Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 210405, 11 pages
http://dx.doi.org/10.1155/2013/210405

Research Article

Association Rule Hiding Based on Intersection Lattice

Hai Quoc Le, Somjit Arch-int, and Ngamnij Arch-int

Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Correspondence should be addressed to Hai Quoc Le; hai_lq@qtttc.edu.vn

Received 12 April 2013; Revised 27 June 2013; Accepted 27 June 2013

Academic Editor: Yang Xu

Association rule hiding has been playing a vital role in sensitive knowledge preservation when sharing data between enterprises. The aim of association rule hiding is to remove sensitive association rules from the released database such that side effects are reduced as low as possible. This research proposes an efficient algorithm for hiding a specified set of sensitive association rules based on intersection lattice of frequent itemsets. In this research, we begin by analyzing the theory of the intersection lattice of frequent itemsets and the applicability of this theory into association rule hiding problem. We then formulate two heuristics in order to (a) specify the victim items based on the characteristics of the intersection lattice of frequent itemsets and (b) identify transactions for data sanitization based on the weight of transactions. Next, we propose a new algorithm for hiding a specific set of sensitive association rules with minimum side effects and low complexity. Finally, experiments were carried out to clarify the efficiency of the proposed approach. Our results showed that the proposed algorithm, AARHIL, achieved minimum side effects and CPU-Time when compared to current similar state of the art approaches in the context of hiding a specified set of sensitive association rules.

1. Introduction

Data mining has been recently applied in many areas of science and business, such as traffic accident detection [1], engineering asset health and reliability prediction [2], assessment of landslide susceptibility [3], enterprises [4], and supply chain management [5]. The discovery of association rules is one of the major techniques of data mining that extracts correlative patterns from large databases. Such rules create assets that organizations can use to expand their businesses, improve profitability, decrease supply chain costs, increase the efficiencies of collaborative product developments, and support more effective marketing [4, 5]. The competitive environment of global economy forces companies, who engage in the same business, to form an alliance for mutual benefits. In the collaboration, companies have to share information in order to shorten processing time dramatically, eliminate value-depleting activities, and improve quality, accuracy, and asset productivity [6]. However, due to legal constraints and/or competition among companies, they do not want to reveal their sensitive knowledge to other parties. *Association rule hiding* is an efficient solution that removes the sensitive association rules from the released database. Thus, the sensitive knowledge can be protected when sharing data between parties.

Many studies in the literature have focused on hiding sensitive association rules by reducing their support or confidence below given thresholds. Association rule hiding algorithms can be divided into three main approach classes [7], namely, border based [8, 9], exact [10, 11], and heuristic [12–22]. The border based and exact approaches aim to protect the revised positive border of frequent itemsets in order to minimize side effects. Although these approaches achieve good results for itemsets hiding, they are not conformable for minimizing the side effects when hiding a specific set of sensitive association rules. The heuristic approach does not guarantee a global optimal solution, but it usually finds a solution close to the best one in a faster response time. In 2012, Hai and Somjit [23] introduced a new direction for hiding a specific set of sensitive association rules named intersection lattice based. This approach concentrated on formulating heuristics for specifying victim items and transactions for data sanitization based on intersection lattice theories.

This study proposes an improvement of the new direction of association rule hiding named intersection lattice-based approach [23–25]. We first introduce in detail theory of intersection lattice of frequent itemsets and prove that it is applicable to the association rule hiding problem. Subsequently, we formulate two heuristics for hiding sensitive association rules with the lowest side effects. The first heuristic determines the victim item that needs to be modified and focuses on maintaining itemsets in the generating set in order to restrict lost rules. The second heuristic assigns a weight to each transaction relying on its degree of safety, the number of sensitive rules, and the number of nonsensitive association rules contained in that transaction. This study provides evidence that removing the victim item from the transactions which have the highest weight minimally produces effects on the nonsensitive association rules and the intersection lattice of frequent itemsets. An experiment is performed on a real dataset to show the performance of the proposed algorithm in real application terms, as well as comparisons with the previous studies.

The rest of this paper is organized as follows. Section 2 presents a brief review of previous works. The problem formulation is provided in Section 3. Section 4 introduces the basic concepts of lattice theory that are applied in this research. The proposed methodology is presented in Section 5. In Section 6, we present the experimental results in order to show the performance of the proposed approach compared with the state of the art approaches. The main contents presented in this study are concluded in Section 7.

2. Related Work

Recently, association rule hiding is classified into four classes, including heuristic, border based, exact based and intersection lattice based. The heuristic approach provides efficient and fast algorithms that select the appropriate transactions and items for hiding sensitive association rules using distortion or blocking technique. The distortion technique adds (or removes) selected items of sensitive association rules to (or from) specified transactions or add dummy transactions [21] to decrease support [8–14] or confidence [12, 13, 15–19] of the rules under the given thresholds in order to hide single or multiple rules [20]. Unlike the distortion, the blocking technique hides a rule by replacing the existing value of some items with an unknown value so as to reduce the support or confidence of the rule [12, 20, 22].

The border-based approach for association rule hiding was first introduced by Sun and Yu [8]. This approach specifies the revised positive and negative borders of all frequent itemsets. It then focuses on the weight of the positive border [8] or the maxmin set [9] to reduce support of the revised negative border while protecting support of the expected positive border so as to maintain the nonsensitive itemsets.

The exact approach transforms the association rule hiding into optimal problem based on the Constraints Satisfactions Problem (CSP). Menon et al. [10] formulated the CSP to specify a minimum number of transactions needed to be modified in order to hide sensitive association rules. Gkoulalas-Divanis and Verykios [11] formulated the CSP based on the revised positive and negative borders to identify candidate items for the hiding process. In this approach, the authors used a process of constraint reduction to formulate CSP in order to make all constraints in CSP to be linear and all variables in CSP to be binary. This allows the use of binary integer programming instead of integer or linear programming for CSP solutions.

The intersection lattice approach for hiding a specific set of association rules was first introduced by Hai and Somjit [23]. The proposed algorithms, ILARH [23] and HSCRIL [24], aim to hide a specific set of sensitive rules in three steps. The first step specifies a set of itemsets satisfying three conditions that (i) contain right-hand side of the sensitive rule, (ii) are maximal sub-itemset of a maximal itemset, and (iii) have minimal support among those subitemsets specified in (ii). An item in the right-hand side of the sensitive rule that is related to the specified maximal support itemset is identified as the victim item. In the second step, a set of transactions supporting sensitive rule is specified. The third step removes the victim items from specified transactions until confidence of the rule is below minimum confidence threshold. In order to reduce side effects, HCSRIL sorts the set of transactions supporting the sensitive rules in ascending order of their size before sanitizing them. Moreover, HCSRIL technically updates the released database such that the sanitization causes least impacts on the generating set. However, the lager transaction may contain fewer nonsensitive association rules. Thus, sorting transactions based on their size is not enough to restrict the lost rules.

Hai et al. [25] assigned a weight to each transaction in order to measure the impacts of hiding process on the nonsensitive association rules. Moreover, the authors formulated the victim item specification based on the measurement of the distance from sensitive rules to the set of maximal itemsets and the nearest nonsensitive association rule. Modifying the victim item on the high-weight transaction can reduce side effects. On the negative side, the constraints between frequent itemsets are not identified in the distances. Thus, modifying the victim item may avoid impacts on some nonsensitive association rules, but it cannot protect the intersection lattice of frequent itemset from being broken. So it may cause more lost rules.

This research takes full advantages of algorithms proposed in [23–25] and proposes an improvement for hiding a specific set of sensitive association rules with the lowest side effects and CPU-Time.

3. Problem Formulation

Let $\mathbf{I} = \{i_1, i_2, \ldots, i_m\}$ be a finite set of m literals. Each member of $\mathbf{I}$ is called an *item*. X is an *itemset* if $X \subseteq \mathbf{I}$. A transaction t is defined by a set of items, namely, $t = \{i_k \mid i_k \in \mathbf{I},\ k \leq m\}$. Let $\mathscr{D}$ be a finite transaction database, namely, $\mathscr{D} = \{t_1, t_2, \ldots, t_n \mid n \in N\}$. An itemset $X \subset \mathbf{I}$ is supported by a transaction $t \in \mathscr{D}$ if $X \subseteq t$. The frequency of an itemset X in database is *support* of X, denoted by $\alpha(X)$, and is defined as

$$\alpha(X) = |X(t)|, \quad \text{where } X(t) = \{t \in \mathscr{D} \mid t \text{ contains } X\}. \tag{1}$$

Table 1: Transaction database.

TID	Itemset
T1	*ABCD*
T2	*ABC*
T3	*ABD*
T4	*BD*
T5	*ABCD*
T6	*AC*
T7	*ABC*

Table 2: Frequent itemsets.

Frequent itemset	α
ABC	4
ABD	3
AB	5
AC	5
BC	4
AD	3
BD	4
A	6
B	6
C	5
D	4

An itemset X is called a *frequent itemset* if $\alpha(X) \geq \sigma$, where σ is the minimum support threshold given by users.

An *association rule* is the implication $X \rightarrow Y$, where $X, Y \subset \mathbf{I}$, and $X \cap Y = \emptyset$.

The *support* of a rule $X \rightarrow Y$ is defined to be the support of itemset $X \cup Y$, that is,

$$\alpha(X \longrightarrow Y) = \alpha(X \cup Y). \quad (2)$$

The *confidence* of a rule $X \rightarrow Y$ is defined as

$$\beta(X \longrightarrow Y) = \frac{\alpha(X \cup Y)}{\alpha(X)}. \quad (3)$$

Example 1. Let a transaction database be given as in Table 1. Let minimum thresholds be given as $\sigma = 3$ and $\delta = 70\%$. Frequent itemsets mined from Table 1 are shown in Table 2, and strong association rules generated from the frequent itemsets are presented in Table 3.

Let σ and δ be the minimum support threshold and the minimum confidence threshold given by users. The association rule $X \rightarrow Y$ is the *strong association rule* if $\alpha(X \rightarrow Y) \geq \sigma$ and $\beta(X \rightarrow Y) \geq \delta$.

Lemma 2 (Apriori property [26]). *Assume that $X, Y \subseteq \mathbf{I}$. If $X \subseteq Y$, then $\alpha(X) \geq \alpha(Y)$.*

The Apriori property shows that if an itemset X is frequent, then all itemsets in the family of subsets of X are frequent.

Table 3: Strong association rules.

Rules	β
$AB \rightarrow C$	80%
$C \rightarrow AB$	80%
$AC \rightarrow B$	80%
$BC \rightarrow A$	100%
$AD \rightarrow B$	100%
$BD \rightarrow A$	75%
$D \rightarrow AB$	75%
$A \rightarrow B$	83%
$B \rightarrow A$	83%
$A \rightarrow C$	83%
$C \rightarrow A$	100%
$C \rightarrow B$	80%
$D \rightarrow A$	75%
$D \rightarrow B$	100%

The association rules discovered from a large database that can be used in the decision-making support process are said to be *sensitive association rules* [14].

Definition 3 (sensitive association rules). Let $\mathcal{D}$ be a transactional database, R be a set of all association rules that are mined from $\mathcal{D}$, and Rules_H be a set of decision support rules that need to be hidden according to some security policies. A set of association rules, denoted by S_R, is said to be sensitive if $S_R \subset R$ and S_R would derive the set Rules_H. $\sim S_R$ is the set of nonsensitive association rules such that $\sim S_R \cup S_R = R$.

A sensitive association rule $I_l \rightarrow I_r$ is hidden if $\alpha(I_l \rightarrow I_r) < \sigma$ or $\beta(I_l \rightarrow I_r) < \delta$. The rule can be hidden by

(i) removing an item $I_i \in I_l I_r$ from some transactions in order to make $\alpha(I_l \rightarrow I_r) < \sigma$,

(ii) adding all items $I_i \in I_l$ to some transactions until $\beta(I_l \rightarrow I_r) < \delta$, or

(iii) removing an item $I_i \in I_r$ from some transactions until $\alpha(I_l \rightarrow I_r) < \sigma$ or $\beta(I_l \rightarrow I_r) < \delta$.

The modifications of any item always cause, however, *side effects* which are the impacts of data modification on the quality of association rule mining, including *lost rules*, *ghost rules*, *false rules*, and *accuracy*.

(i) *Lost rule* is a nonsensitive association rule that is discovered from the original database but cannot be mined from the released database.

(ii) *Ghost rule* is a nonsensitive association rule that cannot be discovered from the original database but can be mined from the released database.

(iii) *False rule* is the sensitive association rule that cannot be hidden by hiding process.

(iv) *Accuracy* is the ratio of distorted data items to total of data items in the original database.

The association rule hiding algorithm is better than the other one if it achieves lower side effects, including lower lost rules, ghost rules, false rules, and higher accuracy, and lower complexity.

The problem of association rule hiding addressed in this paper can be stated as follows.

Let a transaction database $\mathscr{D}$, a minimum support threshold σ, and a minimum support threshold δ be given. Let us assume that R is a set of association rules mined from $\mathscr{D}$, whose support and confidence are not less than σ and δ, respectively. Suppose that a set of certain association rules in R regarded as being sensitive, denoted by S_R, can be specified. The problem is how to transform $\mathscr{D}$ into a released database $\mathscr{D}'$ in such a way that all sensitive association rules in S_R are hidden, while nonsensitive association rules can still be mined from $\mathscr{D}'$ and the side effects are minimal.

We apply method (iii) to a heuristic association rule hiding algorithm based on the intersection lattice of frequent itemsets in order to reduce the side effects.

4. Background

In this section, we recall some concepts in lattice theory that are applied in the present study. Lattice theory was developed by George Grätzer [27]. It singles out a special type of order for details of investigation. The basic concepts of lattice theory that are related to our research are presented as follows.

Let V be a nonempty set. A binary relation θ on V is said to be an order relation if θ satisfies the properties *reflexivity*, *antisymmetry*, and *transitivity*, namely,

(1) *reflexivity*: $a\theta a$,

(2) *antisymmetry*: $a\theta b$ and $b\theta a$ imply that $a = b$,

(3) *transitivity*: $a\theta b$ and $b\theta c$ imply that $a\theta c$.

We usually use $\leq$ to denote an order and $(V;\leq)$ to denote an ordered set.

Let $(V;\leq)$ be an ordered set. An element $a \in P$ is an *upper bound* of $H \subseteq V$ if a majorizes all $h \in H$. An upper bound a of H is the *least upper bound* of H or *supremum* of H if a is majorized by all upper bounds of H. In this case, we will write $a = \sup H$.

The dual concepts of upper bound and least upper bound are the *lower bound* and the *greatest lower bound*, respectively, which are defined by duality. The *greatest lower bound* or the *infimum* of H is denoted by $\inf H$.

Definition 4 (lattice). An ordered set $(L;\leq)$ is said to be a lattice if for all $a, b \in L$, $\inf\{a, b\}$ and $\sup\{a, b\}$ always exist and are denoted by $a \vee b$ and $a \wedge b$, respectively.

Definition 5 (semilattice). Let $(A; o)$ be an algebra with one binary operation o. The algebra $(A; o)$ is a semilattice if o is idempotent, commutative, and associative.

An algebra $(L;\wedge,\vee)$ is said to be a lattice if L is a nonempty set, $(L;\wedge)$ and $(L;\vee)$ are semilattices, and the two absorption identities are satisfied. A lattice as algebra and a lattice as an order are proved "equivalent" concepts [27].

Let U be a finite nonempty set. It is obvious that the power set of U, denoted by $\mathrm{Poset}(U)$, is an ordered set under the inclusion relation $\subseteq$. It can be verified that $(\mathrm{Poset}(U);\subseteq)$ forms a lattice, where $\sup\{A, B\} = A\cup B$ and $\inf\{A, B\} = A\cap B$. If $L \subseteq U$ and $(L;\subseteq)$ is a lattice satisfying the properties that $\sup\{A, B\} = A\cup B$ and $\inf\{A, B\} = A\cap B$, for all A and B, then $(L;\subseteq)$ is called a set lattice. Similarly, if the ordered set $(L;\subseteq)$ is a semilattice under intersection operation "$\cap$" satisfying $\inf\{A, B\} = A \cap B$, for all A and B in L, then $(L;\subseteq)$ is said to be an *intersection lattice*.

5. The Proposed Approach for Association Rule Hiding Based on Intersection Lattice

In this section, we specifically introduce the intersection lattice theory applied in association rule hiding that was basically presented in [23–25]. Firstly, we analyze the characteristics of the intersection lattice of frequent itemsets. Then, we improve heuristics for minimizing the side effects of association rule hiding process. Finally, we propose an efficient algorithm for hiding a specific set of sensitive association rules.

5.1. Intersection Lattice of Frequent Itemsets. In this subsection, we formulate intersection lattice theory for the set of frequent itemsets and prove the applicability of this theory into association rule hiding. Let $\mathscr{D}$ be a given transaction database on a finite set of items $\mathbf{I}$ and let σ be a given minimum support threshold. Consider the lattice $(\mathrm{Poset}(\mathbf{I});\subseteq)$ and the set $P(\sigma)$, denoted by a set of frequent itemsets that are mined from $\mathscr{D}$ and satisfy the given threshold σ; we have the following statements.

Theorem 6 (intersection lattice of frequent itemset). *Let $\mathscr{D}$ be a given transaction database on a finite set of items $\mathbf{I}$ and σ be a given minimum support threshold. Then, $(P(\sigma);\subseteq)$ forms an intersection lattice, denoted by $\mathbf{L}(\mathscr{D},\sigma)$.*

Proof. For all $X, Y \in P(\sigma)$, assume that $Z = X \cap Y$; then we have $Z \subseteq X$. By Lemma 2, we have $\alpha(Z) \geq \alpha(X) \geq \sigma$, so $Z \in P(\sigma)$. In other words, we have $\inf\{X, Y\} = X \cap Y$.

On the other hand, the ordered set $(P(\sigma);\subseteq)$ is a semilattice under the intersection operator $\cap$. Indeed, for all $X, Y \in P(\sigma)$, we always have the following.

(i) $\cap$ is idempotent because $X \cap X = X$.

(ii) $\cap$ is commutative. Consider an arbitrary item $x \in I$. Then by the definition of set intersection, we have $x \in X \cap Y$

$\Leftrightarrow x \in X \wedge x \in Y$

$\Leftrightarrow x \in Y \wedge x \in X$ (by the commutativity of meet operation)

$\Leftrightarrow x \in Y \cap X$.

Hence, by universal generalization, every item which is in $X \cap Y$ is also in $Y \cap X$.

Hence, $X \cap Y = Y \cap X$.

(iii) $\cap$ is associative. Similar to (ii), we have $(X \cap Y) \cap Z = X \cap (Y \cap Z)$.

In other words, the ordered set $(P(\sigma); \subseteq)$ is a semilattice under the intersection operation such that for all $X, Y \in P(\sigma)$, $\inf(X, Y) = X \cap Y$. Hence, $(P(\sigma); \subseteq)$ is an intersection lattice. □

Definition 7 (the generating set). The generating set of $\mathbf{L}(\mathscr{D}, \sigma)$, denoted by $\mathbf{G}_L$, is the smallest subset of $\mathbf{L}(\mathscr{D}, \sigma)$ such that each element of $\mathbf{L}(\mathscr{D}, \sigma)$ can be represented as the (finite) intersection of some elements of $\mathbf{G}_L$, namely,

$$\mathbf{L}(\mathscr{D}, \sigma) = \left\{ X \mid X = \bigcap_{k \in N^*} X_k, X_k \in \mathbf{G}_L \right\}. \tag{4}$$

Definition 7 indicates that each element of $\mathbf{L}(\mathscr{D}, \sigma)$ can be generated by an intersection of a finite number of certain elements of $\mathbf{G}_L$.

Lemma 8. *For all $X, Y, Z \in \mathbf{G}_L$, if $X \neq Z$, $Y \neq Z$, and $X \neq Y$, then $X \cap Y \neq Z$.*

Proof. It can easily be seen that the statement "$X, Y, Z \in GL$, $X \neq Z$, $Y \neq Z$, and $X \neq Y$ then $X \cap Y \neq Z$" is an immediate consequence of Definition 7. Since in the opposite case, $Z = X \cap Y$, then $\mathbf{G}_L \setminus \{Z\}$ is obviously also a generating set of $\mathbf{L}(\mathscr{D}, \sigma)$. This means that $Z \notin \mathbf{G}_L$, a contradiction. □

Theorem 9. *For every $\mathbf{L}(\mathscr{D}, \sigma)$, the set $\mathbf{G}_L$ is unique.*

Proof. It is obvious that if $P(\sigma) = \emptyset$, then $\mathbf{G}_L = \emptyset$. Since $P(\sigma) \neq \emptyset$, to hold Theorem 9, we have to prove two affirmations as follows.

(i) $\mathbf{L}(\mathscr{D}, \sigma)$ always contains a $\mathbf{G}_L$. For all $X \in P(\sigma)$, we have for all $X' \in \text{Poset}(X)$, $X' \in P(\sigma)$ (Lemma 2). By Definition 7, for all $X \in \mathbf{L}(\mathscr{D}, \sigma)$, there is a finite number of itemsets $Y_k \in \mathbf{G}_L$ such that

$$X = \bigcap_{k \in N^*} Y_k. \tag{5}$$

If $k = 1$, then $X = Y_k$; thus, we imply that $X \in \mathbf{G}_L$. By Lemma 8, if $k \geq 2$, then $X \notin \mathbf{G}_L$ and X is generated by an intersection of itemsets $Y_k \in \mathbf{G}_L$. Hence, by universal generalization, for any itemset $X \in \mathbf{L}(\mathscr{D}, \sigma)$, there is a set $\mathbf{G}_L$ such that either $\mathbf{G}_L$ contains X or $\mathbf{G}_L$ contains a finite set of itemsets which can generate X by taking an intersection of those itemsets. In other words, $\mathbf{G}_L$ always exists for every intersection lattice $\mathbf{L}(\mathscr{D}, \sigma)$.

(ii) $\mathbf{G}_L$ is unique in $\mathbf{L}(\mathscr{D}, \sigma)$. Assume that $\mathbf{G}'_L$ is the other generating set of $\mathbf{L}(\mathscr{D}, \sigma)$. We show that $\mathbf{G}'_L = \mathbf{G}_L$. First, we prove that $\mathbf{G}_L \subseteq \mathbf{G}'_L$. Indeed, take any $X \in \mathbf{G}_L$, by the definition of $\mathbf{G}'_L$,

$$X = \bigcap_{h \in N^*} Y'_h, \tag{6}$$

for some sets $Y'_h \in \mathbf{G}'_L$, which implies that $X \subseteq Y'_h$. By Lemma 8, if $X \neq Y'_h$, then $Y'_h \notin \mathbf{G}_L$.

On the other hand, we have

$$Y'_h = \bigcap_{j \in N^*} X'_j, \tag{7}$$

by the definition of $\mathbf{G}_L$. Consequently, we obtain the inclusion

$$X \subseteq Y'_h = \bigcap_{j \in N^*} X'_j. \tag{8}$$

By Lemma 8, we infer that the set of indexes N^* is single and, therefore, $X = Y'_h$; therefore, $X \in \mathbf{G}'_L$, which shows that $\mathbf{G}_L \subseteq \mathbf{G}'_L$.

Similarly, we also have $\mathbf{G}'_L \subseteq \mathbf{G}_L$. In other words, $\mathbf{G}'_L = \mathbf{G}_L$. □

Theorem 10. *The set $\mathbf{G}_L$ is calculated as follows:*

$$\begin{aligned} &\mathbf{G}_L = \{X \in \mathbf{L}(\mathscr{D}, \sigma) \mid d(X) \leq 1\}, \\ &\textit{where } d(X) = |\{Y \in \mathbf{L}(\mathscr{D}, \sigma) \mid X \subset Y\}|. \end{aligned} \tag{9}$$

Proof. Let X be an itemset in $\mathbf{L}(\mathscr{D}, \sigma)$. Assume that $X \in \mathbf{G}_L$ and $d(X) \geq 2$. Then, X can be generated by the intersection of some itemsets in $\mathbf{G}_L$, namely,

$$X = \bigcap_{k \in N^*, k \geq 2} V_k, \quad \text{where each } V_k \in \mathbf{G}_L. \tag{10}$$

By Lemma 8, $X \notin \mathbf{G}_L$. This contradicts the assumption $X \in \mathbf{G}_L$. Therefore, if $X \in \mathbf{G}_L$, then $d(X) \leq 1$. □

Example 11. Let a transaction database $\mathscr{D}$ be given as Table 1 and $\mathbf{L}(\mathscr{D}, \sigma)$ be computed as Table 2. The set $\mathbf{G}_L$ can be computed by applying (9), namely, $\mathbf{G}_L = \{ABC, ABD, AC, BC, AD, BD\}$.

Definition 12 (set of maximal elements). An element Y of $\mathbf{L}(\mathscr{D}, \sigma)$ is said to be a maximal element, if for all $X \in \mathbf{L}(\mathscr{D}, \sigma)$ and $Y \subseteq X$; then $Y = X$. A set of maximal elements of $\mathbf{L}(\mathscr{D}, \sigma)$ is denoted by $\text{MAX}(\mathbf{L}(\mathscr{D}, \sigma))$.

Lemma 13 (the maximal set in the intersection lattice and generating set). *Given an intersection lattice $\mathbf{L}(\mathscr{D}, \sigma)$, then* $\text{MAX}(\mathbf{G}_L) = \text{MAX}(\mathbf{L}(\mathscr{D}, \sigma))$.

Proof. Assuming that $X \in \text{MAX}(\mathbf{G}_L)$, then $X \in \mathbf{L}(\mathscr{D}, \sigma)$. Let $Y \in \mathbf{L}(\mathscr{D}, \sigma)$; then

$$Y = \bigcap_{k \in N^*} V_k, \quad \text{where each } V_k \in \mathbf{G}_L. \tag{11}$$

Assuming that $X \subseteq Y$, we have $X \subseteq V_k$, where $k \in N^*$. By Definition 12, $X = V_k$, $k \in N^*$; hence, $X = Y$. Therefore, $X \in \text{MAX}(\mathbf{L}(\mathscr{D}, \sigma))$. In other words, $\text{MAX}(\mathbf{G}_L) \subseteq \text{MAX}(\mathbf{L}(\mathscr{D}, \sigma))$ $(^*)$.

Conversely, assuming that $X \in \text{MAX}(\mathbf{L}(\mathscr{D}, \sigma))$, then

$$Y = \bigcap_{i \in N^*} S_i, \quad \text{where each } S_i \in \mathbf{G}_L. \tag{12}$$

Assuming that $X \subseteq Y$, we have $X \subseteq S_i$, $i \in N^*$. By Definition 12, $X = S_i$, $i \in N^*$. Thus, since $S_i \in \mathbf{G}_L$, we have

$X \in \mathbf{G}_L$. Then, we imply that $X \in \mathrm{MAX}(\mathbf{G}_L)$. In other words, $\mathrm{MAX}(\mathbf{L}(\mathscr{D},\sigma)) \subseteq \mathrm{MAX}(\mathbf{G}_L)$ $(^{**})$.

By $(^{*})$ and $(^{**})$, we imply that $\mathrm{MAX}(\mathbf{G}_L) = \mathrm{MAX}(\mathbf{L}(\mathscr{D},\sigma))$. □

Definition 14 (coatom). Each item of $\mathbf{L}(\mathscr{D},\sigma)$ is called an atom and each element of the set $\mathrm{MAX}(\mathbf{L}(\mathscr{D},\sigma))$ is called a coatom of $\mathbf{L}(\mathscr{D},\sigma)$. A set of all coatoms of $\mathbf{L}(\mathscr{D},\sigma)$ is denoted by $\mathbf{C}_L$.

By Lemma 13 and Definition 14, we can infer the property of $\mathbf{C}_L$ as follows.

Lemma 15 (characteristics of coatom in the intersection lattice). *For every intersection lattice* $\mathbf{L}(\mathscr{D},\sigma)$*, one always has* $\mathbf{C}_L = \mathrm{MAX}(\mathbf{G}_L)$

The set $\mathbf{C}_L$ of $\mathbf{L}(\mathscr{D},\sigma)$ can be calculated by applying Theorem 10 to find $\mathbf{G}_L$ and then find the maximal itemsets of the set $\mathbf{C}_L$.

Lemma 16. *For each itemset* $X \in \mathbf{L}(\mathscr{D},\sigma)$*,* $\mathrm{Poset}(X)$ forms a lattice and $\mathrm{Poset}(X)\setminus\{X\}$ has a generating set, denoted by $\mathbf{G}_X$, including itemsets in $\mathrm{MAX}(\mathrm{Poset}(X)\setminus\{X\})$.

Proof. By Lemma 2, if $X \in \mathbf{L}(\mathscr{D},\sigma)$ and $Y \subseteq X$, then $Y \in \mathbf{L}(\mathscr{D},\sigma)$. It is obvious that $(\mathrm{Poset}(X); \subseteq)$ is a lattice and $\mathrm{MAX}(\mathrm{Poset}(X)\setminus\{X\}) = \{X\setminus\{I_k\} \mid I_k \text{ is an item of } X\}$. Moreover, for item $I_k \in X$, $k = 1,2,\ldots|X|$, every itemset formed by $X\setminus\{I_k\}$ lacks only one item, so it has a unique containing itemset in $\mathrm{Poset}(X)$ and it has no containing itemset in $\mathrm{Poset}(X)\setminus\{X\}$. All remaining subsets in $\mathrm{Poset}(X)$ lack more than one item, so they have at least two containing itemsets in $\mathrm{Poset}(X)\setminus\{X\}$. By Definition 7, $\mathbf{G}_X$ includes itemsets in $\mathrm{MAX}(\mathrm{Poset}(X)\setminus\{X\})$. □

For example, itemset ABE forms a lattice and $\mathbf{G}_{ABE} = \{AB, AE, BE\}$.

Lemma 17. *If every itemset of* $\mathbf{C}_L$ *is not hidden, then no itemset in* $\mathbf{L}(\mathscr{D},\sigma)$ *is hidden.*

Proof. By Lemma 15, we have for all $X \in \mathbf{L}(\mathscr{D},\sigma) \exists\, Y \in \mathbf{C}_L$ such that $X \subseteq Y$, so $\alpha(X) \geq \alpha(Y)$ (Lemma 2). Since $\alpha(Y) \geq \sigma$, we have $\alpha(X) \geq \sigma$. □

In order to hide a sensitive association rule, this study focuses on decreasing support and confidence of the rule by removing an item belonging to its right-hand side. However, the modification of an item always affects some itemsets in $\mathbf{L}(\mathscr{D},\sigma)$. By (2) and (3), when the support of an itemset is reduced by modifying some items, the support and confidence of association rules that contain these items will be changed. This may lead those rules to be hidden. Moreover, when an itemset is hidden, all association rules generated from this itemset are also hidden. If the hidden rules are not sensitive rules, then they are lost rules. The efficient method that allows the reduction of lost rules restricts itemsets in $\mathbf{L}(\mathscr{D},\sigma)$ from being hidden.

By Definition 7, each itemset of intersection lattice $\mathbf{L}(\mathscr{D},\sigma)$ can be created by an intersection of some itemsets in $\mathbf{G}_L$. Lemma 17 indicates that all itemsets in $\mathbf{L}(\mathscr{D},\sigma)$ are still frequent if every itemset in $\mathbf{C}_L$ is maintained. The generating set $\mathbf{G}_L$ and coatom set $\mathbf{C}_L$ therefore need to be protected from the hiding process in order to maintain $\mathbf{L}(\mathscr{D},\sigma)$. It is possible to propose a heuristic that hides sensitive association rules with lower side effects based on $\mathbf{G}_L$ and $\mathbf{C}_L$ maintenance.

5.2. The Heuristics for Minimizing Side Effects of Association Rule Hiding Algorithm. In this research, we apply method (iii) to hide the rule $I_l \rightarrow I_r$ by removing an item belonging to I_r from some transactions that support the rule until $\alpha(I_l \rightarrow I_r) < \sigma$ or $\beta(I_l \rightarrow I_r) < \delta$. The impacts of the hiding process on $\mathbf{L}(\mathscr{D},\sigma)$ depend on the item and transactions selection for the data modifications [24]. This study proposes an efficient improvement of the intersection lattice approach [23–25] based on two heuristics for minimizing the side effects of association rule hiding process. In this study, we prove the correctness and efficiency of the heuristic for specifying victim item that was presented in [23, 24] and propose an improvement heuristic for specifying transactions [25]. These heuristics are presented as follows.

Heuristic 1 (specifying victim item for data modifications). For each item $I_i \in I_r$, modifying I_i affects support of $|X|-1$ itemsets in $\mathbf{G}_X$, where $X \in \mathbf{C}_L$. It is obvious that the itemset which has the smallest support in $\mathbf{G}_X$ is the easiest to be hidden. This heuristic aims to protect those itemsets in order to restrict the impacts of the hiding process to $\mathbf{L}(\mathscr{D},\sigma)$. Firstly, it identifies itemsets $Y \in \mathbf{G}_X$, where $X \in \mathbf{C}_L$ and $I_lI_r \subseteq X$, which are the most vulnerable to the modification of each item in I_r.

Definition 18 (victim candidate). The victim candidate for hiding a sensitive rule $I_l \rightarrow I_r$, denoted by $\mathbf{M}_{\min}(I_i, X)$, is a set of tuples, where each tuple contains four values: $I_i \in I_r$, itemset $X \in \mathbf{C}_L$ such that $I_lI_r \subseteq X$, itemset $Y \in \mathbf{G}_X$ such that $I_i \subseteq Y$ and Y has minimum support in $\mathbf{G}_X$, and $\alpha(Y)$. It is computed as follows:

$$\begin{aligned}\mathbf{M}_{\min}(I_i, X) = \{(I_i, X, Y, \lambda) \mid \lambda & \\ = \min\{\alpha(Y) \mid I_i \subseteq I_r \cap Y,& \\ Y \in \mathbf{G}_X,\ I_lI_r \subseteq X, X \in \mathbf{C}_L\}\}&.\end{aligned} \tag{13}$$

In order to maintain the set $\mathbf{G}_L$ and $\mathbf{C}_L$, the modification is required with item in the same tuple with the itemsets that have maximum support among elements of $\mathbf{M}_{\min}(I_i, X)$. Such an item is said to be the *victim item* and is defined as follows.

Definition 19 (victim item). The victim item for hiding the sensitive rule $I_l \rightarrow I_r$, denoted by I_{victim}, is an item needed to be modified in order to hide the rule such that the modification causes the lowest impacts on $\mathbf{L}(\mathscr{D},\sigma)$, and it is computed as follows:

$$\begin{aligned}&\mathbf{M}_{\text{maxmin}}(I_l \longrightarrow I_r) \\ &= \{(I_{\text{victim}}, X, Z, \mu) \mid \mu \\ &\quad = \max\{\lambda \mid (I_i, X, Z, \lambda) \in \mathbf{M}_{\min}(I_i, X)\}\}.\end{aligned} \tag{14}$$

Function $\mathbf{M}_{\text{maxmin}}(I_l \rightarrow I_r)$ shows that the item I_{victim} needs to be removed from transactions that support the rule $I_l \rightarrow I_r$. If there are more than two tuples in $\mathbf{M}_{\text{maxmin}}(I_l \rightarrow I_r)$, then the victim item is selected randomly from those tuples.

Theorem 20. *Equation* (14) *always returns a victim item for association rule hiding.*

Proof. By Lemmas 13 and 15, for every rule $I_l \rightarrow I_r$, there is an itemset $X \in \mathbf{C}_L$ such that $I_l I_r \subseteq X$. Let $Z \in \mathbf{G}_X$, by Lemma 16, $|Z| = |X| - 1$. In addition, $|I_r| \leq |I_l I_r| - 1$ so that $|I_r| \leq |Z|$; therefore, there are $|X| - 1$ itemsets $Y \in \mathbf{G}_X$ such that $I_i \subseteq (I_r \cap Y)$. This indicates that the set $\mathbf{M}_{\min}(I_i, X)$ can always be specified. Obviously, we can find a tuple $(I_{\text{victim}}, X, Y, \mu)$ where $\mu = \max\{\alpha(Y) \mid Y \in \mathbf{M}_{\min}(I_i, X)\}$. In other words, the function $\mathbf{M}_{\text{maxmin}}(I_l \rightarrow I_r)$ always returns the victim item I_{victim}. □

Theorem 21. *Modifying the victim item returned by* (14) *causes minimal impacts on the intersection lattice of frequent itemsets.*

Proof. According to (13), the set $\mathbf{M}_{\min}(I_i, X)$ contains all items $I_i \in I_r$ and itemset in $\mathbf{G}_L$ which is the most vulnerable to the modification of item I_i. Obviously, modifying an item which is contained in the same tuple with the itemset that has maximum support in $\mathbf{M}_{\min}(I_i, X)$ produces the lowest impacts on $\mathbf{G}_L$. Consequently, modifying I_{victim} returned by (14) causes minimal impacts on $\mathbf{L}(\mathcal{D}, \sigma)$. □

Heuristic 2 (specifying transaction for data modifications). Assuming that both nonsensitive association rules $X \rightarrow Y$ and sensitive association rules $I_l \rightarrow I_r$ are supported by transaction t, the rule $X \rightarrow Y$ is still strong if $\alpha(X \rightarrow Y) \geq \sigma$ and $\beta(X \rightarrow Y) \geq \delta$. Let a positive integer k be assigned as the number of transactions required to be modified. To maintain the nonsensitive rule $X \rightarrow Y$, k must satisfy the conditions $\alpha(X \rightarrow Y) - k \geq \sigma$ and $(\alpha(X \rightarrow Y) - k)/\alpha(X) \geq \delta$.

Thus, we have $k \leq \alpha(X \rightarrow Y) - \sigma$ and $k \leq \alpha(X \rightarrow Y) - [\alpha(X)^* \delta]$.

The maximal number of transactions that can be modified without hiding the nonsensitive association rules $X \rightarrow Y$ is

$$N(X \longrightarrow Y) = \min\{\alpha(X \cup Y) - \sigma,\ \alpha(X \cup Y) - [\alpha(X)^* \delta]\}. \quad (15)$$

Transaction t is *safe* to the hiding process if no nonsensitive rule supported by t is hidden. We formulate the *safety degree* of transaction t, denoted by $\text{SD}(t)$, as follows:

$$\text{SD}(t) = \min\{N(X \longrightarrow Y) \mid X \longrightarrow Y \in R \setminus S_R \wedge I_r \cap Y \neq \emptyset \wedge XY \subseteq t\}. \quad (16)$$

Accordingly, no nonsensitive rule supported by t is hidden if $\text{SD}(t)$ is above zero. In other words, we need to maintain $\text{SD}(t)$ during the hiding process in order to restrict the nonsensitive rules from being hidden. As a result, transaction that has high safety degree should be modified first.

Let n_{-}trans be the minimum number of transactions that need to be modified in order to hide the sensitive rule r. Then, n_{-}trans can be computed as follows:

$$n_{-}\text{trans} = \min\left\{\alpha(r) - \sigma + 1, \alpha(r) - \left\lceil \alpha(l_r)^* \delta \right\rceil + 1\right\}, \quad (17)$$

where l_r is left hand side of r.

Let T_r be a set of transactions that supports the rule r. Let R_t be a set of nonsensitive association rules supported by transaction $t \in T_r$, namely, $R_t = \{X \rightarrow Y \in R \setminus S_R \mid XY \subseteq t\}$. It is obvious that removing victim item from the transaction t that supports the lowest $|R_t|$ and greatest $|S_R|$ and $\text{SD}(t)$ causes the lowest impacts on $\mathbf{L}(\mathcal{D}, \sigma)$ and nonsensitive association rules.

For each transaction $t \in T_r$, a weight $w(t)$ was assigned to measure ability of removing victim item from t so as to hide the sensitive rule r, but the modification causes the least impact on R_t:

$$w(t) = \begin{cases} \infty & \text{if } R_t = \phi, \\ \dfrac{|S_R|^* \text{SD}(t)}{|R_t|} & \text{if } R_t \neq \phi. \end{cases} \quad (18)$$

Since transaction $t \in T_r$ does not support any nonsensitive association rule corresponding with r, $w(t)$ will be assigned maximal value, because modifying such transaction t does not affect any nonsensitive rule. As a result, modifying the high-weight transaction contributes to restricting the lost rules.

5.3. The Proposed Algorithm. Based on the heuristics that are presented in Section 5.2, we propose a new algorithm, denoted by AARHIL (algorithm of association rule hiding based on intersection lattice), that includes two steps as follows.

Step 1 (initiation). AARHIL computes $\mathbf{G}_L$ and $\mathbf{C}_L$ of the intersection lattice of frequent itemsets $\mathbf{L}(\mathcal{D}, \sigma)$ using Theorem 10 and Lemma 15, respectively.

Step 2 (hiding process). AARHIL executes three sub-steps for each sensitive association rule r.

Step 2.1. AARHIL specifies a set of transactions, denoted by T_r, that fully support the sensitive rule r. The algorithm computes the weight of each transaction in T_r using (16) and (18). Then, it sorts T_r in descending order of weight.

Step 2.2. AARHIL specifies victim item using (13) and (14).

Step 2.3. The victim item will be changed when support of itemset in the same tuple with I_{victim} less than

$$\max\{\alpha(Y) \mid (I_i, X, Y, \lambda) \in \mathbf{M}_{\min}(I_i, X),\ \alpha(Y) \neq \alpha(Z),\ X \in \mathbf{C}_L\}. \quad (19)$$

Input: *Original database $\mathscr{D}$, thresholds σ and δ, $\mathbf{L}(\mathscr{D}, \sigma)$, association rules R and sensitive association rules S_R, .*
Output: *Released database $\mathscr{D}'$.*
Method:
For each $X \subseteq \mathbf{L}(\mathscr{D}, \sigma)$ **Do**
  Calculate $d(X) = |\{Y \in \mathbf{L}(\mathscr{D}, \sigma) \mid X \subset Y\}|$;
$\mathbf{G}_L = \{X \in \mathbf{L}(\mathscr{D}, \sigma) \mid d(X) \leq 1\}$;
$\mathbf{C}_L = \mathrm{MAX}(\mathbf{G}_L)$;
For each rule $r \in S_R$
  Compute T_r;
  For each transaction $t \in T_r$, compute $w(t)$;
  Sort T_r in descending order of $w(t)$;
  n_trans = min $\{\alpha(r) - \sigma + 1, \alpha(r) - \lceil \alpha(\mathbf{l}_r) * \delta \rceil + 1\}$;
  While n_trans > 0
    Specify victim item $\mathbf{I}_{\text{victim}}$: $\mathbf{M}_{\text{maxmin}}(r) = (\mathbf{I}_{\text{victim}}, X, Z, \mu)$;
    k_trans $= \alpha(Z) - \max\{\alpha(Y) \mid (\mathbf{I}_i, X, Y, \lambda) \in \mathbf{M}_{\min}(\mathbf{I}_i, X), \alpha(Y) \neq \alpha(Z), X \in \mathbf{C}_L\} + 1$;
    Remove $\mathbf{I}_{\text{victim}}$ from first max {min {k_trans, n_trans}, 1} transactions in T_r;
    Remove first max {min {k_trans, n_trans} , 1} transactions from T_r;
    n_trans = n_trans − max {min {k_trans, n_trans} , 1};
    Update $\mathbf{L}(\mathscr{D}', \sigma), \alpha(r), \beta(r), \mathbf{G}_{L'}, \mathbf{C}_{L'}$;
  End
End

Algorithm 1: The AARHIL algorithm.

Thus, to save the time needed for updating $L(\mathscr{D}', \sigma)$, $\alpha(r)$, $\beta(r)$, $G_{L'}$, $C_{L'}$, the victim item needs to be updated from k_trans transactions in T_r, where

$$k\_\text{trans} = \alpha(Z) - \max\{\alpha(Y) \mid (I_i, X, Y, \lambda) \in \mathbf{M}_{\min}(I_i, X), \alpha(Y) \neq \alpha(Z), X \in \mathbf{C}_L\} + 1. \tag{20}$$

Next, AARHIL updates itemsets in $\mathbf{L}(\mathscr{D}', \sigma)$, $\mathbf{G}_L$, and $\mathbf{C}_L$.

Since the victim item I_{victim} is removed from transaction t, the support of every itemset that is supported by t and contains I_{victim} is decreased one unit. The intersection lattice $\mathbf{L}(\mathscr{D}', \sigma)$ can be updated by removing all itemsets that have support less than σ from $\mathbf{L}(\mathscr{D}, \sigma)$. The generating set of $\mathbf{L}(\mathscr{D}', \sigma)$, denoted by $\mathbf{G}_{L'}$, can be updated as follows.

For each itemset $X \in \mathbf{G}_L$ such that $\alpha(X) < \sigma$,

$$\mathbf{G}_{L'} = \mathbf{G}_L \setminus \{X\} \cup \left\{ Y \mid Y \in \mathbf{G}_X, Y \neq \bigcap_{k \in N^*, k \geq 2} Z_k, Z_k \in \mathbf{G}_L \setminus \{X\} \right\}. \tag{21}$$

Then, $\mathbf{C}_{L'}$ of $\mathbf{L}(\mathscr{D}', \sigma)$ is updated by taking the maximal itemsets of $\mathbf{G}_{L'} : \mathbf{C}_{L'} = \mathrm{MAX}(\mathbf{G}_{L'})$.

AARHIL then computes $\alpha(r)$ and $\beta(r)$. The algorithm repeats this step until $\alpha(r) < \sigma$ or $\beta(r) < \delta$.

The details of AARHIL algorithm are presented in Algorithm 1.

The correctness of AARHIL was proved by Theorem 20. Moreover, by Theorem 21 and the second heuristic, AARHIL hides a set of sensitive association rules with the lowest lost rules while maintaining a high accuracy. The complexity of AARHIL is computed in Theorem 22.

Theorem 22. *Computational complexity of algorithm AARHIL is $O(k_F^2 + n + n_r \log n_r + k_{\text{tmax}}^2)$, where k_F is the number of frequent itemsets, k_{tmax} is the largest transaction, n_r is the greatest number of transactions supporting the sensitive rule, and n is the size of database (total number of transactions).*

6. Experimental Results and Discussion

In order to measure the efficiency of proposed model, we compared our algorithm with MaxMin2 [9], WSDA [22], the algorithm proposed by Jain [15], denoted by JA (Jain Algorithm), and HCSRIL proposed by Hai et al. [24]. Moustakides and Verykios [9] showed that MaxMin2 is a more efficient method compared with the previous border-based approach [8], which has achieved better results compared with the heuristic Algorithm 2(b) in [13]. The WSDA algorithm applies heuristic to select the appropriate transactions for modifying an item on the right-hand side of the sensitive rules. The experimental results have indicated that WSDA is more efficient compared with Algorithm 1(b) in [13]. Jain at al. [15] proposed the new algorithm (JA) that overcomes ISL and DSR algorithms [28]. The HCSRIL algorithm applied heuristic on victim item selection based on intersection lattice theory.

The experiment was run on Windows 7 operating system with a Pentium Core i5 and 4 GB of RAM. Our experiments were executed using the Retail.dat dataset, which was donated by Brijs [29]. This dataset contains the retail market basket data from an anonymous Belgian retail store. It contains 88,162 transactions on 16,469 items. In order to examine the performance of the proposed algorithm compared with

Table 4: Configuration of datasets and number of association rules satisfy $\sigma = 1\%$ and $\delta = 10\%$.

Number of transactions	Number of items	Largest transaction	Number of association rules
30 k	12,142	75	340
40 k	13,462	75	316
50 k	14,413	75	256
60 k	14,997	75	239
70 k	15,780	77	238
80 k	16,014	77	236
88.162 k	16,469	77	236

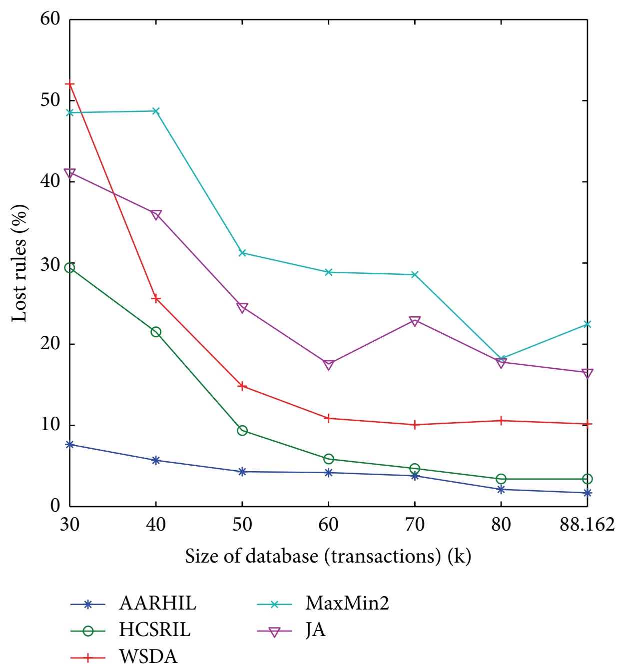

Figure 1: Lost rules comparison.

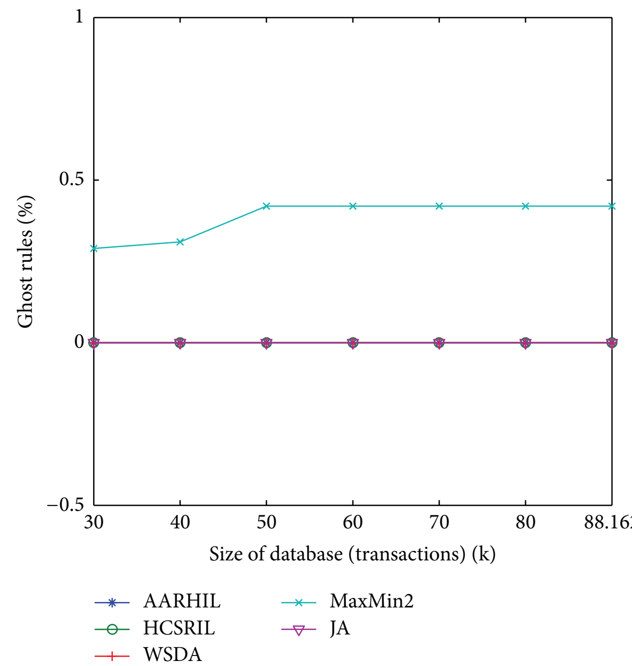

Figure 2: Ghost rules comparison.

the previous works, we started the experiments with 30,000 transactions of dataset on 12,142 corresponding items and then extended the dataset up to the maximum. The configurations of datasets are presented in Table 4.

We selected two sensitive association rules for the experiments. The performances of these algorithms are illustrated in the following figures.

Figure 1 shows that AARHIL algorithm produced the lowest lost rules in every dataset. In other words, AARHIL achieved the best results in minimizing the lost rules compared with HCSRIL, WSDA, JA, and MaxMin2 algorithms. By applying the support reduction method (i), MaxMin2 produced many lost rules. JA combines methods (ii) and (iii), but it does not apply a heuristic to select victim items and transactions. Thus, it produced more lost rules compared with WSDA, which applied a heuristic to select transactions for data modification. AARHIL applies two heuristics to select appropriate victim items and transactions for data modification using the combination of methods (i) and (iii). Moreover, AARHIL applies a heuristic to compute weight of transactions and sort them before modifying, so it attained the lower lost rules compared with HCSRIL.

Figure 2 indicates that these algorithms produce very few ghost rules. The AARHIL, HCSRIL, WSDA, and JA algorithms did not create ghost rules, whereas the number of ghost rules created by MaxMin2 is more than 0.4 percent.

There was no false rule produced by these algorithms when dealing with the selected sensitive association rules for every case of dataset.

Figure 3 shows the comparison of these algorithms on the aspect of accuracy of released dataset. With two rules for hiding being selected, the accuracy of released dataset was very high. This means the hiding process caused a few changes in the released dataset compared with the original dataset. Moreover, by modifying the same number of data items, AARHIL and HCSRIL algorithms achieved the same accuracy, but this accuracy is highest compared to other algorithms in every dataset.

The execution times for these algorithms are shown in Figure 4. These algorithms required only 2000 seconds for running 88,162 transactions of 16,469 items, whereas the

TABLE 5: Average side effect and CPU-Time produced by AARHIL, WSDA, and MaxMin2.

Algorithm	Lost rule (%)	False rule (%)	Ghost rule (%)	Accuracy (%)	CPU-Time (s)
AARHIL	4.20	0	0	99.74	55
HCSRIL	11.09	0	0	99.74	228
WSDA	19.18	0	0	99.59	692
JA	24.48	0	0	99.40	191
MaxMin2	32.37	0	0.38	99.35	1219

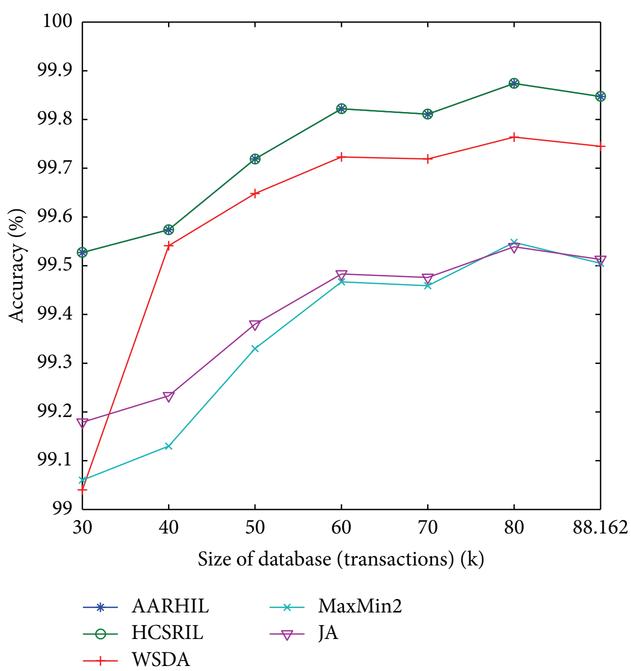

FIGURE 3: Accuracy comparison.

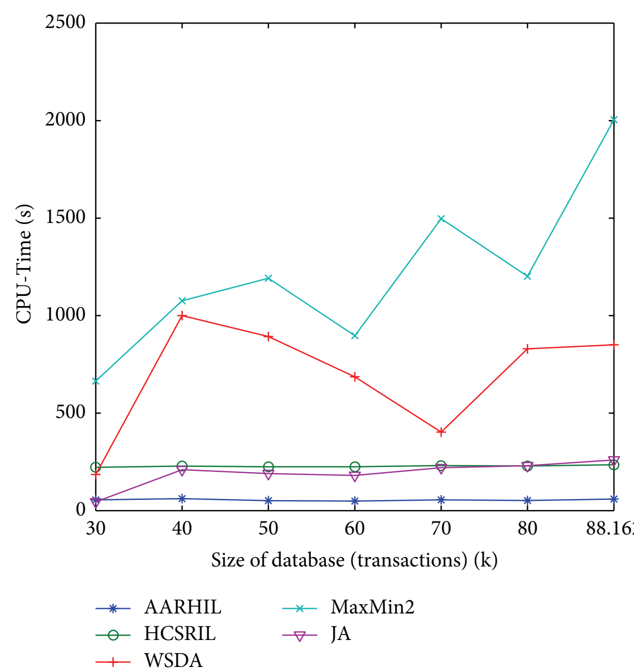

FIGURE 4: Required execution time.

MaxMin2 algorithm required more times compared with the others. The difference between execution times of HCSRIL and JA algorithms is not significant. By reducing the time to access database and the time to compute $\mathbf{G}_L$, AARHIL achieved lowest CPU-Time.

Table 5 shows the performance of these algorithms in the average case. Accordingly, AARHIL achieved the best results in the side effects minimization. On average, AARHIL achieved 4% lost rule compared with 11% of HCSRIL, 19% of WSDA, 24% of JA, and 32% of MaxMin2. These algorithms attained the same performance in the remaining side effects, whereas MaxMin2 produced 0.38 percent of ghost rules. Moreover, AARHIL achieved the lowest CPU-Time compared with the others.

In summary, the results show that the AARHIL algorithm outperforms the HCSRIL, JA, MaxMin2, and WSDA in minimizing the side effects and computational complexity. Hence, this algorithm is suitable for application in the real world.

7. Conclusion

This study introduced in detail the theories of intersection lattice of frequent itemsets, denoted by $\mathbf{L}(\mathscr{D}, \sigma)$, and proposed an improvement to minimize size effects and complexity of intersection lattice-based approach. In order to minimize side effects, two heuristics are formulated relying on the properties of the generating set $\mathbf{G}_L$ of $\mathbf{L}(\mathscr{D}, \sigma)$. The first heuristic aims at specifying the victim item for data distortions such that the modification causes the least impacts on $\mathbf{L}(\mathscr{D}, \sigma)$. The improvement is applied in the second heuristic that computes the weight to each transaction relying on their safety degree, the number of sensitive rules, and the number of nonsensitive association rules contained by that transaction. Removing the victim item from the minimum number of specified transactions that have the highest weight contributes to achieving the lowest lost rules and highest accuracy and to restricting ghost rules. The experimental results showed that the proposed algorithm, AARHIL, achieved minimum side effects and CPU-Time compared with HCSRIL, MaxMin2, WSDA, and JA algorithms in the context of hiding a specified set of sensitive association rules.

Acknowledgment

The authors wish to acknowledge the support of the Department of Computer Science, Faculty of Science, Khon Kaen University Publication Clinic, Research and Technology Transfer Affairs, Khon Kaen University, for their assistance.

References

[1] J. Xi, Z. Gao, S. Niu, T. Ding, and G. Ning, "A hybrid algorithm of traffic accident data mining on cause analysis," *Mathematical Problems in Engineering*, vol. 2013, Article ID 302627, 8 pages, 2013.

[2] M. Dong, "A tutorial on nonlinear time-series data mining in engineering asset health and reliability prediction: concepts, models, and algorithms," *Mathematical Problems in Engineering*, vol. 2010, Article ID 175936, 22 pages, 2010.

[3] C. Gokceoglu, H. A. Nefeslioglu, E. Sezer, A. S. Bozkir, and T. Y. Duman, "Assessment of landslide susceptibility by decision trees in the metropolitan area of Istanbul, Turkey," *Mathematical Problems in Engineering*, vol. 2010, Article ID 901095, 15 pages, 2010.

[4] Z. R. Adam, *An Introduction To Ecommerce*, DAI-AGILE, 2000.

[5] D. Y. Zhang, Y. Zeng, L. Wang, H. Li, and Y. Geng, "Modeling and evaluating information leakage caused by inferences in supply chains," *Computers in Industry*, vol. 62, no. 3, pp. 351–363, 2011.

[6] K. I. Ronald, *Supply Chain Collaboration: How To Implement CPRT and Other Best Collaborative Practices*, J.ROSS Publishing, Plantation, Fla, USA, 2005.

[7] A. Gkoulalas-Divanis and V. S. Verykios, *Association Rule Hiding For Data Mining*, LaVergne, Tenn, USA, 2011.

[8] X. Sun and P. S. Yu, "Hiding sensitive frequent itemsets by a border-based approach," *Computing Science and EngInerring*, vol. 1, no. 1, pp. 74–94, 2007.

[9] G. V. Moustakides and V. S. Verykios, "A MaxMin approach for hiding frequent itemsets," *Data and Knowledge Engineering*, vol. 65, no. 1, pp. 75–89, 2008.

[10] S. Menon, S. Sarkar, and S. Mukherjee, "Maximizing accuracy of shared databases when concealing sensitive patterns," *Information Systems Research*, vol. 16, no. 3, pp. 256–270, 2005.

[11] A. Gkoulalas-Divanis and V. S. Verykios, "An integer programming approach for frequent itemset hiding," in *Proceedings of the 15th ACM Conference on Information and Knowledge Management (CIKM '06)*, pp. 748–757, November 2006.

[12] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, "Disclosure limitation of sensitive rules," in *Proceedings of the Workshop on Knowledge and Data Engineering Exchange*, 1999.

[13] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association rule hiding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 434–447, 2004.

[14] S. R. M. Oliveira and O. R. Zaïane, "A unified framework for protecting sensitive association rules in business collaboration," *International Journal of Business Intelligence and Data Mining*, vol. 1, no. 3, pp. 247–287, 2006.

[15] Y. K. Jain, V. K. Yadav, and G. S. Panday, "An efficient association rule hiding algorithm for privacy preserving data mining hiding," *International Journal on Computer Science and Engineering*, vol. 3, no. 7, pp. 2792–2798, 2011.

[16] E. T. Wang and G. Lee, "An efficient sanitization algorithm for balancing information privacy and knowledge discovery in association patterns mining," *Data and Knowledge Engineering*, vol. 65, no. 3, pp. 463–484, 2008.

[17] E. D. Pontikakis, A. A. Tsitsonis, and V. S. Verykios, "An experimental study of distortion-based techniques for association rule hiding," *IFIP International Federation for Information Processing*, vol. 144, pp. 325–339, 2004.

[18] P. Gulwani, "Association rule hiding by positions swapping of support and confidence," *Information Technology and Computer Science*, vol. 4, pp. 54–61, 2012.

[19] D. Jain, A. Sinhal, N. Gupta et al., "Hiding sensitive association rules without altering the support of sensitive item(s)," *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 3, no. 2, pp. 75–84, 2012.

[20] Y. Saygin, V. S. Verykios, and C. Clifton, "Using unknowns to prevent discovery of association rules," *SIGMOD Record*, vol. 30, no. 4, pp. 45–54, 2001.

[21] T.-P. Hong, C.-W. Lin, C.-C. Chang, and S.-L. Wang, "Hiding sensitive itemsets by inserting dummy transactions," in *Proceedings of the IEEE International Conference on Granular Computing (GrC '11)*, pp. 246–249, November 2011.

[22] V. S. Verykios, E. D. Pontikakis, Y. Theodoridis, and L. Chang, "Efficient algorithms for distortion and blocking techniques in association rule hiding," *Distributed and Parallel Databases*, vol. 22, no. 1, pp. 85–104, 2007.

[23] L. Q. Hai and A. Somjit, "A Conceptual framework for privacy preserving of association rule mining in E-commerce," in *Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications (ICIEA '12)*, pp. 1999–2003, 2012.

[24] L. Q. Hai, A. Somjit, N. X. Huy, and A. Ngamnij, "Association rule hiding in risk management for retail supply chain collaboration," *Computers in Industry*, vol. 64, pp. 776–784, 2013.

[25] L. Q. Hai, A. Somjit, and A. Ngamnij, "Association rule hiding based on distance and intersection lattice," in *Proceedings of the 4th International Conference on Computer Technology and Development (ICCTD '12)*, pp. 227–231, 2012.

[26] C. Zhang and S. Zhang, *Association Rule Mining: Models and Algorithms*, vol. 2307 of *Lecture Notes in Artificial Intelligence*, Springer, New York, NY, USA, 2002.

[27] G. Grätzer, *Lattice Theory: Foundation*, 2010 Mathematics Subject Classification, Springer Basel AG, 2011.

[28] S.-L. Wang, Y.-H. Lee, S. Billis, and A. Jafari, "Hiding sensitive items in privacy preserving association rule mining," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 3239–3244, October 2004.

[29] T. Brijs, "Retail market basket data set," Workshop on Frequent Itemset Mining Implementations (FIMI '03), 2003, http://fimi.ua.ac.be/data/.