Y) \geq MCT$ and
- $\text{Supp}(X \to Y) \geq MST$.

A rule is hidden if its support is less than MST or its confidence is less than MCT. It means we cannot discover these rules in the sanitized database by data mining techniques.

Table 2. Decreasing confidence or support below thresholds for hiding sensitive rules [7]

Before hiding	**After hiding**	**Outcome**
$\text{Supp}(r) \geq MST$ and $\text{Conf}(r) \geq MCT$ and $r \in Rs$	$\text{Supp}(r) < MST$ or $\text{Conf}(r) < MCT$	r is hidden

The rule hiding problem can be formulated as follows.

Let D be a transaction database and R be the set of strong rules that can be mined from D with given MST and MCT. Let R_Ss denote a set of sensitive rules that need to be hidden, $R_S \subset R$, and R_N be the set of non-sensitive rules, we have $R_N \cup R_S = R$. The hiding problem is that how to transform D into a sanitized database D' such that only the rules which belong to R_N can be mined from D'. Let R' denote the strong rules mined from the sanitized database D with the same MST and MCT.

The non-sensitive rules or pre-strong rules in D may be affected by the hiding process. A rule that is considered as pre-strong rule if its support is greater than or equal to MST and its confidence is less than MCT. A pre-strong rule becomes strong if its confidence is greater than MCT. For a non-sensitive rule in D, it is not strong if its support is less than MST or its confidence is less than MCT due to removing the item.

- The number of sensitive rules in D' that are not hidden (Hiding failure)

$$S - N - H = \{r \in Rs | r \in R'\}.$$

- The number of non-sensitive rules found in the original database D and not in the sanitized database D' (Lost rules)

$$N - S - L = \{r \in RN | r \notin R'\}.$$

- The number of ghost rules generated in the sanitized database D' (Ghost rules)

$$F - S - G = \{r \notin R | r \in R'\}.$$

3. THE PROPOSED METHOD

3.1. The preprocess

In the hiding process, different orders of sensitive rules may lead to different results. In the proposed method, the sensitive rules are sorted in increasing order of support. In the case of two frequent itemsets, we consider the longer sensitive itemset firstly. It means we start the hiding process from the sensitive rule with the minimum support and continue with the next itemset in that order until they are done with the hiding of every sensitive itemset.

In the hiding process, instead of considering the large number of non-sensitive frequent itemsets in each transaction that fully support the sensitive rule, the proposed method

Table 3. Side effects caused in the hiding process

Before hiding	**After hiding**	**Outcome**
Supp(r) $\geq$ MST and Conf(r) $\geq$ MCT and $r \in R_S$	Supp(r) $\geq$ MST or Conf(r) $\geq$ MCT	r is sensitive but not hidden
Supp(r) $\geq$ MST and Conf(r) $\geq$ MCT and $r \in R_N$	Supp(r) $<$ MST or Conf(r) $<$ MCT	r is non-sensitive and falsely hidden
Supp(r) $<$ MST and Conf(r) $<$ MCT and $r \notin R$	Supp(r) $\geq$ MST or Conf(r) $\geq$ MCT	r is a new generated spurious

focuses on non-sensitive maximal frequent itemsets in each transaction. In the preprocess, the algorithm identifies the set of non-sensitive maximal frequent itemsets based on all frequent itemsets and the generating itemset of sensitive rules before the hiding process. We then use this set to determine and calculate weight values for transactions that support sensitive rules. Thus the number of frequent itemsets that need to be considered will be significantly reduced. In addition, the support of the maximal frequent itemset is relatively low and sensitive to the sanitization. So, focusing on the most sensitive part of the frequent itemset can effectively avoid the significant change during the hiding process.

3.2. The hiding process

The basic strategy for rule hiding is that we remove some items from the database such that the supports or the confidences for all sensitive rules are below the user-defined threshold MST or MCT accordingly. We solve the following two phases before we remove the items:

- Identifying critical transactions that fully support for sensitive rules to be modified.
- Identifying some sensitive items to be removed from the modified transactions.

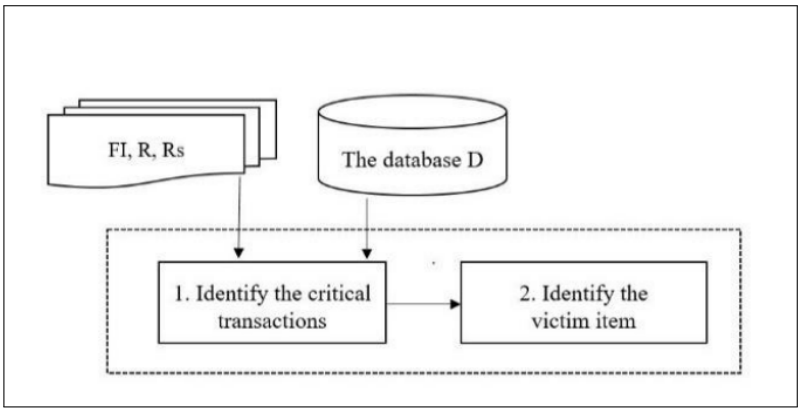

Figure 1. Two phases of association rule hiding

In the first phase, we realized that sensitve items exist only in some transactions of the database, and manipulating all transactions is a time-consuming and useless task. Therefore, it is necessary to have an effective strategy in identifying transactions that can conceal all sensitive rules and reduce data modifications and limit the side effects.

A critical transaction that needs to be modified is the transaction that fully supports one or more sensitive rules. It is not adequate to randomly select and filter transactions that fully support any sensitive rule for modification. We define some measures for evaluating the relevance of different supporting transactions to find the transactions that give least side effects. Based on Relevance_Sorting [4] algorithm, the paper proposes a new way to calculate "the relevance", evaluate and choose critical transactions that need to be modified. The relevance value in the proposed method is not calculated according to all non-sensitive frequent itemsets like in [4], it is calculated by the number of non-sensitive maximal frequent itemsets that contain at least one item to the consequent of the rule. Because the maximal frequent itemsets can present and deduce non-sensitive frequent itemsets that are greatly reducing the number of elements to be considered. The support of element in the maximal itemsets is relatively low, close to the minimum threshold given by the user, sensitive to sanitization. Focusing on the most sensitive part of non-sensitive frequent itemsets can effectively avoid the significant change. Besides, these itemsets contain at least one item to the consequent of the rule that can be chosen as a victim item, so these itemsets can be directly affected when removing the victim item, thereby evaluating the relative significance of the transactions that need to be modified. The negative side of this approach is that non-sensitive frequent itemsets do not precalculate for each rule, because it depends on a given sensitive rule that need to be hidden, so it takes longer execution time than the original algorithm.

We sort these transactions by their relevance values. In case of there are two transactions that have the same relevance value, they are sorted in increasing order of the transaction length. We modify transactions that have less non-sensitive maximal frequent itemsets, that hold the highest relevance values. Assume $NUM_{non\_sen(t)}$ is the number of non-sensitive maximal itemsets supported by transaction t, the relevance of t is calculated as

$$\text{Relevance}(t) = 1/[1 + NUM_{non\_sen(t)}]. \tag{4}$$

In order to hide the association rule, we can reduce the support, or the confidence of the sensitive rule that drops below the user-specified threshold: MST or MCT. The authors in [4] combined these two strategies together to calculate the minimum number of transactions, and to reduce more non-sensitive rules that are falsely lost and the fewer ghost rules generated when they are independently implemented. In [4] the following properties were given.

Property 1. Let $X \cup Y$ be the set of all transactions that support $X \to Y$. In order to decrease the confidence of the rule below MCT, the minimal number of transactions which need to be modified in $X \cup Y$ is

$$NUM_1 = \lceil (\text{Supp}(X \cup Y) - \text{Supp}(X) \times MCT \times |D| \rceil + 1. \tag{5}$$

Property 2. Let $X \cup Y$ be the set of all transactions that support $X \to Y$. In order to decrease the support of the generating itemset of $X \to Y$ below MST, the minimal number

of transactions which need to be modified in $X \cup Y$ is

$$NUM_2 = \lceil (\mathrm{Supp}(X \cup Y) - MCT \times |D| \rceil + 1. \tag{6}$$

Based on Property 1 and Property 2, we can infer the minimum number of transactions to be modified to hide the sensitive rule is

$$\begin{aligned} \min\{NUM_1, NUM_2\} = \min\{ &\lceil (\mathrm{Supp}(X \cup Y) - \mathrm{Supp}(X) \times MCT \times |D| \rceil + 1, \\ &\lceil (\mathrm{Supp}(X \cup Y) - MCT \times |D| \rceil + 1\}. \end{aligned} \tag{7}$$

During the hiding process, the iterations to hide sensitive rules can be updated when some items from transactions are removed to hide one rule that shares the common items with other sensitive rules. Doing like that helps reduce the iterations that must be performed to hide sensitive rules, which improves the performance of the algorithm.

In the second phase, the algorithm identifies an appropriate item to remove from the identified transaction so that the side effects are minimized. In order to hide the rule $X \to Y$, if the confidence of the rule is reduced by modifying X on the transaction that contains both X and Y, then both the numerator and the denominator decrease, thus making the convergence rate of the algorithm low. Therefore, the proposed algorithm reduces the confidence or the support of the rule by modifying Y on the transaction that contains both X and Y, then the frequency of the numerator is decreased but the denominator remains unchanged. The convergence speed of the algorithm will be faster. Thus, we need to select the appropriate item A in Y and delete A to hide $X \to Y$ so that the side effects are minimized.

In some algorithms, for example, MinFIA [15] that selects an item with the lowest support, because these items generate less non-sensitive patterns than other items so modifying this item causes the least impact on the non-sensitive patterns. The algorithm in [4] chooses the item that has the highest support to delete from the sensitive transactions because non-sensitive patterns containing the item with the highest frequency have high support, so these patterns are minimally affected by sanitization process. Unlike these algorithms, the proposed method chooses the item to the consequent of the rule to remove such that:

If the rule has only one item to the right side so the algorithm will remove this item from the identified transaction. If there are more than one item to the right side of the rule, the algorithm has to identify the appropriate item that needs to be removed. For each item to the consequent of a sensitive rule, from the list of the identified non-sensitive maximal frequent itemsets for each transaction, the algorithm lists the elements that contain the corresponding items and their support greater than MST specified by the data owner. Then, the algorithm selects the itemset from the above lists with the lowest support, denoted as min_Itemset. If there are more than one the lowest itemset with the same support, the proposed method selects the longer itemset, and we have the set of elements with the lowest support. Comparing between these itemsets, we select the itemset with the highest support, denoted as max_min. The corresponding item with max_min element is a victim item and will be removed from the identified transaction. We need to choose like that because when selecting the item in the maximal itemset from the list of the lowest frequent itemsets, we can minimally affect to other non-sensitive itemsets. Since then, the algorithm can control the effects so that the side effects are minimized. In addition, the algorithm extends the selection of appropriate victim item when hiding the rules that are longer than 2-itemsets. If there are more than one max_min element, the algorithm will select the item with the lower support to remove. The proposed algorithm is given below.

Algorithm 1 Sorts of Critical Transactions

Input: D, MST, MCT, FIs, R, Rs
Output: The sanitized database D'
Main Method

Step 1: Initialization
1. Sort sensitive rules $\in Rs$ by their support in increasing order and their length in descending order
2. Identify the maximal frequent itemsets, denoted as M
3. Filter out all transactions supporting at least one sensitive rule, denoted as $\sum$

Step 2: The hiding process
For each rule r_i in R_S
{
1. Filter out transactions which fully support r_i, denoted as $\sum_i$
$\sum_i = \{t \in \sum | t \text{ fully supports } r_i\}$
2. For each transaction $t \in \sum_i$
{
+ Determine the list of non-sensitive maximal itemsets $\in M$, that contain at least one item to the consequent of the rule r_i supported by t
+ Determine the relevance value of the transaction t according to Eq(4)
Relevance $(t) = 1/[1 + NUM_{non\_sen(t)}]$
}
3. Sort the transactions in $\sum_i$ by the relevance values in descending order
4. Use the Eqs (5), (6), (7) to calculate the minimum number of transactions that need modifications to hide the rule r_i, denoted as N_iterations
5. For $i := 1$ to N_iterations do
{
- Choose the transaction in $\sum_i$ with the highest relevance value, $t = \sum_i$ [1]
- Choose the item j corresponding to the consequent of the rule r_i to remove that the support of non-itemset maximal itemset contains it is the greatest
- Remove the item j from the transaction t
- Update the support and confidence of r_i
- Update the support and confidence of other affected rules which are originally supported by t and contain the item j
- $\sum_i = \sum_i t$
}
}

3.3. A demonstrative example

The database table in Table 1 is used in this example. The frequent itemsets corresponding to the database and strong association rules are listed in Table 4 and Table 5, respectively with $MST = 40\%$ and $MCT = 70\%$.

We assume the following rules {a→ bc} and {e → f} are regarded as sensitive rules that will be hidden using the proposed method as follows:

Step 1: The preprocess

Table 4. Frequent itemsets FIs

TID
abc: 4, def: 3
ab: 4, ac: 4, bc: 4, be:4, bg:3, ch:3, de:3, df:4, ef:3
a: 5, b: 6, c: 5, d: 4, e: 4, f: 4, g:3, h:3

Table 5. Association rules R

Rule	**Supp**	**Conf**	**Rule**	**Supp**	**Conf**
f→de	3	0.75	e→b	4	0.75
a → c	4	0.8	c→ ab	4	0.8
ab → c	4	1.0	ac → b	4	1.0
bc → a	4	1.0	ac → b	4	0.8
e → f	3	0.75	df → e	3	0.75
e → df	3	0.75	d → f	3	1.0
a → b	4	0.8	d → ef	3	0.75
d → e	3	0.75	c → b	4	0.8
g → b	4	1.0	e → d	3	0.75
ef → d	3	1.0	h → c	3	1.0
f → e	3	0.75	de → f	3	1.0
c → a	4	0.8	f → d	3	1.0

The algorithm firstly sorts sensitive rules by their support in increasing order and their length in descending order. So the order of sensitive ones as {e → f, a →bc}.

Based on all frequent itemset FIs and the generating itemsets of sensitive rules, the algorithm identifies maximal itemsets as

$$M = \{\text{ab, bc, ac, be, bg, ch, def}\}.$$

Before we calculate the relevance value for transactions, all transactions that support at least one sensitive rule will be filtered out to reduce the iterations to access to the database. The obtained set is $\sum = \{1, 2, 3, 4, 5, 7\}$.

Step 2: The hiding process

Regarding the following rule {e→f}: We calculate and sort transactions by the relevance values that show in Table 6. Because transactions have the same relevance value so they are sorted by transaction length in increasing order.

Table 6. The relevance value of supporting transactions for the rule e {→} f

ID	**Relevance**	**Itemset contains r_{RHS}**
2	0,5	def
4	0,5	def
7	0,5	def

According to Eqs (5), (6), and (7), the smallest of the modified transactions that hide e $\{\rightarrow\}$ f is calculated as follows $N = \min\{(38 \times 0.4 + 1), (3 - 4 \times 0.7 + 1)\} = 1$.

Regarding the following rule e $\{\rightarrow\}$ f , we only need to modify one supporting transaction that is the first one in the sorted list having the highest relevance value. Because this rule only has one item to the consequent of the rule so we select the second transaction from the database and modify it by removing the item {f}. The result is Supp(ef) $< MST$ so the rule is hidden.

Regarding the following rule a $\{\rightarrow\}$ bc, we also calculate and sort transactions by the relevance values that show in Table 7.

Table 7. The relevance value of supporting transactions for the rule a $\{\rightarrow\}$ bc

ID	Relevance	Itemset contains r_{RHS}
1	0,2	ab, ac, bc, bg
3	0,2	ab, ac, bc, ch
7	0,2	ab, ac, bc, be
5	0,167	ab, ac, bc, bg, ch

According to Eqs (5), (6), and (7), the minimum number of the modified transactions to hide $\{a \rightarrow bc\}$ that are calculated as $N = \min(48 \times 0.4 + 1), (4 - 5 \times 0.7 + 1) = 2$.

For the rule a $\{\rightarrow\}$ bc, we modify there are two supporting transactions that have the highest relevance values, the 1st and 3rd transactions. Because this rule has more than one item to the consequent so the victim item is being selected as follows:

- At the 1st iteration of the 1st transaction: Item {b}: {ab, bc, bg} $\rightarrow$ min_Itemset: {bg}; Item {c}:{ac, bc} $\rightarrow$ min_Itemset: {ac}. Because min_Itemset {bg} contains the item {c} that has the higher support so we will remove {c} from the 1st transaction. Update the support and the confidence of the affected itemsets.
- At the 2rd iteration of the 3rd transaction: Item {b}: {ab, bc} $\rightarrow$ min_Itemset: {bc}; Item{c}: {ac, bc, ch}$\rightarrow$ min_Itemset: {ac}. The item {c} is being selected because the max_min elements of two items have the same support so the item has the higher support will be removed from the 3rd transaction. The result as Supp(abc) $< MST$ so the rule is hidden.

The sanitized database D' and association rules R' after hiding are shown in Table 8 and Table 9.

4. PERFORMANCE EVALUATION

4.1. Measurements

Hidden rate: Hidden rate is to measure the quality of sensitive association rule hiding. It measures the percentage of hidden association rules among all of the sensitive rules

$$HF = \frac{|R_s(D')|}{|R_s(D)|}. \tag{8}$$

Table 8. The sanitized database D'

TID	Items
1	a, b, c, g
2	b, d, e, f
3	a, b, c, h
4	a, d, e, f
5	a, b, c, d, e, f
6	b, e, g
7	a, b, c, d, e, f
8	c, d, f, h

Table 9. The association rules R'

Rule	Supp	Conf
c→a	3	0.75
d→e	3	0.75
e→b	3	0.75
d→f	3	0.75
g→b	3	1.0
e→d	3	0.75
f→d	3	1.0
c→h	3	0.75
h→c	3	1.0

The lower hidden-rate is the better. The best situation of hidden rate is 100%, that means that all the sensitive rules can be hidden at the same (or higher) thresholds.
Lost rate: Lost rate is to measure the side effect of hiding. It measures the percentage of lost association rules among all non-sensitive rules

$$LR = \frac{|R_N(D)| - |R_N(D')|}{|R_N(D)|}. \tag{9}$$

The lower lost rate is the better. There should be no lost rules in the sanitized database which are arrived support and confidence in the original database at the same (or higher) thresholds.
False rate is to measure the side effect, either. It measures the percentage of false association rules among all of rules whose confidence is below the pre-defined minimum confidence threshold

$$GR = \frac{|R'| - |R \cap R'|}{|R'|}. \tag{10}$$

The lower false rate is the better. No false rules should be produced when mining the sanitized database at the same (or higher) thresholds.

Distortion degree is the number of items that are modified or removed during the hiding process.

4.2. Experimental results

The experiments were carried out on the same platform as Java, and implemented on the same PC with Intel (R) Core i7 CPU 2.5 GHz and 4GB RAM, Windows 10 (64-bit). To measure the performance, the proposed method in the paper will be compared with the original Relevance_Sorting algorithm [4]. Many real datasets are considered to evaluate the effectiveness of data sanitization. We evaluated the proposed algorithm using four well-known real datasets that are Mushroom, Chess, Bms-1 and Bms-2. The Mushroom dataset prepared by Bayardo [7] is publicly available through the FIMI repository at http:// fimi.cs.helsinki.fi/. The Chess dataset was generated and described by Shapiro Alen and published publicly through UCI repository at https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King-Pawn%29. The Bms-1 and Bms-2 datasets used in the KDD Cup of 2000 [8] contained click stream data from the legwear and legcare retailer website. These four datasets show varying characteristics with respect to the number of available items and the number of transactions as well as with respect to the average transaction length that are summarized in Table 10. For each dataset, we randomly choose from the association rules 5 sensitive rules to hide and do some iterations to evaluate the performance of the proposed algorithm. We appropriately set *MST* for each dataset to ensure we could generate a large amount of frequent itemsets. The setting is relevant to the density of a dataset. Besides, the paper also experimented with different numbers of sensitive rules to evaluate, compared with Relevance_Sorting algorithm and evaluate the effectiveness of the proposed algorithm.

Table 10. Datasets

Datasets	**Count of transactions**	**Count of items**	**Avg. trans. length**	***MST***
Mushroom	8,124	119	23	0.05
Bms-1	59,602	497	2.5	0.001
Bms-2	77,512	3,340	5.0	0.002
Chess	3,196	75	40.2	0.6

We randomly choose 5 sensitive rules from the association rules to hide and do some iterations to evaluate the performance of the proposed algorithm. Table 11 presents the average results after some iterations.

Compare side effects values

According to the experimental results, it is noticed that both algorithms could completely hide sensitive rules $HF = 0$. Compared with the original algorithm, the proposed algorithm can achieve better result on *LR* and *GR* values in most cases on the three datasets Mushroom, Chess and Bms-1. However, in Bms-2 dataset, the number of generated ghost rules is increased in a very small percentage, but not significantly compared to the Relevance_Sorting algorithm.

Compare LR values with increasing numbers of sensitive rules

Table 11. Comparative results

Datasets	$\|R_s\|$	**Relevance_Sorting**			**Sorts of Critical Transactions**		
		(S-N-H, N-S-L, F-S-G)	The number of modified Items	Time (seconds)	(S-N-H, N-S-L, F-S-G)	The number of modified Items	Time (seconds)
mushroom	5	(0, 21.8,2.73)	1862	2145.06	(0,18.73,2)	1701	2388.73
	10	(0,49.4,5.3)	5016	9100.2	(0,39.6,4)	4609	16017
	20	(0,89.7,6.3)	7921	22230.5	(0,71.7,4.8)	6973	26760.9
Bms-1	5	(0,2.13,0.73)	125	483	(0,2.07,0.67)	125	607.2
	10	(0,4.8,1.5)	247	627.7	(0,4.3,1.5)	246	1356.7
	20	(0,9.1,2.4)	741	1344.5	(0,7.6,2.4)	677	2567
Bms-2	5	(0,4.73,0.93)	416	654.6	(0,4.6,1.07)	415	693
	10	(0,10.5,2.3)	889	891.7	(0,9.2,2.4)	884	1209.8
	20	(0,20.8,3.5)	1733	1449.7	(0,16.4,4.3)	1645	1923
Chess	5	(0,108.07,0)	2777	1040.4	(0,94.47,0)	2515	1523.47
	10	(0,211,0)	5333	1824.2	(0,173.6,0)	4516	3075.3
	20	(0,342.8,0)	9730	2575.1	(0,265.2,0)	7128	5478.2

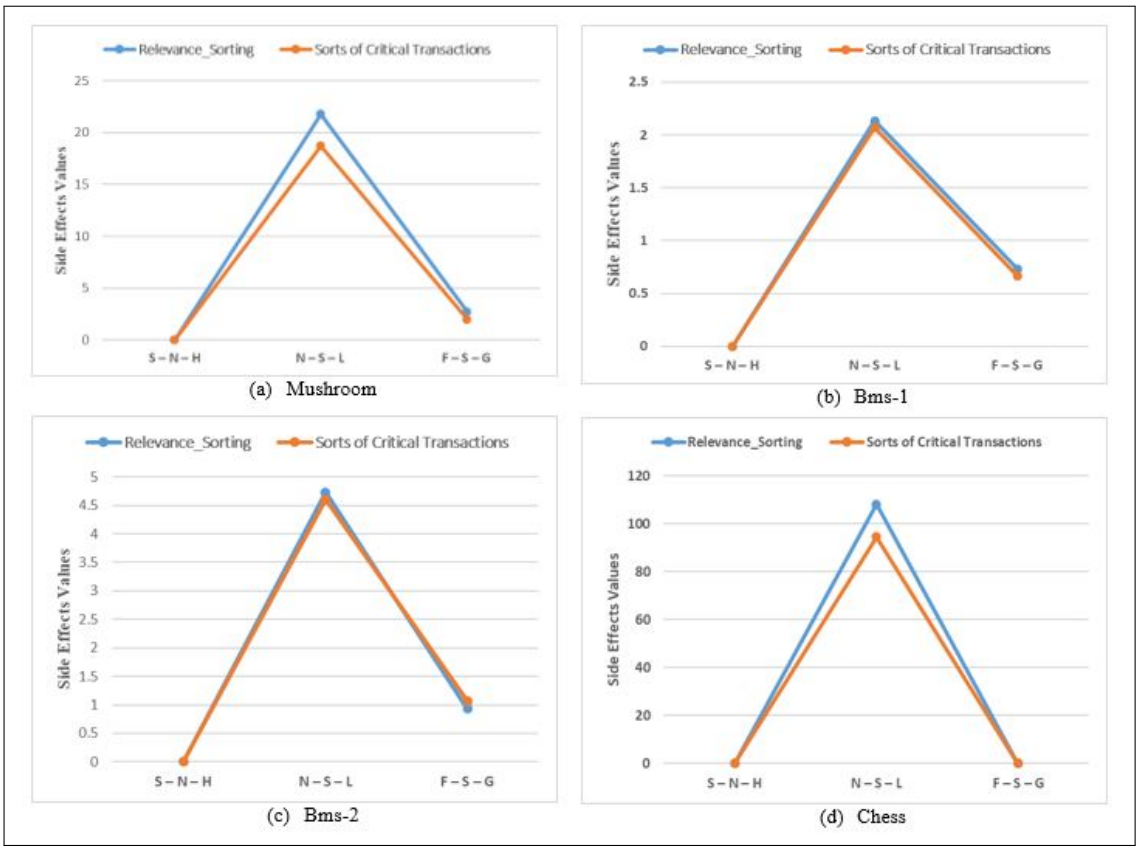

Figure 2. Side effects

We carried out extended experiments to assess the influence of set of different sensitive rules and the influence of increasing size of set of sensitive rules. We randomly selected 5, 10, 20 rules on each dataset as sensitive rules to be concealed that means the more transactions to be sanitized, the more non-sensitive rules may be affected.

We found more side effects are produced on each dataset with the increasing number of sensitive rules. We just compare on *LR* value, and notice that the proposed algorithm achieves better results than the original algorithm. On most datasets when hiding with 20 rules, the proposed method can effectively reduce the missing non-sensitive rules. Since each supporting transaction's weight is evaluated by counting the number of non-sensitive maximal itemsets that can be directly affected, and updates the iterations when hiding the rule shares some items with other sensitive rules, we need less the iterations to hide the rule, help improve the performance and achieve better results.

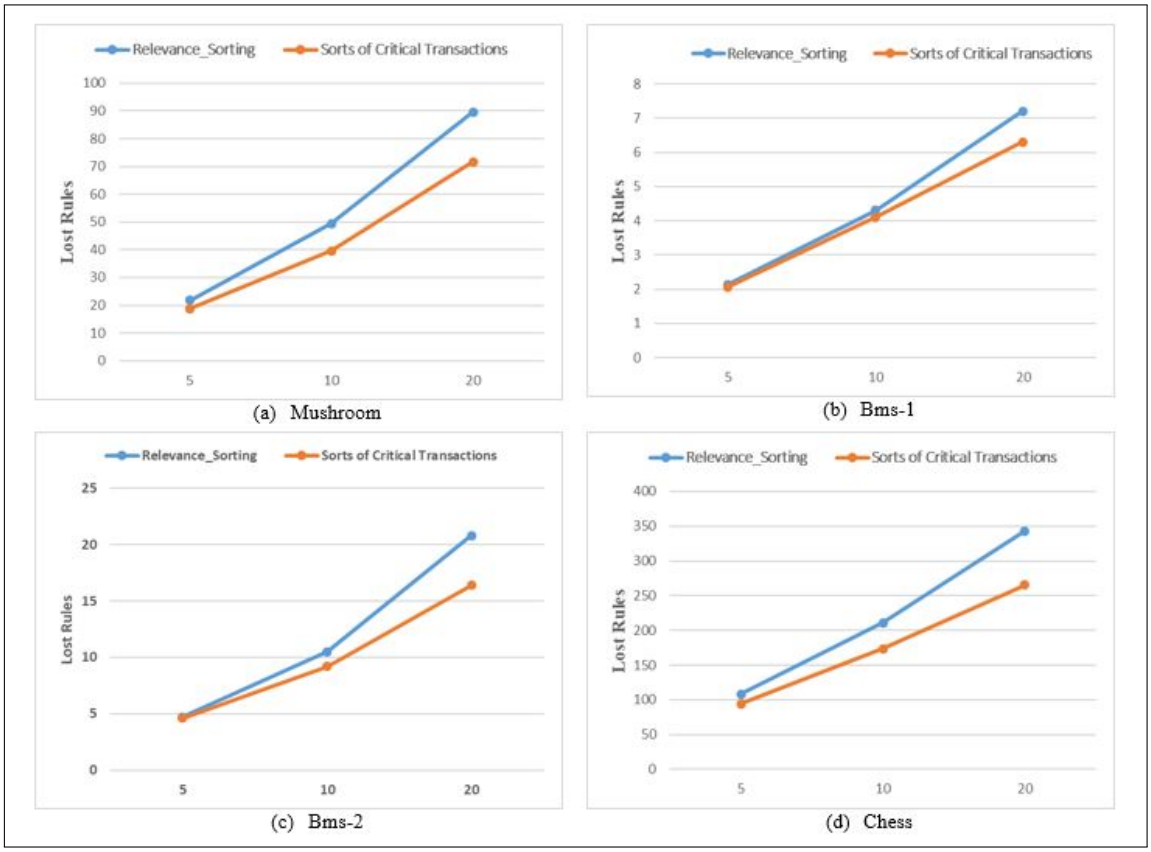

Figure 3. *LR* values

Distortion degree

According to Fig 4, the number of modified items that need to hide 5 sensitive rules on all four datasets, we notice that the proposed method modifies fewer items than the original algorithm does. On Mushroom and Chess datasets, the number of modified items are more than that on Bms-1 and Bms-2 datasets.

Running time

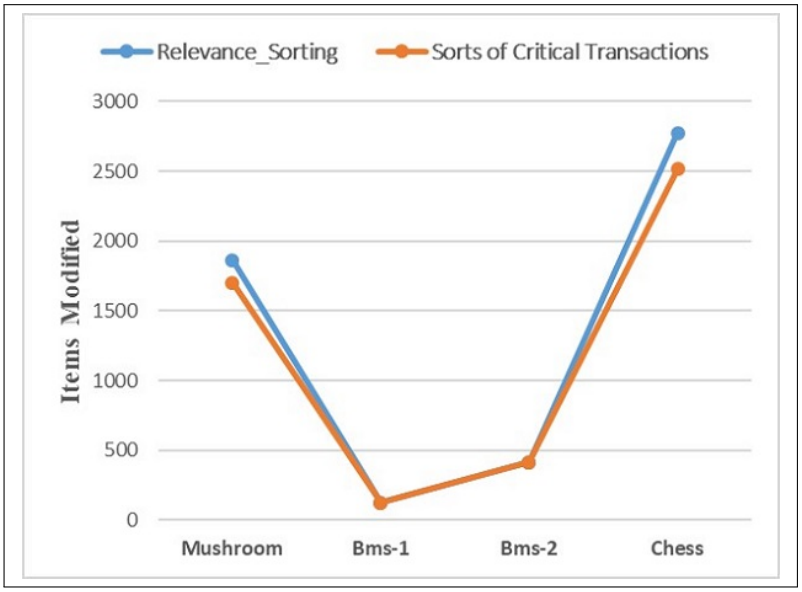

Figure 4. Distortion degree values on 4 datasets

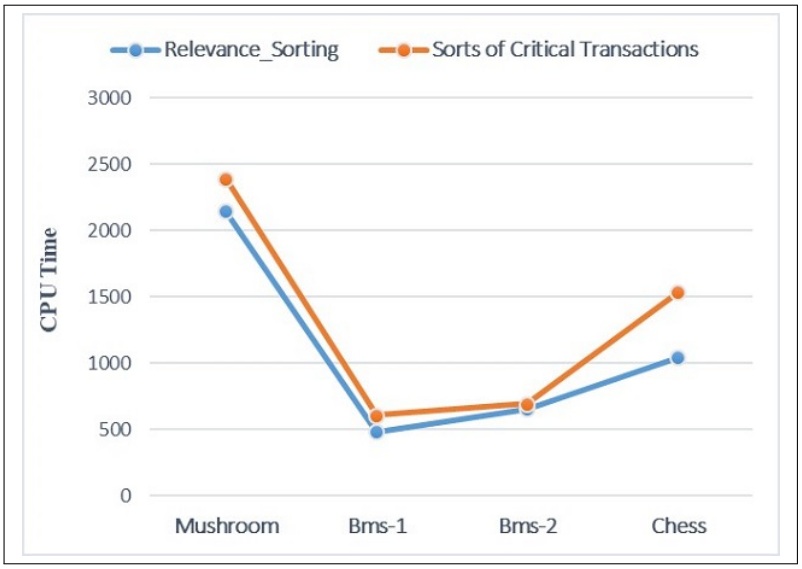

Figure 5. Running time on 4 datasets

The execution time of the proposed method is not less than that of the original algorithm [4] on all datasets. The original algorithm determined the list of non-sensitive itemsets supported by each transaction and calculated the relevance value of the transaction before considering which sensitive rule to hide, and did not repeat these steps to hide different sensitive rules. While the proposed algorithm depends on the sensitive rule that need to be hidden, it just lists the number of non-sensitive maximal itemsets and calculates the

relevance value when it considers that rule. This method repeats these operations for each sensitve rule, so it consumes more time than Relevance_Sorting algorithm.

The experimental results may be affected by the density of a dataset. From the above figures, we notice that the proposed method achieves better result on denser datasets than on sparser datasets. On Mushroom and Chess datasets, the number of missing non-sensitive rules is higher than that on Bms-1 and Bms-2 datasets. For denser datasets, a transaction may contain more association rules or more itemsets, so non-sensitive rules may be affected more than sparser datasets. Besides, on denser datasets, the number of itemsets share common items more than sparser datasets, the proposed algorithm update iterations to hide rules that help to reduce the missing non-sensitive rules.

5. CONCLUSIONS

Privacy preserving in associaton rule hiding is known as an important topic in the database security research area. This paper has proposed a new effective method that solves the association rule hiding problem for data sharing and reduce side effects on the sanitized database. The algorithm has proposed a new way to evaluate the critical transactions, calculate the minimum number of transactions that need to hide sensitive rules and determine the appropriate item to remove from the modified transactions. Experimental results on real datasets have demonstrated the effectiveness of the proposed method in hiding sensitive rules. For further investigation, we will study other approaches, especially evolutionary approaches, that can hide multiple rules at the same time, improve the performance of the proposed algorithm to reduce the number of missing non-sensitive rules and improve the execution time to achieve optimal solutions.

ACKNOWLEDGEMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2018.307

REFERENCES

[1] M. H. Afshari, M. N. Dehkordi, and Akbari, "Association rule hiding using cuckoo optimization algorithm," *Expert Systems with Applications*, vol. 64, pp. 340–351, 2016.

[2] A. Amiri, "Dare to share: Protecting sensitive knowledge with data sanitization," *Decision Support Systems*, vol. 43, no. 1, pp. 181–191, 2007.

[3] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and Verykios, "Disclosure limitation of sensitive rules," *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99) (Cat. No.PR00453)*, pp. 45–52, 1999. [Online]. Available: DOI:10.1109/KDEX.1999.836532

[4] 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, pp. 295–306, 2015.

[5] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," *Information Hiding 4th International Workshop, IH 2001 Pittsburgh*, pp. 369–383, 2001.

[6] A. Divanis, V. Verykios, and Volos, "An integer programming approach for frequent itemset hiding," *Proceeding CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 748–757, 2006.

[7] R. Javier and J. Bayardo, "Efficiently mining long patterns from databases," *Proceeding SIGMOD '98 Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp. 85–93, 1998. [Online]. Available: doi⟩10.1145/276304.276313

[8] C. E. Kohavi, R. Brodley, and et al., "Kdd-cup 2000 organizers' report: Peeling the onion," *SIGKDD Explorations*, vol. 2, no. 2, pp. 86–93, 2000. [Online]. Available: http://www.acm.org/sigkdd/explorations/

[9] C. W. Lin, T. Hong, and S. W. K.T. Yang, and, "The GA-based algorithms for optimizing hiding sensitive itemsets through trans-action deletion," *Appl. Intell.*, vol. 42, no. 2, pp. 210–230, 2015.

[10] C. Lin, T. P. Hong, J. W. Wong, G. Lan, and W. Lin, "A GA-based approach to hide sensitive high utility itemsets," *The Scientific World Journal*, 2014.

[11] C. Lin, B. Zhang, K. Yang, and T. Hong, "Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms," *The Scientific World Journal*, 2014. [Online]. Available: http://dx.doi.org/10.1155/2014/398269

[12] J. C.-W. Lin, T.-Y. Wu, P. Fournier-Viger, G. Lin, J. Zhan, and M. Voznak, "Fast algorithms for hiding sensitive highutility itemsets in privacy-preserving utility mining," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 269–284, 2016.

[13] S. Menon, S. Sarkar, and S. Mukherjee, "Maximizing accuracy of shared databases when concealing sensitive patterns," *Information Systems Research*, vol. 16, no. 3, pp. 235–330, 2005.

[14] G. Moustakides and V. Verykios, "A max-min approach for hiding frequent itemsets," *Data and Knowledge Engineering*, pp. 75–89, 2008.

[15] S. Oliveira and O. Zaiane, "Privacy preserving frequent itemset mining," *Proceedings of the IEEE international conference on privacy, security and data mining*, pp. 43–54, 2002.

[16] X. Sun and P. Yu, "A borderbased approach for hiding sensitive frequent itemsets," *Proceedings of the 5th IEEE international conference on datamining*, pp. 426–433, 2005.

[17] B. Taleb and M. N. Dehkordi, "Sensitive association rules hiding using electromagnetic field optimization algorithm," *Expert Systems With Applications*, vol. 114, pp. 155–172, 2018.

[18] A. Telikani and A. Shahbahrami, "Data sanitization in association rule mining: An analytical review," *Expert Systems with Applications*, vol. 96, pp. 406–426, 2018.

Received on August 07, 2019
Revised on Octorber 19, 2019