Hiding Sensitive Rules with Minimal Compromise of Data Utility

Geetika M. Kalra[1], Hitesh Chhinkaniwala[2]

[1]Department of Computer Engineering, Ganpat University, Mehsana, India
[2]Dean, Shankersinh Vaghela Bapu Institute of Technology, Gandhinagar, India

Abstract- Data mining extracts valuable knowledge from large amounts of data. It is a powerful new technology to help companies focus on the most important information in their data warehouses. Several issues need to be addressed when mining on data is performed that are bulk at size and geographically distributed at various sites. Privacy preserving data mining has emerged as promising way to mining knowledge from large databases securely. The techniques are classified as: data distribution, data modification, and data mining algorithm, rule hiding and privacy preservation. Our approach is based on heuristics of Association Rule Mining. The techniques involves modifying database to prevent sensitive rules from getting disclosed which leads to information loss of non-sensitive data. We propose an algorithm that hides sensitive item on either side of rule by selective modification of database with minimal information loss. The prime objective is the accuracy of algorithm in terms of rule hiding, ghost rules and missing rules.

Index Terms- privacy preserving data mining, association rule, confidence, support.

I. INTRODUCTION

Progress in digital data acquisition and storage technology has resulted in the growth of huge database. The result of this lead to growth in the possibility of tapping these data and extracting from them information that might be of value to the owner of the database. This discipline concerned with the task has become known as data mining. [1] Extraction of useful information from database is definitely a fruitful investment for the owner but it is also a threat for them as this information might get an unauthorized access. This generates the need to provide privacy for it. Privacy preserving data mining solutions aim at achieving a data mining algorithm[2] to use data *without* ever actually "seeing" it.[3,4]

Association analysis is a powerful tool for discovering relationships which are hidden in large database. Mining of association rules between set of items in a large database is carried out using support and confidence as key heuristics. [6,7,8] Apriori algorithm is used for discovering all significant association rules between items in a large database.

Actually any given specific rules to be hidden, many approaches for hiding association, classification and clustering rules have been proposed. Maximum researchers have worked on the basis of reducing the support and confidence of sensitive association rules.[10-19] The association rule items whether in Left Hand Side (LHS) or Right Hand Side (RHS) of the generated rule, should not be disclosed by mining process. [9] ISL and DSR are the common approaches used to hide the sensitive rules.

In our proposed approach we are hiding contain some sensitive information which can be on the Right hand side or Left hand side of the rule, so that rules containing confidential item can't be reveal. This is done on the basis of selective modification of the data. Selective modification achieves higher utility for the modified data given that the privacy is not jeopardized. The technique should minimize hiding failure, ghost rules and lost rules reducing execution time.The rest of the paper is organized as follows. Section 2 presents the statement of the problem and the notation used in the paper. Section 3 presents the proposed algorithms for hiding sensitive items. Section 4 shows example of the proposed algorithm. Section 5 analyses the result of the efficiency of proposed algorithm and shows how it is better than previously defined methods. Concluding remarks and future works are described in section 6.

II. PROBLEM STATEMENT

The goal is to transform a given data set D into modified version D'' that satisfies a given privacy requirement. A database D of transactions can describe the problem in Fig 1. We have a set of transactions T. Let I = {I1, 12, 13, ... Im} be a set of literals, called items. As shown in Fig. 1, the circled items are frequent and the rest are infrequent. Each transaction T is a set of items such that T ⊆ I. Each transaction is formed from the items in the set.

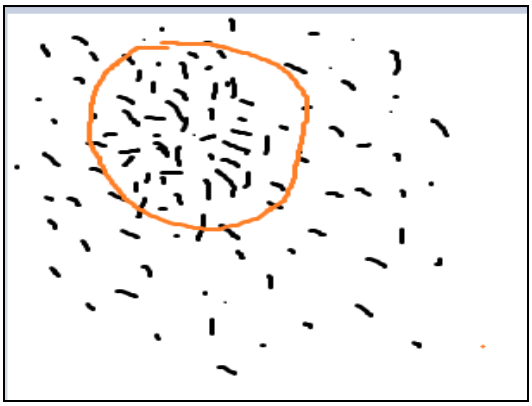

Figure1: A Database D containing set of transactions T.

Find all sets of items (itemsets) from the database transactions that have transaction support above minimum

support. The support for an itemset is the number of transactions that contain the itemset. Itemsets with minimum support are called large itemsets, and all others small itemsets. Use the large itemsets to generate the desired rules that satisfy minimum confidence.

An association rule is an expression X→Y, where X ⊆ I, Y ⊆ I and X∩Y = empty. LHS and RHS of a rule is formed by items in the data set which doesn't have anything in common. The X and *Y* are called the body (left hand side) and head (right hand side) of the rule, where N is the total transactions of a database. The heuristics that are used for hiding sensitive items are:[20-22]

1. Confidence of a rule measures the degree of the correlation between item sets.

$$Confidence = \frac{|X \cup Y|}{|X|}$$

2. Support of a rule measures the significance of the correlation between item sets.

$$Support = \frac{|X \cup Y|}{No.\ of\ transactions}$$

III. Proposed Algorithm

Association rule hiding aims to hide the sensitive rule. A rule is said to be sensitive if its disclosure risk is above a certain privacy threshold. Our focus is on hiding sensitive rules with minimal compromise of data utility. The approach for rule hiding is based on selectively modifying the database transactions. The modification is performed by data owners before publishing their data.

See following sample Table I which is the original dataset S. The sensitive item is FALSE. The minimum support & confidence is 40% & 50%. The instances in sample dataset are 14.

Table I: Original Dataset S before applying Algorithm

[1] No.	[2] Outlook	[3] Temperature	[4] Humidity	[5] Windy	[6] Play
[7] 1	[8] Sunny	[9] Hot	[10] High	[11] FALSE	[12] No
[13] 2	[14] Sunny	[15] Hot	[16] High	[17] TRUE	[18] No
[19] 3	[20] Overcast	[21] Hot	[22] High	[23] FALSE	[24] Yes
[25] 4	[26] Rainy	[27] Mild	[28] High	[29] FALSE	[30] Yes
[31] 5	[32] Rainy	[33] Cool	[34] Normal	[35] FALSE	[36] Yes
[37] 6	[38] Rainy	[39] Cool	[40] Normal	[41] TRUE	[42] No
[43] 7	[44] Overcast	[45] Cool	[46] Normal	[47] TRUE	[48] Yes
[49] 8	[50] Sunny	[51] Mild	[52] High	[53] FALSE	[54] No
[55] 9	[56] Sunny	[57] Cool	[58] Normal	[59] FALSE	[60] Yes
[61] 10	[62] Rainy	[63] Mild	[64] Normal	[65] FALSE	[66] Yes
[67] 11	[68] Sunny	[69] Mild	[70] Normal	[71] TRUE	[72] Yes
[73] 12	[74] Overcast	[75] Mild	[76] High	[77] TRUE	[78] Yes
[79] 13	[80] Overcast	[81] Hot	[82] Normal	[83] FALSE	[84] Yes
[85] 14	[86] Rainy	[87] Mild	[88] High	[89] TRUE	[90] No

Data owners want the transformation to best preserve those domain-specific data properties that are critical for building meaningful data mining models, thus maintaining mining task specific data utility of the published datasets. [23]We will be working on data sets from UCI machine learning repository. In this work, we assume that only sensitive items are given and propose the algorithm to modify data. Given a transaction database *D*, a minimum support, a minimum confidence and a set of items *H* to be hidden, the objective is to modify the database *D* such that no association rules containing *H* on the any side of rule will be discovered. As an example, for a given database in Table 1

Algorithm- perturbation algorithm used for securing sensitive rules and provides data privacy.
Input- Dataset D
Threshold value of support and confidence
Output- Association rules R'
Procedure-

```
Apriori Process ()
{
    Date d1 = new Date ();
Start1 = d1.getTime ();
updated = 0;
  While (Entry flag==1)
  {
 level++;
    generate Candidates (level);
```

```
 If (C level != empty)
      calculate Support (level);
 Else
      Set Entry flag to 0;
  If (level>=2 && frequent set! =0)
      generate Rules (level);
  }
 If (final rules not empty)
     Take an input index of sensitive item;
     Find the sensitive item using index from C 1;
     count sensitive item (sensitive item);
 Else
     Display Dead End;
 If (count sensitive item ==0)
     Display: Item not present in Rules;
     d1 = new Date ();
      End1 = d1.getTime ();
Time = (End1 – Start1)/1000;
Display: Time;
  Display: Association Rules R';
  Display: Rules having sensitive item;
  d2 = new Date ();
    Start2 = d2.getTime ();
 If (count sensitive item>0)
 {
     Entry flag=1;
     While (Entry flag==1)
     {
        level++;
        generate Candidates (level);
       If (C level != empty)
    calculate Support (level);
      Else
   Set Entry flag to 0;
      If F 'level contains sensitive item
   update transaction (level, index, sensitive item);
 If (level>=2 && frequent set! =0)
    generate Rules (level);
}
    count sensitive item (sensitive item);
  }
d2 = new Date ();
End2 = d2.getTime ();
Time = (End2 – Start2)/1000;
 Display: Time;
 Display: Association Rules R';
Display: Rules having sensitive item;
}
```

generate Candidates (int level)

```
{
   If (level==1)
              Scan the dataset D and store the candidates in set C level;
          Else if (level==2)
    concate items from set F' (level-1) to form a pair of two items and store it in set C level;
      Else
    generate all possible combinations of items from set F '(level-1) and store each pair in set Clevel;
            If (C level is empty)
             Entry flag=0;
}
```

calculate Support (int level)

```
{
          If (level==1)
   For each item j in set C level, count its occurrence in each Transaction T of Dataset D and store in array count[j];
          Else
{
    For each item j in set C level
          {
            Tokenize item j such that j = {k1, k2, k3, …, kn};
           where k1, k2… are single items
            For each Transaction T in Dataset D
           {
             Find the occurrence of k1, k2, … ,kn and accordingly set match++;
            If (match==n)
             Set count[j] ++;
           }
          }
    For each item j in set C level
   {
           support = (count[j]/no. of transactions);
           If (support >= min_supp)
          Store item j in set F level ;
}

If (F level! = empty && level >1)
{
          Store F level in Final F' level;
}

Else
Display message: No more frequent candidates generated.
}
```

generate Rules (int level)

```
{
          For each candidate in set F' level
          {
      Tokenize items of candidate m such that = {k1, k2, …, kn};
  Generate all possible combination of rules of the form X → Y for item m and store in set R; where |X| = n-1 and |Y|=1; n is size of item m
           }
   For each rule r in set R
   {
            r : X→ Y
           confidence = count (X ∪ Y) / count (X);
           If (confidence >= min_conf)
            Store the rule r in set R';
   }
   }
```

count sensitive item (sensitive item)

```
{
    For each rule in set R'
        {
   Tokenize items of rule r such that = {r1, r2, …, rn};
```

```
        If (tokens r_i  contains sensitive item)
                Set count++;
            }
  return: count;
}
update transaction (level, index, sensitive item)
{
If (level==1)
{
   For each Transaction T in Dataset D
            {
 For each candidate in set F'1
                {
          Find the occurrence of each item k1, k2, … ,kn of F' 1 in
T and accordingly set match++;
                  If (match=level)
                     Set count line [T] ++;
                   If (T contains sensitive item)
                      Set count sensitive++;
                      Set line sensitive [T] ++;
                  }
     }
For each count c in count sensitive
sensitive array [count line[line sensitive[c]]];
Reorder sensitive array in descending order;
Reorder line sensitive accordingly;
Find support of sensitive item = (count sensitive/no.  of
transactions);
Find difference of support= support of sensitive item – minimum
support;
If (difference ==0)
     difference= difference + 1;
extra occurrence= (difference * no. of transactions)/100;
Take drift as input from user;
Transactions to modify = (drift * extra occurrence)/100;
   }
    initial= Transactions to modify * (level -1);
   updation = Transactions to modify + updated;
   For some i ranging from initial to updation
            If (T == line sensitive [i])
             Set update flag=1;
             Set updated ++;
If (update flag==1)
    Replace sensitive item with null string in T;
Else
   Continue;
Increment T;
}
```

IV. EXAMPLE

Table I is the sample dataset for simulation. We begin the execution by applying algorithm on the original dataset S, generating candidates and frequent sets. Minimum support is 40% & confidence is 50%. Thus for a candidate to pass the minimum support, it has to be present in 6 transactions at least. The sensitive item is FALSE.

Table II: 1-Candidates set of sample dataset S.

[91] Candidate Items	[92] Support Count
[93] **Sunny**	[94] 5
[95] **Overcast**	[96] 4
[97] **Rainy**	[98] 5
[99] **Hot**	[100]4
[101]**Mild**	[102]**6**
[103]**Cool**	[104]4
[105]**High**	[106]**7**
[107]**Normal**	[108]**7**
[109]**TRUE**	[110]**6**
[111]**FALSE**	[112]**8**
[113]**Yes**	[114]**9**
[115]**No**	[116]5

Table III: 1-Frequent set of sample dataset S.

[117]Candidate Items	[118]Support Count
[119]**Mild**	[120]6
[121]**High**	[122]7
[123]**Normal**	[124]7
[125]**TRUE**	[126]6
[127]**FALSE**	[128]8
[129]**Yes**	[130]9

Table II & III explains how first candidate & frequent set is generated. Repeat steps 2 and 3 till candidates are generating. Table IV & shows 2nd level candidate & frequent set.

In the sample dataset, no candidates can be generated at third level because no prior item-set is common to form a third itemset. Thus final frequent set is obtained at 2nd level. Generate association rules from 2-frequent set using confidence.

Table IV: 2-Candidates set generated from 1-frequent set.

[131]Candidate Items	[132]Support Count
[133]**{Mild, High}**	[134]4
[135]**{Mild, Normal}**	[136]2
[137]**{Mild, TRUE}**	[138]3
[139]**{Mild, FALSE}**	[140]3
[141]**{Mild, yes}**	[142]4
[143]**{High, TRUE}**	[144]3
[145]**{High, FALSE}**	[146]4
[147]**{High, Yes}**	[148]3

[149]**{Normal, TRUE}**	[150]3
[151]**{Normal, FALSE}**	[152]4
[153]**{Normal, Yes}**	[154]**6**
[155]**{TRUE, Yes}**	[156]3
[157]**{FALSE, Yes}**	[158]**6**

Table V: 2-Frequent set generated from 1-frequent set.

[159]**Candidate Items**	[160]**Support Count**
[161]**{Normal, Yes}**	[162]6
[163]**{FALSE, Yes}**	[164]6

Table VI: Association Rules

[165]**Candidate Items**	[166]**Support Count**
[167]**Yes → Normal**	[168]66.66%
[169]**Normal → Yes**	[170]85.71%
[171]**Yes → FALSE**	[172]66.66%
[173]**FALSE → Yes**	[174]75.00%

Table VII: Final association Rules satisfying minimum confidence

[175]**Candidate Items**	[176]**Support Count**
[177]**Yes → Normal**	[178]66.66%
[179]**Normal → Yes**	[180]85.71%
[181]**Yes → FALSE**	[182]66.66%
[183]**FALSE → Yes**	[184]75.00%

We can see that Table VI displays the rules generated. Table VII contains the rules satisfying minimum confidence. As all association rules have confidence greater than minimum confidence value (50%). Select a sensitive item, say, FALSE from the rules. There are two rules having FALSE. The support of FALSE in the dataset S (Figure 4.1):

$$Support\ (FALSE) = \frac{8}{14} \times 100$$
$$= 57.14\%$$

This means out of 14 transactions, FALSE is present in 8 of them. The difference between its support and minimum support is 17.14%. This difference indicates by how much percentage the item has crossed the threshold value of support.

We can find the no. of transactions from the difference percentage.

$$= \left| \frac{Difference \times No.\ of\ transactions}{100} \right|$$
$$= \left| \frac{17.14 \times 14}{100} \right|$$
$$= 3$$

Deciding the drift percentage which will be used to indicate in how many transactions modification is done at each level of item set generation. Suppose we choose drift percentage equals to 25%. So it will modify 25% of difference value at each level, till the item occurs in the frequent set. Transaction to modify at each level is given by:

$$= \left| \frac{Drift \times Difference\ value}{100} \right|$$
$$= \left| \frac{25 \times 3}{100} \right|$$
$$= 1$$

We now begin the process again using modification algorithm. Generate candidates and frequent set. Check if the sensitive item occurs in 1-frequent set.

Table VIII: Transactions Count Value

[185]**Transaction No.**	[186]**Support Count**
[187]**1**	[188]**2**
[189]**2**	[190]2
[191]**3**	[192]**3**
[193]**4**	[194]**4**
[195]**5**	[196]**3**
[197]**6**	[198]2
[199]**7**	[200]3
[201]**8**	[202]**3**
[203]**9**	[204]**3**
[205]**10**	[206]**4**
[207]**11**	[208]4
[209]**12**	[210]4
[211]**13**	[212]**3**
[213]**14**	[214]3

Table IX: Transactions having sensitive item FALSE.

[215]**Transaction No.**	[216]**Support Count**
[217]**1**	[218]**2**
[219]**3**	[220]**3**
[221]**4**	[222]**4**
[223]**5**	[224]**3**
[225]**8**	[226]**3**

[227]**9**	[228]**3**
[229]**10**	[230]**4**
[231]**13**	[232]**3**

Find the count value of transactions using 1-frequent set. Reorder transactions based on descending order of count. Modification is done in the transaction having maximum value of count as calculated above. Table VIII shows the count value of all transactions. Table IX contains only those transactions and their count having sensitive item, FALSE.

Table X: Descending order of count value & re-arranged index of transactions.

[233]**Transaction No.**	[234]**Support Count**
[235]**4**	[236]4
[237]**10**	[238]4
[239]**5**	[240]3
[241]**8**	[242]3
[243]**9**	[244]3
[245]**3**	[246]3
[247]**13**	[248]3
[249]**1**	[250]2

We select transactions based on Table X. Modify the transaction no. 4, replacing sensitive item with null string. Now continue process of generating candidates from step 3 followed by step 4. The candidates and frequent set generated now will be different from the one generated before. The reason behind this is the modification done. If they are same, it means we need to perform more modifications. If we get a frequent set without sensitive item, the modification process ceases. After first modification process continue, generating 2-candidate using dataset transactions which have been modified, followed by 2-frequent set.

Table XI: 2-candidates set after applying algorithm.

[251]**Candidate Items**	[252]**Support Count**
[253]**{Mild, High}**	[254]4
[255]**{Mild, Normal}**	[256]2
[257]**{Mild, TRUE}**	[258]3
[259]**{Mild, FALSE}**	[260]2
[261]**{Mild, yes}**	[262]4
[263]**{High, TRUE}**	[264]3
[265]**{High, FALSE}**	[266]3
[267]**{High, Yes}**	[268]3
[269]**{Normal, TRUE}**	[270]3
[271]**{Normal, FALSE}**	[272]4
[273]**{Normal, Yes}**	[274]**6**
[275]**{TRUE, Yes}**	[276]3
[277]**{FALSE, Yes}**	[278]5

Table XII: 2-frequent set after applying algorithm.

[279]**Candidate Items**	[280]**Support Count**
[281]**{Normal, Yes}**	[282]6

Table XI contains 2-candidate after first modification done on the dataset. Table XII contains transaction with count greater than minimum count.

The rows marked bold are the ones that indicate how performing modification has changed the count value of candidate item-sets. Previously, the counts of these candidates were 3, 4 and 6. The change is due to selective modification done on the transaction. Frequent set will have only one candidate which is different from previous results of 2 candidates.

FALSE, the sensitive item does not occurs in the frequent set. It contains only one item pair {Normal Yes}. This indicates that support of sensitive item is reduced, which affects the candidate pairs in the subsequent sets. Thus the candidates with that item will no longer be generated. At last association rules are generated satisfying minimum confidence, the rules will no longer have sensitive item present in them.

Table XIII: Modified Dataset S' after applying Algorithm

[283]**No.**	[284]**Outlook**	[285]**Temperature**	[286]**Humidity**	[287]**Windy**	[288]**Play**
[289]**1**	[290]Sunny	[291]Hot	[292]High	[293]FALSE	[294]No
[295]**2**	[296]Sunny	[297]Hot	[298]High	[299]TRUE	[300]No
[301]**3**	[302]Overcast	[303]Hot	[304]High	[305]FALSE	[306]Yes
[307]**4**	[308]**Rainy**	[309]**Mild**	[310]**High**	[311]**null**	[312]**Yes**
[313]**5**	[314]Rainy	[315]Cool	[316]Normal	[317]FALSE	[318]Yes
[319]**6**	[320]Rainy	[321]Cool	[322]Normal	[323]TRUE	[324]No
[325]**7**	[326]Overcast	[327]Cool	[328]Normal	[329]TRUE	[330]Yes

[331]**8**	[332]Sunny	[333]Mild	[334]High	[335]FALSE	[336]No
[337]**9**	[338]Sunny	[339]Cool	[340]Normal	[341]FALSE	[342]Yes
[343]**10**	[344]Rainy	[345]Mild	[346]Normal	[347]FALSE	[348]Yes
[349]**11**	[350]Sunny	[351]Mild	[352]Normal	[353]TRUE	[354]Yes
[355]**12**	[356]Overcast	[357]Mild	[358]High	[359]TRUE	[360]Yes
[361]**13**	[362]Overcast	[363]Hot	[364]Normal	[365]FALSE	[366]Yes
[367]**14**	[368]Rainy	[369]Mild	[370]High	[371]TRUE	[372]No

Association rules after algorithm are: {Yes ** Normal, Normal ** Yes}. If the item exists, it means the hiding failure occurs. The hiding failure indicates that sensitive item is not hidden completely from the dataset S. We can calculate the percentage of hiding failure. Time taken to execute the algorithm prior to modification is 4 seconds and after selective modification is 5 seconds. Thus the total taken to complete execution on sample dataset is 9 seconds.

V. Analysis

This section analyzes some of the characteristics of the proposed algorithm. The table shows the association rules generated before algorithm and rules generated after selective modification on dataset. We have used Adult & Bank Marketing dataset.[24,25,26]

Results for Adult Dataset

Table XIV(1) & XIV(2). Adult Dataset with support=50%, confidence= 70% & drift= 40%; varying sensitive item

Sensitive Item	Before applying algorithm		After applying algorithm	
	No. of Rules	No. of Rules with sensitive item	No. of Rules	No. of Rules with sensitive item
White	25	20	5	0
Fnlwgt0	25	20	5	0
Cl0	25	25	20	0

Execution time		Analysis		
before applying algorithm (sec)	after applying algorithm (sec)	Hiding Failure	Ghost Rules	Lost Rules
78.42	57.86	N	0	0
76.95	63.56	N	0	0
79.61	51.76	N	0	0

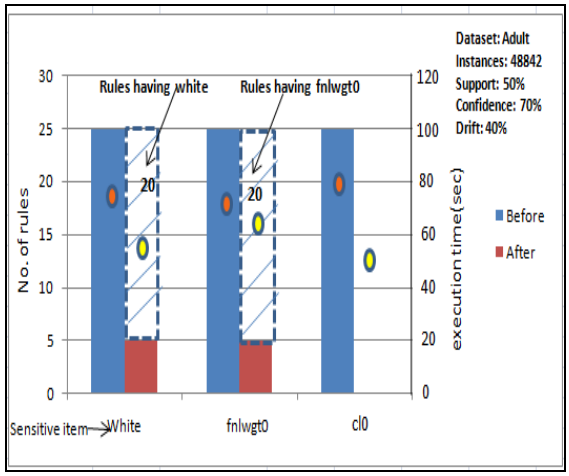

Figure 2: Adult Dataset: Changing sensitive item; keeping support, confidence, drift same.

The results produced are shown in column chart. X-axis indicating different parameters such as sensitive item, support, confidence, drift. Y-axis indicates the no. of rules generated at the last level. The level here represents the one generated prior to applying the proposed algorithm. Suppose before applying algorithm last level goes till 5th frequent set, leading to form rules. Out of which all rules are sensitive. So results of applying algorithm will remove all those rules which will cause it to reach up to 4th level. Execution time is displayed in seconds.

Table XV(1) & XV(2). Adult Dataset with confidence= 70%, drift= 40% and sensitive item is White; varying support

Support (%)	Before applying algorithm		After applying algorithm	
	No. of rules	No. of rules with sensitive	No. of rules	No. of rules with sensitive
50	25	20	5	0
60	5	5	12	0
70	27	12	15	0

Execution time		Analysis		
before applying algorithm (sec)	after applying algorithm (sec)	Hiding failure	Ghost rules	Lost rules

78.42	57.86	N	0	0
37.49	41.50	N	0	0
17.36	27.22	N	0	0

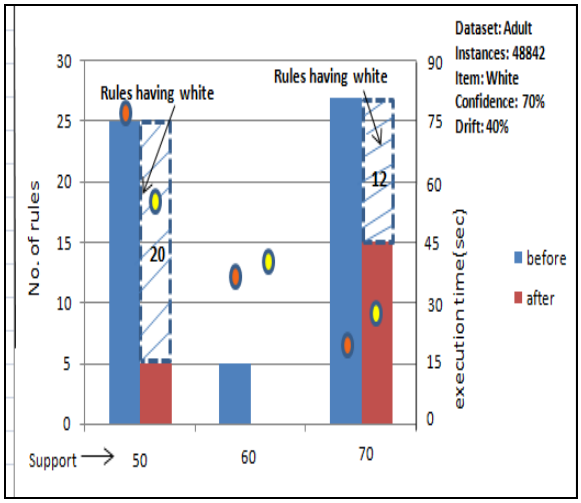

Figure 3: Adult Dataset: Changing support; keeping item, confidence, drift same

The graph shows the no. of rules before and after applying algorithm. Rules displayed are the one at last level of before applying proposed algorithm. It may happen before applying the proposed algorithm it reached up to 5th level but later goes up to 4th level.

Table XVI(1) & XVI(2). Adult Dataset with support = 60%, drift= 40% and sensitive item is White; varying confidence

Confidence (%)	Before applying algorithm		After applying algorithm	
	No. of rules	No. of rules with sensitive item	No. of rules	No. of rules with sensitive
70	5	5	12	0
80	5	5	12	0
90	3	3	6	0

Execution time		Analysis		
before applying algorithm (sec)	after applying algorithm (sec)	Hiding failure	Ghost rules	Lost rules
37.49	41.50	N	0	0
31.76	37.02	N	0	0
28.32	33.74	N	0	0

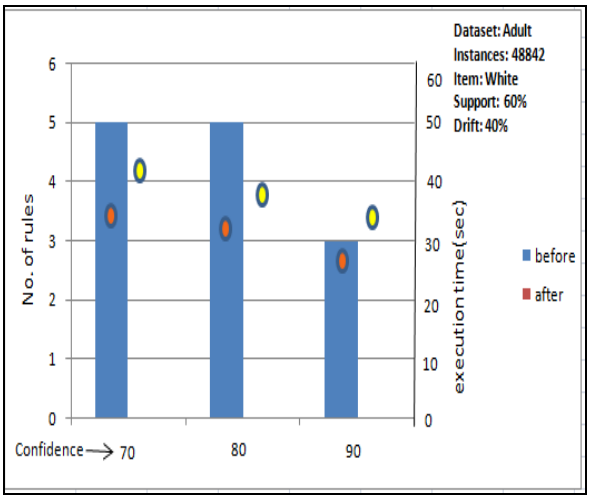

Figure 4: Adult Dataset: Changing confidence; keeping item, support, drift same.

Table XVII(1) & XVII(2). Adult Dataset with support = 80%, confidence= 80%, sensitive item is cl0; varying drift.

Drift (%)	Before applying algorithm		After applying algorithm	
	No. of rules	No. of rules with sensitive item	No. of rules	No. of rules with sensitive
40	12	8	6	2
45	12	4	10	2
50	12	8	4	0

Execution time		Analysis		
before applying algorithm (sec)	after applying algorithm (sec)	Hiding failure	Ghost rules	Lost rules
16.96	33.43	Y	0	0
15.19	32.10	Y	0	0
7.88	19.20	N	0	0

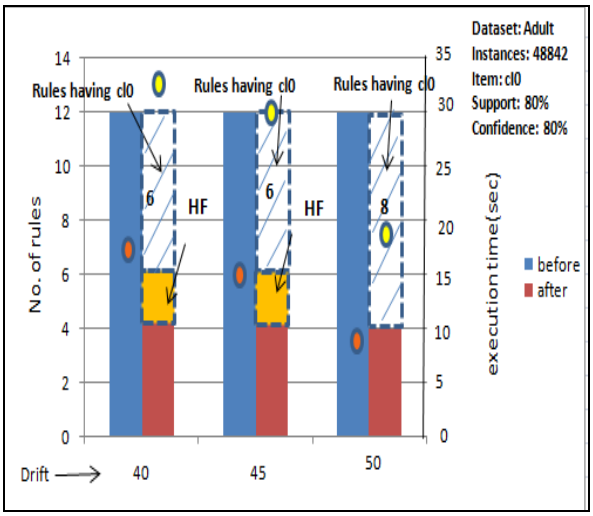

Figure 5: Adult Dataset: Changing drift; keeping support, confidence, item same.

The results clearly indicate that in some cases hiding failure occurs. For example, if we vary drift value hiding failure can be overcome. It is clear that there are no ghost rules generated no missing rules effect.

In Table XIV to XVII we can see the results on adult dataset having 48842 records. The table displays the rules generated before and after applying proposed algorithm along with before and after execution time of proposed algorithm and evaluation parameters such as hiding failure, lost rules and ghost rules. First table show the result of varying sensitive item keeping support, confidence & drift constant. Table XV displays the results of altering support with confidence, drift and sensitive item constant. Table XVI displays the results of varying confidence with support, item and drift constant. Last Table XVII shows results when drift is changed with same support, confidence and sensitive item.

VI. Conclusion & Future Work

The proposed algorithm carries out pre-processing of the dataset and performing selective modification to hide sensitive rules from being disclosed As from our example we see that our approach is better in the way that it hides any rule which contains the user specified sensitive item without looking whether it is on LHS or RHS. The approach has been evaluated based on accuracy achieved in terms of sensitive rule hiding, data utility preserved, ghost rules and missing rules. We have obtained favourable results on both the datasets in terms of desired accuracy and affordable time complexity. The approach is easy to implement. Future work can include hiding items on specific side of association rule such as LHS or RHS. One can also focus on improving the execution time.

References

[1] D. Hand, H. Mannila and P. Smyth. “Principles of Data Mining”, The MIT Press, 2001.

[2] J. Han and M. Kamber. ”Data Mining, Concepts and Techniques”, Morgan Kaufmann Publishers, Champaign: CS497JH, www.cs.sfu.ca/~han/DM_Book.html.

[3] R. Agrawal and R. Srikant. ”Privacy preserving data mining”, In ACM SIGMOD Conference on Management of Data, May 2000.

[4] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. “Privacy preserving mining of association rules”. In Proc. Of the 8th ACM SIGKDD Int’l Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, July 2002.

[5] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, Y. Theodoridis. “State-of-the-art in Privacy Preserving Data Mining”. SIGMOD Record, Vol. 33, No. 1, March 2004.

[6] D. Agrawal and C. C. Aggarwal. “On the design and quantification of privacy preserving data mining algorithms”. In Proceedings of the 20th Symposium on Principles of Database Systems, Santa Barbara, California, USA, May 2001.

[7] R. Agrawal, T. Imielinski, and A. Swami. “Mining Association Rules between Sets of Items in Large Databases”. In Proceedings of ACM SIGMOD International Conference on Management of Data, Washington DC May 1993.

[8] Agrawal R, Srikant R. “Fast algorithms for mining association rules in large databases”. In: Proceedings of the 20th International Conference on Very Large Databases (VLDB), pp 487–499, 1994.

[9] Yogendra Kumar Jain, “An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining”, International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 7 July 2011.

[10] Dasseni, E., Verykios, V., Elmagarmid, A., & Bertino, E.. “Hiding association rules by using confidence and support”. In Proceedings of 4th information hiding workshop, Pittsburgh, PA, 2000.

[11] Clifton, C. “Using sample size to limit exposure to data mining”. Journal of Computer Security, 2000.

[12] C. Clifton, “Protecting Against Data Mining through Samples”, in Proceedings of the Thirteenth Annual IFIP WG 11.3 Working Conference on Database Security, 1999.

[13] S. L. Wang, Yu-Huei Lee, Steven Billis, Ayat Jafari. "Hiding Sensitive Items in Privacy Preserving Association Rule Mining" International Conference on Systems, Man and Cybernetics, 2004.

[14] S. Oliveira, O. Zaiane, “Algorithms for Balancing Privacy and Knowledge Discovery in Association Rule Mining”, Proceedings of 7th International Database Engineering and Applications Symposium (IDEAS03), Hong Kong, July 2003.

[15] S. Oliveira, O. Zaiane, “Protecting Sensitive Knowledge by Data Sanitization”, Proceedings of IEEE International Conference on Data Mining, November 2003.

[16] Y. Saygin, V. Verykios, and C. Clifton, “Using Unknowns to Prevent Discovery of Association Rules”, SIGMOND Record 30(4): 45-54, December 2001.

[17] S. L.Wang, B. Parikh, A. Jafari, “Hiding informative association rule sets”, Expert Systems with Applications, 2007.

[18] C. Clifton and D. Marks, ‘Security and Privacy Implications of Data Mining”, in SIGMOD Workshop on Research Issues on Data Mining and knowledge Discovery, 1996.

[19] Alexandre Evfimievski, “Randomization in Privacy Preserving Data Mining”, SIGKDD Explorations, 4(2), Issue 2, 43-48, Dec. 2002.

[20] Alexandre Evfimievski, Johannes Gehrke and Ramakrishnan Srikant, “Limiting Privacy Breaches in Privacy Preserving Data Mining”, PODS 2003, San Diego, CA, June 9-12,2003.

[21] R. C. Belwal, J. Varshney, S. A. Khan and A. Sharma, “Hiding Sensitive Association Rules Efficiently By Introducing New Variable Hiding Counter”, IEEE transaction, 2008.

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

[23] Keke Chen, Ling Liu,”A Survey of Multiplicative perturbation for privacy preserving data mining”.

[24] “The Weka Machine Learning Workbench”, http://www.cs.waikato.ac.nz/ml/weka.

[25] Ronny Kohavi and Barry Becker, "Data Mining and Visualization", Silicon Graphics.

[26] "UCI Machine Learning Repository" for working with data sets. http://archie.ics.uci.edu/ml/

AUTHORS

First Author – Geetika M. Kalra, B.E(C.E), M.Tech(C.E), U.V.Patel college of engineering, geetika.kal248@gmail.com.
Second Author – Hitesh Chhinkaniwala, PhD, 2Dean,Shankersinh Vaghela Bapu Institute of Technology.