An Optimization Based Modified Maximum Sensitive Item-Sets Conflict First Algorithm (MMSICF) for Hiding Sensitive Item-Sets

D.Jaya Kumari
Research scholar, CS &SE Dept
Andhra University, Visakhapatnam
Assoc.Professor, RRSCET, HYD

N.V.E.S.Murthy
Department of CS & SE
Andhra University
Vishakapatnam-530003, India

S.S.Suresh
Department of AS & CT
I^{2}IT, Pune - 411057, India

ABSTRACT

In privacy preserving data mining, utility mining plays an important role. In privacy preserving utility mining, some sensitive itemsets are hidden from the database according to certain privacy policies. Hiding sensitive itemsets from the adversaries is becoming an important issue nowadays. The existing paper utilized two algorithms; such as HHUIF and MSICF are conceal the sensitive itemsets, so that the adversaries cannot mine them from the modified database. But, the performance of this method lacks if the utility value of the items are same. To solve this problem, in this paper a Modified MSICF algorithm (MMSICF) is proposed. The proposed MMSICF algorithm is a modified version of existing MSICF algorithm. The MMSICF algorithm computes the sensitive itemsets by utilizing the user defined utility threshold value. The threshold value selection plays a major role in this paper and it is determined by the hybridization of Artificial Bee Colony (ABC) and Genetic Algorithm (GA). In order to hide the sensitive itemsets, the frequency value of the items is changed. The proposed MMSICF 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.

General Terms

Data Mining, Privacy

Keywords

Utility Mining, Privacy Preserving Utility Mining, Sensitive Itemsets, Utility Value, Frequency Value, Maximum Sensitive Itemsets Conflict First (MSICF) .

1. INTRODUCTION

The collection of digital information by governments, corporations, and individuals has created tremendous opportunities for knowledge-based decision making. Driven by mutual benefits, or by regulations that require certain data to be published, there is a demand for the exchange and publication of data among various parties [1]. Today's globally networked society demands the dissemination and sharing of large amount of sensitive data. Such data is an important asset to business organizations and governments for decision making processes and providing social benefits [2]. These are the collection of data known as data mining. Data mining is the process of extracting patterns from data. It has become an increasingly important tool for transforming data into information. However, with the rapid development of data mining technologies, preserving data privacy posed and increasing challenge to data mining applications in many areas [3]. Some of the organization needs privacy for the original data. So recently all the organizations are utilizing the Privacy preserving Utility Mining (PPUM) for the security purpose. Many data mining applications deal with privacy- sensitive data. It is randomly perturbing the data while preserving the underlying probabilistic properties [4]. Privacy is usually measured using some form of disclosure risk, while the data utility is traditionally measured as information loss between the original data and the transformed sanitized data [5].

The problem of privacy preserving data mining has become more important in recent years because of the increasing ability to store personal data about users and the increasing sophistication of data mining algorithm to leverage this information. A number of techniques have been suggested in recent years in order to perform privacy preserving data mining [6]. PPUM research usually takes one of the three philosophical approaches: (i) data hiding (ii) rule hiding (iii) secure multiparty computation. Its main goal is to develop efficient algorithm that allow one to extract relevant knowledge from a large amount of data, while prevent sensitive data and information from disclosure or inference [7]. The former notion of data quality is strictly related to the use the data are intended for. Moreover, some of those algorithms can be computationally very expensive and thus cannot be used when very large sets of data need to be frequently released. Therefore, in addition to data quality, performance also needs to be carefully assessed [8]. The first type of privacy, termed as output privacy, is that the data is altered so that the mining result will conserve certain privacy. Many modification techniques such as perturbation, blocking, aggregation, swapping and sampling are used for this type of privacy [9] [10]. The second type of privacy, labeled as input privacy, is that the data is manipulated so that the mining result is not affected or less affected. The cryptography based and reconstruction based techniques are used for this type of privacy [11] [12].

2. REVIEW OF RELATED WORKS

In 2009, Mohammad Naderi Dehkordi *et al.* [13] have presented the Extracting of knowledge form large amount of data was an important issue in data mining systems. One of most important activities in data mining was association rule mining and the new head for data mining research area was privacy of mining. A lot of researches have done in the area but most of them focused on perturbation of original database heuristically. Therefore the final accuracy of released database falls down intensely. In addition to accuracy of database the main aspect of security in this area was privacy of database that is not warranted in most heuristic approaches, perfectly. They introduced new multi-objective method for hiding sensitive association rules based on the concept of genetic algorithms. The main purpose of the method was fully supporting security of database and keeping the utility and certainty of mined rules at highest level.

In 2011, Vijayarani *et al.* [14] have discussed about the association rule hiding problem. Association rule mining, one of the very important data mining techniques. The process of discovering itemsets that frequently co-occur in a transactional database so as to produce significant association rules that hold for the data was known as Association rule mining. This process was modifying the original database by hiding the sensitive data to protect the sensitive association rules. In the paper, they have proposed Artificial Bee Colony optimization algorithm for hiding the sensitive association rules. They analyzed the efficiency of the Artificial Bee Colony optimization technique by using various performance factors.

In 2011, Xinjun Qi *et al.* [15] have proposed the privacy preserving data mining issue. They also make a classification for the privacy preserving data mining, and analyze some works in this field. Data distortion method for achieving privacy protection association rule mining and privacy protection data release were focused on discussion. Detailed evaluation criteria of privacy preserving algorithm was illustrated, which include algorithm performance, data utility, privacy protection degree, and data mining difficulty. Finally, the development of privacy preserving data mining for further directions is prospected.

In 2010, Jieh-Shan Yeh *et al.* [16] have proposed the privacy preserving utility mining (PPUM) with two novel algorithms, HHUIF and MSICF, to achieve the goal of hiding sensitive itemsets so that the adversaries could not mine them from the modified database. The work also minimizes the impact on the sanitized database of hiding sensitive itemsets. The experimental results show that HHUIF achieves lower miss costs than MSICF on two synthetic datasets. On the other hand, MSICF generally has a lower difference ratio than HHUIF between original and sanitized databases.

3. THE PROPOSED MODIFIED PRIVACY PRESERVING UTILITY MINING ALGORITHM

The proposed method is modified Privacy Preserving Utility Mining algorithm for selecting the minimum itemsets from the original database. The process is mentioned below,

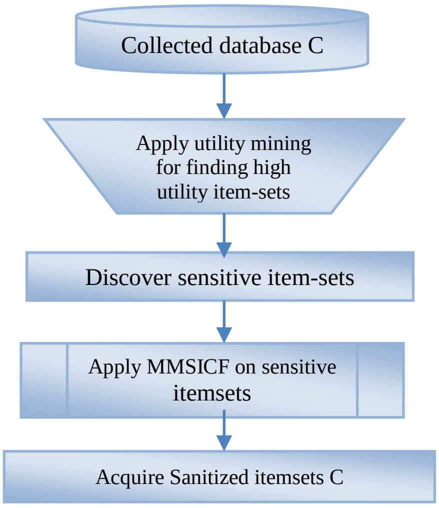
Collected database C
Apply utility mining for finding high utility item-sets
Discover sensitive item-sets
Apply MMSICF on sensitive itemsets
Acquire Sanitized itemsets C

Figure 1: Structure of the Proposed MMSICF algorithm

Let C be the transaction database, containing a set of transactions $C = \{T_1, T_2, T_3, \cdots T_Y\}$, where y is the total number of transactions. The database D contains a set of items, which is denoted as $J = \{i_1, i_2, i_3, \cdots i_X\}$, where x is the total number of items in the database. Each transaction T_y 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, $E = \{e(i_1), e(i_2), e(i_3), \cdots e(i_X)\}$ where $e(i_X)$ is the external utility value of an item $i_X ; i_X \subseteq J$ the frequency value of each item i_X in transaction T_Y is $v(i_X, T_Y)$ is the number of items i_m acquired in transaction T_Y.

3.1 Utility Mining Algorithm

Utility mining is used to find all the itemset's utility values. The utility value of item i_X in transaction T_Y is defined as,

$$a(i_X, T_Y) = e(i_Y) \times b(i_X, T_Y) \qquad (1)$$

The utility value of an itemset P in transaction T_Y is denoted as,

$$a(P, T_Y) = \sum_{i_X \in P} a(i_X, T_Y) \qquad (2)$$

Then, find the itemsets whose utility value is higher than the user specified threshold value ε, where, the minimum utility threshold is ε. The itemset P is a high utility itemset, if $a(P) \geq \varepsilon$. These high utility itemsets are stored in $K = \{s_1, s_2, \cdots s_m\}$ 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 [16] has utilized two algorithms: HHUIF and MSICF. Amongst these two algorithms, MSICF produces lower DS than the HHUIF and the MSICF algorithm is described below.

3.2 Obtainable MSICF Algorithm

The main objective of the MSICF algorithm is to diminish the utility value of each sensitive itemset by modifying the quantity values of items which has the maximum conflict count among items in the sensitive itemsets. The main process in this algorithm is to provide a new Icount values for the formatted data set. After selecting the Icount values arrange them in any predefined order. The pseudo code of the MSICF algorithm is given below:

Algorithm 1: MSICF Algorithm

```
Input: the original database $C$ ; the minimum utility threshold $\varepsilon$ ; the
        sensitive itemsets $K = \{s_1, s_2, \cdots s_m\}$
Output: the sanitized database $C'$ so that $s_m$ cannot be mined
Calculate $Icount_{i_m}(K)$ for all S
Arrange $i_m$ by decreasing order of $Icount_{i_m}(K)$
for each sensitive itemset $s_m \in C$
    $diff = a(s_m) - \varepsilon$ // the utility value needs to be reduced
    while $(diff > 0)$     {
        $o(i_X, T_Y) = \arg\max_{(i \in s_m, s_m \subseteq T)} (a(i,T))$
        modify $o(i_X, T_Y)$ with
        $o(i_X, TY) = \begin{cases} 0 & ,if \quad a(i_X, T_Y) < diff \\ o(i_X, T_Y) - \left\lceil \frac{diff}{s(i_X)} \right\rceil & ,if \quad a(i_X, T_Y) > diff \end{cases}$
        $diff = \begin{cases} diff - a(i_X, T_Y) & ,if \quad a(i_X, T_Y) < diff \\ 0, & ,if \quad a(i_X, T_Y) > diff \end{cases}$
    }
return the sanitized database $C'$
```

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

3.3 Problem Formulation

The MSICF algorithm hides the sensitive itemsets having high utility value. But, the drawback of this algorithm is that, if the items in the sensitive itemsets having same utility value, then it will decrease the hiding performance. For example, $\{A, B\}$ is a sensitive itemset $(\varepsilon = 120)$, 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, in this paper a Modified MSICF algorithm with Item Selector (IS) (MMSICF) is proposed. 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 is modified. The developed IS will reduce the computation complexity as well as improves the hiding performance of the itemsets.

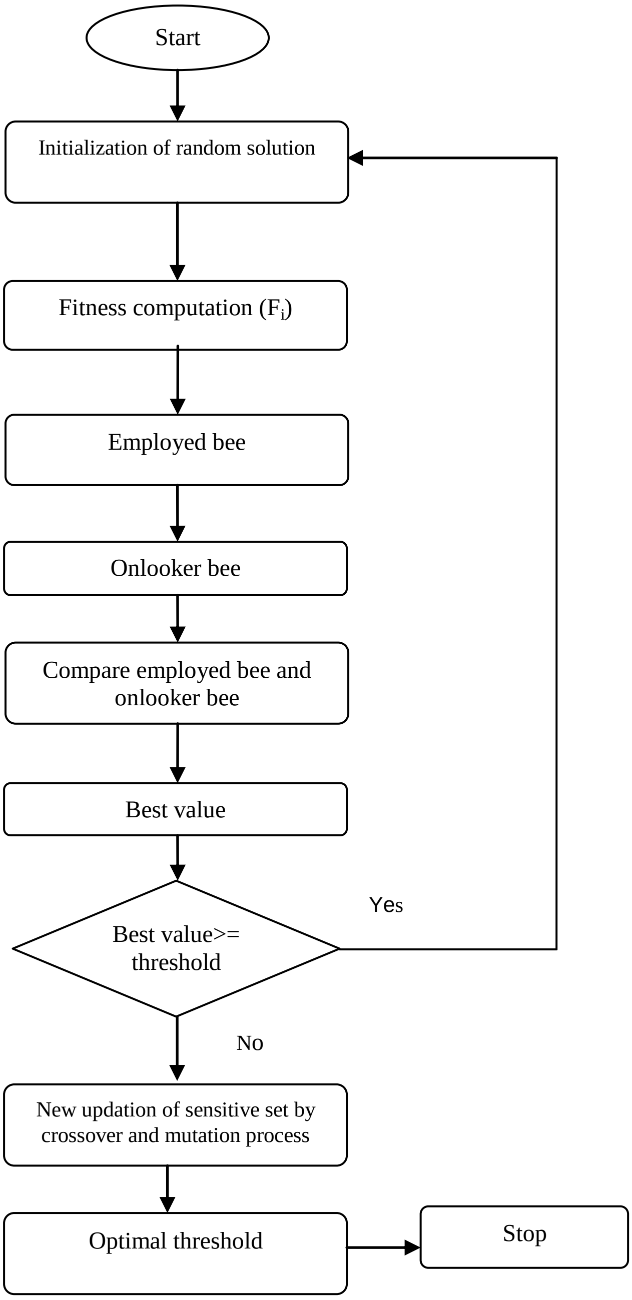
Start
Initialization of random solution
Fitness computation (F$_i$)
Employed bee
Onlooker bee
Compare employed bee and onlooker bee
Best value
Best value>= threshold
Yes
No
New updation of sensitive set by crossover and mutation process
Optimal threshold
Stop

Fig 2: flow chart of in threshold proposed method evaluation

4. MODIFIED MSICF ALGORITHM WITH ITEM SELECTOR (MMSICF)

The main objective of the MMSICF 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 ε. This hiding process is repeated the until all sensitive itemsets utility value become lower than the threshold value ε. The proposed MMSICF algorithm is described below.

Algorithm 2: Modified MSICF Algorithm

Input: the original database C ; the minimum utility threshold ε ; the sensitive itemsets $K = \{s_1, s_2, \cdots s_m\}$

Output: the sanitized database C' so that S_m cannot be mined.

Finding threshold value $\varepsilon = \alpha \sum_{n=0}^{X} \sum_{m=0}^{Y} (T_n(m) * T_n(m)')\beta$

The aforementioned threshold computation process is done with the aid of hybrid technique of artificial bee colony (ABC) and genetic algorithm (GA),

```
1.  Calculate $Icount_{i_m}(K)$ for all S
2.  Arrange $i_m$ by decreasing order of $Icount_{i_m}(K)$
3.  for each sensitive itemset $s_m \in K$
4.  $diff = a(s_m) - \varepsilon$ // the utility value needs to be reduced
5.  while $(\text{diff} > 0)$ {
6.   if $S_m$ contains two items $s_m \subseteq (i_Z, i_X)$
7.   Compare $a(i_Z, T_Y)$ and $a(i_X, T_Y)$
8.   if $a(i_X, T_Y) = a(i_Z, T_Y)$ go to step 9 otherwise go to step 17
9.  Select $i_Z, i_X$ items frequency values $b(i_X, T_Y)$ and $b(i_Z, T_Y)$
10.  Sort $b(i_X, T_Y)$ and $b(i_Z, T_Y)$ frequency values in descending order and sored in $S_X$ and $S_Z$
11.  Select top $n^{b(i_X, T_Y)}, n^{b(i_Z, T_Y)}$ values from $S_X$ and $S_Z$
12. Compute frequency value $f^{n^{b(i_X, T_Y)}}, f^{n^{b(i_Z, T_Y)}}$ for each $n^{b(i_X, T_Y)}, n^{b(i_Z, T_Y)}$ value
13.  compute $P^{i_X}, P^{i_Z}$
       $P^{i_X} = \sum_{n=1}^{l} n^{b(i_X, T_Y)} * f^{n^{b(i_X, T_Y)}}$
       $P^{i_Z} = \sum_{n=1}^{p} n^{b(i_Z, T_Y)} * f^{n^{b(i_Z, T_Y)}}$
14. If $P^{i_X} \geq P^{i_Z}$ then change $b(i_X, T_Y)$ otherwise change $b(i_Z, T_Y)$
15.
     $o(i_X, T_Y) = \max_{(i_X \in s_m, T_Y \in n^{b(i_X, T_Y)})}(a(i, T))$
16. modify $o(i_X, T_Y)$ with
     $o(i_X, T_Y) = \begin{cases} 0 & , if \;\; a(i_X, T_Y) < diff \\ o(i_X, T_Y) - \left\lceil \frac{diff^2}{s(i_X) * \varepsilon} \right\rceil & , if \;\; a(i_X, T_Y) > diff \end{cases}$
17.
     $o((i_X, T_Y), (i_Z, T_Y)) = \max_{(i \in s_m, s_m \subseteq T)}(a(i, T))$
     repeat again 16.
18.
     $diff = \begin{cases} diff - a(i_X, T_Y) & , if \;\; a(i_X, T_Y) < diff \\ 0, & , if \;\; a(i_X, T_Y) > diff \end{cases}$
19. return the sanitized database $C'$
```

In this threshold value formula, where

$T_n(m)$ - Transaction value,

$T_n(m)'$ - Utility value, α, β - Weight age value.

The proposed algorithm shows that the item-set S_m Contains two items and their utility values are found and check for higher utility value. If suppose both items utility values are same, then find frequency values for each item-sets otherwise it will move to the final condition modified step. The same utility items frequency values are sorted in descending order and top most value is selected. The top most items frequency values are determined and find two parametric values are found by multiplying the top most items frequency with top most items value. After that, we compare both parametric values based on the item chosen and change that item frequency value.

The proposed MMSICF algorithm hides the sensitive item-sets having high utility values, so the adversaries cannot mine such sensitive item-sets from the database.

Example

Let us consider a transaction database with three numbers 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	3	0	6
T4	1	0	1
T5	0	1	1
T6	1	0	0
External Utility Value	6	10	3

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

Utility Item Set	Utility Values
AB	16
AC	45
BC	13
A	36
B	20
C	27

High Utility Item Set	Utility Values
AC	45
A	36

In this example we set the threshold value = 30.

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

TID Item	A	C
T3	3	6
T4	1	1

The utility value of the items A and C is high in transaction T3and both items have the same value as 18. So we select any one of the items value to be changed by using the MMSICF 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 m (here: m = 1) values and find the frequency value of the m values.

In our example $n^{b(A;Tn)} = 3$, $n^{b(C;Tn)} = 6$ and their frequency value $f n^{b(A,,Tn)}, f n^{b(C,,Tn)} = 1$. Then P^A, P^C as,

$P^A = 3*1=3$
$P^C = 6*1=6$

The P^C value is greater than the P^A (6>3), so the P^A value to be changed in the data set. Based on the new value of C the item sets AC utility value to be calculated and compared with the threshold value.

5. EXPERIMENTAL RESULTS

The proposed MMSICF algorithm is implemented in the working platform of MATLAB version 7.12. The performance of the proposed MMSICF algorithm is measured by conducting experiments on one dataset. In the dataset, the proposed MMSICF algorithm finds the sensitive item-sets that have high utility than our specified minimum utility threshold value. The sensitive item-sets are mined from the dataset and the corresponding item-sets items utility value is changed by utilizing IS.

5.1 Dataset Description

In this paper, two Datasets are utilized for the performance analysis of proposed MMSICF algorithm. The dataset I contains 100 transactions with 10 different items and dataset II contains 200 transactions with 10 different items. Dataset is described in Table 2.

Table 2: Dataset Description

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

5.2 Performance Analysis

The effectiveness of proposed technique is analyzed by invoking some performance measures given in [17]. Moreover, the proposed MMSICF algorithm performance is compared with the conventional MSICF algorithm. The performance analysis is carried out by changing the minimum utility threshold as 1000, 1500, 2000, 2500 and 3000. The performance measures of the proposed and conventional algorithms are shown in the following Table 2 and Table 3. 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,

$$HF = \frac{|H(C')|}{|H(C)|} \qquad (3)$$

In Eqn. (3), $H(C)$ and $H(C')$ represents the sensitive itemsets from original database C and the sensitive itemsets from sanitized database C', respectively.

(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,

$$MC = \frac{|\delta(C) - \delta(C')|}{|\delta(C)|} \qquad (4)$$

Where, $\delta(C)$ and $\delta(C')$ denotes the non-sensitive itemsets discovered from the original database C and the sanitized database C', respectively.

(iii) Dissimilarity (Diff)

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

$$DS = \frac{1}{\sum_{m=1}^{X} \varphi_C(m)} \left(\sum_{m=1}^{X} [\varphi_C(m) - \varphi_C{}'(m)] \right) \qquad (5)$$

Where, $\varphi_C(m)$ and $\varphi_{C'}(m)$ represents the frequency of the m^{th} item in the database C and the frequency value of the m^{th} item in the database C'.

Table 3: Performance comparison between proposed MMSICF algorithm and Conventional MSICF algorithm for data set I

TH Range	MMSICF				MSICF[16]			
	'TH'	'HF'	'MC'	'Dis'	'TH'	'HF'	'MC'	'Dis'
100-500	469	0.66255	1.40740	0	500	0.88983	1.36864	0
500-1000	994	0.503546	3.148936	2.40E-217	1000	0.810219	3.080292	0
1000-1500	1414	0.464789	7.239437	6.27E-106	1500	0.633803	6.873239	3.95E-140
1500-2000	1912	0.071429	19.89286	1.68E-11	2000	0.071429	18.96429	1.17E-11
2000-2500	2471	0	21.5	2.15E-09	2500	0	20.5	1.56E-09
2500-3000	2916	0	21.5	1.70E-07	3000	0	20.5	7.57E-08

As can be seen from Table 3, the performance measure shows that the proposed algorithm has offered higher performance compared to the conventional algorithm. The hiding failure value of MMSICF algorithm is lower than the conventional MSICF algorithm. The low value of HF shows that the proposed technique hides the sensitive items more efficiently than the conventional MSICF algorithm [16]. Similarly, the dissimilarity values of our proposed MMSICF algorithm are also low than the conventional MSICF algorithm. But Miss Costs of our proposed system has high compared to conventional MSICF algorithm.

The following figures 3, 4 and 5 shows the graphical representation of the proposed and conventional techniques performance in HF, MC and DS performance measures for different minimum threshold values.

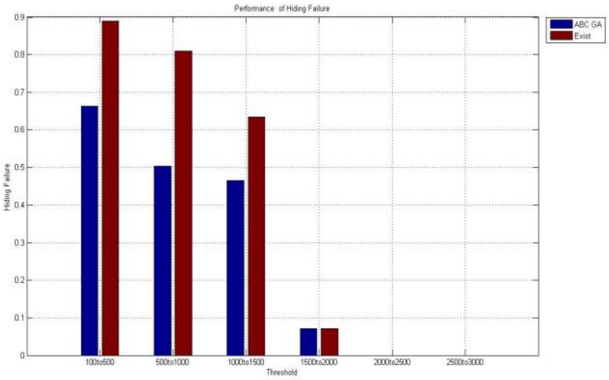

Figure 3: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of Hiding Failure (HF)

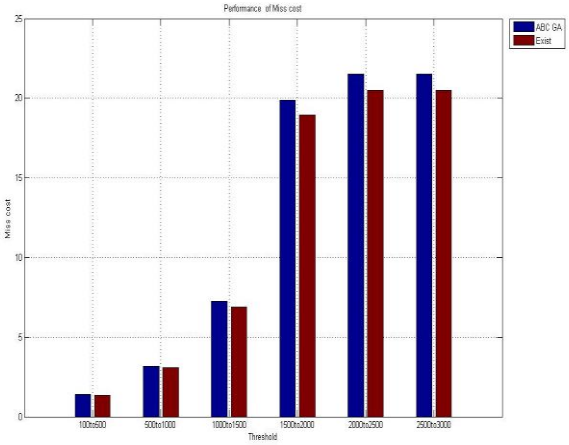

Figure 4: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of Miss cost (MC)

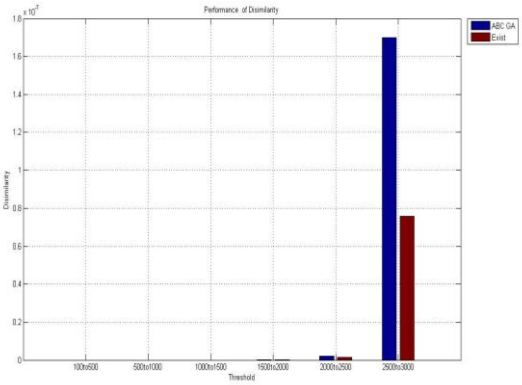

Figure 5: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of DS

Table 4: Performance comparison between proposed MMSICF algorithm and Conventional MSICF algorithm for data set II

TH	MMSICF				MSICF[16]			
Range	'TH’	'HF'	'MC'	'Dis'	'TH’	'HF'	'MC'	'Dis'
100-500	494	0.31671	1.02244	0	500	0.94416	1.00254	0
500-1000	1015	0.148936	4.751773	3.55E-47	1000	0.851351	4.331081	0
1000-1500	1492	0.081633	15.55102	1.25E-43	1500	0.604167	15.5	4.24E-87
1500-2000	1998	0.294118	46.70588	1.16E-12	2000	0.352941	46.05882	1.40E-20
2000-2500	2500	0	114.8571	6.90E-08	2500	0	113.8571	6.57E-08
2500-3000	3000	0	201.75	4.34E-09	3000	0	200.75	3.91E-09

The following figures 6, 7 and 8 shows the graphical representation of the proposed and conventional techniques performance in HF, MC and DS performance measures for different minimum threshold values.

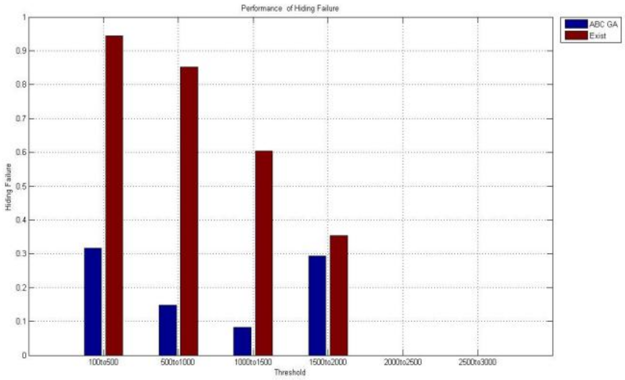

Figure 6: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of Hiding Failure (HF)

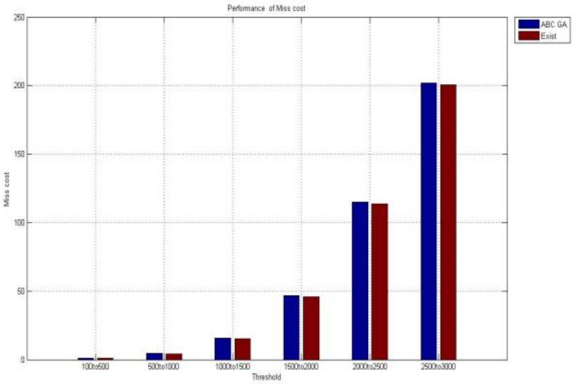

Figure 7: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of Miss cost (MC)

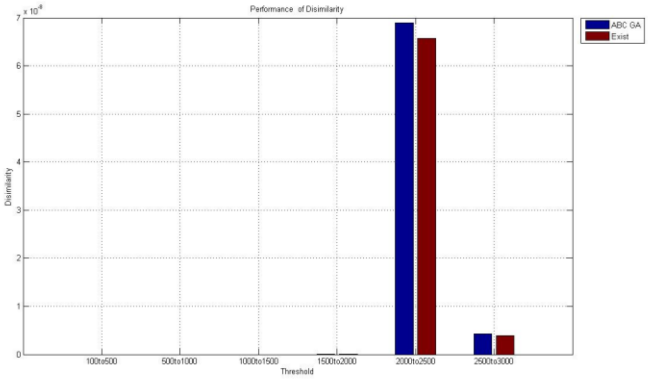

Figure 8: Graphical representation of proposed MMSICF and existing MSICF algorithms performance in terms of DS

The fig. 3, 4, and 5 are shown the performance of MMSICF and MSICF algorithms in different utility threshold values with different performance measures for dataset 1. From the graph the proposed algorithm is shown in the graph as first bar (blue bar) and second bar (brown bar) is the existing algorithm for both dataset 1 and 2. The performance measure HF value of our proposed technique is low when compared to MSICF. This low value illustrates that our MMSICF algorithm performs the sanitization process perfectly than the MSICF. 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 MSICF algorithm. Also, the DS measure shows that our MMSICF algorithm removes the sensitive items and its corresponding sensitive transactions. In the figure 5, the dissimilarity value are very small compared with the other values so it is not visible in the graph and its values are 2.40E-217, 6.27E-106 as well as in the existing also minimum (0,0 and 3.95E-140) or have zero values. 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 MMSICF algorithm efficiently hides the sensitive itemsets form the original database and provides a database with the non sensitive itemsets. The above figures 6, 7 and 8 are shown the graphical representation of our proposed and conventional techniques performance in HF, MC and DS performance measures for different threshold values for dataset 2. The performance measure HF value of proposed technique is low when compared to MSICF. This low value illustrates that the proposed MMSICF with ABC_GA algorithm performs the sanitization process perfectly than the existing MSICF algorithm. Similarly the miss cost and dissimilarity are highest compared with the existing algorithm. In figure 8 the values are very low compared with the other values so that values are not shown in the graph for simplicity and the values are (3.55E-47, 1.25E-43, and 1.16E-12) and (4.24E-87 and 1.4E-20). It is illustrated that the proposed algorithm is given the perfect sanitation process than the existing algorithm.

6. CONCLUSION

In this proposed technique, MMSICF privacy preserving utility mining algorithm for hiding the high utility sensitive item sets by utilizing exploiting the Item Selector (IS). The enhanced MMSICF algorithm successfully hides the sensitive item sets from the adversaries even though the items utility value is similar or non similar. Initially our proposed technique presents a privacy preserving utility mining (PPUM) model and builds up an MMSICF algorithm to reduce the impact on the source database of privacy preserving utility mining. In this paper the proposed part is the estimation of threshold value for the optimal threshold selection. For the selection of threshold values the hybridization of ABC along with GA is proposed. The main advantage of this proposed part of GA is to provide the optimal solution since the value of crossover and mutation are being set to a constant value. This algorithm modifies the database transactions containing sensitive item sets 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 the proposed MMSICF algorithm was better than the conventional MSICF algorithm. In future, by making small modifications in computing threshold process or by changing the optimization algorithm these results can be improved. This in case reduces the computation time taken for the whole process and retrieve better results.

7. REFERENCES

[1] Benjamin C. M. Fung, Ke Wang, Rui Chen And Philip S. Yu, “Privacy-Preserving Data Publishing: A Survey of Recent Developments”, ACM Computing Surveys, Vol. 42, No. 4, Article 14, pp. 1-53, June 2010.

[2] Paramjeet, V. Ravi, Naveen Nekuri and Chillarige Raghavendra Rao, “Privacy preserving data mining using particle swarm optimization trained auto-associative neural network: an application to bankruptcy prediction in banks", Int. J. Data Mining, Modelling and Management, Vol. 4, No. 1, pp. 39-56, 2012

[3] Guang Li and Yadong Wang, “A Privacy-Preserving Classification Method Based on Singular Value Decomposition”, The international Arab Journal of Information Technology, Vol.9, No.6, pp. 529-534, 2012

[4] Hillol Kargupta, Souptik Datta, Qi wang and Krishnamoorthy Sivakumar, “Random Data Perturbation Techniques and A Privacy Preserving Data Mining”, in proceedings of IEEE International Conference on Data Mining, pp. 1-23, 2003

[5] Michal Sramka, "Data Mining as a tool in privacy preserving data publishing", Tatra M t. Math. Publications, Vol.45, pp. 151–159, 2010

[6] Md. Riyazuddin, Dr.V.V.S.S.S.Balaram, Md.Afroze, Md.JaffarSadiq and M.D.Zuber, “An Empirical Study on Privacy Preserving Data Mining”, International Journal of Engineering Trends and Technology, Vol.3, No.6, pp.687-693, 2012

[7] Xiaodan Wu, Chao-HsienChu, Yunfeng Wang, Fengli Liu and Dianmin Yue, "Privacy preserving data mining research: current status and key issues", Springer -Lncs, pp. 762-772, 2007

[8] Elisa Bertino, Igor Nai Fovino and Loredana Parasiliti Provenza, “A Framework for Evaluating Privacy Preserving Data Mining Algorithms”, Data Mining and Knowledge Discovery, Vol.11, pp. 121–154, 2005

[9] C. Clifton, “Using Sample Size to Limit Exposure to Data Mining”, Journal of Computer Security, Vol. 8, pp. 281- 307, Dec. 2000.

[10] Y. Saygin, V.S. Verykios, C. Clifton, “Using Unknowns to Prevent Discovery of Association Rules”, SIGMOD Record, Vol. 30, pp. 45- 54, Dec. 2001

[11] A. Evfimievski, “Randomization in Privacy Preserving Data Mining”, in Proceedings of the SIGKDD Explorations, Vol. 4, pp. 43- 48, Dec. 2002

[12] M. Kantarcioglu, C. Clifton, “Privacy-preserving distributed mining of association rules on horizontally partitioned data”, ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Jun. 2002

[13] Mohammad Naderi Dehkordi, Kambiz Badie and Ahmad Khadem Zadeh, “A Novel Method for Privacy Preserving in Association Rule Mining Based on Genetic Algorithms”, Journal of Software, Vol. 4, No. 6, pp. 555-562, August 2009

[14] M.Sathiya Prabha and S.Vijayarani, “Association Rule Hiding using Artificial Bee Colony Algorithm”, International Journal of Computer Applications, Vol. 33, No.2, pp. 41-47, November 2011

[15] Xinjun Qi and Mingkui Zong, “An Overview of Privacy Preserving Data Mining” , in proceedings of International Conference on Environmental Science and Engineering, Procedia Environmental Sciences, Vol.12, pp. 1341-1347 , 2012

[16] Jieh-Shan Yeh and Po-Chiang Hsu, “HHUIF and MSICF: Novel algorithms for privacy preserving utility mining”, Expert Systems with Applications, Vol. 37, pp. 4779–4786, 2010

[17] Stanley R. M. Oliveira and Osmar R. Zaiane, "Privacy Preserving Frequent Itemset Mining", In Proceedings of the IEEE international conference on Privacy, security and data mining, Vol. 14, pp. 43-54, 2002