MODIFIED HIDING HIGH UTILITY ITEM FIRST ALGORITHM (MHHUIF): NOVEL ALGORITHMS FOR PRIVACY PERSERVING DATA MINING

Priyank Gupta

*Lecturer, SOS Mathematics and Allied Sciences (Computer Science), Jiwaji University, Gwalior
Madhya Pradesh, India*

Abstract

Privacy preserving data mining is introduced to show the negative side of the data mining. Because in data mining when anybody is able to access the sensitive information through mining but with this he/she is also able to access the private or personal information too. So it is the problem of the data mining to preserve the personal information. To protect private or personal data we use privacy preserving techniques.That's by privacy preserving data mining is a popular topic in the research community and many algorithms are proposed by researchers to protect private or personal data. The main thing is how to strike a balance between privacy protection and knowledge discovery in the sharing process is an important issue. In this paper, we focus on privacy preserving utility mining and present a novel algorithm MHHUIF is to achieve the goal of hiding sensitive itemsets with less time duration. The experimental results are same as HHUIF [1] algorithm on different quantity of synthetic datasets.

Key Words: *Privacy preserving, Utility mining, Data mining.*

1. INTRODUCTION

One of the most important methodologies in data mining, traditional association rule mining, discovers all itemsets witch support values are greater than the given threshold. The literature discloses may algorithms for discovering the frequent itemsets. The Apriori algorithm [2][3][4] is the most famous one. In order to measure how useful an itemset is in the database, researchers recommend utility mining [5]. Utility mining overcomes the shortcoming of traditional association rule mining. Which ignores the sale quantity and price or profitability among items in a transaction.

In the past decade, privacy preserving data mining [6][7] became a popular research direction for data mining[8][9]. First Showed that data mining threatens databasesand suggested possible solutions to achieve privacy protection from data mining[10][11] discussed privacy preserving mining of association rules.

Evfimievski et al. discussed a motivation example [12]. Suppose a server has many clients and each client has its own data. The clients expert the server to gather statistical information from all client's data about association among items to provide recommendations to their customers.

However, the clients do not like the server to take itemsets containing highly sensitive knowledge. Thus, when a client delivers its database to the server, some sensitive itemsets are hidden from the database according to specific privacy policies. Ther server only gathers statistical information from the modified database.

However, privacy preserving utility mining is not discussed in the literature. Therefore, this study focuses on privacy preserving utility mining and present a novel algorithm MHHUIF, to achieve the goal of hiding sensitive itemsets so adversaries cannot mine them from the modified database. The procedure of transforming the original database into the sanitized on is called the sanitizing process. The sanitizing process acts on the data to remove a small number of items in some transactions containing sensitive itemsets.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 offers the MHHUIF algorithm to improve the balance between privacy protection and knowledge discovery. Section 4 presents our experimental results and evaluates the performance of the proposed algorithm. Finally, Section 5 presents the study's conclusions.

2. RELATED WORKS

In this section we explain in detail utility mining and Privacy preserving mining on association rules to understand MHHUIF algorithm, its working and its advantages.

2.1 Utility mining

Utility mining searches all itemsets whose utility values are equal to or greater than a user specified threshold in a transaction database. However, the utility values of itemsets do not satisfy the downward closure property. That is, a subset of a high utility itemset may not be a high utility itemsdet. The challenge of tulity mining is in restricting the size of the candidate sets and simplifying the computation for calculation the utility. Recently, Li et al. developed some efficient approaches, including the FSM, SuFSM, ShFSM, and DCG methods for share mining [13][14][15]. Share mining is an item count and external utility of items; share mining is equivalent to utility mining. In the meanwhile, [16] also presented the two-phase(TP) algorithm for fast discovering all high utility itemsets. By using and isolated items discarding strategy, [17] proposed an efficient algorithm for discovering high utility itemsets.

The following set of terms are defined and given in [17] for the utility mining problem.

Let I = $\{i_1, i_2, \ldots, i_m\}$ be a set of items, where m is the total number of items. Let DB = $\{T_1, T_2, \ldots, T_n\}$, the task-relevant database, be a set of transactions where each transaction T_q is a set of items, that is, $T_q$is a set of items, that is, $T_q \subseteq I$. Set of items is also referred as an itemset. An itemset that contains k-items is called a k-itemset.

- The item count of item $i_p \subseteq I$ in transaction T_q, $c(i_p, T_q)$, is the number of item i_p purchased in transaction T_q. For example, $c(A, T_1) = 0$, $c(B, T_1) = 0$, and $c(C, T_1) = 18$ in Table1 (a).

- Each item i_p has an associated set of transactions $T_{ip} = \{T_q \in DB \backslash i_p \in T_q\}$.

- A k-itemset X = $\{x_1. x_2, \ldots, x_k\}$ is a subset of 1, where $1 \leq k \leq m$.

- Each k-itemset X has an associated set of transaction $T_X = \{T_q \in DB \backslash X \in T_q\}$.

- The external utility of item of $i_p \subseteq l$, eu (i_p), is the value associated with item $i_p$in the external utility table. This value reflects the importance of an item, which is independent of transactions. For example, in Table 1(b), the external utility of item A, eu(A), is 3.

- The utility of item $i_p \subseteq l$ in transaction T_q, $u(i_p, T_q)$, is the quantitative measure of utility for item $i_p$in transaction T_q, defined as $eu(i_p)$ x $c(i_p, T_q)$.

- The utility of itemset X in transaction T_q, $u(X, T_q)$, is$\sum_{ip \in x} u(i_p, T_q)$, where $X \subseteq T_q$.

- The utility of itemset X, u(X), is defined as $\sum_{X \subseteq Tq \in \mathbf{DB}} u(X, T_q)$.

Utility mining is to find all the itemsets whose utility values are beyond a user specified threshold. Itemset X is a high utility itemset, if u (X) $\geq \varepsilon$, where ε is the minimum utility threshold. In Table 1, u({A,D}) = u({A,D},T_4) +u({A,D}, T_8) = 9+27=36, and u({A,D,E}) =u({A,D,E},T_4) +u({A,D,E}, T_8) = 14+32=46. If ε is set to be 40, {A,D} is a low utility itemset and {A,D,E} is a high utility itemset. That is, the “downward closure property” does not hold in the utility mining model.

Table – 1: An example of transaction database

TID	**A**	**B**	**C**	**D**	**E**
(a) Transaction table					
T1	0	0	18	0	1
T2	0	6	0	1	1
T3	2	0	1	0	1
T4	1	0	0	1	1
T5	0	0	4	0	2
T6	1	1	0	0	0
T7	0	10	0	1	1
T8	3	0	25	3	1
T9	1	1	0	0	0
T10	0	6	2	0	2

ITEM	**PROFIT ($) (per unit)**
(b) The external utility table	
A	3
B	10
C	1
D	6
E	5

2.2 Privacy preserving mining on association rules

Researchers divide sanitizing algorithm for privacy preserving mining on association into two categories: **(1) Data-Sharing approach** and **(2) Pattern-Sharing approach** [6]. Fig. 1 illustrates the taxonomy of the sanitizing algorithms for PPDM.

Data-Sharing approach: The sanitizing process removes or hides the group of restrictive rules that contain sensitive knowledge; Researchers further subdivide algorithms of the Data-Sharing approach into the following sub-categories [19]. First technique is "item Restriction-Based" [20], and second technique is "Item Addition" [6], and third technique is "Item Obfuscation-Based" [21][6].

Pattern-Sharing approach: The sanitizing algorithm acts on the rules mined from a database rather than acting on the data. Regarding pattern-sharing techniques, Rule Restriction-Based method, introduced by [6], is the only known approach that falls into this category. This approach blocks some interface channels to ensure that an adversary cannot reconstruct restrictive rules from the non-restrictive ones. In doing so, this method reduces the inference channels and minimizes the side effect, Examples of this include the DSA algorithm proposed by [22] and MINSS, MINNS, SIMBLK, and BINFCH by [23][24].

Other motivating heuristics privacy preserving mining an association rules can be sought in [25][26][27].

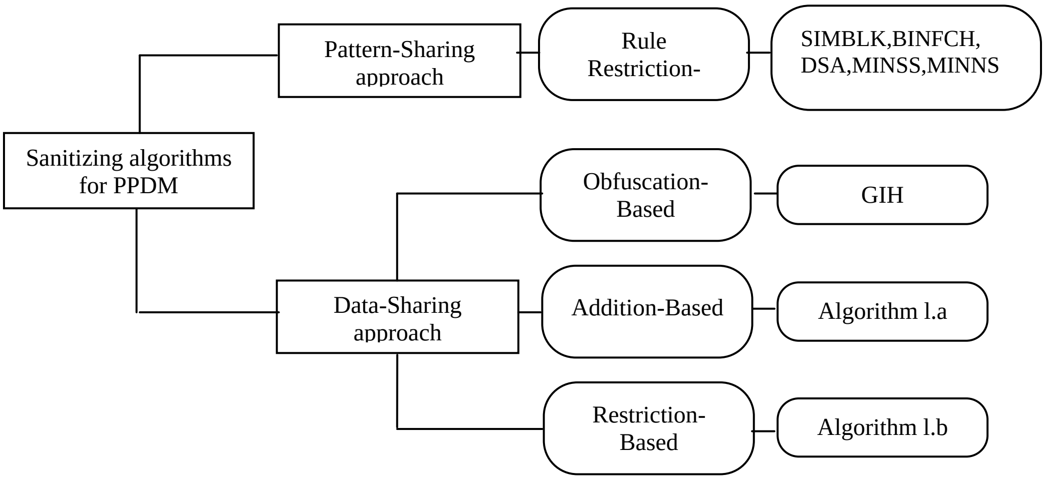


Fig – 1: Taxonomy of sanitizing algorithms and the corresponding algorithms

3. Proposed algorithms

As before we say that the algorithm [1] is takes a lot of extra time. That's why in this section presents the sanitizing process in detail and propose the following algorithm for privacy preserving utility mining **Modified Hiding High Utility Item First Algorithm (MHHUIF)**. First Definitions of some notations used in the rest of the paper follow.

Definition 1 (sensitive itemsets): Let I={i_1, i_2,…,i_m} be a set of items, where m is the total number of items. Let DB = {T_1, T_2,…,T_n} be a set of transactions, ε be the minimum utility threshold, and L be a set of all high utility itemsets for ε. Let U = {S_1, S_2,…,S_l} be a subset of L, where $S_{l,}$ called **sensitive itemset,** is an itemset that should be hidden according to some security policies.

Definition 2 (Conflict Count): Theconflict count of item i_p in U, denoted as Icounti$_p$ (U), is the number of sensitive itemsets containing i_p. That is, Icounti$_p$ (U)= |{S_i € U|i_p € S|$_i$}|.

Definition 3 (privacy threshold): Oliveira et al. first proposed the concept of the privacy threshold ψ [18], which is the proportion sensitive patterns that are still discovered from the sanitized database.

The privacy threshold ψ ranges from 0% to 100%. When ψ = 0% no sensitive patterns can be discovered, When ψ = 100%, there are no restriction on the sensitive patterns and all sensitive patterns can be discovered. The advantage of this threshold mechanism is that users can balance privacy and the disclosure of information. After that Yeh and Hsu et al. also proposed an algorithm for privacy preservation called HHUIF [1]; but it takes lot of extra time. To reduce the process time we also proposed an algorithm called Modified Hiding High Utility Item First Algorithm **(MHHUIF).**

3.1 The sanitization process

In general, the sanitization process for PPUM consists of the following three steps: (1) apply utility mining algorithm on collected database to obtain all high utility itemsets as given; (2) find sensitive itemsets based on user requirements; and (3) apply MHHUIF algorithm to generate the sanitized database which has no any utility value which produce high utility value from threshold.

Step 1: first collect all high utility itemset in a separate temporary table.
Step 2: find minimum utility threshold.
Step 3: those values are greater than threshold values to be decreases. These values user will not be release publicly because these are sensitive data.
Step 4:apply MHHUIF algorithm on sensitive database.
Step 5: The main goal of the sanitizing algorithm is to decrease the utility value of each sensitive itemset by modifying item quantity values in the sensitive itemset.

Bringing the utility values of all sensitive itemsets under the minimum utility threshold ε completes the sanitized database. As long as the adversary selected utility threshold is smaller than or equal to ε, the sanitizing algorithm guarantees the sanitized database will not reveal any sensitive itemsets.

3.2 Modified Hiding High Utility Item First (MHHUIF) algorithm

The main goal of the MHHUIF algorithm is to decrease the utility value of each sensitive itemset by modifying the quantity values of items contains in the sensitive itemset with less time duration process as compare to HHUIF [1].

To decreases the utility value of each sensitive itemset S, MHHUIF modifies the item quantity value with the highest utility value in some transaction containing S. the process repeats until the utility values of all sensitive itemsets are below the minimum utility threshold. The pseudo-code of the MHHUIF algorithm is as follows:

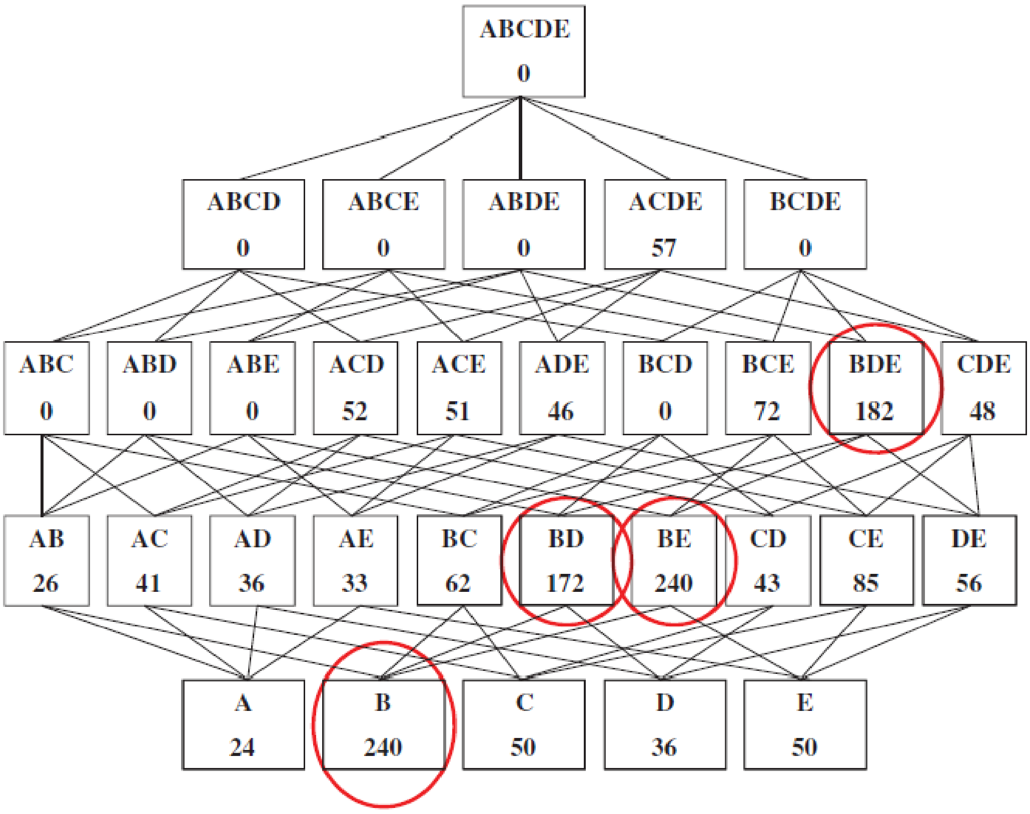
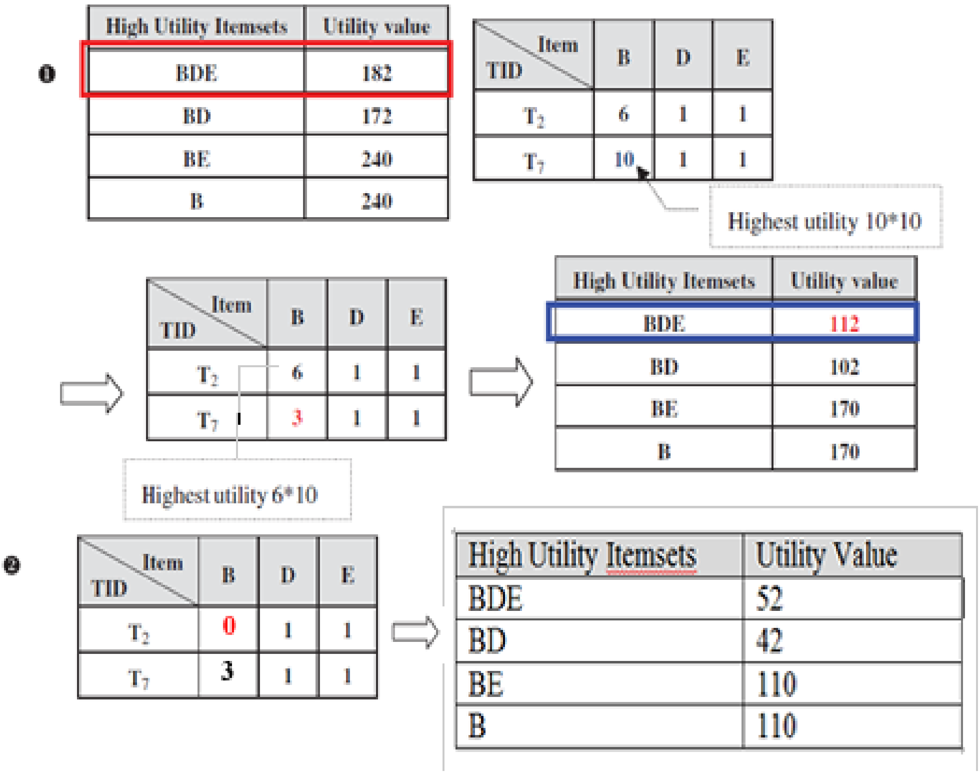


Fig. – 3: Itemsets lattice related to the example in Table 1 with ε = 120. Itemsets in solid circles are the high utility itemsets.



High Utility Itemsets	Utility value
BDE	182
BD	172
BE	240
B	240

TID \ Item	B	D	E
T2	6	1	1
T7	10	1	1

TID \ Item	B	D	E
T2	6	1	1
T7	3	1	1

High Utility Itemsets	Utility value
BDE	112
BD	102
BE	170
B	170

TID \ Item	B	D	E
T2	0	1	1
T7	3	1	1

High Utility Itemsets	Utility Value
BDE	52
BD	42
BE	110
B	110

Fig. – 4: The detailed steps of the MHHUIF algorithm for Example 1

Algorithm MHHUIF
Input: the original database DB; the minimum utility threshold ε; the sensitive itemsets U = {S_1, S_2, ..., S_l}.
Output: the sanitized database DB' so that S_i cannot be mined.

```
1.  Start loop Si times which is subset of U
2.  Find diff of $u(S_i)$- ε
3.  Start loop until diff is greater than 0
        • (conflict item, Transition) = find avg max value of (u(i,T))
        • Modify o(conflict item, Transition)
        a. If u(conflic item, Transition)<diff than set o(conflict item, Transition) with 0
        b. If u(conflic item, Transition)>diff than set o(conflict item, Transition) with o(conflict item, Transition) – diff/$s(i_p)$
4.  return the sanitized database DB'
```

Line 2 calculates the difference between the utility of an itemset S_i and the minimum utility threshold ε. Diff is the amount of the utility value needs to be reduced. In Lines 3, **MHHUIF** sanitizes the transactions containing the sensitive itemsets repeatedly until diff $\leq$ 0. First find the values which are greater than average value of threshold after that modify these values using MHHUIF algorithm. After that again check minimum utility threshold transaction T_q. if $u(i_p, T_q)$ < diff, that is, the utility of i_p on T_q is less than diff, this method reduces the quantity of item i_p in T_q to 0 and continues to modify the quantity of the next item until diff $\leq$ 0. If $u(i_p, T_q)$ > diff, the quantity of i_p in T_q does not have to be reduced to 0. The quantity value of i_p in T_q is set as $o(i_p, T_q)$- $[\frac{\text{diff}}{s(ip)}]$. The process continues until the utility value of each sensitive itemset is below ε.

Example 1: To illustrate how the HHUIF algorithm works. Consider the sample transactional database in Table 1. For the given minimum utility threshold ε=120, the high utility itemsets are {B}, {B, D}, {B, E}, and {B, D, E} as listed in Fig. Suppose {B, D, E} and {B, E} are the chosen sensitive itemsets. The transactions containing the sensitive itesmets {B, D, E} and {B, E} are {$T_{2,}$ T_7,} and {T_2, T_7, T_{10}}, respectively. Since item B has the highest utility 10 x 10 in transaction T_7 among all items in {B, D, E} and all transaction T_7 and modifies its quantity from 10 to 3. In doing so, the utility value of sensitive itemset {B, D, E} become 112 which is below the minimum utility threshold. Similarly, for the sensitive itemset {B, E}, MHHUIF repeats the above steps until the utility value of sensitive itemset {B, E} is smaller than 120. In fact, the high utility itemsets {B, D} and are {B} are hidden by accident after sanitizing. The MHHUIF algorithm generates to artificial itemsets from the sanitized database. Fig 4 illustrates the detailed steps of the MHHUIF algorithm for example 1.

In this section we get the find the results using HHUIF and MHHUIF algorithm and we can check easily MHHUIF algorithm produce same results as HHUIF but it takes less time for processes those algorithms.

4.1 Experimental results

To measure the effectiveness of the HHUIF algorithms, experiments were conducted on two synthetic datasets. All experiments were performed on a Sony Vaio workstation with a 2.53 GHz Intel CORE i3 processor and 4 GB of main memory, running Windows 7 Home Premium. We first applied the HHUIF algorithm to extract all high utility itemsets from the datasets. From the high utility itemsets found in each dataset, this experiment randomly selected two sets of sensitive itemsets with a size of five and ten. Next, we sanitized sensitive itemsets in the Microsoft MS-Access 2007 database and developed on Java 1.6. In most cases, the data receiver can choose different thresholds for mining high utility itemsets on the released database. After that we applied the MHHUIF algorithm and find new same results but the processing time is much reduced. Users control the proportion of restrictive patterns still discovered from the sanitized database can be controlled by users with the privacy threshold ψ. This proportion ranges from 0% to 100%. When ψ = 0%, no restrictive patterns can be discovered. When ψ = 100%, there are no restrictions on the restrictive patterns and all restrictive patterns can be discovered. The advantage of this threshold mechanism is that users can balance privacy and the disclosure of information. The current experiments adopt the minimum utility threshold (MinUtility) ε for the data deliverer and introduce another parameter called Expecting Minimum Utility Threshold d for the data receiver.

4.2 Datasets

This study used the synthetic data to generate a synthetic datasets. The dataset contains 10, 50, 100, 200, 500, 1000 and 2000 transactions with 5 distinct items.

Table - 2: Result MHHUIF and HHUIF

Existing Algorithm		Proposed Algorithm	
Transactions	**Time(in milliseconds)**	**Transactions**	**Time(in milliseconds)**
10	2056	10	1366
100	17834	100	11983
200	26393	200	17888
500	65876	500	57742
1000	216528	1000	132452
2000	502732	2000	38519

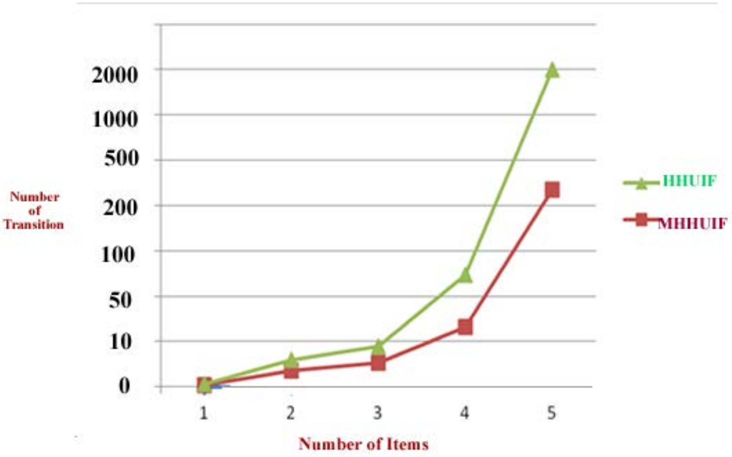


Fig. – 5: Plot data and number of items over time, which is executes on HHUIF and MHHUIF algorithm for Example 1

It is observe that our proposed work give the better performance in terms of execution of the algorithm over the existing concepts. The performance has shown by the above figure 5.

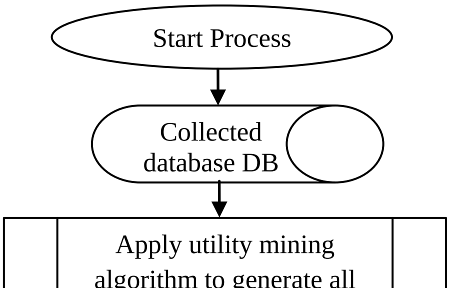


Fig. – 2: illustrate the sanitizing process.

5. Conclusion

Data quality plays an important role in the mining process. Accurate input data brings meaningful mining results. If sensitive data is lost so it is very harm full to any organization. On the other hand, preserving privacy is also a vital issue. In strategic alliance cases, organizations need to share information with others and protect their own business confidentiality as well. Therefore, research needs to offer and investigate an effective privacy preserving mining model. This study first discusses a privacy preserving utility mining (PPUM) model and presents the MHHUIF algorithms to reduce the impact on the source database of privacy preserving utility mining with less time. This algorithm modify the database transactions containing sensitive itemsets to reduce the utility value can below the given threshold while preventing reconstruction of the original database from the sanitized one. Experimental results show that MHHUIF has the lower miss costs than MSICF on two synthetic datasets. On the other hand, MSICF has a lower difference ratio than HHUIF between original and sanitized databases.

6. Future work

In future we will further work this algorithm using some different parameters like hiding failure, miss cost, DBDR etc.

(a) Hiding failure (HF): the ratio of sensitive itemsets that are disclosed before and after the sanitizing process. The hiding failure is calculated as follows:

$$HF = \frac{|U(DB')|}{|U(DB)|}$$

where U(DB) and U(DB') denote the sensitive itemsets discovered from the original database DB and the sanitized database DB', respectively. The cardinality of a set S is denoted as |S|.

(b) Miss cost (MC): the difference ratio of legitimate itemsets found in the original and the sanitized databases. The misses cost is measured as follows:

$$MC = \frac{|\sim U(DB) - \sim U(DB')|}{|\sim U(DB)|}$$

where ~ U(DB) and ~ U(DB') denote the non-sensitive itemsets discovered from the original database DB and the sanitized database DB0, respectively.

(c) Database difference ratio (DBDR): the difference ratio between the original database DB and the sanitized database DB' is given by:

$$DBDR = \frac{|DB - DB'|}{|DB|}$$

REFERENCES

[1]. J.-S.Yeh, P.-C. Hsu, "HHUIF and MSICF: Novel algorithms for privacy preserving utility mining", Expert Systems with Application, vol. - 37, pp. - 4779-4786, 2010.

[2]. R. Agrawal, T. Imielinski, A. N. Swami, "Mining association rules between sets of items in large databases", In Proceedings of the 1993 ACM SIGMOD international conference on management of data, pp. - 207-216, 1993.

[3]. R. Agrawal, R. Srikant, "Fast algorithms for mining association rules", In Proceedings of the 20th international conference on very large data bases, pp. - 487-499, 1994.

[4]. H. Mannila, H. Toivonen, A. I. Verkamo, "Efficient algorithms for discovering association rules", In Proceedings of AAAI workshop on knowledge discovery in databases (KDD'94), pp. - 181-192, 1994.

[5]. H. Yao, H. J. Hamilton, C. J. Butz, "A foundational approach to mining itemsetutilities from databases", In Proceedings of the 4th SIAM international conference on data mining", pp. - 484-486, 2004.

[6]. V. Verykios, E. Bertino, I. N. Foyino, L. P. Provenza, Y. Saygin, Y. Theodoridis, "State-of-the-art in privacy preserving data mining", ACM SIGMOD Record, vol. - 33, no.-1, pp. - 50-57, 2004.

[7]. V. S. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, E. Dasseni, "Association rules hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. - 10, no.-4, pp. - 434-447, 2004.

[8]. C. Clifton, D. Marks, "Security and privacy implications of data mining", In Proceedings of the 1996 ACM SIGMOD workshop on data mining and knowledge discovery, pp. - 15-19, 1996.

[9]. C. Clifton, M. Kantarcioglu, J. Vaildya, "Defining privacy for data mining", In National Science Foundation Workshop on Next Generation Data Mining (WNGDM), pp. - 126-133, 2002.

[10]. S. J. Rizvi, J. R. Haritsa, "Maintaining data privacy in association rule mining", In Proceedings of the 28th

international conference on very large data bases, pp. - 682-693, 2002.

[11]. Y. Saygin, V. S. Verykios, A. K. Elmagamid, "Privacy-preserving association rule mining", In Proceedings of the 12th international workshop on research issues in data engineering: Engineering E-commerce/E-business systems (RIDE'02), pp. - 151, 2002.

[12]. A. Evfimievski, R. Srikant, R. Agrawal, J. Gehrke, "Privacy preserving mining of association rules", In Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining, pp. - 217-228, 2002.

[13]. Y. C. Li, J. S. Yeh, C. C. Chang, "Efficient algorithms for mining share-frequent itemsets", In Proceedings fo fuzzy logic, soft computing and computational intelligence – 11th world congress of international fuzzy systems association (IFSA 2005), pp. - 534-539
, 2005a.

[14]. Y. C. Li, J. S. Yeh, C. C. Chage, "A fast algorithm for mining share-frequent itemsets", Lecture Notes in Computer Science, Vol. 3399, pp. - 417-428, 2005b.

[15]. Y. C. Li, J. S. Yeh, C. C. Chang, "Direct candidates generation: A novel algorithm for discovering complete share-frequent itemsets", Lecture Notes in Artificial Intelligence, Vol. - 3614, pp. - 551-560, 2005c.

[16]. Y. Liu, W. K. Liao, A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets", Lecture Notes in Computer Science, Vol. - 3518, pp. - 689-695, 2005.

[17]. Y. C. Li, J. S. Yeh, C. C. Chang, "Isolated items discarding strategy for discovering high utility itemsets", Data & Knowledge Engineering, Vol. - 64, no. - 1, pp. – 198-217, 2008.

[18]. S. R. M. Oliveira, O. R. Zaiane, Y. Saygin, "Secure association rule sharing", In Proceedings of 8th Pacific-Asia conference on knowledge discovery and data mining (PAKDD'04), pp. – 74-85, 2004.

[19]. S. J. Rizvi, J. R. Haritsa, "Maintaining data privacy in asspciation rule mining", In Proceedings of the 28th international conference on very large data bases, pp. – 682-693, 2002.

[20]. V. S. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, E. Dasseni, "Association rules hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. - 16, no. – 4, pp. – 434-447, 2004.

[21]. Y. Saygin, V. S. Verykios, C. Clifton, "Using unknowns to prevent discovery of association rules", ACM SIGMOD Record, Vol. – 30, no. – 4, pp. – 45-54, 2001.

[22]. S. R. M. Oliveira, O. R. Zaiane, Y. Saygin, "Secure association rule sharing", In Proceedings of 8th Pacific-Asia conference on knowledge discovery and data mining (PAKDD'04), pp. – 74-85, 2004.

[23]. Z. Wang, W. Wang, B. Shi, S. H. Boey, "Preserving private knowledge in frequent pattern mining" In Proceedings of 6th IEEE international conference on data mining – workshops (ICDMW'06), pp. – 530-534, 2006.

[24]. Z. Wang, W. Wang, B. Shi, "Blocking inference channels in frequent pattern sharing", In Proceedings of 2007 IEE 23rd international conference on data engineering, pp. – 1425-1429, 2007.

[25]. A. Gkoulalas-Divanis, V. S. Verykios, "hiding sensitive knowledge without side effects", Knowledge and Information Systems. Doi:10.1007/s10115-008-0178-7, 2009.

[26]. M. Kantarcioglu, C. Clifton, "Privacy-perserving distributed mining of association rules on horizontally partitioned data". IEEE transactions on Knowledge Data Engineering, Vol. – 16, no. – 9, pp. 1026-1037, 2004.

[27]. J. Vaidya, C. Clifton, "Privacy preserving association rule mining in vertically partitioined data", In Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, pp. 639-644, 2002.

BIOGRAPHIE

Priyank Gupta has working as Lecturer in SOS Computer Science Jiwaji University, Gwalior, Madhya Pradesh, India.