Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 235837, 12 pages
http://dx.doi.org/10.1155/2014/235837

Research Article

Reducing Side Effects of Hiding Sensitive Itemsets in Privacy Preserving Data Mining

Chun-Wei Lin,[1,2] Tzung-Pei Hong,[3,4] and Hung-Chuan Hsu[3]

[1] *Innovative Information Industry Research Center (IIIRC), School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China*
[2] *Shenzhen Key Laboratory of Internet Information Collaboration, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China*
[3] *Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan*
[4] *Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung 804, Taiwan*

Correspondence should be addressed to Tzung-Pei Hong; tphong@nuk.edu.tw

Received 16 January 2014; Accepted 26 February 2014; Published 10 April 2014

Academic Editors: T. Cao and M. Ivanovic

Data mining is traditionally adopted to retrieve and analyze knowledge from large amounts of data. Private or confidential data may be sanitized or suppressed before it is shared or published in public. Privacy preserving data mining (PPDM) has thus become an important issue in recent years. The most general way of PPDM is to sanitize the database to hide the sensitive information. In this paper, a novel hiding-missing-artificial utility (HMAU) algorithm is proposed to hide sensitive itemsets through transaction deletion. The transaction with the maximal ratio of sensitive to nonsensitive one is thus selected to be entirely deleted. Three side effects of hiding failures, missing itemsets, and artificial itemsets are considered to evaluate whether the transactions are required to be deleted for hiding sensitive itemsets. Three weights are also assigned as the importance to three factors, which can be set according to the requirement of users. Experiments are then conducted to show the performance of the proposed algorithm in execution time, number of deleted transactions, and number of side effects.

1. Introduction

With the rapid growth of data mining technologies in recent years, useful information can be easily mined to aid mangers or decision-makers for making efficient decisions or strategies. The derived knowledge can be simply classified into association rules [1–5], sequential patterns [6–8], classification [9, 10], clustering [11, 12], and utility mining [13–16], among others. Among them, association-rule mining is the most commonly used to determine the relationships of purchased items in large datasets.

Traditional data mining techniques analyze database to find potential relations among items. Some applications require protection against the disclosure of private, confidential, or secure data. Privacy preserving data mining (PPDM) [17] was thus proposed to reduce privacy threats by hiding sensitive information while allowing required information to be mined from databases. Privacy information includes some personal or confidential information in business, such as social security numbers, home address, credit card numbers, credit ratings, purchasing behavior, and best-selling commodity. In PPDM, data sanitization is generally used to hide sensitive information with the minimal side effects for keeping the original database as authentic as possible. The intuitive way of data sanitization to hide sensitive information is directly to delete sensitive information from amounts of data. Three side effects of hiding failure, missing cost, and artificial cost are then generated in data sanitization process but most approaches are designed to partially evaluate the side effects. Infrequent itemset is, however, not considered in the evaluation process, thus raising the probability of artificial itemsets caused. Besides, the differences between

the minimum support threshold and the frequencies of the itemsets to be hidden are not considered in the above approaches.

In this paper, a hiding-missing-artificial utility (HMAU) algorithm is proposed for evaluating the processed transactions to determine whether they are required to be deleted for hiding sensitive itemsets by considering three dimensions as hiding failure dimension (HFD), missing itemset dimension (MID), and artificial itemset dimension (AID). The weight of each dimension in evaluation process can be adjusted by users. Experimental results showed that the proposed HMAU algorithm has good performance in execution time and the number of deleted transactions. Besides, the proposed algorithm can thus generate minimal side effects of three factors compared to the past algorithm for transaction deletion to hide the sensitive itemsets.

This paper is organized as follows. Some related works are reviewed in Section 2, including the data mining techniques, the privacy preserving data mining, and the evaluated criteria of PPDM. The proposed HMAU algorithm to hide the sensitive itemsets for transaction deletion is stated in Section 3. An illustrated example of the proposed HMAU algorithm is given in Section 4 step by step. Experiments are conducted in Section 5. Conclusion and future works are mentioned Section 6.

2. Review of Related Works

In this section, privacy preserving data mining (PPDM) techniques and evaluated criteria of PPDM are respectively reviewed.

2.1. Privacy Preserving Data Mining Techniques. Data mining is used to extract useful rules from large amounts of data. Agrawal and Srikant proposed Apriori algorithm to mine association rules in two phases to firstly generate the frequent itemsets and secondly derive the association rules [3]. Han et al. then proposed the Frequent-Pattern-tree (FP-tree) structure for efficiently mining association rules without generation of candidate itemsets [18]. The FP-tree was used to compress a database into a tree structure which stored only large items. It was condensed and complete for finding all the frequent patterns. The construction process was executed tuple by tuple, from the first transaction to the last one. After that, a recursive mining procedure called FP-Growth was executed to derive frequent patterns from the FP-tree.

Through various data mining techniques, information can thus be efficiently discovered. The misuse of these techniques may, however, lead to privacy concerns and security problems. Privacy preserving data mining (PPDM) has thus become a critical issue for hiding private, confidential, or secure information. Most commonly, the original database is sanitized for hiding sensitive information [19–21].

In data sanitization, it is intuitive to directly delete sensitive data for hiding sensitive information. Leary found that data mining techniques can pose security and privacy threats [22]. Amiri proposed the aggregate, disaggregate, and hybrid approaches to, respectively, determine whether the transactions or the items are to be deleted for hiding sensitive information [23]. The approaches considered the ratio of sensitive itemsets to nonsensitive frequent itemsets to evaluate the side effects of hiding failures and missing itemsets. Oliveira and Zaïane designed the sliding window algorithm (SWA) [24], in which the victim item with the highest frequency in the sensitive rules related to the current sensitive transaction is selected. Victim items are removed from the sensitive transaction until the disclosure threshold equals 0. Hong et al. proposed a lattice-based algorithm to hide the sensitive information through itemset deletion by a lattice structure to speed up the sanitization process [25]. All the sensitive itemsets are firstly used to build the lattice structure. The sensitive itemsets are then gradually deleted bottom-up form the lowest levels to the highest ones until the frequencies of the sensitive itemsets are lower than the minimum support threshold. Different strategies for hiding sensitive itemsets are still designed in progress to find better results considering of side effects and the dissimilarity of database [21, 26–30].

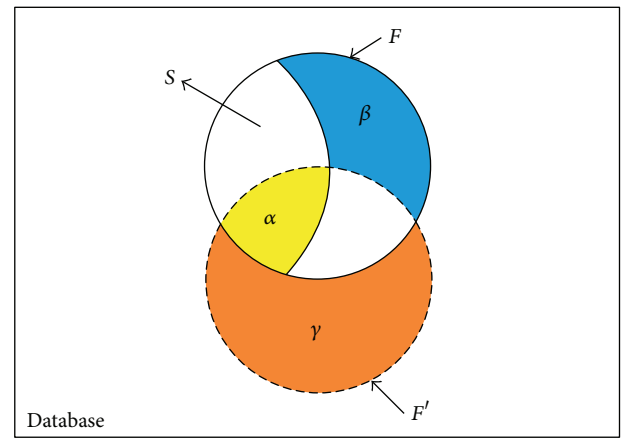

Figure 1: Relationship between the side effects and mined rules of the original database and sanitized one.

2.2. Evaluation Criteria. In data sanitization, the primary goal is to hide the sensitive information with minimal influences on databases. Three side effects of hiding failures, missing itemsets, and artificial itemsets are used to evaluate the performance of data sanitization. for data distortion [28, 31, 32] of sensitive itemsets in PPDM. The relationships between the side effects and mined itemsets of the original database and sanitized one are shown in Figure 1.

In Figure 1, F represents the frequent itemsets mined from the original database, F' represents the frequent itemsets mined from the sanitized database, and S represents the sensitive itemsets that should be hidden. The α part is concerned as hiding failures that fail to hide the sensitive itemsets. Thus, α is the intersection of S and F' ($= S \cap F'$). β part is concerned as missing itemsets that mistakenly to delete the nonsensitive frequent rules. Thus, β is the difference between F, S, and F' ($= F - S - F'$). γ part is concerned as artificial itemsets which is unexpectedly generated. Thus, γ is

the difference between F' and $F\,(=F'-F)$. In PPDM, it is intuitive to delete transactions with sensitive itemsets in the sanitization process. In this paper, α, β, and γ with adjustable weights are considered to evaluate whether the processed transactions are required to be deleted. Besides the above side effects, the number of deleted transactions or items is also a criterion to evaluate the data distortion [32, 33].

3. Proposed Hiding-Missing-Artificial Utility Algorithm

3.1. Definition of Formulas. Data sanitization is the most common way to protect sensitive knowledge from disclosure in PPDM. To avoid the side effects of hiding failures, missing itemsets, and artificial itemsets, minimal distortion of the databases is thus necessary. In this paper, a hiding-missing-artificial utility (HMAU) algorithm is proposed to hide sensitive itemsets through transaction deletion. Three dimensions of hiding failure dimension (HFD), missing itemset dimension (MID), and artificial itemset dimension (AID) are thus concerned to evaluate whether the transactions are required to be deleted for hiding the sensitive itemsets. The transactions with any of the sensitive itemset are first evaluated by the designed algorithm to find the minimal HMAU values among transactions, The transaction with minimal HMAU value will be directly removed from the database. The procedure is thus repeated until all sensitive itemsets are hidden. In order to avoid exposing the already hidden sensitive itemsets again, the minimum count is dynamically updated during the deletion procedure.

The value of each dimension is set from 0 to 1 ($0 <$ value ≤ 1). In the proposed formulas, the differences between minimum support threshold and the frequencies of the sensitive itemsets are thus considered to evaluate whether the transactions are required to be deleted instead of only the presence of the itemsets in the transactions.

First, the HFD is used to evaluate the hiding failures of each processed transaction in the sanitization process. When a processed transaction T_k contains a sensitive itemset hs_x, the HFD value of the processed transaction is calculated as

$$\mathrm{HFD}^k\left(\mathrm{hs}_x\right) = \frac{\mathrm{MAX}_{\mathrm{HS}} - \mathrm{freq}\left(\mathrm{hs}_x\right) + 1}{\mathrm{MAX}_{\mathrm{HS}} - \lceil |D| \times \lambda \rceil + 1}, \tag{1}$$

where λ is defined as the percentage of the minimum support threshold, sensitive itemset hs_x is from the set of sensitive itemsets HS, $\mathrm{MAX}_{\mathrm{HS}}$ is the maximal count of the sensitive itemsets in the set of sensitive itemsets HS, $|D|$ is the number of transactions in the original database D, and $\mathrm{freq}(\mathrm{hs}_x)$ is the occurrence frequency of the sensitive itemset hs_x.

Second, the MID is used to evaluate the itemsets of each processed transaction in the sanitization process. When a processed transaction T_k contains a frequent itemset fi_x, the MID value of the processed transaction is calculated as

$$\mathrm{MID}^k\left(\mathrm{fi}_x\right) = \frac{\mathrm{MAX}_{\mathrm{FI}} - \mathrm{freq}\left(\mathrm{fi}_x\right) + 1}{\mathrm{MAX}_{\mathrm{FI}} - \lceil |D| \times \lambda \rceil + 1}, \tag{2}$$

where an itemset fi_x is a frequent itemset from the set of large (frequent) itemsets FI, $\mathrm{MAX}_{\mathrm{FI}}$ is the maximal count of the large itemsets in the set of FI, and $\mathrm{freq}(\mathrm{fi}_x)$ is the occurrence frequency of the large itemset fi_x.

Third, the AID is used to evaluate the artificial itemsets of each processed transaction in the sanitization process. In AID, only the small 1-itemsets are considered in the sanitization process since it is a nontrivial task to keep all infrequent itemsets. When a processed transaction T_k contains a small 1-itemset si_x, the AID value of the processed transaction is calculated as

$$\mathrm{AID}^k\left(\mathrm{si}_x\right) = \frac{\mathrm{freq}\left(\mathrm{si}_x\right) - \mathrm{MIN}_{\mathrm{SI}^1} + 1}{\lceil |D| \times \lambda \rceil - \mathrm{MIN}_{\mathrm{SI}^1}}, \tag{3}$$

where a small 1-itemset si_x is from the set of small 1-itemsets SI^1, $\mathrm{MIN}_{\mathrm{SI}^1}$ is the minimal count of the small 1-itemsets in the set of SI^1, and $\mathrm{freq}(\mathrm{si}_x)$ is the occurrence frequency of the small 1-itemset si_x.

In this paper, a risky bound is designed to speed up the execution time of the proposed HMAU algorithm by avoiding the evaluation of all large itemsets and small 1-itemsets by considering MID and AID. A parameter μ is set as the percentage used to find the upper and lower boundaries of the minimum support threshold. Only the large itemsets and infrequent 1-itemsets within the boundaries are used to determine whether the processed transactions are required to be deleted. For the large itemsets, the minimum support threshold is set as the lower boundary, and the upper boundary is set as

$$\mathrm{freq}\left(\mathrm{fi}_j\right) \leq \left\lceil \lceil |D| \times \lambda \rceil \times (1+\mu) \right\rceil, \tag{4}$$

where $|D|$ is the number of transactions in the original database D, λ is the minimum support threshold, μ is the risky bound, and $\mathrm{freq}(\mathrm{fi}_j)$ is the occurrence frequency of the large itemset fi_j.

For small 1-itemsets, the minimum support threshold is set as the upper boundary, and the lower boundary is set as

$$\mathrm{freq}\left(\mathrm{si}_a\right) \geq \left\lfloor \lceil |D| \times \lambda \rceil \times (1-\mu) \right\rfloor, \tag{5}$$

where $\mathrm{freq}(\mathrm{si}_a)$ is the occurrence frequency of the small 1-itemset si_a.

The flowchart of the proposed HMAU algorithm is depicted in Figure 2.

3.2. Notation. See Table 1.

Details of the proposed HMAU algorithm are illustrated as follows.

Proposed HMAU Algorithm.

Input. This includes an original database D, a minimum support threshold ratio λ, a risky bound μ, a set of large (frequent) itemsets $\mathrm{FI} = \{\mathrm{fi}_1, \mathrm{fi}_2, \ldots, \mathrm{fi}_p\}$, a set of small

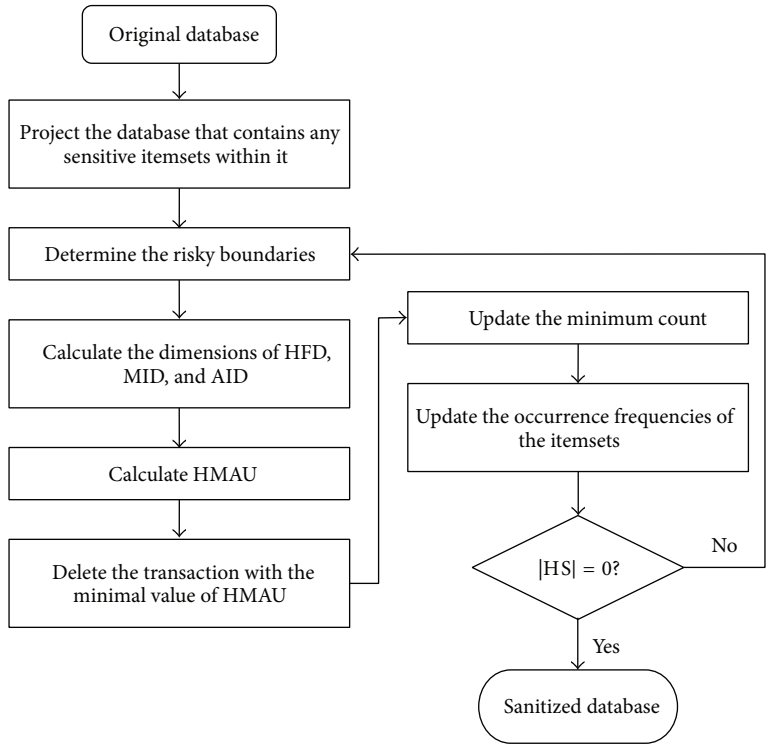

Figure 2: Flowchart of the proposed HMAU algorithm.

(nonfrequent) 1-itemsets $SI^1 = \{si_1, si_2, \ldots, si_q\}$, and a set of sensitive itemsets to be hidden $HS = \{hs_1, hs_2, \ldots, hs_r\}$.

Output. This includes a sanitized database D^* with no sensitive information.

Step 1. Select the transactions to form a projected database D', where each transaction T_k in D' consists of sensitive itemsets hs_i within it, where $1 \le i \le r$.

Step 2. Process each frequent itemset fi_j in the set of FI to determine whether its frequency satisfies the condition $\text{freq}(fi_j) \le \lceil \lceil |D| \times \lambda \rceil \times (1 + \mu) \rceil$, where $|D|$ is the number of transactions in the original database D and $\text{freq}(fi_j)$ is the occurrence frequency of the large itemset fi_j. Put the fi_j that do not satisfy the condition into the set of FI_{tmp}.

Step 3. Process each small 1-itemset si_a in the set of SI^1 to determine whether its frequency satisfies the condition $\text{freq}(si_a) \ge \lfloor \lceil |D| \times \lambda \rceil \times (1 - \mu) \rfloor$, where $\text{freq}(si_a)$ is the occurrence frequency of the small 1-itemset si_a. Put the si_a that do not satisfy the condition into the set of SI^1_{tmp}.

Step 4. Calculate the maximal count (MAX_{HS}) of the sensitive itemsets hs_i in the set of HS as

$$\text{MAX}_{\text{HS}} = \max\left\{\text{freq}\left(\text{hs}_i\right),\ \forall \text{hs}_i, 1 \le i \le r\right\}, \tag{6}$$

where $\text{freq}(hs_i)$ is the occurrence frequency of the sensitive itemset hs_i in the set of HS.

Step 5. Calculate the HFD of each transaction T_k. Do the following substeps.

Substep 5.1. Calculate the HFD of each sensitive itemset hs_i within T_k as

$$\text{HFD}^k\left(\text{hs}_i\right) = \frac{\text{MAX}_{\text{HS}} - \text{freq}\left(\text{hs}_i\right) + 1}{\text{MAX}_{\text{HS}} - \lceil |D| \times \lambda \rceil + 1}. \tag{7}$$

Substep 5.2. Sum the HFDs of sensitive itemsets hs_i within T_k as

$$\text{HFD}^k = \frac{1}{\sum_{i=1}^{r} \text{HFD}^k\left(\text{hs}_i\right) + 1}. \tag{8}$$

Substep 5.3. Normalize the HFD^k for all transactions T_k in D'.

Step 6. Calculate the maximal count (MAX_{FI}) of the large itemsets fi_j in the set of FI as

$$\text{MAX}_{\text{FI}} = \max\left\{\text{freq}\left(\text{fi}_j\right),\ \forall \text{fi}_j, 1 \le j \le p\right\}. \tag{9}$$

Table 1: The notations used in the proposed HMAU algorithm are described below.

D	An original database, $D = \{T_1, T_2, \ldots, T_x, \ldots, T_n\}$, in which each T_x represents a transaction
D'	A projected database, $D' = \{T_1, T_2, \ldots, T_y, \ldots, T_k\}$, in which each T_y contains sensitive itemsets
D^*	A sanitized database, from which no sensitive information can be mined
λ	The minimum support threshold ratio
μ	The risky bound parameter, using which the itemsets within the boundary are used to evaluate the processed transactions
FI	The set of frequent itemsets FI = $\{\text{fi}_1, \text{fi}_2, \ldots, \text{fi}_j, \ldots, \text{fi}_p\}$ in D, in which each itemset fi_j is larger than or equal to the minimum support threshold
fi_j	A frequent itemset
SI^1	The set of infrequent 1-itemsets $\text{SI}^1 = \{\text{si}_1, \text{si}_2, \ldots, \text{si}_a, \ldots, \text{si}_q\}$ in D, in which each itemset si_a is below the minimum support threshold
si_a	A small (infrequent) 1-itemset
HS	A set of sensitive itemsets HS = $\{\text{hs}_1, \text{hs}_2, \ldots, \text{hs}_i, \ldots, \text{hs}_r\}$, in which each element represents an itemset that should be hidden in the original database
hs_i	A sensitive itemset
HS_{tmp}	The temporary set of sensitive itemsets outside the boundary
FI_{tmp}	The temporary set of large itemsets outside the boundary
SI^1_{tmp}	The temporary set of small 1-itemsets outside the boundary
HFD	The hiding failure dimension used to consider the side effects of hiding failures
MID	The missing itemset dimension used to consider the side effects of missing itemsets
AID	The artificial itemset dimension used to consider the side effects of artificial itemsets
$\text{HFD}^k(\text{hs}_i)$	The value of the sensitive itemset hs_i in transaction T_k
$\text{MID}^k(\text{fi}_j)$	The value of the large itemset fi_j in transaction T_k
$\text{AID}^k(\text{si}_a)$	The value of the small 1-itemset si_a in transaction T_k
MAX_{HS}	The maximal count of the sensitive itemsets in the set of HS
$\text{freq}(\text{hs}_i)$	The occurrence frequency of the sensitive itemset hs_i in the set of HS
MAX_{FI}	The maximal count of the large itemsets in the set of FI
$\text{freq}(fi_j)$	The occurrence frequency of the large itemset fi_j
MIN_{SI^1}	The minimal count of the small 1-itemsets in the set of SI^1
$\text{freq}(\text{si}_a)$	The occurrence frequency of the small 1-itemset si_a
w_b	The weights for HFD, MID, and AID, in which $0 < w_b \leq 1$
HMAU	The utility value used to determine whether the processed transactions should be deleted

Step 7. Calculate the MID of each transaction T_k. Do the following substeps.

Substep 7.1. Calculate the MID of each large itemset within T_k as

$$\text{MID}^k\left(\text{fi}_j\right) = \frac{\text{MAX}_{\text{FI}} - \text{freq}\left(\text{fi}_j\right) + 1}{\text{MAX}_{\text{FI}} - \lceil |D| \times \lambda \rceil + 1}. \tag{10}$$

Substep 7.2. Sum the MIDs of large itemsets fi_j within T_k as

$$\text{MID}^k = \sum_{j=1}^{p} \text{MID}^k\left(\text{fi}_j\right). \tag{11}$$

Substep 7.3. Normalize the MID^k for all transactions T_k in D'.

Step 8. Calculate the minimal count (MIN_{SI^1}) of the small 1-itemsets si_a in the set of SI^1 as

$$\text{MIN}_{\text{SI}^1} = \min\left\{\text{freq}\left(\text{si}_a\right), \ \forall \text{si}_a, 1 \leq a \leq q\right\}. \tag{12}$$

Step 9. Calculate the AID of each transaction T_k. Do the following substeps.

Substep 9.1. Calculate the AID of each small 1-itemset within T_k as

$$\text{AID}^k\left(\text{si}_a\right) = \frac{\text{freq}\left(\text{si}_a\right) - \text{MIN}_{\text{SI}^1} + 1}{\lceil |D| \times \lambda \rceil - \text{MIN}_{\text{SI}^1}}. \tag{13}$$

Substep 9.2. Sum the AIDs of small 1-itemsets si_a within T_k as

$$\text{AID}^k = \frac{1}{\sum_{a=1}^{q} \text{AID}^k\left(\text{si}_a\right) + 1}. \tag{14}$$

Table 2: Original database.

TID	Item
T_1	*a, b, c, e*
T_2	*e*
T_3	*b, c, e, f*
T_4	*d, f*
T_5	*a, b, d*
T_6	*b, c, e*
T_7	*a, b, c, d, e*
T_8	*a, b, e*
T_9	*c, e*
T_{10}	*a, b, c, e*

Substep 9.3. Normalize the AID^k for all transactions T_k in D'.

Step 10. Calculate the HMAU for HFD, MID, and AID of each transaction T_k as

$$\text{HMAU}^k = w_1 \times \text{HFD}^k + w_2 \times \text{MID}^k + w_3 \times \text{AID0} \quad (15)$$

where w_1, w_2, and w_3 are the predefined weights by users.

Step 11. Remove transaction T_k with $\min\{\text{HMAU}^k, \forall T_k, 1 \leq k \leq |D'|\}$ value.

Step 12. Update the minimum count ($= \lceil |D| \times \lambda \rceil$) of sanitized database.

Step 13. Update the occurrence frequencies of all sensitive itemsets in the sets of HS and HS_{tmp}. Put hs_i into the set of HS_{tmp} if $\text{freq}(\text{hs}_i) <$ minimum count ($= \lceil |D| \times \lambda \rceil$), and put hs_i into the set of HS otherwise.

Step 14. Update the occurrence frequencies of all large itemsets in the sets of FI and FI_{tmp}. Put fi_j into the set of FI_{tmp} if $\text{freq}(\text{fi}_j) <$ minimum count ($= \lceil |D| \times \lambda \rceil$), and put fi_j into the set of FI otherwise.

Step 15. Update the occurrence frequencies of all small 1-itemsets in the sets of SI^1 and SI^1_{tmp}. Put si_a into the set of SI^1_{tmp} if $\text{freq}(\text{si}_a) \geq$ minimum count ($= \lceil |D| \times \lambda \rceil$), and put si_ainto the set of SI^1 otherwise.

Step 16. Repeat Step 2 to Step 15 until the set of HS is empty ($|\text{HS}| = 0$).

4. An Illustrated Example

In this section, an example is used to illustrate the proposed algorithm step by step. Consider a database with 10 transactions (tuples) and 6 items (denoted as a to f) shown in Table 2. Each transaction can be considered a set of purchased items in a trade. The minimum support threshold is initially set at 40%, and the risky bound is set at 10%. A set of sensitive itemsets, $\text{HS} = \{be : 6, abe : 4\}$, is considered to be hidden by the sanitization process.

Based on an Apriori-like approach [3], the large (frequent) itemsets and small 1-itemsets are mined. The results are, respectively, shown in Tables 3 and 4.

The proposed algorithm then proceeds as follows to sanitize the database for hiding all sensitive itemsets in HS.

Step 1. The transactions in D are selected with any of the sensitive itemsets in HS. In this example, the transactions 1, 3, 6, 7, 8, and 10 are selected to form the database shown in Table 5.

Step 2. The frequent itemsets in FI are processed to check whether the condition is satisfied, which is calculated as $\text{freq}(\text{fi}_j) \leq \lceil \lceil 10 \times 0.4 \rceil \times (1 + 0.1) \rceil$ ($= \text{freq}(\text{fi}_j) \leq 5$). The itemsets $\{a, ab, ae, bc, bce\}$ satisfy the condition and are kept in FI; the remaining itemsets, $\{b, c, e, ce\}$, are put into the set of FI_{tmp}.

Step 3. The infrequent 1-itemsets in SI^1 are then processed to check whether the condition is satisfied, which is calculated as $\text{freq}(\text{si}_a) \geq \lfloor \lceil 10 \times 0.4 \rceil \times (1 - 0.1) \rfloor$ ($= \text{freq}(\text{si}_a) \geq 3$). The itemset $\{d\}$ satisfies the condition and is kept as SI^1; the other itemset, $\{f\}$, is put into the set of SI^1_{tmp}.

Step 4. The maximal count (MAX_{HS}) among the sensitive itemsets in the set of HS is then calculated. In this example, the maximal count of the sensitive itemsets $\{be\}$ and $\{abe\}$ is calculated as $\text{MAX}_{\text{HS}} = \max\{6, 4\} = 6$.

Step 5. The HFD of each transaction is calculated to evaluate the side effects of hiding failures of the processed transaction. In this example, transaction 7 is used to illustrate the following steps. According to formula (1), the HFD is calculated as $\text{HFD}^7(be) = (6 - 6 + 1)/(6 - 4 + 1) = 0.33$ and $\text{HFD}^7(abe) = (6 - 4 + 1)/(6 - 4 + 1) = 1$. The HFD of transaction 7 is calculated as $\text{HFD}^7 = 1/(0.33 + 1 + 1) = 0.43$. The other transactions are processed in the same way. The results are shown in Table 6.

The HFDs for all transactions are then normalized as shown in Table 7.

Step 6. The maximal count (MAX_{FI}) among the large itemsets in the set of FI is then calculated. In this example, the large itemsets are $\{a, ab, ae, bc, bce\}$, and the MAX_{FI} is calculated as $\text{MAX}_{\text{FI}} = \max\{5, 5, 4, 5, 5\}$ (=5).

Step 7. The MID of each transaction is calculated to evaluate the side effects of missing itemsets of the processed transaction. The frequent item $\{a\}$ in transaction 7 is used as an example to illustrate the steps. According to formula (2), the MID of the item $\{a\}$ is calculated as $\text{MID}^7(a) = (5 - 5 + 1)/(5 - 4 + 1) = 0.5$. The other frequent itemsets ab, ae, bc, and bce in transaction 7 are calculated in the same way, with $\text{MID}^7(ab) = 0.5$, $\text{MID}^7(ae) = 1$, $\text{MID}^7(bc) = 0.5$, and $\text{MID}^7(bce) = 0.5$. The MID of transaction 7 is then calculated as $\text{MID}^7 = 0.5 + 0.5 + 1 + 0.5 + 0.5$ (= 3). The other

Table 3: Large itemsets.

Large 1-itemset	Count	Large 2-itemset	Count	Large 3-itemset	Count
a	5	*ab*	5	*abe*	4
b	7	*ae*	4	*bce*	5
c	6	*bc*	5		
e	8	*be*	6		
		ce	6		

Table 4: Small 1-itemsets.

Small 1-itemset	Count
d	3
f	2

Table 5: Projected database D'.

TID	Item
T_1	*a, b, c, e*
T_3	*b, c, e, f*
T_6	*b, c, e*
T_7	*a, b, c, d, e*
T_8	*a, b, e*
T_{10}	*a, b, c, e*

Table 6: Hiding failure dimension for all transactions.

TID	HFD
T_1	0.43
T_3	0.75
T_6	0.75
T_7	0.43
T_8	0.43
T_{10}	0.43

Table 7: Normalization of HFDs for all transactions.

TID	HFD
T_1	0.57
T_3	1
T_6	1
T_7	0.57
T_8	0.57
T_{10}	0.57

Table 8: Missing itemset dimension for all transactions.

TID	MID
T_1	3
T_3	1
T_6	1
T_7	3
T_8	2
T_{10}	3

Table 9: Normalization of MIDs for all transactions.

TID	MID
T_1	1
T_3	0.33
T_6	0.33
T_7	1
T_8	0.67
T_{10}	1

transactions are processed in the same way. The results are shown in Table 8.

The MIDs for all transactions are then normalized as shown in Table 9.

Step 8. The minimal count (MIN_{SI^1}) among the small 1-itemsets in the set of SI^1 is then calculated. In this example, the small 1-itemset has only $\{d\}$, and the minimal count of the small 1-itemset is calculated as $\text{MIN}_{\text{SI}^1} = \min\{3\} = 3$.

Step 9. The AID of each transaction is calculated to evaluate the side effects of artificial itemsets of the processed transaction. Small 1-itemset $\{d\}$ in transaction 7 is used as an example to illustrate the steps. According to formula (3), the AID of the small 1-itemset $\{d\}$ is calculated as $\text{AID}^7(d) = (3 - 3 + 1)/(4 - 3) = 1$; since there is only one itemset in the set of SI^1, no other calculations are necessary. The AID of transaction 7 is calculated as $\text{AID}^7 = 1/(1 + 1) = 0.5$. The other transactions are processed in the same way. The results are shown in Table 10.

The AIDs for all transactions are then normalized as shown in Table 11.

Step 10. The three dimensions for evaluating the selected transactions are then organized as in Table 12. The weights of hiding failures, missing itemsets, and artificial itemsets are, respectively, set to 0.5, 0.4, and 0.1. Note that these values can be defined by users to decide the importance among the dimensions. In this example, the HMAU of transaction 7 is calculated as

$$\text{HMAU}^7 = 0.5 \times 0.57 + 0.4 \times 1 + 0.1 \times 0.5 \ (= 0.735). \quad (16)$$

The other transactions are processed in the same way. The results are shown in the last column of Table 12.

Step 11. The selected transactions in Table 12 are then evaluated to find a transaction with the minimal HMAU value.

Table 10: Artificial itemset dimension for all transactions.

TID	AID
T_1	1
T_3	1
T_6	1
T_7	0.5
T_8	1
T_{10}	1

Table 11: Normalization of AIDs for all transactions.

TID	AID
T_1	1
T_3	1
T_6	1
T_7	0.5
T_8	1
T_{10}	1

Table 12: Three dimensions of each transaction in projected database.

TID	HFD	MID	AID	HMAU
T_1	0.57	1	1	0.785
T_3	1	0.33	1	0.733
T_6	1	0.33	1	0.733
T_7	0.57	1	0.5	0.735
T_8	0.57	0.67	1	0.652
T_{10}	0.57	1	1	0.785

Table 13: Sanitized database.

TID	Item
T_2	e
T_4	d, f
T_5	a, b, d
T_7	a, b, c, d, e
T_9	c, e
T_{10}	a, b, c, e

Table 14: Large itemsets of the sanitized database.

Large 1-itemset	Count	Large 2-itemset	Count
a	3	ab	3
b	3	ce	3
c	3		
e	4		

In this example, transaction 8 has the minimal value and is directly removed from Table 12.

Step 12. Transaction 8 is deleted in the dataset in this example. The minimum count is updated as $\lceil |10 - 1| \times 0.4 \rceil$ (= 4).

Step 13. The occurrence frequencies of all sensitive itemsets in the sets of HS and HS_{tmp} are, respectively, updated. Since the original database with transaction 8 consisted of the sensitive itemsets $\{be, abe\}$, which was deleted in Step 11, the counts of $\{be, abe\}$ in the set of HS are, respectively, updated as $\{be\}$ (= 6 − 1) (= 5) and $\{abe\}$ (= 4 − 1) (= 3). In this example, the set of HS_{tmp} is empty, so there is nothing to be done in this step. After the updating process, the itemset $\{abe\}$ is put into the set of HS_{tmp} since its count is below the minimum count (3 < 4).

Step 14. The occurrence frequencies of all large itemsets in the sets of FI and FI_{tmp} are, respectively, updated. Since the original database with transaction 8 consisted of the large itemsets $\{a, b, e, ab, ae\}$, which was deleted in Step 11, the counts of $\{a, b, e, ab, ae\}$ in the set of FI and FI_{tmp} are, respectively, updated as $\{a\}$ (= 5 − 1) (= 4), $\{b\}$ (= 7 − 1) (= 6), $\{e\}$ (= 8 − 1) (= 7), $\{ab\}$ (= 5 − 1) (= 4), and $\{ae\}$ (= 4 − 1) (= 3). After the updating process, the itemset $\{ae\}$ is put into the set of FI_{tmp} since its count is below the minimum count (3 < 4).

Step 15. The occurrence frequencies of all small 1-itemsets in the sets of SI^1 and SI^1_{tmp} are, respectively, updated. Since the original database with transaction 8 did not consist of any of the small 1-itemsets in SI^1 and SI^1_{tmp}, nothing is done in this step.

Step 16. In this example, the sensitive itemset $\{abe\}$ is already hidden, but the occurrence frequency of sensitive itemset $\{be\}$ is larger than the minimum count. Steps 2 to 15 are repeated until the set of sensitive itemsets HS is empty ($|HS| = 0$). After all Steps are processed, the sanitized database is obtained as shown in Table 13.

Comparing the original database and the sanitized one, transactions 1, 3, 6, and 8 are removed from the original database, and the minimum count is updated as 3. The updated frequent itemsets of the sanitized database are shown in Table 14.

Comparing the large itemsets in Table 3, the sensitive itemsets $\{be\}$ and $\{abe\}$ are hidden and no artificial itemset is generated. Three itemsets, $\{ae, bc, bce\}$, are, however, missing itemsets of the sanitized database. In this example, the side effects of hiding failures, missing itemsets, and artificial itemsets are 0, 3, and 0, respectively.

5. Experimental Results

Experiments are conducted to show the performance of the proposed HMAU algorithm compared to that of the aggregate algorithm [23] for hiding sensitive itemsets through transaction deletion. The experiments were coded in C++ and performed on a personal computer with an Intel Core i7-2600 processor at 3.40 GHz and 4 GB of RAM running 64-bit Microsoft Windows 7. The real database BMS-WebView-1 [34] and a synthetic database (T7I7N200D20K) [35] from IBM data generator in which T symbolizes the average length of the transactions, I symbolizes the average maximum size

Table 15: Details of real and synthetic databases.

Dataset	Number of transactions	Number of items	Maximum transaction size	Average transaction size
BMS-WebView-1	59,602	497	267	2.5
T7I7N200D20K	15,351	200	26	8.7

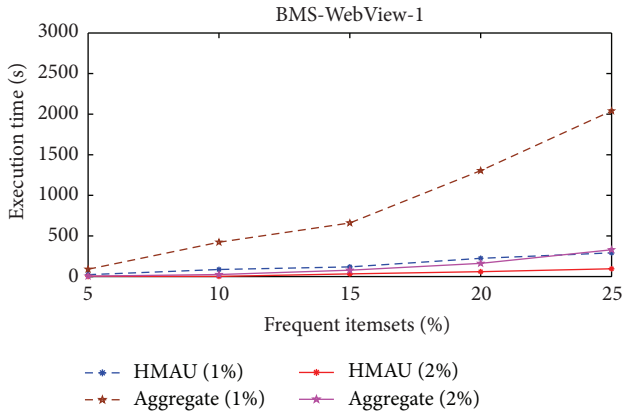

Figure 3: Comparison of execution time in BMS-WebView-1 database.

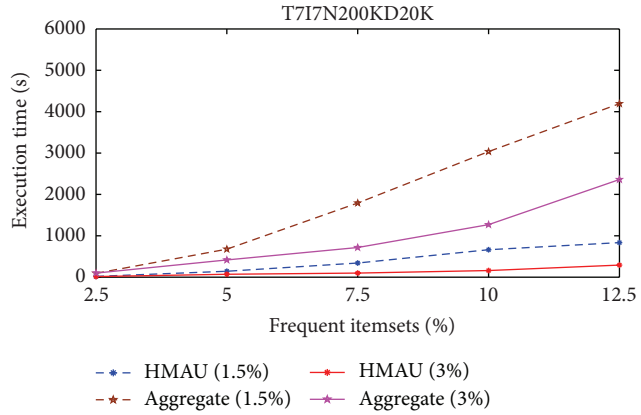

Figure 4: Comparison of execution time in T7I7N200D20K database.

of frequent itemsets, N symbolizes the number of differential items, and D symbolizes the size of database were used in the experiments. The details of the two databases are shown in Table 15.

For the BMS-WebView-1 database, the minimum support thresholds were, respectively, set at 1% and 2% to evaluate the performance of the proposed approach, and the percentages of sensitive itemsets were sequentially set from 5% to 25% of the number of frequent itemsets in 5% increments. In the experiments, the weights of HFD, MID, and AID in the proposed algorithm were, respectively, set at 0.5, 0.4, and 0.1.

For the T7I7N200D20K database, the minimum support thresholds were, respectively, set at 1.5% and 3%, and the percentages of sensitive itemsets were sequentially set at 2.5% to 12.5% of the number of frequent itemsets in 2.5% increments. In the experiments, the weights of HFD, MID, and AID in the proposed algorithm were, respectively, set at 0.5, 0.4, and 0.1.

5.1. Comparisons of Execution Time. Figure 3 shows the execution time of two algorithms in BMS-Web-View-1 database. Different minimum support thresholds of two algorithms are then compared in various sensitivity percentages of the frequent itemsets.

The execution time of the proposed HMAU algorithm is faster than those of the aggregate algorithm whether the minimum support threshold is set at 1% or 2%. Experiment is then conducted in T7I7N200D20K database and the results are shown in Figure 4.

From Figures 3 and 4, it is obvious to see that the proposed HMAU algorithm is faster than those of the aggregate method in two different databases.

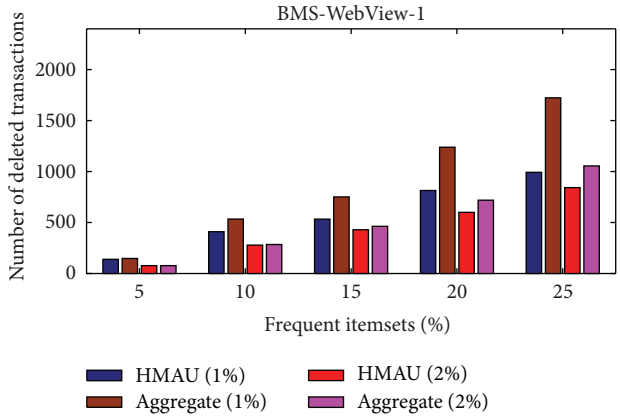

Figure 5: Comparison of number of deleted transactions in BMS-WebView-1 database.

5.2. Comparisons of Number of Deleted Transactions. Experiments were also conducted to evaluate the number of deleted transactions of the proposed algorithm in two different databases. For the BMS-WebView-1 database, the results are shown in Figure 5.

From Figure 5, it is obvious to see that the proposed HMAU algorithm deletes fewer transactions than the aggregate algorithm whether the minimum support threshold is set at 1% or 2% in BMS-WebView-1 database, thus achieving lower data distortion. For the T7I7N200D20K database, the results are shown in Figure 6.

From Figure 6, it is obvious to see that when the sensitive itemsets were set at 10% of the frequent itemsets with 1.5% minimum support threshold in T7I7N200D20K database, the proposed HMAU algorithm produced more transactions to be deleted for hiding sensitive itemsets. Since the proposed

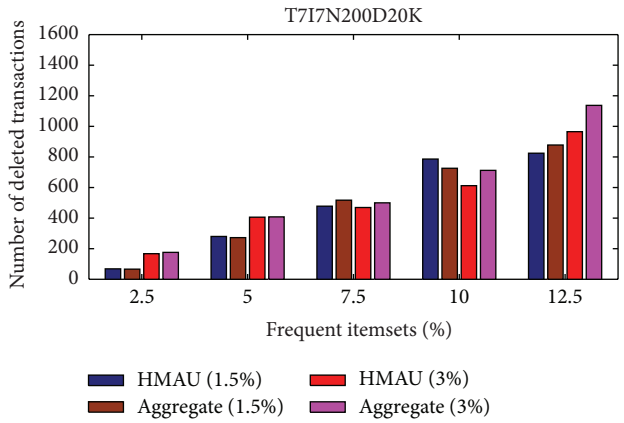

Figure 6: Comparison of number of deleted transactions in T7I7N200D20K database.

HMAU algorithm considers the three dimensions together, the selected transactions for deletion may consist of fewer large transactions rather than many sensitive itemsets.

5.3. Comparisons of Side Effects. Three side effects are then compared to show the performance of the proposed algorithm in two different databases.

The side effects of hiding failures, missing itemsets, and artificial itemsets are, respectively, symbolized as α, β, and γ. In Table 16, it can be seen that when the minimum support threshold was set at 1%, the proposed HMAU algorithm produces no side effects whereas the aggregate algorithm produces some artificial itemsets since the criteria of artificial itemsets are not considered in aggregate algorithm. Both the two algorithms produce no side effects when the minimum support threshold was set at 2%. The results to evaluate the side effects of the proposed HMAU algorithm in T7I7N200D20K database are shown in Table 17.

From Table 17, it is obvious to see that when the minimum support threshold was set at 1.5%, the proposed HMAU algorithm produces fewer artificial itemsets and missing itemsets than the aggregate algorithm for various sensitivity percentages of the frequent itemsets. The proposed HMAU algorithm produces no side effects at 3% minimum support threshold whereas the aggregate algorithm produces some artificial itemsets.

To summarize the above results for BMS-WebView-1 and T7I7N200D20K databases, the proposed HMAU algorithm outperforms the aggregate algorithm in terms of the execution time, the number of deleted transactions, and the number of side effects.

6. Conclusion and Future Works

In this paper, the HMAU algorithm is proposed for hiding sensitive itemsets in data sanitization process by reducing the side effects through transaction deletion. The formulas of three dimensions as HFD, MID, and AID are defined to evaluate the correlation between the processed transactions and side effects. The weights of three evaluation dimensions of HFD, MID, and AID can be set by users' interests. In the experiments, both the real dataset and synthetic dataset are used to, respectively, evaluate the performances of the two proposed algorithms. Experimental results showed that the proposed HMAU algorithm outperforms the aggregate algorithm in terms of execution time, number of deleted transactions, and number of side effects.

In the future, the sensitive itemsets to be hidden can be extended to the sensitive association rules to be hidden. More considerations are necessary to be concerned to decrease not only the supports of sensitive itemsets but also the confidence of sensitive association rules. Other distortion approaches such as the noise addition and data modification are also the important issues to hide the sensitive information in PPDM.

Table 16: Comparison of side effects in BMS-WebView-1 database.

Sensitive percentage of FIs (minimum support threshold)	HMAU			Aggregate		
	α	β	γ	α	β	γ
5% (1%)	0	0	0	0	0	0
10% (1%)	0	0	0	0	0	1
15% (1%)	0	0	0	0	0	1
20% (1%)	0	0	0	0	0	3
25% (1%)	0	0	0	0	0	2
5% (2%)	0	0	0	0	0	0
10% (2%)	0	0	0	0	0	0
15% (2%)	0	0	0	0	0	0
20% (2%)	0	0	0	0	0	0
25% (2%)	0	0	0	0	0	0

Table 17: Comparison of side effects in T7I7N200D20K database.

Sensitive percentage of FIs (Minimum support threshold)	HMAU			Aggregate		
	α	β	γ	α	β	γ
2.5% (1.5%)	0	0	1	0	1	2
5% (1.5%)	0	0	0	0	3	4
7.5% (1.5%)	0	0	3	0	3	7
10% (1.5%)	0	0	0	0	2	6
12.5% (1.5%)	0	0	1	0	3	6
2.5% (3%)	0	0	0	0	0	0
5% (3%)	0	0	0	0	0	2
7.5% (3%)	0	0	0	0	0	1
10% (3%)	0	0	0	0	0	1
12.5% (3%)	0	0	0	0	0	2

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Science Council of the Republic of China under Contract no. NSC-102-2923-E-390-001-MY3, and by the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology under Grant HIT.NSRIF.2014100.

References

[1] R. Agrawal, T. Imielinski, and A. Sawmi, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993, Proceedings of the ACM SIGMOD International Conference on Management of Data.

[2] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: a performance perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914–925, 1993.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 487–499, June1994.

[4] T. Hong, C. Lin, and Y. Wu, "Incrementally fast updated frequent pattern trees," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2424–2435, 2008.

[5] C. W. Lin, T. P. Hong, and W. H. Lu, "The Pre-FUFP algorithm for incremental mining," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9498–9505, 2009.

[6] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the IEEE 11th International Conference on Data Engineering*, pp. 3–14, March 1995.

[7] H. Cheng, X. Yan, and J. Han, "IncSpan: incremental mining of sequential patterns in large database," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 527–532, August 2004.

[8] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology*, pp. 3–17, 1996.

[9] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," in *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, 2007.

[10] J. R. Quinlan, *C4. 5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[11] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, pp. 25–71, Springer, 2006.

[12] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. 22, no. 11, pp. 1025–1034, 1973.

[13] Y. Liu, W. K. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Advances in Knowledge Discovery and Data Mining*, vol. 3518 of *Lecture Notes in Computer Science*, pp. 689–695, Springer, 2005.

[14] C. Lin, T. Hong, and W. Lu, "An effective tree structure for mining high utility itemsets," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7419–7424, 2011.

[15] G. Lan, T. Hong, and V. S. Tseng, "Discovery of high utility itemsets from on-shelf time periods of products," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5851–5857, 2011.

[16] C. Lin, G. Lan, and T. Hong, "An incremental mining algorithm for high utility itemsets," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7173–7180, 2012.

[17] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000.

[18] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[19] 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*, pp. 45–52, July 1999.

[20] C. W. Lin, T. P. Hong, C. C. Chang, and S. L. Wang, "A greedy-based approach for hiding sensitive itemsets by transaction insertion," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 4, pp. 201–227, 2013.

[21] Y. Wu, C. Chiang, and A. L. P. Chen, "Hiding sensitive association rules with limited side effects," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 29–42, 2007.

[22] D. E. O. Leary, "Knowledge discovery as a threat to database security," in *Knowledge Discovery in Databases*, pp. 507–516, AAAI/MIT Press, 1991.

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

[24] S. R. M. Oliveira and O. R. Zaïane, "Protecting sensitive knowledge by data sanitization," in *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 613–616, November 2003.

[25] T. Hong, C. Lin, K. Yang, and S. Wang, "A lattice-based data sanitization approach," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 2325–2329, October 2011.

[26] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding association rules by using confidence and support," in *Proceedings of the 2001 International Workshop on Information Hiding*, pp. 369–383, 2001.

[27] K. Duraiswamy, D. Manjula, and N. Maheswari, "Advanced approach in sensitive rule hiding," *CCSE Modern Applied Science*, vol. 3, no. 2, pp. 98–107, 2009.

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

[29] S. P. Patil and T. M. Patewar, "A novel approach for efficient mining and hiding of sensitive association rule," in *Proceedings of the 2012 Nirma University International Conference on Engineering*, pp. 1–6, 2012.

[30] C. Wu, Y. Huang, and J. Chen, "Privacy preserving association rules by using greedy approach," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, vol. 4, pp. 61–65, April 2009.

[31] T. Hong, C. Lin, C. Chang, and S. 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.

[32] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, "Using TF-IDF to hide sensitive itemsets," *Applied Intelligence*, vol. 38, no. 4, pp. 502–510, 2013.

[33] B. Dai and L. Chiang, "Hiding frequent patterns in the updated database," in *Proceedings of the International Conference in Information Science and Applications (ICISA '10)*, pp. 1–8, April 2010.

[34] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 401–406, August 2001.

[35] IBM Quest Data Mining Project, "Quest synthetic data generation code," http://www.almaden.ibm.com/cs/quest/syndata.html.