Indian Journal of Science and Technology, Vol 9(27), DOI: 10.17485/ijst/2016/v9i27/97476, July 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Hybrid Algorithm based on Heuristic Method to Preserve Privacy in Association Rule Mining

Narges Jamshidian Ghalehsefidi* and Mohammad Naderi Dehkordi

Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran;
Jamshidian_n@sco.iaun.ac.ir, Naderi@iaun.ac.ir

Abstract

By developing technology and competition in different fields, preserving sensitive data is considered as a problematic issue for users. As long as users do not need to share their data, they preserve them in different ways, such as encryption and hiding them in personal devices like cell phones and computers. When users find it necessary to share their personal data, privacy preserving data mining will help them. In the present study, we introduce two algorithms called ISSDD (Intelligent Selection of Sanitization in Dense Database) and ISSSD (Intelligent Selection of Sanitization in Sparse Database) to decrease side effects such as hiding failure, losing non-sensitive rules, making new rules and also hiding sensitive rules without any restriction in the number of items in the left and right hand. In the suggested algorithms distortion technique is used to hide sensitive rules by declining confidence-based and support-based of rules.

Keywords: Hiding Sensitive Rules, Privacy Preserving Data Mining, Sensitive Pattern

1. Introduction

The aim of data mining is to discover useful patterns among a bulk of data in transactional databases that can be appointed by standard algorithms such as Apriori algorithm, FP-Tree algorithm and so on. Data mining techniques involve predictive type (classification, regression, time series analysis), descriptive type (clustering, association rules and sequential patterns). Association rule technique extracts useful patterns as a rule. Extracted useful patterns involve sensitive patterns that the owner of database tries to hide them and non-sensitive patterns that their sharing is not dangerous for the owner of database. The purpose of privacy preserving in data mining is to prevent sensitive patterns from being revealed that can be safe by hiding from any commercial and confidential misuse of the owner of database. In spite of some methods such as inferencing, sensitive data cannot be absolutely preserved. Moreover, hiding sensitive data makes side effects that are dangerous for sensitive databases like medical database. Side effect is in fact a cost that the owner of the database will pay to hide sensitive data that involves losing non-sensitive rules, making new rule, failure in hiding and so on. Generally, two kinds of methods[1] can be viewed in hiding sensitive rules: 1-method based on support: in this method by declining support in one item of the components of the sensitive rule, the support level can be reduced in that rule. 2-method based on confidence: in this method the level of support can be reduced by increasing the degree of support of left hand items in sensitive rule. For both mentioned methods distortion ways, a process by which value 1 is replaced to 0 and vice versa, and, blocking ways, by which cardinal numbers are replaced by? are used. In this article we use distortion method or both mentioned methods to hide sensitive rules. In suggested algorithms one of the methods is used based on the selected item to make security. In the proposed algorithm, an appropriate item is selected by applying non-sensitive rules, supporting sensitive item and the degree of its sensitivity. The selected transaction has a remarkable impact on the degree of side effects that are selected by the degree of side effects that are selected by considering the sensitive and non-sensitive items. In the present study the proposed algorithm is compared by

*Author for correspondence

RRLR, DSRRC, MDSRRC, ISL, DSR, Hybrid (DSR, ISL) algorithms over 3 real data sets. The second part of rule frame describes the dependency. The third part investigates about relevant works. The fourth part deals with proposed algorithm, terms, and explains its process. In section 5 comparison and evaluation is made between the proposed algorithm and other six algorithms and in the sixth section conclusion is presented.

2. Frame of Association Rules

Extracted rules from data mining represent relationship among different databases. The criterion of extraction for support and confidence are rules. The issue related to extracting association rule was presented by[2]. Imagine $I=\{i_1,i_2,\ldots, i_m\}$ a group of elements and database $D=\{T_1,T_2,\ldots, T_n\}$ a group of transactions. Every $T\in D$ transaction involves I in its subcategory. The total frame of association rule is $X\rightarrow Y$. If X and Y are subcategories of I and $X\cap Y=\emptyset$. X is called antecedent or LHS rule and Y is called consequent or RHS.

The support of $X\rightarrow Y$ is defined by computing the proportion of simultaneous frequency of X and Y in transactions over the total number of transactions of database.

The confidence of a rule $X\rightarrow Y$ is defined by computing the proportion of simultaneous frequency of X and Y in transactions over the number of X frequency alone in transaction of database.

Minimum Support Threshold (MST) and Minimum Confidence Threshold (MCT) are used to extract useful rules from database. If support $(X\rightarrow Y) >= MST$ and confidence $(X\rightarrow Y) >= MCT$, so $X\rightarrow Y$ will be significant and will be extracted from database in data mining.

3. Relevant Works

In[3] presented two algorithms to hide association rules. The first algorithm called ISL decreases confidence rule by increasing support in sensitive rule in the left hand side elements as they select rules that have sensitive items in their left hand and insert sensitive items in transactions that don't contain this rule. This algorithm has lots of failure in hiding and making new rules. The second algorithm called DSR decreases support rule by decreasing support in right hand side elements as they select rules that have sensitive items in their right hand and removes sensitive items from those transactions which contain this rule. Failure in this algorithm is close to zero and many non-sensitive rules will be lost[3,4].

In[5] presented an algorithm called DSRRC that applied clustering for right hand common items in hiding. The drawback of this algorithm is in hiding those items which have one element in their right side, it's dependent on arrangement in transactions and shows different results by changing the orders of transactions in database, it requires arrangement after deleting every item and it is not appropriate for large databases. There are lots of lost rules in this algorithm[5].

In[6] presented an algorithm called ADSRRC for improving DSRRC algorithm. This algorithm also hides those rules which have single RHS and arrangement is made once only. In addition, in this article, an algorithm called RRLR is suggested that hides those rules which have single LHS. In this algorithm, to hide sensitive rules both support and confidence are decreased as in transaction with high degree of sensitivity, left hand item is deleted and insert a transaction which has partially sensitive rule[6].

In[7] proposed MDSRRC algorithm to eliminate restriction in the number of left and right items. This algorithm selects the best item for deletion based on its frequency on the right side of the sensitive rule and supporting that item. This algorithm contains the minimum side effect compared with DSRRC. Failure in hiding is close to zero[7].

In[8] combined ISL and DSR algorithms together and made the main purpose on declining the number of changes in database and decreasing the time to hide sensitive rules[8].

In[9] presented a heuristic algorithm called ABS. in this algorithm selection of transaction is randomly done. Its idea is originated from the way honey bees looking for the source of food. In this algorithm a support-based method is used[9].

In[10] proposed two algorithms called Random, Round Robin that its base is on selecting an item to preserve that is done in order or randomly[10].

In[11] proposed an algorithm called SRH by which they could decrease time and memory complexity by computing the number of required transactions for hiding sensitive rule[11].

In[12] proposed an algorithm with an accurate focus called integer programming and blanket, intelligent strategies. The advantage of this algorithm was in hiding rate, assuring the best accuracy level, formulating for measurement and solving the problems in an optimizing way[12].

In[13] proposed WSDA and BA algorithms. WSDA hides sensitive rules by distorting technique and BA hides by blocking technique. WSDA algorithm concentrates on optimizing hiding techniques to minimize side effects and have the least complexity in hiding. This algorithm is not appropriate for large databases. The aim for BA algorithm is to hide rules that cannot be discoverable and to minimize the number of lost association rule and ghost rule[13].

In[14] proposed aggregate, disaggregate and hybrid algorithms that hide sensitive rules based on support-based method. In the first algorithm called aggregate supporting sensitive rule is decreased by deleting some transactions. The second algorithm called disaggregate declines supporting degree of sensitive rules by deleting some sensitive elements. The third algorithm called Hybrid determines the identified transactions by aggregate method and then specifies the required elements for deleting by disaggregate method[14].

Introduced a new method to preserve privacy based on genetic algorithm, to make sure no ghost rule or lost rule is made. This algorithm is based on rules and items with the least amount of side effect through hiding strategy. Three strategies for selection of an item and three strategies for Crossover of an item are suggested in this algorithm[15].

In[13] presented two strategies and five algorithms that decline the degree of support for productive sensitive rules to reach to less than minimum amount of support. This is done in two ways: 1-deleting an item that contains maximum supporting degree of a transaction with the least length. 2- Sorting a group of sensitive productive rules according to their length and support, and hiding them by rotation[16].

4. Proposed Algorithms

In proposed algorithms distorting technique is used to hide association rules by decreasing support and confidence in sensitive rule. These algorithms have 4 major purposesrespectively:

1. Hiding sensitive rules without any restriction in the number of right and left hand side items.
2. Reducing failure in hiding sensitive rules
3. Reducing the number of lost rules due to hiding.
4. Reducing the number of ghost rules due to hiding.

In this way the number of changes in database will be reduced. First of all, used terms in these algorithms are introduced, and then steps of proposed algorithms are explained.

Sensitive item and item sensitivity: Presented items in sensitive rules are called sensitive item and the number of its frequency in sensitive rules is called item sensitivity.

Non-sensitive item: Presented items in ghost rule in which at least one sensitive item was presented in it. These rules have less support or confidence compared with MST and MCT defined by user.

Degree of conflict in transaction: The number of sensitive rules in a transaction. In fact, the transaction contains all presented items in sensitive rule.

Non-sensitive rule: A rule that contains a sensitive item and their support and confidence is larger than MST and MCT. For instance, if MCT=60 and MST=30 the rules that their support is between MST=30 and MST=40 or their confidence is between MCT=60 and MCT=70 will be selected.

The steps in proposed algorithm:

The base of proposed algorithms is on selecting an appropriate item and transaction for security. In this algorithm two preserving items are defined that will be selected according to selection of an item. Selecting an appropriate item is based on the degree of its support in database, item sensitivity, the number of frequency of an item in RHS and LHS of non-sensitive rules. Transactions in proposed algorithms are divided into two parts. The first part contains transactions that support at least one rule completely that their conflict degree is at least 1 and the second part which contains transactions that their conflict degree is zero. In the first group item deletion and in the second group item insertion is done. Transactions that are located in the first group are sorted according to their conflict degree and length and transactions that are located in the second group are sorted according to the number of non-sensitive items, sensitive items and their length.

In the proposed algorithms if there is no way to alter any transaction and sensitive rule has not been hidden, a new transaction is made in the first group that involves left hand items in sensitive rules. This transaction doesn't have any sensitive rule completely and if a sensitive rule is made in it an item must be deleted from that transaction. Selecting an item for deletion is made if the numbers of frequency for the first left hand item rule in the total number of sensitive rule is less than the numbers of frequency for the first right hand item rule, which will be deleted. Making a new transaction is made once only

and if a rule is not able to be hidden, we consider it as a failure. The advantage of new transaction is to decline simultaneous decrease in support and confidence for all sensitive rules. But the new transaction is dangerous for non-sensitive rules that their support degree is equal to MST and leads to their disappearance. For this reason, we can determine those rules that their support is equal to MST and participated items in transactional condition that has zero conflict degree must also be observed.

In these algorithms, first of all data mining completed to recognize sensitive and non-sensitive rules. The number of ghost rule depends on the user. The more items are available the less degree of ghost rule resulted from hiding is observed. To find ghost rule in Apriori algorithm we consider the degree of support and confidence less than MST and MCT. For instance, if MST=60 and MCT=80 and support=50 and confidence=70 then we select items that their support and confidence is between two ranges. Figure 1 illustrates proposed algorithms flowchart and Algorithm1 illustrates pseudo code for ISSDD.

Figure 1. Proposed algorithm flowchart. **(Image Problem)**

Algorithm 1: Pseudo code for ISSDD

Input: Source Database D, MCT, MST, Sensitive rule (RH), Non sensitive rule (RNS), Ghost rule (RG)

Output: The Sanitized Database D

1. Find sensitivity of each item ∈ RH set IS
2. Find count repeat of each item∈ IS in RHS,LHS of each R∈RNS
3. Find Support of each item∈IS in D
4. Find Non sensitivity of each item∈RG and item∉IS set INS
5. Find conflict T∈D set TS if a T no R∈RH Set TNS
6. Sort TS by Conflict ASC and Length T∈TS ASC
7. Sort TNS by INS∈T ASC and IS∈T ASC and Length T∈TNS ASC
8. Sort IS by Sensitivity DESC and Support DESC and RHS count ASC and LHS count ASC
9. Sort RH by decreasing order of their confidence
10. Hiding
11. While(hiding all the sensitive rule ≠True){
    a. For i=0 to No. rules∈RH{
        i. If RH[i] Status=False{
            1. Next:
            2 For j=0 to No. items∈IS{
                a. If LHS's RH[i] contains IS[j]{
                    i. Index=j;
                    ii. Chang=Start Delete LHS
                b.}else if RHS's RH[i] contains IS[j]{
                    i. Index=j;
                    ii. Chang=Start Delete RHS
                c. }
            3. }
            4. If !change{
                a. Sensitivity IS[index]-1
                b. Index++
                c. If Index>IS count{
                    i. If ! insert transaction{
                        1. Start insert transaction; start update support & confidence;
                        2. Insert transaction=true;
                    ii. }else{
                        1. Status RH[i]=Failure;
                    iii. }
                d. }else{
                    i. Go to Next;
                e. }
            5. }else{
                a. Index=0
                b. Start Update support& Confidence
                c. i =-1
            6. }
        ii. }
    b. }
12 }
1. Start Delete RHS
2. For i=0 to No. TS{
                a. If(itemset xyz ∈TS[i]){
                    i. Remove itemselected from TS[i]
                    ii. Change=true

```
                a. }
3. }
1. Start Delete LHS
2. For i =0 to No. TS{
                a. If(itemset xyz ∈TS[i]){
                    i. Remove itemselected from TS[i]
                    ii. Start Insert LHS
                    iii. Change=true
                b. }
3. }
4. Insert LHS
5. For i =0 to No. TNS{
                b. If RHS of rule (yz) is partially to
                   TNS[i] and LHS of rule(x) is
                   partially TNS and itemseleced does
                   not belongs to TNS[i]
                    i. Insert itemselected in TNS[i] ]//
                       if multi LHS insert All
                    ii. Change=true
6. }
7. If change=false{
8. For i =0 to No. TS{
                a. If RHS of rule (yz) is partially to
                   TS[i] and LHS of rule(x) is partially
                   TS and itemseleced does not
                   belongs to TS[i]
                    i. Insert itemselected in TS[i]
                       //if multi LHS insert All
                    ii. Change=true
                b. }
9. }
1. Start Update Support & confidence
2. For i=0 to No. RH{
                a. If (support of RH[i]<MST
                   or Confidence of RH[i]<MCT){
                    i. If Status RH[i]=False{
                        1. Status RH[i]=True
                        2. Index=0
                        3. Update sensitivity of each
                           item ∈IS
                    ii. }
                b. }else{
                    ii. If Status RH[i]=True{
                        1. Update sensitivity of each
                           item ∈IS
                        2. Status RH[i]=False
                    ii. }
                c.}
3. }
```

Sort IS by Sensitivity DESC and Support DESC and RHS count ASC and LHS count ASC

RH is a sensitive rule that the owner of database is about to hide it and RNS is non-sensitive rule and RG is ghost Rule. In the first step, find item sensitivity in sensitive rule and put them in IS category in the second step, compute the frequency number for sensitive items in the right and left hand side of the non-sensitive rule. In the third step, support finds sensitive items. The fourth step is for finding non-sensitive items in ghost rule and putting them in INS category. All presented items in ghost rule except for sensitive items are considered as non-sensitive item. In the fifth step, conflict degree of transactions is found and those transactions which have at least a conflict degree equal to 1 are put in TS category and those which have conflict degree equal to zero are put in TNS category. The sixth step is for ordering TS category based on ascendant in conflict degree and length. The seventh step is for arranging TNS category based on ascendant non-sensitive item, Ascendant sensitive item and ascendant length for transaction. The eighth step is for ordering IS category based on their item sensitivity and support decently and the frequency number of items in RHS and LHS is ascendant. The ninth step is for arranging sensitive rules according to their descent confidence. Security is made from line 10 to 12.

An essential item here is to determine the position of sensitive rule. It has been ether hidden so it is equal to True or it has not been hidden therefore it is equal to false. As long as all rules are not fixed to true position, preserving will continue. The first sensitive rule is selected then selects the first present item in the rule. If the selected item is located on the left side security preserving is made on declining confidence-based and if it is located on the right side security preserving is made on declining support-based. Index keeps the position of selected item. In preserving with selected item, if there is no way to alter, the sensitivity level of selected item will be decreased in one unit, then adds one unit to index so that another Item will be selected subsequently. If the value of index gets larger than the number of items in IS category, it shows that the item is no larger appropriate for changes. If a new transaction has not inserted yet, a new transaction will be inserted otherwise it changes the position of rule to failure. If index is smaller than the number of items in IS category, it jumps to next position and selects another item.

If there is a possibility to alter preserving by the selected item, then change index value to zero and update support

and confidence for sensitive rules and put i=-1 since by applying changes for the first rules, false may happen in the first so all rules will be checked once again and if they are false, they will be selected for preserving. Delete RHS will be recalled when the item in the right side of the rule is selected. It will check full support of the rule transaction from TS first selected transaction then the selected item will be deleted from it. Delete LHS will be recalled when selected item is on the left side. First of all delete selected item according to the mentioned rule from TS category. If deletion is done, insertion will be recalled in which the first transaction from TNS category is selected and insertion will be done according to mentioned condition. If there are multiple items on the left side, all of them must be inserted. If there is not any possibility to insert in TNS category, a transaction will be selected according to insertion rule from TS category. Update support and confidence will be recalled when preserving is made. If a rule is hidden, its position will be changed to True and its item sensitivity will be updated and if a rule has already been hidden and appeared again, it will change its position to False and update its items sensitivity at the ends rearrange IS category.

ISSDD algorithm will have better result on dense database. In order to have better result on sparse databases, ISSSD algorithm was introduced that is similar to ISSDD only some codes have been changed. Changed codes are illustrated in Algorithm2.

Algorithm 2: Changed code ISSDD

1. Sort TS by ***Conflict DESC*** and Length T∈TS ASC
2. Sort IS by Sensitivity DESC and ***Support ASC*** and RHS count ASC and LHS count ASC
4. Delete RHS
5. For i=0 to No. TS{
    a. If(itemset xyz ∈TS[i]){
        i. Remove itemselected
        i. Support itemselected -1
        ii. Change=true
    b. }
6. }
10. Insert LHS
11. For i=0 to No. TNS{
    a. If RHS of rule (yz) is partially to TNS[i] and LHS of rule(x) is partially TNS and itemseleced does not belongs to TNS[i]
        i. Insert itemselected TNS[i]] if multi LHS insert All
        ii. Update support LHS rule
        ii. Change=true
12. }
13. If change=false{
14. For i=0 to No. TS{
    a. If RHS of rule (yz) is partially to TS[i] and LHS of rule(x) is partially TS and itemseleced does not belongs to TS[i]
        i. Insert itemselected TS[i] if multi LHS insert All
        iii. Update support LHS rule
        ii. Change=true
    b. }
15. }
16. If change=false{
    b. Support itemselected -1
17. }

Consider Table 1 with MST=25 and MCT=60. The extracted rules are 156. In order to have algorithms in equal condition and be able to hide rules; sensitive rules are selected as single elements in the left and right side. Sensitive rules involve d⟶f, f⟶d, e⟶f, g⟶d, d⟶h, h⟶e. Hiding steps are illustrated in Tables 2 to 8. The results obtained from hiding these rules are illustrated in Figure 2. As it can be seen in Figure 5, a failure happened in algorithm and as proposed algorithms don't have any failure will have better result compared with other algorithms. In the second test, those items which have multiple items on their left and right side are selected that involve fh⟶bd, cd⟶fh, de⟶fh, bdf⟶eh. The results obtained from this test are displayed in Figure 3.

Table 1. Sample database

TID	Items
0	c d e g
1	b d f h
2	c f h
3	a b f
4	a b c d e f h
5	b c
6	f g
7	b d
8	e g d
9	a g

10	b c d e f h
11	c d e
12	b d e f h
13	a c f g
14	b c f g h

Table 2. Input algorithm

Sensitive rule	d⟶f, f⟶d, e⟶f, g⟶d, d⟶h, h⟶e
Non-sensitive rule	143 rules: MST=25+10, MCT=60+10
Ghost rule	61 rules: MST=between 20 – 25, MCT= between 50 - 60
MST	25
MCT	60

Table 3. Item sensitive

Item	Support	Sensitive	LHS	RHS
d	9	4	53	53
f	9	3	55	51
e	7	2	51	36
h	6	2	73	42
g	6	1	0	0

Table 4. Item non-sensitive

Item	Sensitive
C	25
B	38

Table 5. Transaction sensitive

TID	Items	Conflict	Length
9	a d g	1	3
0	c d e g	1	4
13	a c e f g	1	5
6	d f g	3	3
2	c d f h	3	4
14	b c d e f h	4	6
1	b d e f h	5	5
12	b d e f h	5	5
10	b c d e f h	5	6
4	a b c d e f h	5	7

Table 6. Transaction non-sensitive

TID	Items	INS	IS	Length
8	e g	0	3	2
11	c	25	0	1
7 b 38			0	1
3	a b f	38	3	3
5	b c	63	0	2

Table 7. Steps sanitization

Rule Selected	Item Selected	Sanitize technique	Transaction Edit	Status rules
d⟶f d⟶f f⟶d d⟶h	d d	Delete & Insert LHS Delete & Insert LHS	D(6), I(8) D(2), I(11)	False True True True
e⟶f	e	Delete & Insert LHS	D(13), I(11)	True
g⟶d d⟶f d⟶h	d	Delete RHS	D(9)	True False False
d⟶f d⟶h	d	Delete & Inser LHS	D(14), I(7)	True True
h⟶e	e	Delete RHS	D(1)	True

Table 8. Final sanitized database

TID	Items
0	c d e g
1	b d e f h
2	c d f h
3	a b f
4	a b c d e f h
5	b c
6	d f g
7	b
8	e g
9	a d g
10	b c d e f h
11	c
12	b d e f h
13	a c e f g
14	b c d f g h

D(6)= Delete item from transaction 6.
I(8)= Insert into Transaction 8.

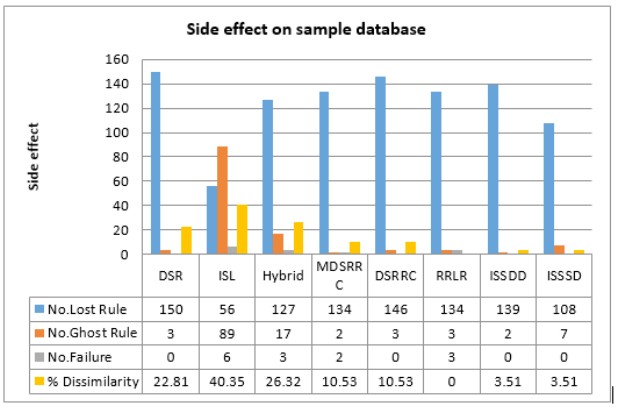

Figure 2. Result of sample dataset with single item in LHS and RHS.

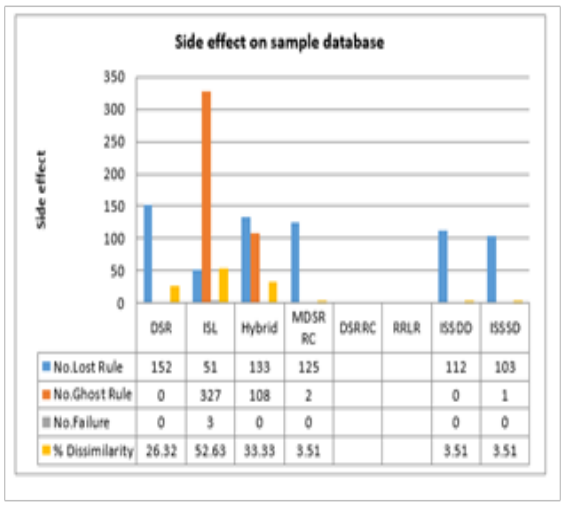

Figure 3. Result of sample dataset with multi item in LHS and RHS.

5. Comparison and Evaluation

Proposed algorithms with familiar algorithms ISL, DSR, Hybrid (ISL-DSR), MDSRRC, RRLR, DSRRC were Implemented and executed on Sony F115FM, CPU core i7, memory 6GB, HDD 500GB and windows 7 operating system with C# program language. Different tests with different rules applied to compare and evaluate on chess, mushroom, retail database. They are available via the FIMI repository (http://fimi.cs.helsinki.fi/). Features of data sets are displayed in Table 9. The obtained results in Figure 4 are attached in appendix.

Table 9. Characteristics of the real database

Database name	Number of transactions	No. items	Avg. trans. Length	Type
Chess	3196	74	37.00	Dense
Mushroom	8124	119	23.00	Sparse
Retail	88162	16470	10.30	Sparse

The rate of side effects resulted from hiding sensitive rules are shown in Figure 5 based on 12 tests on chess database. Support sensitive rules refer to sensitive rules that the algorithm has the ability to hide. They are represented in percentage form.

As can be seen, DRLR and DSRRC algorithms cannot hide all sensitive rules because they are multiple item rules. Therefore, the number of lost rules in these algorithms is fewer than that in other algorithms.

ISL algorithm does not operate well in dense database and fails in hiding rules for 91.66%, on average. The algorithms should be assessed in the same conditions. The missing rules might be reduced in an algorithm but the number of sensitive rules it can hide might be fewer compared to other algorithms. The comparison is not fair in such a situation.

The rate of side effects resulted from hiding sensitive rules are shown in Figure 6 based on 12 tests on mushroom database. MDSRRC, Hybrid, ISL, DSR and DSRRC algorithms do not present good results in sparse database. All of these five algorithms failed in hiding rules in consequence.

The rate of side effects resulted from hiding sensitive rules are shown in Figure 7 based on 12 tests on mushroom database.

6. Conclusion

Considering the present results, it can be concluded that proposed algorithms have less side effects on dense and sparse databases. In compared algorithm due to lack of control on hidden rules, there is a possibility for failure in specific conditions and it has been solved by controlling sensitive rules. Failure in hiding proposed algorithm is equal to zero. The value of ghost rules depends on non-sensitive items and the length of LHS in sensitive rule. If an item, which was selected for preserving is on the left side and there are lots of items on the left side, inserting all of them will certainly cause making ghost rule. Non-

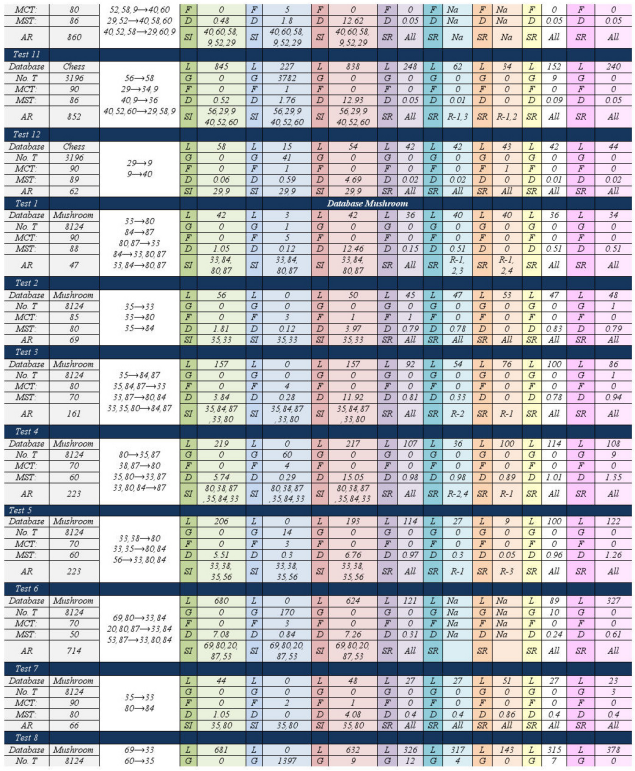

Figure 4. Result of tests.

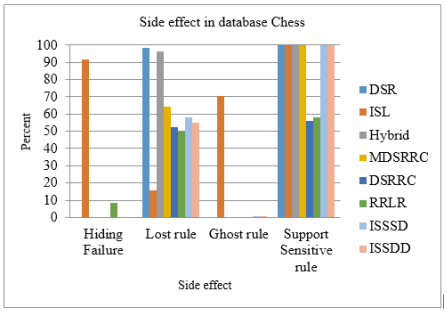

Figure 5. Result from hiding sensitive rules on 12 tests on Chess database.

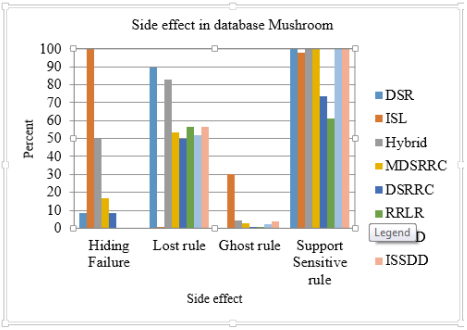

Figure 6. Result from hiding sensitive rules on 12 tests on Mushroom database.

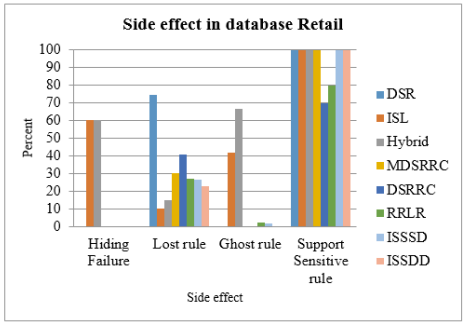

Figure 7. Result from hiding sensitive rules on 5 tests on Retail database.

sensitive rules are always in danger of missing even if insertion is only used for hiding like ISL algorithm that non-sensitive rules will be lost. Some of non-sensitive rules will be lost eventually with every kinds of technique in hiding. For instance, "hcd" itemset involves h⟶cd, d⟶hc, c⟶dh, c⟶d, h⟶d the support of this item will be declined unwillingly with every type of technique in hiding and these rules may be lost in proposed algorithms. It was attempted to select the best items that are present in minimum non sensitive rules.

6. References

1. Verykios VS, Bertino E, Fovino IN, Provenza LP, Saygin Y, Theodoridis Y. State of the art in privacy preserving data mining, SIGMOD Rec. 2004; 33:50–7.
2. Atallah M, Bertino E, Elmagarmid A, Ibrahim M, Verykios V. Disclosure limitation of sensitive rules, Knowledge and Data Engineering Exchange (KDEX). 1999; 43:45–52.
3. Wang SL, Jafari A. Hiding sensitive predictive association rules. Systems, Man and Cybernetics. 2005; 1:164–9.
4. Wang SL, Parikh B, Jafari A. Hiding informative association rule sets. Expert Systems with Applications. 2007; 33:316–23.
5. Modi CN, Rao UP, Patel DR., Maintaining privacy and data quality in privacy preserving association rule mining, Computing Communication And Networking Technologies. 2010; 32:1–6.
6. Shah K, Thakkar A, Ganatra A. Association rule hiding by heuristic approach to reduce side effects and hide multiple R.H.S. items. International Journal of Computer Applications. 2012; 45:1–7.
7. Domadiya NH, Rao UP. Hiding sensitive association rules to maintain privacy and data quality in database, Advance Computing Conference (IACC). 2012; 32:1306–10.
8. Jain YK, Yadav VK, Panday GS. An efficient association rule hiding algorithm for privacy preserving data mining, International Journal on Computer Science and Engineering. 2011; 3:2792–8.
9. Vijayarani S, Prabha MS. Association rule hiding using artificial bee colony algorithm, International Journal of Computer Applications. 2011; 33:41–7.
10. Oliveira SM, Za¨ıane OR. Algorithms for balancing privacy and knowledge discovery in association rule mining. Seventh International Database Engineering and Applications Symposium, 2003. Proceedings. 2003; 56:54–63.
11. Duraiswamy K, Manjula D, Maheswari NA. New approach to sensitive rule hiding. Stud Comp Intell. 2008; 1:107–11.
12. Menon S, Sarkar S, Mukherjee S. Maximizing accuracy of shared databases when concealing sensitive patterns. Information System Research. 2005; 16:256–570.
13. Verykios VS, Pontikakis ED, Theodoridis Y, Chang L. Efficient algorithms for distortion and blocking techniques in association rule hiding, Distributed and Parallel Databases. 2007; 22:85–104.
14. Amiri A. Dare to share: Protecting sensitive knowledge with data sanitization. Decision Support Systems. 2007; 43:181–91.
15. Dehkordi MN, Badie K, Zadeh AK. A novel method for privacy preserving in association rule mining based on genetic algorithms. Journal of Software. 2009; 4:555–62.
16. Ramakrishnan M. Switch pattern encryption based WBAN security in an IOT environment. Indian Journal of Science and Technology. 2015; 8:67–98. DOI: 10.17485/ijst/2015/v8i34/85274.