Journal of Information Hiding and Multimedia Signal Processing ©2013 ISSN 2073-4212
Ubiquitous International Volume **4**, Number **4**, October **2013**

A Greedy-based Approach for Hiding Sensitive Itemsets by Transaction Insertion

Chun-Wei Lin[1], Tzung-Pei Hong[2,4], Chia-Ching Chang[2], and Shyue-Liang Wang[3]

[1]Innovative Information Industry Research Center (IIIRC)
School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus Shenzhen University Town Xili, Shenzhen 518055 P.R. China

[2]Department of Computer Science and Information Engineering

[3]Department of Information Management
National University of Kaohsiung
Kaohsiung, 811, Taiwan, R.O.C.

[4]Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, 804, Taiwan, R.O.C.
Corresponding author

jerrylin@ieee.org, tphong@nuk.edu.tw, snoopy.smile@msa.hinet.net,
slwang@nuk.edu.tw
Received March, 2013; revised April, 2013

Abstract. *Data mining technology is designed to derive useful knowledge from large database, which is used to aid decision making. The process of data collection and dissemination may, however, causes privacy concerns. Sensitive or personal information and knowledge of individuals, industries and organizations must be kept private before they are publicly shared or published. Thus, privacy-preserving data mining (PPDM) has become an important issue to efficiently hide sensitive information. In this paper, a greedy-based approach is proposed to hide sensitive itemsets by transaction insertion. The proposed approach first computes the maximal number of transactions to be inserted into the original database for totally hiding sensitive itemsets. The fake items of the transactions to be inserted are thus designed by the statistical approach, which can greatly reduce side effects in PPDM. Experiments are also conducted to evaluate the performance of the proposed approach.*
Keywords: Privacy preservation, data mining, greedy approach, data sanitization, transaction insertion

1. **Introduction.** Privacy-preserving data mining (PPDM) [1, 2, 6, 7, 11, 12, 16, 22] has become an important issue due to the rapid proliferation of electronic data in governments, industries, and organizations. Secure data may implicitly contain confidential information and lead to privacy concerns if the provided information is misused. Confidential information includes income, medical history, address, credit card numbers, phone number, and purchasing behavior. Some shard information among companies may be extracted and analyzed by other partners, which may not only decrease the benefits of the companies but also cause threats to sensitive data. This has led to increasing concerns

about the privacy of the underlying data and the implicit knowledge contained in the data.

Many approaches have been proposed in PPDM, but most of the approaches hide the sensitive information by deleting transactions or itemsets [16, 22]. In real-world applications, however, deleting transactions from the original database may greatly affect decision-making, thus causing the serious damage especially in medical decision. In this paper, a greedy-based approach for inserting new transactions into the original database is thus proposed. The empirical rules in standard normal distribution is respectively applied to determine the number of newly inserted transactions, the lengths of the inserted transactions, and the itemsets to be added into the inserted transactions. Several experiments are conducted to evaluate the performance in terms of execution time, the number of inserted transactions required for hiding sensitive information, and the side effects of the proposed algorithm.

The rest of this paper is organized as follows. Some related works are described in Section 2. The proposed greedy-based algorithm is explained in Section 3. An example is given in Section 4. Experimental results are shown in Section 5. The conclusion and future work are given in Section 6.

2. **Review of Related Work.**

In this section, studies concerning data mining approaches and data sanitization are briefly reviewed.

2.1. **Data Mining Approach.**

Data mining [3-5, 9, 10, 14, 15, 17-18, 20] is the most commonly used in attempts to induce potential information from transaction data. The most common data mining approach is association-rule mining that indicates the presence of certain items in a transaction will imply the presence of some other items. To achieve this purpose, the Apriori algorithm [4-5] and the FP-growth algorithm [13] are recommended as the efficient approaches to derive frequent itemsets in association-rule mining. The former one is a level-wise approach to generate-and-test candidates and the later one uses a tree structure to keep the frequent itemsets without candidate generation, thus reducing the computational cost of rescanning database. In Apriori algorithm, the database is firstly scanned to find the frequencies of items. An item is then considered as a large (frequent) item since its count (frequency) is larger than or equal to the minimum count threshold. Next, the candidate itemsets obtained two items are then formed from the large items in combination process. The generated candidate itemsets are then determined to check the counts of the 2-itemsets larger than or equal to the minimum count threshold. This process was repeated until all large itemsets had been found. Association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than the minimum confidence were output as the desired association rules.

2.2. **Data Sanitization.**

Data mining techniques can pose security problems and lead to privacy concerns. PPDM techniques have thus become a critical research issue for hiding confidential or secure information. The goal in PPDM is to hide the sensitive itemsets with the minimal side effects. The relationship of itemsets before and after the PPDM process can be depicted in Figure 1, where L represents the large itemsets of D, S represents the sensitive itemsets defined by users that are large, S represents the non-sensitive itemsets that are large, and L' is the large itemsets after some records are inserted.

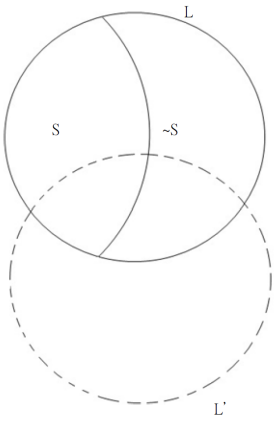

FIGURE 1. The relationship of itemsets before and after the PPDM approach is processed

Let α be the number of sensitive itemsets that fail to be hidden. Thus, the sensitive itemsets still appear after the sanitization process. The sensitive itemsets should ideally become zero after the PPDM. The set of sensitive itemsets can be shown in Figure 2, in which the α part is the interaction of S and L'.

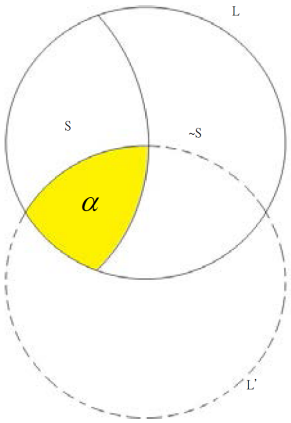

FIGURE 2. The set of sensitive itemsets that fail to be hidden

Similarly, let β be the number of missing itemsets for another criteria in evaluation process. A missing itemset is a non-sensitive large itemset in the original database, but is not derived from the sanitized database. This side effect of β is shown in Figure 3, in which β is the difference of S and L'.

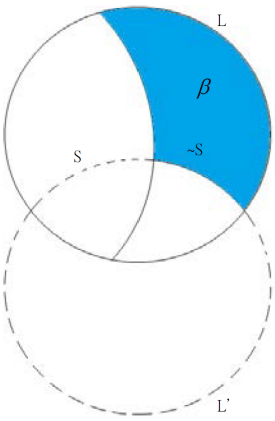

FIGURE 3. The set of missing itemsets

The γ is then defined as the last criteria in evaluation process as the number of artificial itemsets. An artificial itemset is a new large itemset appearing in the sanitized database but not in the original database. This side effect of γ is shown in Figure 4, in which γ is the difference of L' and L.

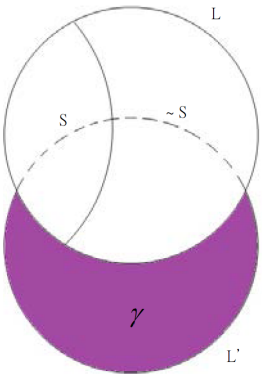

FIGURE 4. The set of artificial itemsets

From Figures 2 to 4, it is obvious to see that $\alpha = S \cap L'$, $\beta = S - L' = (L - S) - L'$, and $\gamma = L' - L$. Thus, the PPDM is used to minimize α, β, and γ by data sanitization for hiding the sensitive itemsets.

Atallah et al. proposed a protection algorithm for data sanitization to avoid inferring of association rules [8]. It uses both addition and deletion procedures to modify databases for hiding sensitive information. Dasseni et al. proposed a hiding algorithm based on the hamming-distance approach to reduce the confidence or support values of association rules [11]. Three heuristic hiding approaches were proposed to respectively increase the supports of antecedent parts, to decrease the supports of consequent parts, and to decrease the support of either the antecedent or the consequent parts. When the supports or the confidences of sensitive association rules are below a minimum support threshold, the sensitive association rules are hidden. Oliveira and Zaïane [19] introduced multiple-rule hiding approach to efficiently hide sensitive itemsets. The database must be scanned twice regardless the number of sensitive itemsets. In the first database scan, an index file is created to efficiently find sensitive itemsets within transactions. Three algorithms are then used in the second database scan to remove minimal individual items. Amiri proposed three heuristic algorithms to hide multiple sensitive rules [7]. The first approach computes

the union of the supporting transactions for all sensitive itemsets to remove the transaction that supports the most sensitive and the least non-sensitive itemsets. The second one aims to remove individual items from transactions instead of removing whole transactions. The third approach combines the previous two approaches to identify sensitive transactions and to selectively delete items from these transactions until the sensitive knowledge has been hidden. Pontikakis et al. [21] proposed two heuristic approaches based on data distortion. The priority-based distortion algorithm (PDA) was designed to reduce the confidences of sensitive rules by decreasing consequent items. The weight-based sorting distortion algorithm (WDA) was then proposed to prioritize the selection of sanitized transactions. The priority values are used to weight the transactions based on effective data structures.

3. **Proposed Greedy-based Algorithm for Transaction Insertion.** In the problem of PPDM, some basic concepts are borrowed from association rule mining. It is thus necessary to review association rule mining before exploring the issues of PPDM. The most commonly used algorithm for association rule mining is the Apriori algorithm proposed by Agrawal et al. [5]. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D be a set of transactions, where each transaction $T \in D$ consists of a set of items, such that $T \subseteq I$ Each transaction T has a unique identifier, called its TID. A set of items $X \subset I$ is called an itemset. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. Usually, Y consists of only a single item.

An association rule $X \Rightarrow Y$ holds in a database D if the following two factors are satisfied. The first one is the support condition, which is defined as at least $s\%$ of the transactions in D that contain $X \cup Y$ It can be thought of as a measure of the frequency of a rule, and is expressed as $\frac{|X \cup Y|}{N} \geq s$, where N is the number of transactions in D. The second factor is the confidence condition, which is defined as at least $c\%$ of transactions with the itemset X that contain Y. It is a measure of the strength of the rule, and is expressed as $\frac{|X \cup Y|}{|X|} \geq c$.

In PPDM, a sensitive itemset $H = \{h_1, h_2, \ldots, h_i\}$ is normally defined by users. Sensitive itemsets that belong to frequent itemsets may consist of confidential information. In this paper, sensitive itemsets are hidden by adding the number of new transactions to increase the minimum count threshold of D. Let the modified database be denoted as D'. Thus, each sensitive itemset will have insufficient support to be frequent in D'. In addition to hiding the sensitive itemsets to prevent them from being mined, there are other goals when the original database is sanitized. For example, all non-sensitive rules should be mined from the sanitized database D'. Rules that are not found in the original database D should not be generated from the sanitized database D'.

In this paper, the sensitive itemsets are then hidden by adding newly transactions into the original database, thus increasing the minimum count threshold to achieve the goal. It is, however, three factors should be taken as the consideration. First, the number of transactions should be serious determined for achieving the minimal side effects to totally hide the sensitive itemsets. In this part, sensitive itemsets are then respectively evaluated to find the maximal number of transactions to be inserted. Second, the length of each newly inserted transaction is then calculated according to the empirical rules in standard normal distribution. Last, the already existing large itemsets are then alternatively added into the newly inserted transactions according to the lengths of transactions which determined at the second procedure. This step is to avoid the missing failure of the large itemsets for reducing the side effects in the PPDM.

A greedy-based approach for data sanitization proposed in this paper consisted of three steps to insert new transactions into original database for hiding sensitive itemsets. In

the first step, the safety bound for each sensitive itemset is then calculated to determine how many transactions should be inserted. Among the calculated safety bound of each sensitive itemset, the maximum operation is then used to get the maximal numbers of inserted transactions. Next, the lengths of inserted transactions are then evaluated through empirical rules in statistics as the standard normal distribution. In the third step, the count difference is then calculated between the sensitive itemsets and non-sensitive frequent itemsets at each k-level (k-itemset). The non-sensitive frequent itemsets are then inserted into the transaction in descending order of their count difference. This property remains that the original frequent itemsets would be still frequent after the numbers of transactions are inserted for hiding sensitive itemsets. The above steps are then repeatedly progressed until all sensitive itemsets are hidden. The proposed algorithm is then shown below.

Proposed algorithm:

INPUT: A transaction dataset $D = \{T_1, T_2, \dots T_m\}$, a set of k frequent itemsets $FI = \{fi, fi_2, \dots fi_k\}$, a set of l infrequent (small) itemsets $I = \{i_1, i_2, \dots i_l\}$, a user-specified minimum support threshold α, and a set of user-specified sensitive itemsets $S = \{si_1, si_2, \dots , si_j, \dots , si_p\}$.

OUTPUT: A sanitized dataset.

STEP 1: Calculate the value of *maximum safety bound* (MSB) for the number of newly inserted transactions as:

$$n = \max_{i=1}^{p}(SB_i) = \left[\frac{|si_i|}{\alpha} - m\right] + 1$$

where SB_i is the *safety bound* of each sensitive itemset, m is the number of original transactions in D, $|si_i|$ is the count of sensitive itemset si_i.

STEP 2: Calculate the length p_n of each inserted transaction in d according to the empirical rules in standard normal distribution, where $d = \{d_1, d_d, \dots , d_n\}$, and n is the number of inserted transactions obtained in STEP 1.

STEP 3: Choose the itemsets to be inserted into each inserted transaction d_n. Do the substeps as follows.

Substep 3-1: Calculate the *count difference* (CD_{fi_k}) of each frequent itemset to be possibly inserted into the new transactions as:

$$CD_{fi_k} = [|fi_k| - (m + n)\alpha]$$

where $|fi_k|$ is the count (frequency) of an item fi_k, m is the number of transactions in D and n is the number of transactions in d.

Substep 3-2: Put the frequent itemsets fi_k with negative CD_{fi_k} into the set of *Insert_Items*.

Substep 3-3: Sort the fi_k in the set of *Insert_Items* in descending order of their lengths.

Substep 3-4: Sort the sorted results obtained in substep 3-3 in descending order of their $|CD_{fi_k}|$.

STEP 4: Process the inserted transactions d_n one-by-one respectively to add the fi_k in the set of *Insert_Items* according to the sorted order obtained in substep 3-4. Note that

the length of an inserted itemset fi_k is no longer than p_n in d_n and the inserted itemset fi_k in a transaction cannot be formed as any super-itemsets of a sensitive itemset in S.

STEP 5: Update (decrease) the value $|CD_{fi_k}|$ and the corresponding sub-itemsets of the processed itemset fi_k by 1.

STEP 6: Repeat the STEPs 4 to 5 until the set of *Insert_Items* is *null* or there is no longer itemsets to be inserted into d_n obtained the constraints in STEP 4.

STEP 7: Add the small items in the set of I into the d_n while d_n remains positions to be added according to empirical rules in standard normal distribution.

4. **An Example.** In this section, an example is then used to illustrate the proposed algorithm step-by-step. Assume a database shown in Table 1 is used as an example. It consists of 8 transactions with 7 items, denoted a to g.

TABLE 1. A database with 8 transactions

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

Assume a set of the user-specified sensitive itemsets S is {*c*:6, *be*:5, *abc*:4}, and the minimum support threshold is set at 50%. The minimum count of this example is calculated as $(0.5 \times 8)(= 4)$. The Apriori approach [3] is then executed to find all frequent itemsets from Table 1 and the results are then shown in Table 2, respectively. Note the sensitive itemsets are marked in red color in Table 2.

TABLE 2. All large itemsets

Large 1-itemset		**Large 2-itemset**		**Large 3-itemset**	
Item	***Count***	***Item***	***Count***	***Item***	***Count***
a	4	*ab*	4	*abc*	4
b	7	*ac*	4	*bce*	4
c	6	*bc*	5		
e	6	*be*	5		
		ce	5		

STEP 1: The number of inserted transactions in this example is first calculated by the proposed approach. In this example, three sensitive itemsets are defined to be hidden. Thus, the *safety bound* for sensitive itemset {*c*} is calculated as $([(6/0.5) - 8] + 1)(=$

5). The *safety bound* for sensitive itemset *be* and *abc* are respectively calculated as $([(5/0.5) - 8] + 1)(= 3)$ and $([(4/0.5) - 8] + 1)(= 1)$. The *maximum safety bound* (MSB) among three sensitive itemsets is thus $max(5, 3, 1)(= 5)$. Thus, the number of newly inserted transactions is initially set at 5.

STEP 2: For five inserted transactions obtained in STEP 1, the length of each transaction is thus computed according to empirical rules in standard normal distribution. In this example, the average length of 8 transactions is calculated as $(5 + 4 + 2 + 4 + 2 + 4 + 4 + 4)/8(= 3.625)$. The standard deviation is then calculated as $\sqrt{\frac{1}{8-1}\left[(5 - 3.625)^2 + (4 - 3.625)^2 + \cdots + (4 - 3.625)^2\right]}(= 1.06)$. The length of transactions in the original database is standardized and shown in Table 3.

TABLE 3. The standardized values of transaction lengths

Length	*Standardized*
2	-1.53
3	-0.59
4	0.35
5	1.3

That is, the probability of length 2, 3, 4, and 5 are calculated as (13.5%, 34%, 34%, 13.5%) shown in Figure 5.

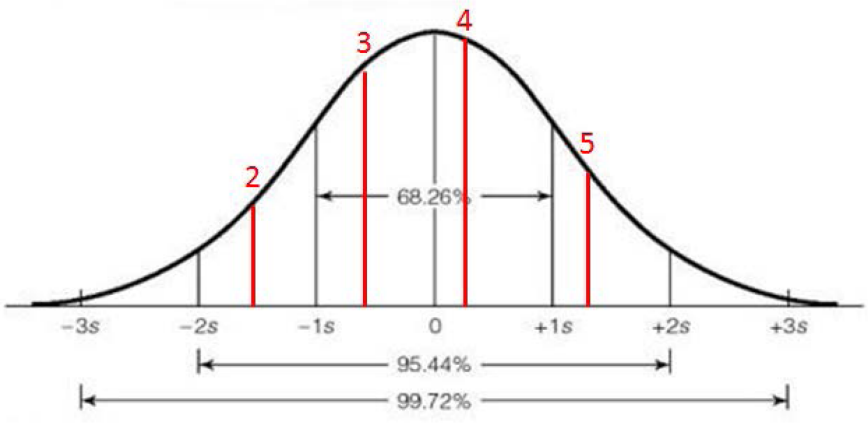

FIGURE 5. The probabilities of different lengths in standard normal distribution

The lengths of five inserted transactions are then assigned according to Figure 5. Thus, the lengths of *TID* 9 to 13 are {4, 3, 4, 2, 3}, respectively.

STEP 3: The *count difference* of each frequent itemset in Table 2 is then respectively calculated. Take item $\{a\}$ as an example to illustrate the step. The count of item $\{a\}$ in the original database is 4. The updated minimum count of item $\{a\}$ in the updated database is then calculated as $(8 + 5) \times 0.5(= 6.5)$. The *count difference* CD_a is thus $[(4 - 6.5)] (= -3)$. The *count differences* of other frequent itemsets are then shown in Table 4.

In Table 4, only the itemsets with negative CD will be considered as the itemsets for insertion. In this example, itemsets $\{a : -3, ab : -3, ac : -3, bc : -2, ce : -2, bce : -3\}$

TABLE 4. The count differences of all frequent itemsets

Large 1-itemset		**Large 2-itemset**		**Large 3-itemset**	
Item	***CD***	***Item***	***CD***	***Item***	***CD***
a	-3	*ab*	-3	*bce*	-3
b	0	*ac*	-3		
e	-1	*bc*	-2		
		ce	-2		

satisfy the condition and are then sorted according to their lengths and $|CD|$ value. After that, the sorted results are then put into the set of *Insert_Items* $= \{bce : 3, ab : 3, ac : 3, bc : 2, ce : 2, a : 3\}$.

STEP 4,5 & 6: The itemsets are then respectively added into the transactions 9 to 13 according the sorted order in the set of *Insert_Items*. For example, the first itemset in *Insert_Items* is $\{bce : 3\}$, indicating the itemset $\{bce\}$ can thus be added into three different inserted transactions by 3 times. The results are then shown in Table 5.

TABLE 5. The process to add an itemset $\{bce\}$

9	*b c e* ○
10	*b c e*
11	*b c e* ○
12	○ ○
13	○ ○ ○

Insert_Items	
bce:3	3 times to be added
ab:3	3 times to be added
ac:3	3 times to be added
bc:2	2 times to be added
ce:2	2 times to be added
a:3	3 times to be added

After an itemset $\{bce\}$ is respectively inserted into transaction 9, 10, and 11, the count of $\{bce\}$ in the *Insert_Items* becomes 0. The corresponding sub-itemsets $\{bc, ce\}$ are then also updated (decreased) as 0. After that, the *Insert_Items* $= \{ab : 3, ac : 3, a : 3\}$. The itemset $\{ab\}$ is then respectively inserted into the transactions 12 and 13. The results are then shown in Table 6.

TABLE 6. The process to add the itemset $\{ab\}$

9	*b c e* ○
10	*b c e*
11	*b c e* ○
12	*a b*
13	*a b* ○

Insert_Items	
ab:1	1 times to be added
ac:3	3 times to be added
a:3	3 times to be added

Since there is only one position in transactions 9, 11, and 13, there is no more spaces for 2-itemsets $\{ab\}$ and $\{ac\}$. Besides, an itemset $\{a\}$ cannot be added into the transactions

9 and 11 due to those two transactions will produce the super-itemsets $\{abce\}$ of the sensitive itemsets $\{abc\}$. Thus, the greedy procedure is terminated.

STEP 7: In Table 2, the small items are $\{d:3, f:2, g:1\}$ in the original database. Since transactions 9, 11, and 13 still remain one position for insertion, the small items are alternative selected by empirical rules in stand normal distribution. In this example, items $\{d\}$, $\{f\}$, $\{g\}$ are respectively added into 3 different transactions. The results are shown in Table 7.

TABLE 7. The process to add the small items

9	*b c e g*
10	*b c e*
11	*b c e f*
12	*a b*
13	*a b d*

Small items	
Item	**Count**
d	3 (+1)
f	2 (+1)
g	1 (+1)

That is, the final updated database is shown in Table 8.

TABLE 8. The process to add the small items

TID	Item
1	*a, b, c, d, e*
2	*a, b, c, e*
3	*c, e*
4	*a, b, c, e*
5	*b, g*
6	*b, d, e, f*
7	*a, b, c, d*
8	*b, c, e, f*
9	*b, c, e, g*
10	*b, c, e*
11	*b, c, e, f*
12	*a, b*
13	*a, b, g*

5. **Experimental Results.** Experiments were made to show the performance of the proposed approach. They were performed on an Intel Core2 CPU with 2GB RAM based on the Windows 7 with 64 bit platform. The details of the three databases used in the experiments were shown in Table 9.

TABLE 9. The details of the two databases

Database	# of Transactions	# of Items	Maximum Transaction Size	Average Transaction Size
BMS-Webview-1	59,602	497	267	2.5
BMS-Webview-2	77,512	3,340	161	5.0

In the experiments, the minimum support thresholds were set at 3% and 2% for the BMS-WebView-1 database and the BMS-WebView-2 [23], respectively. The numbers of sensitive itemsets are then defined by the percentages of the frequent itemsets in the databases, which is more flexible to see the performance of the proposed algorithm.

For the proposed greedy-based algorithm, the relationships between the numbers of inserted transactions, the execution, and the side effects are then compared in three different databases. The numbers of newly inserted transactions are then computed for two different databases show in Figure 6.

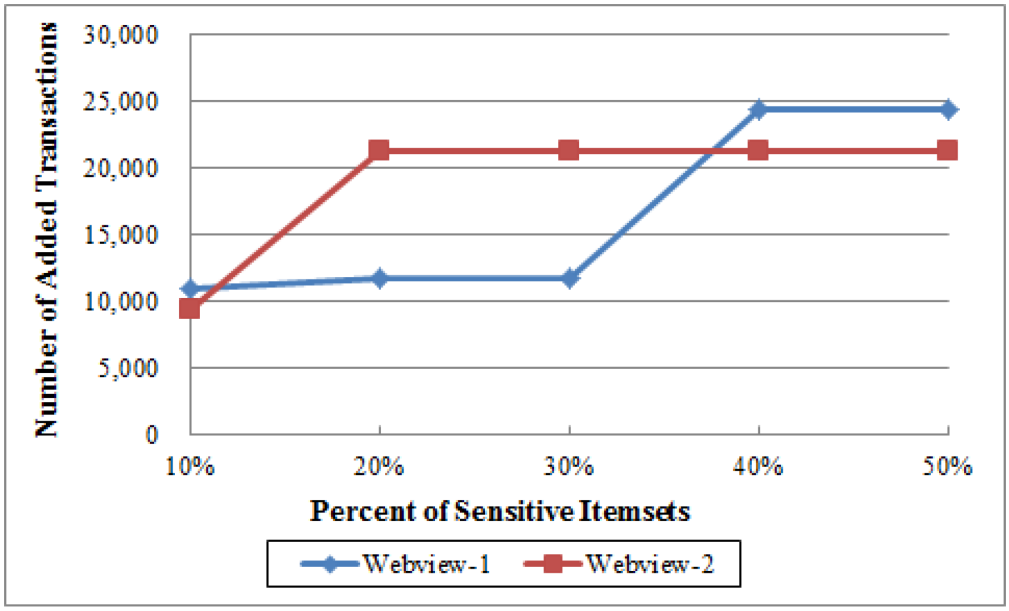

FIGURE 6. Numbers of added transactions among two databases in different percentages of sensitive itemsets

Besides, the execution times are compared among two different databases in different percentages of sensitive itemsets. The results are then shown in Figure 7.

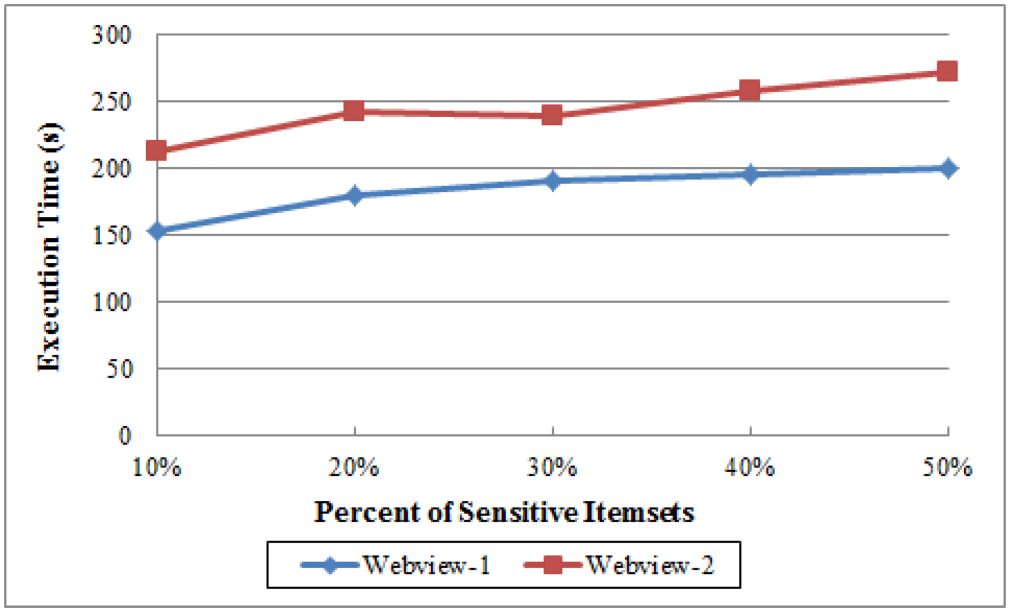

FIGURE 7. Execution times among two databases in different percentages of sensitive itemsets

The side effects of the proposed greedy-based approach are also evaluated, including the hiding failure for the number of sensitive itemsets, the number of the missing non-sensitive itemsets, and the number of artificial itemsets. The number of side effects for databases Webview-1 and Webview-2 are then evaluated to show the performance in different percentage of sensitive itemsets. The results are then respectively shown in Figure 8 and Figure 9. From Figure 8 and Figure 9, it is obvious to see that the proposed greedy-based approach can thus totally hide the sensitive itemsets without any side effects of hiding failure.

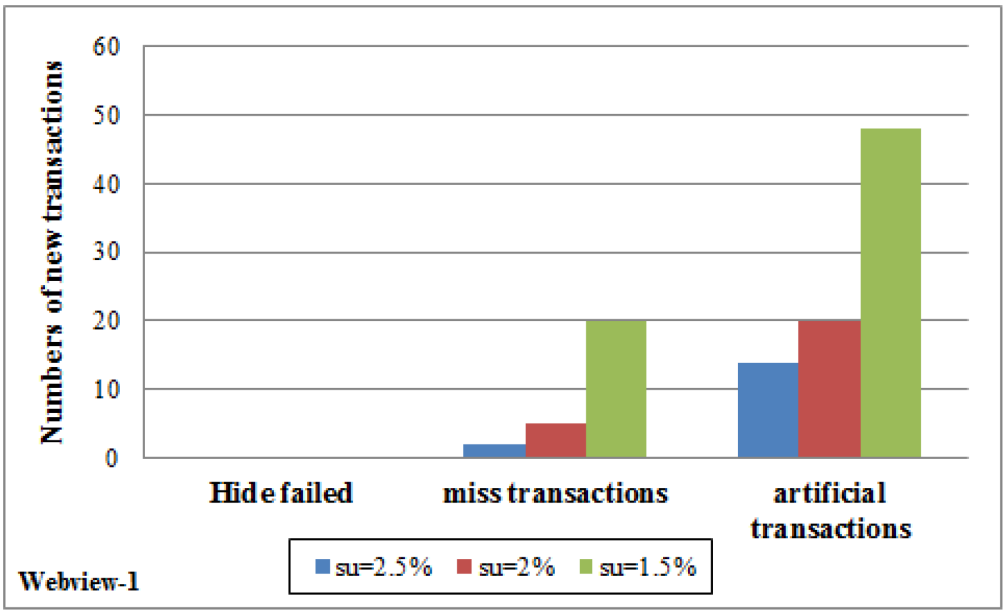

FIGURE 8. The evaluation of three side effects in Webview-1

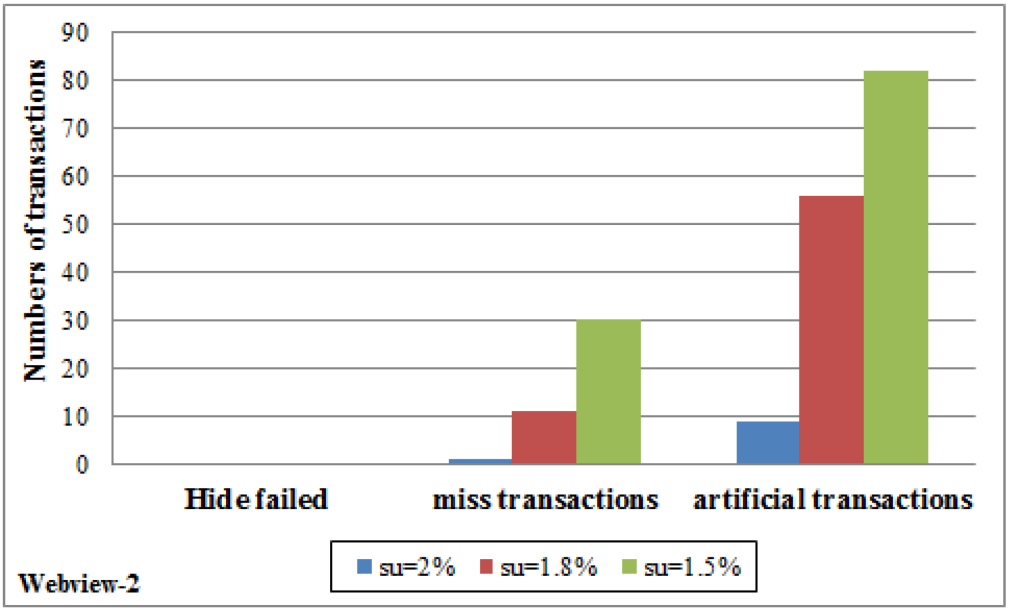

FIGURE 9. The evaluation of three side effects in Webview-2

6. **Conclusion and Future Work.** In the research of privacy-preserving data mining (PPDM), it normally can be classified into two removal approaches. The first one is to remove items from transactions, and the second one is to remove the transactions from the database. In real-world applications, however, it may cause unpredictable damages to industries or organizations if the important information or rules are removed. In this paper, a greed-based algorithm is thus proposed to insert newly transactions into the original database for efficiently hiding the sensitive itemsets. The number of newly inserted transactions and the length of each inserted transaction can be thus determined by empirical rules in standard normal distribution. The large itemsets in the original database are respectively added into the inserted transactions, for reducing the side effects of missing rules. The above procedure is repeated until the set of sensitive itemsets become null or there is no longer large itemsets to be added in the inserted transactions. After that, the small items are then added into the inserted transactions with the remaining positions to be filled according to empirical rules in standard normal distribution. The experimental results are then shown the performance of the proposed greedy-based approach for inserting new transactions.

In this paper, new transactions are then inserted into the original database for hiding the sensitive itemsets. In real-world applications, the modified itemsets within the transactions can be also considered as another research issue. How to improve the performance of the proposed greedy-based approach can also be considered as a critical research issue in PPDM.

REFERENCES

[1] R. Agrawal, and R. Srikant, Privacy-preserving data mining, *Proc. of the 2000 ACM SIGMOD international conference on Management of data*, pp. 439-450, 2000.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, Approximation algorithms for k-anonymity, *Proc. of the International Conference on Database Theory*, pp. 1-18, 2005.

[3] R. Agrawal, T. Imielinski, and A. Swami, Database mining: a performance perspective, *IEEE Trans. Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914-925, 1993.

[4] R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between sets of items in large databases, *Proc. of the 1993 ACM SIGMOD international conference on Management of data Pages*, pp. 207-216, 1993.
[5] R. Agrawal, and R. Srikant, Fast algorithms for mining association rules, *Proc. of the 20th Very Large Data Bases Conference*, pp. 487-499, 1994.
[6] R. Agrawal, and R. Srikant, Privacy-preserving data mining, *Proc. of the 2000 ACM SIGMOD international conference on Management of data*, pp. 439-450, 2006.
[7] A. Amiri, Dare to share: protecting sensitive knowledge with data sanitization, *Journal of Decision Support Systems*, vol. 43, no. 1, pp. 181-191, 2007.
[8] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios, Disclosure limitation of sensitive rules, *Proc. of the 1999 Workshop on Knowledge and Data Engineering Exchange*, pp. 45-52, 1999.
[9] F. Berzal, J. C. Cubero, N. Marín, and J. M. Serrano, *Journal of Data & Knowledge Engineering*, vol. 37, no. 1, pp. 47-64, 2001.
[10] M. S. Chen, J. Han, and P. S. Yu, Data mining: an overview from a database perspective, *IEEE Trans. Knowledge and Data Engineering*, vol. 8, pp. 866-883, 1996.
[11] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, Hiding association rules by using confidence and support, *Proc. of the 4th International Workshop on Information Hiding Pages*, pp. 369-383, 2001.
[12] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, Privacy preserving mining of association rules, *Proc. of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 217-228, 2002.
[13] J. Han, J. Pei, Y. Yin, and R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, *Journal of Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.
[14] T. P. Hong, C. W. Lin, and Y. L. Wu, Incrementally fast updated frequent pattern trees, *Journal of Expert Systems with Applications*, vol. 34, no. 4, pp. 2424-2435, 2008.
[15] T. P. Hong, and C. H. Wu, An improved weighted clustering algorithm for determination of application nodes in heterogeneous sensor networks, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 173-184, 2011.
[16] T. P. Hong, C. W. Lin, K. T. Yang, and S. L. Wang, Using TF-IDF to hide sensitive itemsets, *Journal of Applied Intelligence* , 2012.
[17] C. W. Lin, T. P. Hong, and W. H. Lu, The pre-fufp algorithm for incremental mining, *Journal of Expert Systems with Applications*, vol. 36, no. 5, pp. 9498-9505, 2009.
[18] G. C. Lan, T. P. Hong, and V. S. Tseng, Discovery of high utility Itemsets from on-shelf time periods of products, *Journal of Expert Systems with Applications*, vol. 38, no. 5, pp. 5851-5857, 2011.
[19] S. R. M. Oliveira, and O. R. Zaïane, Privacy preserving frequent itemset mining, *Cryptology ePrint Archive 2012/490*, Proc. of the IEEE international conference on Privacy, security and data mining, pp. 43-54, 2002.
[20] J. S. Park, M. S. Chen, and P. S. Yu, Using a hash-based method with transaction trimming for mining association rules, *IEEE Trans. Knowledge and Data Engineering*, vol. 9, no. 5, pp. 813-825, 1997.
[21] E. D. Pontikakis, A. A. Tsitsonis, and V. S. Verykios, An experimental study of distortion-based techniques for association rule hiding, *Journal of IFIP Advances in Information and Communication Technology*, vol. 144, pp. 325-339, 2004.
[22] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, Association rule hiding, *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 4, pp. 434-447, 2004.
[23] Z. Zheng, R. Kohavi, and L. Mason, Real world performance of association rule algorithms, *Proc. of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 401-406, 2001.