International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **12**, December **2013**

 ISSN 1349-4198
pp. 4851–4862

A MODIFIED HIDING HIGH UTILITY ITEM FIRST ALGORITHM (HHUIF) WITH ITEM SELECTOR (MHIS) FOR HIDING SENSITIVE ITEMSETS

Rajalakshmi Selvaraj[1] and Venu Madhav Kuthadi[2]

[1]Department of Computer Science
Botswana International University of Science and Technology
OCAAT Plot 1055 Modipane Road Oodi, Botswana
rajalakshmi0878@gmail.com

[2]Department of Applied Information Systems
University of Johannesburg
No. 89, 6th Avenue, Melville, Johannesburg, South Africa

Received December 2012; revised April 2013

Abstract. *In privacy preserving data mining, utility mining plays an important role. In privacy preserving utility mining, some sensitive itemsets are concealed from the database according to certain privacy policies. Hiding sensitive itemsets from the adversaries is becoming an important issue nowadays. Also, only very few methods are available in the literature to hide the sensitive itemsets in the database. One of the existing privacy preserving utility mining methods utilizes two algorithms, HHUIF and MSICF to conceal the sensitive itemsets, so that the adversaries cannot mine them from the modified database. To accomplish the hiding process, this method finds the sensitive itemsets and modifies the frequency of the high valued utility items. However, the performance of this method lacks if the utility value of the items are the same. The items with the same utility value decrease the hiding performance of the sensitive itemsets and also it has introduced computational complexity due to the frequency modification in each item. To solve this problem, in this paper a modified HHUIF algorithm with Item Selector (MHIS) is proposed. The proposed MHIS algorithm is a modified version of existing HHUIF algorithm. The MHIS algorithm computes the sensitive itemsets by utilizing the user defined utility threshold value. In order to hide the sensitive itemsets, the frequency value of the items is changed. If the utility values of the items are the same, the MHIS algorithm selects the accurate items and then the frequency values of the selected items are modified. The proposed MHIS reduces the computation complexity as well as improves the hiding performance of the itemsets. The algorithm is implemented and the resultant itemsets are compared against the itemsets that are obtained from the conventional privacy preserving utility mining algorithms.*

Keywords: Utility mining, Privacy preserving utility mining, Sensitive itemsets, Utility value, Frequency value

1. **Introduction.** Data mining techniques and algorithms play an important role in knowledge discovery. Data miners often require a full access to the data for building accurate models [1]. Data mining technique has many promising advantages like discover valuable, non-obvious information from large database [15,19], cannot lead to any misuse [2]. Data mining is performed with large databases, where it may contain some sensitive information [6]. As long as security of sensitive data against unauthorized access is a long term goal for the database security research community and government statistical agencies [7]. Hence, the security issues have become a more important area of research

in the field of data mining. While on the other hand one of the biggest barriers in facing data mining projects today is the "inability to release data" due to privacy concerns [1].

The concept of preserving the privacy of data mining has recently been discussed in response to the above concerns [3]. Privacy is a term which is associated with the mining task so that we are able to hide some crucial information which we do not want to disclose to the public. So, the concept of privacy preserving data mining is the process of preserving the personal information from data mining algorithms [4]. There are two types of privacy in data mining. The first type of privacy is output privacy, where the data is minimally altered so that the mining result will preserve certain privacy. The second type of privacy is input privacy, in which the data is manipulated so that mining result is not affected or minimally affected [12].

Privacy issues are further exacerbated by the Internet, which makes it easy for everyone to collect the new data automatically and add it to databases [5]. The goal of privacy preservation in data mining is to develop algorithms to modify the original data in some way, so that the private data and knowledge remain confidential even after the mining process [8]. The main concern in privacy preserving data mining is twofold. First, sensitive raw data like identifiers, names, addresses and more should be modified or trimmed out from the original database. Second, sensitive knowledge mined from a database by using data mining algorithms should also be excluded, because such knowledge can equally well compromise data privacy [9].

In privacy preservation of data mining, utility based mining will play an important role. Utility mining is used to find out the high utility itemsets. User-defined utility is based on the information not available in the transaction dataset. It often requires user preference and then it can be represented by an external utility table or utility function. Utility table (or function) determines the utilities of all items in a given database [11]. Utility mining discovers all itemsets whose utility values are equal to or greater than a user specified threshold in a transaction database [3]. Protection of privacy in utility mining will automatically minimize the number of non-sensitive patterns lost [10].

Some literary works based on privacy preserving data mining are discussed in the literature. Hence, this study focuses on privacy preserving utility mining and presents one novel algorithm MHIS, to achieve the goal of hiding sensitive itemsets, and so the antagonists cannot extract them from the modified database. The process of converting the original database into the sanitized one is called sanitization. The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 discusses the utility mining algorithm and the problem present in the existing HHUIF algorithm. Section 4 describes the proposed MHIS algorithm for avoiding the drawbacks in the existing HHUIF algorithm. Section 5 discusses the experimental results and evaluates the performance of the proposed algorithm. Finally, Section 6 concludes the paper.

2. **Related Works.**

Poovammal and Ponnavaikko [13] have proposed a simple technique to transform the categorical and numeric sensitive data by using a mapping table and graded grouping technique, respectively. The typical data mining tasks like classification, clustering, and association and rule mining have been performed on both the original and transformed tables. The rules/results/patterns of both the tables have been compared and the utility of the transformed data has been evaluated.

Poovammal and Ponnavaikko [14] have developed an approach for the designing of a micro data sanitization technique in order to preserve privacy against some factors such as proximity and divergence attack and also to preserve the utility of the data for any type of mining task. This proposed approach has applied a graded grouping transformation on numerically sensitive attributes and a mapping of the table based transformation on

categorical sensitive attribute. They have also conducted some experiments on adult dataset and compared the results of the original and transformed table to show that the proposed task of independent technique preserves privacy, information, and utility.

Yeh and Hsu [20] have focused on privacy preserving utility mining (PPUM) and proposed two algorithms called HHUIF (Hiding High utility item First Algorithm) and MSICF (Maximum Sensitive Itemsets Conflict First algorithm), in order to achieve the goal of hiding sensitive itemsets, so that the adversaries cannot mine them from the modified database. On the other hand, they have also minimized the impact on the sanitized database of hiding sensitive itemsets. The experimental results have shown that the HHUIF achieved a lower miss cost than MSICF on two synthetic datasets. On the other hand, MSICF generally has a lower difference ratio between original and sanitized databases than the HHUIF.

Rani and Revathi [16] have proposed that anonymized publications on static micro data can be achieved with heavy information loss by Generalization. An enhanced advantage of Generalization known as Angelization has produced the same level of anonymization but with very small information loss. In reality, there is a need to publish another version of micro data, after insertions and deletions. Anonymization was applicable to various generalization principles like k-anonymity, l-diversity and t-closeness. Incremental m-invariance with Angelization has also preserved the privacy in re-publication of dynamic micro data after insertions and deletions. Mondrian algorithm has been used in the technique for the partitioning in Angelization. The publication of marginal's from the generalized micro data is also supported by m-invariance. KL-divergence has been used for quantifying the discrepancy of two distributions. The reconstruction error has been measured as the KL-divergence between the reconstructed distribution and the original distribution. Data reconstruction error was minimal in m-invariance with enhanced utility of Generalization.

Li and Wang [17] have proposed a PPDM algorithm based on weighted SVD (Singular Value Decomposition) technique. In this proposed method, each sample has a weight and different samples have been treated with different weights. The experimental results have shown that while maintaining the data utility, the weighted SVD-based method has significantly improved over the original SVD-based method in privacy protection.

Contributions of the paper

The main contributions of the paper are

- To find the sensitive itemsets.
- To execute the MHIS algorithm process by comparing the sensitive itemsets, items utility value.
- Analyze the results with the existing HHUIF algorithm.

3. The Proposed Modified Privacy Preserving Utility Mining Algorithm.

Let D be the transaction database, containing a set of transactions $D = \{T_1, T_2, T_3, \cdots, T_n\}$, where n is the total number of transactions. The database D contains a set of items, which is denoted as $I = \{i_1, i_2, i_3, \cdots, i_m\}$, where m is the total number of items in the database. Each transaction T_n is a set of items and a set of items is termed as an itemset. Moreover, the external utility value of each item in the database is stored in the external utility table, which is referred as, $E = \{e(i_1), e(i_2), e(i_3), \cdots, e(i_m)\}$ where $e(i_m)$ is the external utility value of an item i_m, $i_m \subseteq I$. The frequency value of each item i_m in transaction T_n is $v(i_m, T_n)$ is the number of items i_m acquired in transaction T_n.

Utility Mining Algorithm

Utility mining is used to find all the itemsets' utility values.

The utility value of item i_m in transaction T_n is defined as,

$$u(i_m, T_n) = e(i_m) \times v(i_m, T_n) \tag{1}$$

The utility value of an itemset X in transaction T_n is denoted as,

$$u(X, T_n) = \sum_{i_m \in X} u(i_m, T_n) \tag{2}$$

Then, find the itemsets whose utility value is higher than the user specified threshold value α, where α is the minimum utility threshold. The itemset X is a high utility itemset, if $u(X) \geq \alpha$. These high utility itemsets are stored in $H = \{s_1, s_2, \cdots, s_i\}$ and such itemsets are sensitive itemsets. The sensitive itemsets should be concealed according to some security strategies. To perform the sanitizing process, the existing method [20] has utilized two algorithms: HHUIF and MSICF. Amongst these two algorithms, HHUIF produces lower miss cost than the MSICF and the HHUIF algorithm is described below.

The Existing HHUIF Algorithm

The main objective of the HHUIF algorithm is to diminish the utility value of each sensitive itemset by modifying the quantity values of items contained in the sensitive itemsets. The pseudo-code of the HHUIF algorithm is given below:

Input: the original database D; the minimum utility threshold α; the sensitive itemsets $H = \{s_1, s_2, \cdots s_i\}$
Output: the sanitized database D' so that s_i cannot be mined.
Step 1: for each sensitive itemset $s_i \in H$
Step 2: $diff = u(s_i) - \alpha$// the utility value needs to be reduced
Step 3: while $(diff > 0)\{$
Step 4: $o(i_m, T_n) = \arg\max_{(i \in s_i, s_i \subseteq T)}(u(i, T))$
Step 5: modify $o(i_m, T_n)$ with

$$o(i_m, T_n) = \begin{cases} 0, & \text{if } u(i_m, T_n) < diff \\ o(i_m, T_n) - \left\lceil \frac{diff}{s(i_m)} \right\rceil, & \text{if } u(i_m, T_n) > diff \end{cases}$$

Step 6:

$$diff = \begin{cases} diff - u(i_m, T_n), & \text{if } u(i_m, T_n) < diff \\ 0, & \text{if } u(i_m, T_n) > diff \end{cases}$$

$\}$
Step 7: return the sanitized database D'

This HHUIF process continues until the utility value of each sensitive itemset becomes lower than α. This existing HHUIF privacy preserving utility mining algorithm has some drawbacks in the hiding process and such drawback is formulated in the following section.

Problem Formulation

The HHUIF algorithm hides the sensitive itemsets having high utility value. However, the drawback of this algorithm is that, if the items in the sensitive itemsets having the same utility value, then it will decrease the hiding performance. For example, $\{A, B\}$ is a sensitive itemset ($\alpha = 120$), having utility value $u(A, B) = 200$. To hide the itemset $\{A, B\}$, here the frequency value of item in the itemset having high utility value is changed. In case, if both A, B have the same utility value as 100 then, any one of the item's value is modified randomly. Hence, this process creates an impact between these items. To solve this drawback, we have proposed a Modified HHUIF algorithm with Item Selector (IS) (MHIS). The Item Selector is used to select the high utility value itemset by using the following algorithm, and subsequently the frequency value of the selected items are modified. The developed IS will reduce the computation complexity as well as improves the hiding performance of the itemsets. The proposed MHIS algorithm has numerous

applications in homeland security, medical database mining, and customer transaction analysis.

4. **Modified HHUIF Algorithm with Item Selector (MHIS).** The main objective of the MHIS algorithm is to select the best items from the sensitive itemsets having same high utility value. The high utility value itemsets are hidden by modifying the frequency values of items contained in the sensitive itemsets based on the minimum utility threshold value α. This hiding process is repeated until all the sensitive itemsets utility value become lower than the threshold value α. The proposed MHIS algorithm is described below.

Input: the original database D; the minimum utility threshold α; the sensitive itemsets $H = \{s_1, s_2, \cdots, s_i\}$
Output: the sanitized database D' so that s_i cannot be mined.
Step 1: for each sensitive itemset $s_i \in H$
Step 2: $diff = u(s_i) - \alpha //$ the utility value needs to be reduced
Step 3: while $(diff > 0)\{$
Step 4: if s_i contains two items $s_i \subseteq (i_k, i_m)$
Step 5: Compare $u(i_k, T_n)$ and $u(i_m, T_n)$
Step 6: if $u(i_k, T_n) = u(i_m, T_n)$ go to Step 7 otherwise go to Step 15
Step 7: Select i_k, i_m items frequency values $v(i_m, T_n)$ and $v(i_k, T_n)$
Step 8: Sort $v(i_m, T_n)$ and $v(i_k, T_n)$ frequency values in descending order and stored in S_m and S_k
Step 9: Select top $j^{v(i_m,T_n)}$, $j^{v(i_k,T_n)}$ values from S_m and S_k
Step 10: Compute frequency value $f^{j^{v(i_m,T_n)}}$, $f^{j^{v(i_k,T_n)}}$ for each $j^{v(i_m,T_n)}, j^{v(i_k,T_n)}$ values
Step 11: Compute g^{i_m}, g^{i_k}

$$g^{i_m} = \sum_{j=1}^{l} j^{v(i_m,T_n)} * f^{j^{v(i_m,T_n)}}$$

$$g^{i_k} = \sum_{j=1}^{p} j^{v(i_k,T_n)} * f^{j^{v(i_k,T_n)}}$$

Step 12: If $g^{i_m} \geq g^{i_k}$ then change $v(i_m, T_n)$ otherwise change $v(i_k, T_n)$
Step 13: $o(i_m, T_n) = \max_{(i_m \in s_i, T_n \in j^{v(i_m,T_n))}}(u(i, T))$
Step 14: modify $o(i_m, T_n)$ with

$$o(i_m, T_n) = \begin{cases} 0, & \text{if } u(i_m, T_n) < diff \\ o(i_m, T_n) - \left\lceil \frac{diff^2}{s(i_m) * \alpha} \right\rceil, & \text{if } u(i_m, T_n) > diff \end{cases}$$

Step 15: $o((i_m, T_n), (i_k, T_n)) = \max_{(i \in s_i, s_i \subseteq T)}(u(i, T))$ repeat Step 14
Step 16:

$$diff = \begin{cases} diff - u(i_m, T_n), & \text{if } u(i_m, T_n) < diff \\ 0, & \text{if } u(i_m, T_n) > diff \end{cases}$$

$\}$
Step 17: return the sanitized database D'

The proposed algorithm shows that the itemset s_i contains two items and we find these two items utility values and check which one is high. If both items utility value is the same, then we change that item's frequency value which is described in Steps 7-14, otherwise we go to Step 15. The same utility items frequency values are sorted in ascending order and top most value is selected. The top most items frequency values are determined and then two parametric values are found by multiplying the top most items frequency with top most items value. After that, we compare both parametric values and based on that we choose the item and change that item's frequency value.

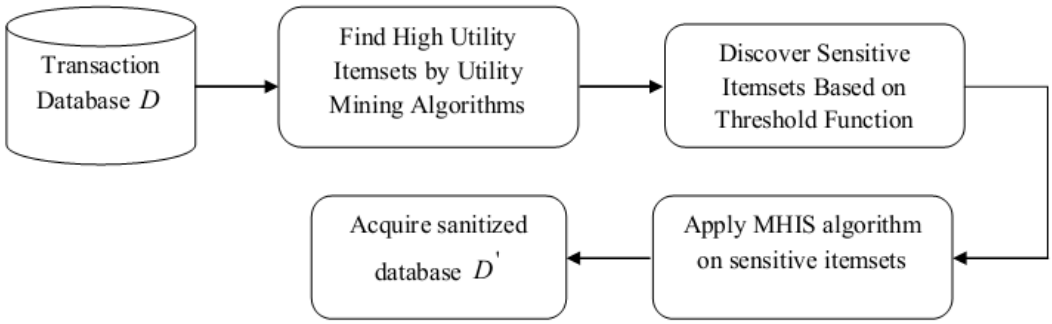


FIGURE 1. Structure of the proposed MHIS algorithm

The proposed MHIS algorithm hides the sensitive itemsets having high utility values, so that the adversaries cannot mine such sensitive itemsets from the database. The proposed MHIS algorithm refinement process is illustrated in Figure 1. The following example illustrates the proposed method process with numerical values.

Example 4.1. *Let us consider a transaction database with five number of transactions and three different items with their external utility values are shown in Table 1.*

TABLE 1. Transaction database with external utility value

TID	**A**	**B**	**C**
T1	0	0	1
T2	1	1	0
T3	6	0	3
T4	1	0	1
T5	0	1	1
External Utility value	3	10	6

By using the above transaction table we find the high utility itemsets are

High Utility Itemsets	Utility Value
AC	45
C	36

In our example we set the threshold value $\alpha = 30$.

After that, we compared the utility values of both items sets A and C in the transactions T3 and T4.

Item / TID	A	C
T3	6	3
T4	1	1

The utility value of the items A and C is high in transaction T3 and both items have the same value as 18. So we select any one of the items value to be changed by using the MHIS algorithm. Initially we find the frequency value of the items A and C and sort the values in descending order. After that we select top j (here: $j = 1$) values and find

the frequency value of the j values. In our example $j^{v(A,T_n)} = 6$, $j^{v(C,T_n)} = 3$ and their frequency value $f^{j^{v(A,T_n)}}$, $f^{j^{v(C,T_n)}} = 1$. Then g^A, g^C as,

$$g^A = 6 * 1 = 6$$

$$g^C = 3 * 1 = 3$$

If the g^A value is greater than the g^C (6>3), so the g^A value to be changed. Based on the new value of A the itemsets AC utility value to be calculated and compared with the threshold value.

5. **Experimental Results.** The proposed MHIS algorithm is implemented in the working platform of MATLAB version 7.12. The performance of our proposed MHIS algorithm is measured by conducting experiments on two datasets. In those datasets, our proposed MHIS algorithm finds the sensitive itemsets that have high utility than our specified minimum utility threshold value α. The sensitive itemsets are mined from the datasets and the corresponding itemsets items utility value is changed by utilizing IS.

5.1. **Dataset description.** In this paper, we have utilized two datasets for the performance analysis of our proposed MHIS algorithm. The dataset I and dataset II contain 100 transactions with 10 and 20 different items. These two datasets are described in Table 2.

Table 2. Dataset description

Dataset	**Number of transactions**	**Distinct items**
Dataset I	100	10
Dataset II	200	20

Table 3. Performance of our proposed MHIS algorithm for different minimum utility threshold values

Threshold values (α)	**Dataset I**			**Dataset II**		
	HF	**MC**	**DS**	**HF**	**MC**	**DS**
2000	7.00	0.00	3.24	8.00	0.00	3.12
2500	0.00	19.89	2.76	0.00	0.00	3.43
3000	0.00	21.50	4.88	0.00	11.23	4.97
3500	0.00	25.56	8.43	0.00	27.43	7.65
4000	0.00	31.26	9.43	0.00	38.54	10.00

Table 4. Performance of conventional HHUIF algorithm for different minimum utility threshold values

Threshold values (α)	**Dataset I**			**Dataset II**		
	HF	**MC**	**DS**	**HF**	**MC**	**DS**
2500	11.00	10.00	1.08	23	0.00	1.06
3000	22.00	14.23	1.03	12	7.00	1.09
3500	7.00	19.34	4.97	8	12.35	2.56
4000	0.00	20.00	7.28	3	25.66	5.45
4500	0.00	24.53	2.9	0.00	36.87	2.44

5.2. **Performance analysis.** The effectiveness of our proposed technique is analyzed by invoking some performance measures given in [18]. Moreover, our proposed MHIS algorithm performance is compared with the conventional HHUIF algorithm. The performance analysis is carried out by changing the minimum utility threshold α as 2000, 2500, 3000, 3500 and 4000. The performance measures of our proposed and conventional algorithms are shown in the following Tables 3 and 4.

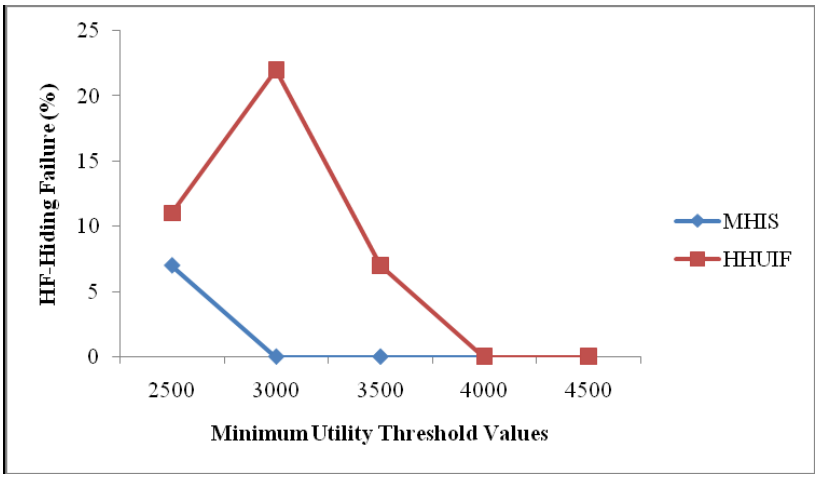
The performance measures are described below,

(i) Hiding Failure (HF)

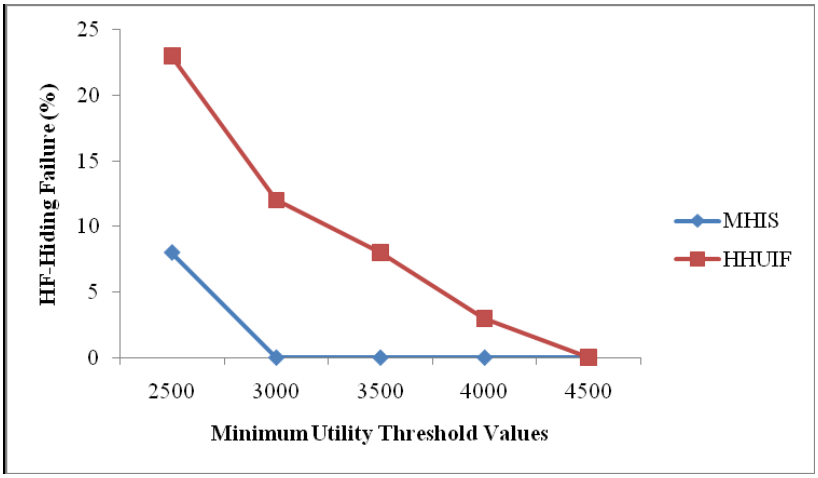
Hiding failure measures the percentage of sensitive itemsets discovered from D'. The HF is measured by the sensitive itemsets of both the original database and the sanitized database, which is stated as follows:

$$HF = \frac{|H(D')|}{|H(D)|} \tag{3}$$

In Equation (3), $H(D)$ and $H(D')$ represent the sensitive itemsets from original database D and the sensitive itemsets from sanitized database D', respectively.

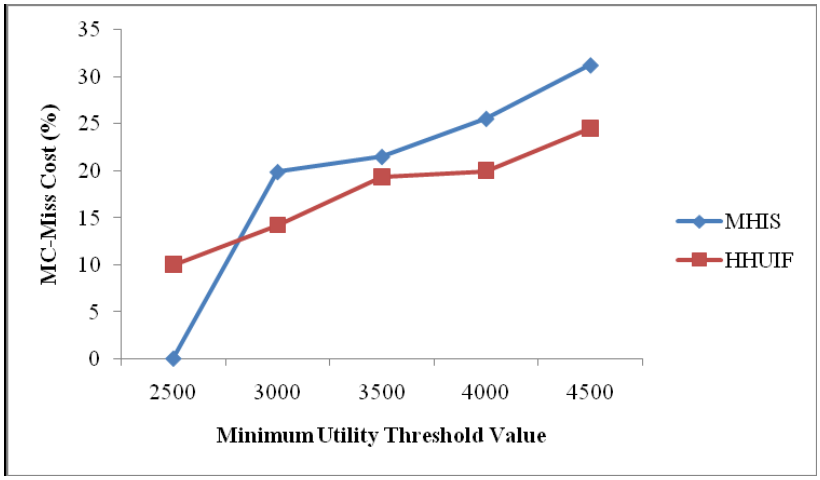


(a)

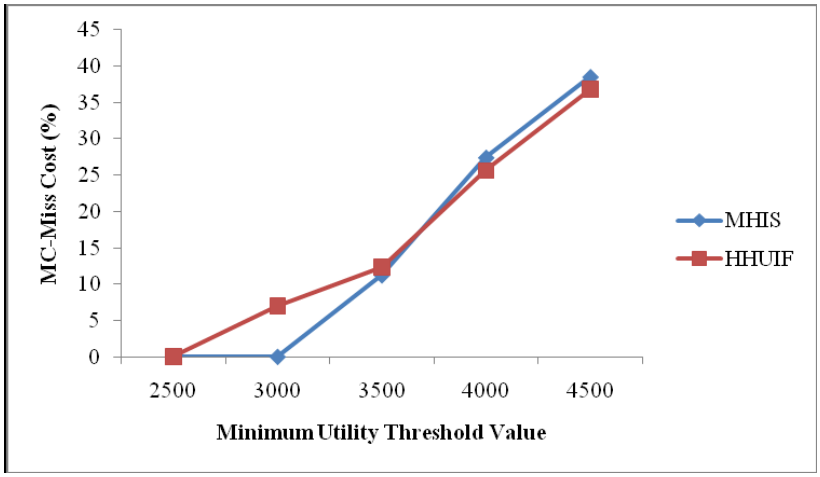


(b)

FIGURE 2. Graphical representation of proposed MHIS and existing HHUIF algorithms performance in terms of their measure HF (a) Dataset I (b) Dataset II



(a)



(b)

FIGURE 3. Graphical representation of proposed MHIS and existing HHUIF algorithms performance in terms of their measure MC (a) Dataset I (b) Dataset II

(ii) Miss Cost (MC)

Miss cost measures the difference ratio of valid itemsets presented in the original database and the sanitized database. The Miss Cost value is computed as,

$$MC = \frac{|n(D) - n(D')|}{|n(D)|} \tag{4}$$

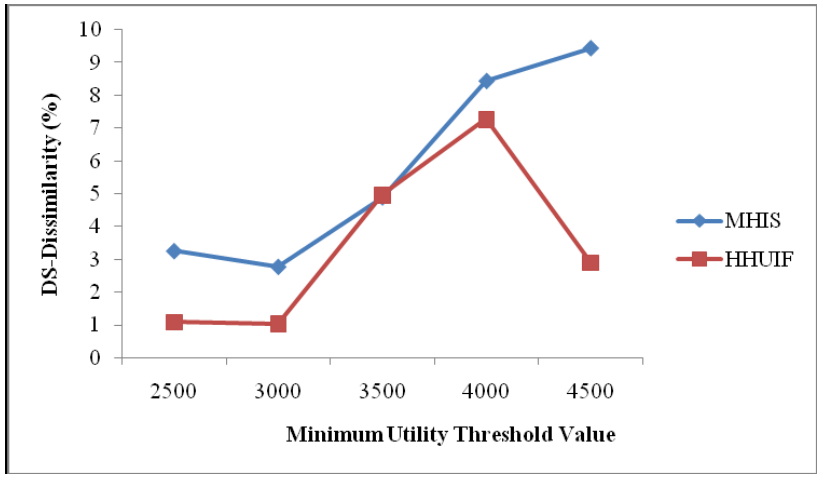
where, $n(D)$ and $n(D')$ denote the non-sensitive itemsets discovered from the original database D and the sanitized database D', respectively.

(iii) Dissimilarity (Diff)

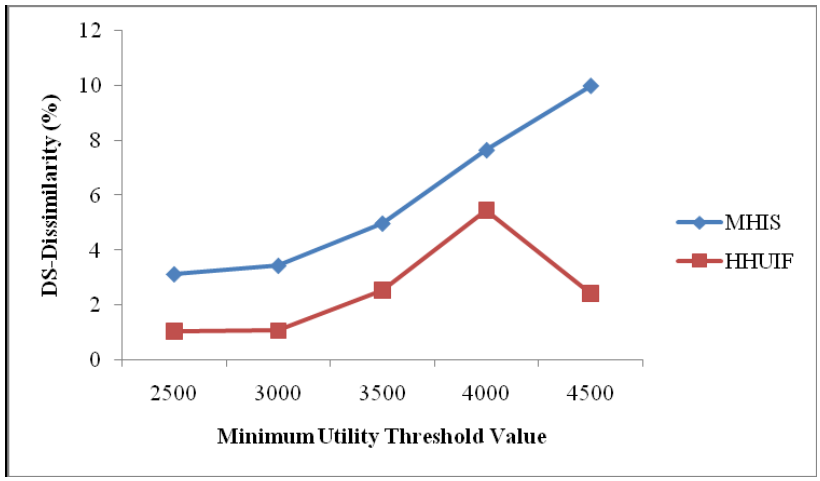
The dissimilarity between the original database D and the sanitized database D' is calculated as,

$$DS = \frac{1}{\sum_{i=1}^{m} f_D(i)} \left(\sum_{i=1}^{m} [f_D(i) - f'_D(i)] \right) \tag{5}$$

where, $f_D(i)$ and $f_{D'}(i)$ represent the frequency of the i^{th} item in the database D and the frequency of the i^{th} item in the database D'.



(a)



(b)

FIGURE 4. Graphical representation of proposed MHIS and existing HHUIF algorithms performance in terms of their measure DS (a) Dataset I (b) Dataset II

As can be seen from Tables 3 and 4, the performance measure shows that our proposed algorithm has offered higher performance compared with the conventional algorithm. The hiding failure value of MHIS algorithm is lower than the conventional HHUIF algorithm. The low value of HF shows that our proposed technique hides the sensitive items more efficiently than the conventional HHUIF algorithm [20]. Similarly, the MC and the dissimilarity values of our proposed MHIS algorithm are also lower than the conventional HHUIF algorithm.

Figures 2, 3 and 4 show the graphical representation of our proposed and conventional technique performance in HF, MC and DS performance measures for different minimum threshold values.

Figures 2, 3 and 4 illustrate the performance of MHIS and HHUIF algorithms in different minimum utility threshold values with different performance measures. The performance measure HF value of our proposed technique is lower when compared with HHUIF. This low value illustrates that our MHIS algorithm performs the sanitization process perfectly than the HHUIF. Moreover, the high MC value shows that our sanitization database contains more valid items than the original database. The MC value is low

for the HHUIF algorithm. Also, the *DS* measure shows that our MHIS algorithm removes the sensitive items and its corresponding sensitive transactions. As we can know from these graphs, our proposed technique has offered high performance in different minimum utility threshold values with different performance measures. Thus, our proposed MHIS algorithm efficiently hides the sensitive itemsets from the original database and provides a database with the non sensitive itemsets.

6. **Conclusion.** In this paper, the MHIS privacy preserving utility mining algorithm for hiding the high utility sensitive itemsets was proposed by exploiting the Item Selector (IS). The successfully enhanced MHIS algorithm hides the sensitive itemsets from the adversaries even when the items utility value is same or different. Our proposed technique first presents a privacy preserving utility mining (PPUM) model and builds up an MHIS algorithm to reduce the impact on the source database of privacy preserving utility mining. This algorithm modifies the database transactions containing sensitive itemsets to minimize the utility value below the given threshold while preventing reconstruction of the original database from the sanitized one. The experimental results proved that the performance of our proposed MHIS algorithm was better than the conventional HHUIF algorithm.

REFERENCES

[1] M. S. Al-Ahmadi, Privacy-preserving data mining for horizontally-distributed datasets using EGADP, *Journal of Communications of the IBIMA*, vol.5, no.2, pp.7-15, 2008.

[2] A. Evfimievski, J. Gehrke and R. Srikant, Limiting privacy breaches in privacy preserving data mining, *Proc. of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database System*, pp.211-222, 2003.

[3] V. Ciriani, S. D. C. di Vimercati, S. Foresti and P. Samarati, *Chapter 1 K-Anonymous Data Mining: A Survey*, Springer, 2008.

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

[5] A. Evfimievskia, R. Srikantb, R. Agrawalb and J. Gehrke, Privacy preserving mining of association rules, *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Canada, pp.1-12, 2002.

[6] D. Saxena, Privacy of data, preserving in data mining, *International Journal of Scientific & Engineering Research*, vol.2, no.3, pp.1-5, 2011.

[7] K. S. Rao and V. Chiranjeevi, Distortion based algorithms for privacy preserving frequent item set mining, *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol.1, no.4, pp.16-27, 2011.

[8] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin and Y. Theodoridis, State-of-the-art in privacy preserving data mining, *Newsletter ACM SIGMOD Record*, vol.33, no.1, pp.1-8, 2004.

[9] I. Cano, S. Ladra and V. Torra, Evaluation of information loss for privacy preserving data mining through comparison of fuzzy partitions, *Proc. of the IEEE World Conference on Computational Intelligence*, Barcelona, Espana, 2010.

[10] R. R. Rajalaxmi and A. M. Nataraja, An effective data sanitization approach for privacy preserving utility itemset mining, *International Journal of Engineering Research & Industrial Applications*, vol.1, no.6, pp.133-143, 2008.

[11] V. Podpecan, N. Lavrac and I. Kononenko, A fast algorithm for mining utility-frequent itemsets, *Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2007.

[12] K. Duraiswamy, D. Manjula and N. Maheswari, Advanced approach in sensitive rule hiding, *Morden Applied Science*, vol.3, no.2, pp.98-107, 2009.

[13] E. Poovammal and M. Ponnavaikko, Utility independent privacy preserving data mining on vertically partitioned data, *Journal of Computer Science*, vol.5, no.9, pp.666-673, 2009.

[14] E. Poovammal and M. Ponnavaikko, Privacy and utility preserving task independent data mining, *International Journal of Computer Applications*, vol.1, no.15, pp.104-111, 2010.
[15] A. George and D. Binu, DRL-PREFIXSPAN a novel pattern growth algorithm for discovering downturn, revision and launch (DRL) sequential patterns, *Central European Journal of Computer Science*, vol.2, no.4, pp.426-439, 2012.
[16] L. Rani and N. Revathi, Enhancing the utility of generalization for privacy preserving re-publication of dynamic datasets, *International Journal of Computer Applications*, vol.13, no.6, pp.42-49, 2011.
[17] G. Li and Y. Wang, A new method for privacy-preserving data mining based on weighted singular value decomposition, *Journal of Convergence Information Technology*, vol.6, no.3, pp.28-34, 2011.
[18] S. R. M. Oliveira and O. R. Zaiane, Privacy preserving frequent itemset mining, *Proc. of the IEEE International Conference on Privacy, Security and Data Mining*, vol.14, pp.43-54, 2002.
[19] A. George and D. Binu, An approach to products placement in supermarkets using PrefixSpan algorithm, *Journal of King Saud University – Computer and Information Sciences*, vol.25, no.1, pp.77-87, 2013.
[20] J.-S. Yeh and P.-C. Hsu, HHUIF and MSICF: Novel algorithms for privacy preserving utility mining, *Expert Systems with Applications*, vol.37, pp.4779-4786, 2010.