Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 4 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 4 Special Issue (2015)
ISSN: 1300-1949

Hiding sensitive rules using SIF-IDF to preserve privacy in extracting association rules

Negar OMIDI[1], Sima EMADI[1,*]

[1]*Department of Computer Engineering, Yazd Branch, Islamic Azad University, Yazd, Iran.*

Received: 22.03.2015; Accepted: 29.05.2015

Abstract. Nowadays, data mining and privacy preserving are two important and fundamental issues for organizations, individuals, and data miners. Data mining discovers the relations among the items of a database. Some of the discovered relations are private for organizations and individuals and must not be available to others. This information is called sensitive information and the database owner tries to hide it. Hiding sensitive information has some side effects for the database and insensitive information including loss of insensitive information, creation of new information that doesn't exist in the original database (ghost rules), dissimilarity in the database, etc. All the presented algorithms for privacy preserving try to sanitize databases with the least side effects. In this paper, an algorithm based on SIF-IDF algorithm in order to hide sensitive rules is proposed. In the proposed algorithm, heuristic technique and support-based approach are used for sanitizing databases. The aim of the proposed algorithm is reducing the side effects of database sanitization including loss of rules, runtime reduction and hiding failure. The proposed algorithm is assessed by 1.b, MDSRRC, and SIF-IDF algorithms and the results show the efficacy of the proposed algorithm.

Keywords: Sensitive information, Data mining, Privacy preserving

INTRODUCTION

Nowadays, due to increasing amount of data and advances in technology, data mining is a popular tool by which users can extract useful information from a database in a short time. Different algorithms like Apriori, FP-tee and Eclat have been presented to extract information and useful relations .These algorithms differ in the time and method of information extraction but the final results of all of them are the same. The extracted information from these algorithms is called association rules that is a subset of descriptive method in data mining and shows the useful relations among the items of a database.

Privacy preserving in data mining tries to sanitize sensitive information by changing the database items or structure. Different methods and techniques have been proposed for database sanitization. Generally three techniques, heuristic, border based and exact, are proposed to sanitize centralized databases. [1] The heuristic technique includes distortion approach in which sensitive information is hidden by deletion or insertion of an item and blocking approach in which sensitive information is hidden by substitution of the database item for question mark (?). For hiding a sensitive rule one can reduce the support of the sensitive rule by deletion of one of the items of the sensitive rule; this is called support-based approach. Or one can reduce the confidence of the sensitive rule by increasing the support of all items on the left side of the sensitive rule, this one is called confidence-based approach.

*Corresponding author. *Email:* emadi@iauyazd.ac.ir

Special Issue: Technological Advances of Engineering Sciences

For the first time, in 1999, Atalla et al. proposed cyclic algorithm with the purpose of hiding sensitive rules .This algorithm tries to reduce the support of sensitive rules by drawing a graph of itemsets . [2] Then Dasseni et al. proposed three ways of hiding sensitive rules. In the first method, the sensitive rules are hidden by decreasing their confidence by inserting the left –side items. The second method reduces the support by deleting the items from the right side of sensitive rules. And the third item reduces support or confidence by choosing and deleting the best item from right or left side. [3] Verykios et al. used two methods, distortion and blocking, to hide sensitive rules. BA algorithm hides sensitive rules using blocking technique and SM criterion; and WSDA do this by distortion technique and item deletion. In these algorithms the main focus is on selecting the suitable transaction. [4] Shah et al. proposed two algorithms, PRLR and ADSRRC to sanitize databases by hiding sensitive rules. ADSRRC algorithm hides the rules that have just one item on their right side using support-based approach. This algorithm tries to hide several rules simultaneously by clustering rules based on the right-side item. PRLR algorithm hides the rules that have one item on the left side using confidence –based approach. Operationally, PRLR is more efficient than ADSRRC.[5]After that, Damandiya and Rao proposed MDSRRC for eliminating the limitations of ADSRRC. In this algorithm the best item is selected based on its sensitivity level in sensitive rules, then transactions are organized based on the selected item. This algorithm tries to hide sensitive rules by support reduction. [6] Hong et al. tried to hide sensitive itemsets to sanitize a database. SLF-IDF hides sensitive itemsets by using support-based approach. In this algorithm, a shorter transaction including more sensitive items is selected to delete the victim item. In this algorithm, the main focus is on selecting the best transaction. The shortcomings of this algorithm are long runtime, having infinite loop, and different results are observed in each sanitization when the order of entering itemsets change. [7] Lin et al. suggest HMAU algorithm to sanitize a database by hiding sensitive itemsets. In this algorithm, some transactions are omitted from the database to reduce the support of the sensitive itemset. The best transaction is selected based on insensitive itemsets, hiding failure and production of ghost itemsets. [8] Ghalesefidi and Dehkordi proposed an algorithm that is a combination of support-based and confidence –based approaches .In this algorithm, the target approach is selected based on the support level of left-side and right-side items. If the support of left-side items is more , the confidence-based approach will be taken. If it is not the case, support-based approach will be adopted. [9] Table 1 shows the side effects of each proposed algorithm.

Table 1. The side effects of each proposed algorithm.

Algorithm	Lost rule / Itemset	Ghost rule/Itemset	Hiding failure	Dissimilarity
Cyclic	√			√
1.a	√	√	√	√
1.b	√		√	√
2.a	√			√
BA	√	√		
WSDA	√			
ADSRRC	√			√
RRLR	√	√	√	
MDSRRC	√		√	√
SIF-IDF	√		√	√
HMAU	√			√
[9]	√			√

Generally all the presented algorithms for privacy preserving try to sanitize databases with the least side effects. It is not possible to eliminate all the side effects simultaneously. In general,

algorithms consider three main aims that are reducing the lost rules, hiding failure and ghost rules. **The main purpose of this research is preventing the disclosure of sensitive rules and controlling side effects of hiding sensitive rules including lost insensitive rules and runtime reduction.** SLF-IDF algorithm presented in [7] selects the best transaction by exact calculations. But due to sanitization based on sensitive itemsets, it leads to losing many insensitive rules. To solve this problem, in this research, hiding is carried out based on SLF-IDF algorithm method and mapping sensitive rules instead of sensitive itemsets. The structure of this paper is as following: section 2 explains the association rules format. Section 3 is about the proposed algorithm, its concepts and explanation of its steps. Section 4 shows a case study of the proposed algorithm. Section 5 compares and assesses the proposed algorithm with 1.b , MDSRRC and SLF-IDF and the conclusion is presented in section 6.

Association rules Format:

The relations of the items of a database are determined by support and confidence criteria. First, by using data mining algorithms, the support level of the database items are identified and the frequent sets are extracted. Then, the relations among frequent sets are discovered using the confidence level. Two criteria, MST (minimum support threshold) and MCT (Minimum Confidence Threshold) are presented to select the useful relations from the discovered relations. If the support level of the item is more than MST , the item will be extracted from the database as a frequent itemset. And if the confidence of the rule is more than MCT, that rule will be extracted as a useful rule. Assume the main database is D and the items of the database are in the set $I=\{i_1,1_2,..,i_n\}$. Every database includes a set of transactions in the form of $T=\{t_1,1_2,...,t_k\}$. Every transaction T includes a subset of items in I. $A\rightarrow B$ if $A\cap B=\emptyset$ is the general format of association rules. For calculating the support level of element I, one should have its frequency in the database and calculate the ratio of the frequency to the number of transactions. [10] Equation (1) shows the support level of AB and the second equation demonstrates the confidence level of $A\rightarrow B$.

$$Support(AB) = \frac{|A \cup B|}{|D|} \quad (1)$$

$$Confidence = \frac{|A \cup B|}{|A|} \quad (2)$$

The proposed algorithm:

The proposed algorithm is presented based on SLF-IDF algorithm in order to sanitize databases. SLF-IDF hides sensitive itemsets but the proposed algorithm hides sensitive rules. Hiding based on sensitive itemsets leads to losing more insensitive rules. Therefore, by mapping on SLF-IDF, sensitive rules are used to sanitize databases. For selecting the suitable transaction, some changes are made in the relations of SLF-IDF algorithm. Also this algorithm is different from SIF-IDF in the selection of the victim item and the process of performing the commands for selecting the best transaction to apply the changes. The proposed algorithm pursues the following goals:

1. Reducing sensitive rules hiding failure.
2 Reducing runtime.
3. Reducing lost rules.

Selecting the transaction and victim item significantly affects the amount of lost rules. In SLF-IDF the shortest transaction with the most sensitive itemsets is selected to eliminate the victim item. Equation (3) shows the transactions in SLF-IDF algorithm.

$$\text{SIF-IDF}(T)= \sum_{i=1}^{n} \left(\frac{|s_{ij}|}{|T|} * \sum_{k=1}^{p} \log \frac{|n|}{|f_k\text{-}MRC_k|}\right) \quad (3)$$

In this equation, S_{ij} is the length of transaction T, the number of sensitive items in the transaction. F_k is the item frequency in the database and n shows the number of transactions in the database. MRC (maximum reduced count) is the maximum number of reductions for each item in the database.

The stages of the proposed algorithm are shown in figure (1) in the shape of a flow chart. Each stage is briefly explained in section 1-3.

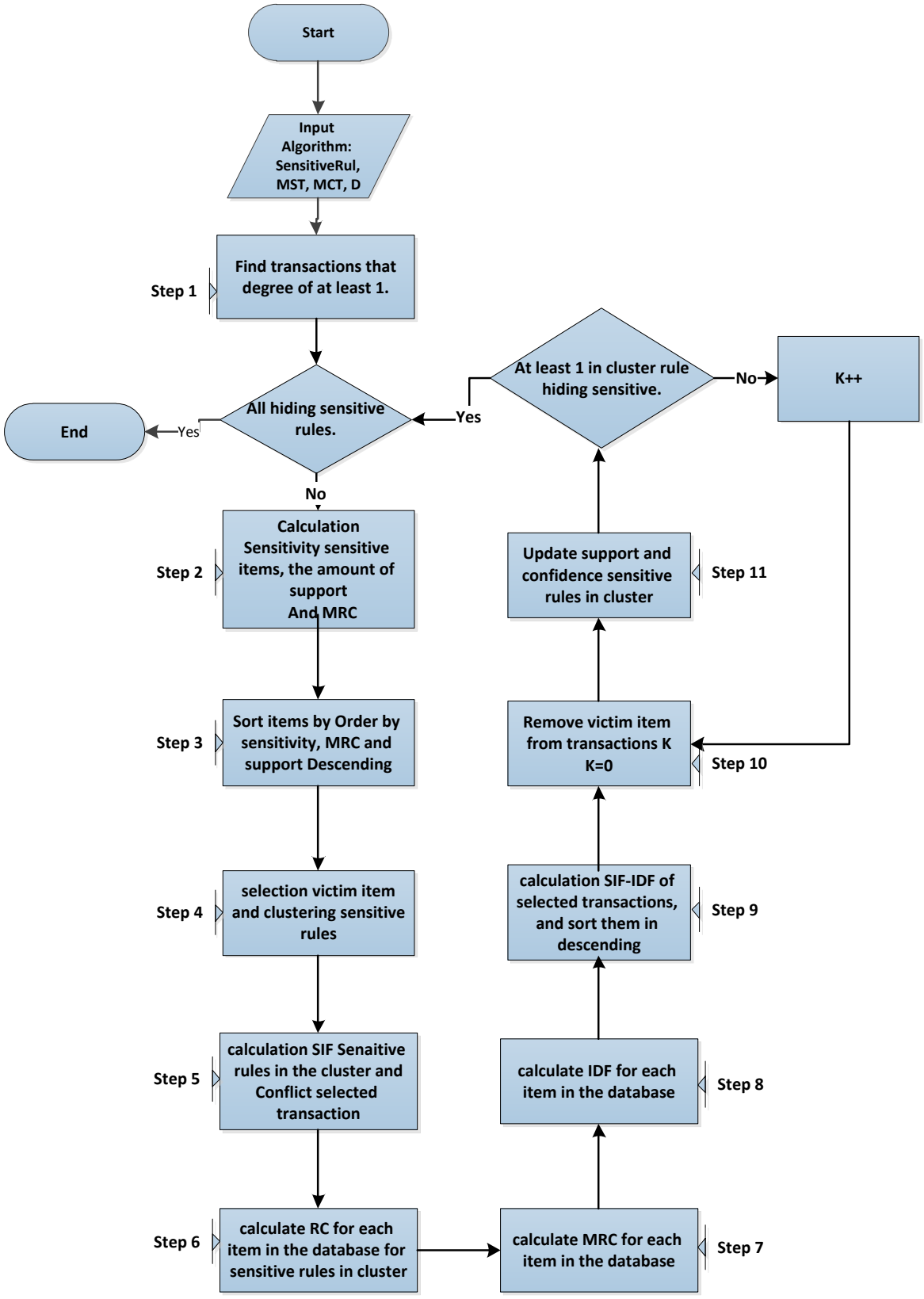

Figure 1. Flowchart of proposed algorithm.

Stages of the proposed algorithm

Stage 1: Selects the transactions which completely support at least one sensitive rule. The number of rules that are completely present in the transaction are called the conflict degree of the transaction. In the proposed algorithm, the shortest transaction with the highest conflict degree is selected to delete the victim item. The calculation time is reduced by pruning transactions based on the conflict degree.

Stage2: Identification of sensitive items, calculating sensitivity, support level and the MRC of sensitive items are done in this stage. The items on the right side of the sensitive rule are the sensitive items in this algorithm. Their sensitivity equals the number of their repetitions in sensitive rules. The way the MRC of sensitive items is calculated is like the way used in SIF-IDF algorithm. But in SIF-IDF algorithm, the maximum reduced count is selected for database items while in the proposed algorithm, minimum reduced account is selected for sensitive items. Equation (4) shows how to calculate MRC and equation (5) shows how to calculate RC.

$$MRC_k = min_{j=1}^{m} RC_{kj} \tag{4}$$

$$RC_{kj} = f_j - s \times n + 1 \tag{5}$$

In this equation, F_j is the frequency of item j in the database, S is MST criterion and n is the number of transactions in the original database.

Stage3: Sensitive items are sorted based on descending sensitivity, descending MRC and descending support. By sorting based on the descending sensitivity, the most repeated victim item in sensitive rules is selected; therefore, the deletion of one item can influence several sensitive rules. When two items have the same sensitivity, they will be sorted based on MRC. MRC shows the number of victim item deletions for hiding the sensitive rule. If they have the same MRC, they will be sorted based on their frequency in the database. The more the MRC and support level of an item are, the less lost rules will exist.

Stage 4: Selecting the victim item and clustering sensitive rules based on the victim item happen. In this algorithm, clustering is done whitin the runtime and based on the presence of the victim item on the right side of the sensitive item. It is possible to have the problem of infinite loop in SIF-IDF algorithm which means the algorithm falls into an infinite loop because the algorithm didn't succeed to delete an element due to the selection of an unsuitable transaction and victim item. When SIF-IDF algorithm sorts the transactions, the victim item may not be present in the first transaction, so it cannot delete the victim item. It repeats the calculations but the same transaction is selected as the best transaction again. For solving this problem and hiding several sensitive rules with the least changes, clustering is presented in the proposed algorithm. The rules that are present in the cluster at each stage are considered for next calculations and other rules don't become involved.

Stage5: Calculating the SIF of each sensitive rule is based on SIF-IDF algorithm. But in the proposed algorithm, in addition to calculating the SIF of the rules, the ratio of the conflict degree of each transaction to all the sensitive rules that are present in the cluster is calculated. This increases the possibility of selecting the transaction which supports more sensitive rules to delete the victim item. The more the conflict degree is, the less side effects will be. Equation (6) [7] shows how to calculate the SIF of sensitive rules and equation (7) shows how to calculate conflict degree of transactions.

$$SIF_{ij} = \frac{|s_{ij}|}{|T_i|} \quad (6)$$

$$Conflict = \frac{|R_{T_i}|}{|RS|} \quad (7)$$

In this equation, S_{ij} is the number of sensitive items in the transaction and T_i is the number of items in the transaction. RS is the overall number of sensitive rules present in the cluster and R_{Ti} shows the number of the sensitive rules that are completely present in the transaction.

Stage 6: Calculating RC, MRC, IDF is like calculating them in SIF-IDF algorithm. Equation (8) shows how to calculate MRC. Equation (9) shows the calculation of IDF for each item in the database. [7]

$$MRC_k = max_{j=1}^{m} RC_{kj} \quad (8)$$

$$IDF_k = log \frac{|n|}{|f_k - MRC_k|} \quad (9)$$

Stage 7: Calculating the SIF-IDF of transactions is performed using equation (10). In fact, the best transaction that is short and includes the most sensitive rules is selected to delete the victim item by multiplying equation (3) by equation (7).

$$\text{SIF-IDF}(T_i) = \sum_{i=1}^{n} \left(\frac{|s_{ij}|}{|T_i|} * \sum_{k=1}^{p} \log \frac{|n|}{|f_k\text{-}MRC_k|} \right) * \text{Conflict}(T_i) \quad (10)$$

Stage8: Transactions are sorted based on SIF-IDF in a descending way.

Stage 9: The victim item is deleted from the transaction when at least one sensitive rule hid. So the runtime sharply decreases. In fact, if we have n sensitive rules, the maximum number of calculation repetitions in the proposed algorithm will equal n and the minimum number will depend on the common item among sensitive rules. While, in DIF-IDF algorithm, for hiding just one sensitive rule the calculation repetition equals [Support (itemset)-MST]+1. Hence, for hiding n sensitive itemsets in SIF-IDF algorithm the maximum number of calculation repetition equals $\sum_{i=1}^{n}$[support($itemset_i$)-MST]+1. When at least one of the sensitive rules present in the cluster is hidden, stages 2 to 10 are repeated for all sensitive rule to become hidden.

CASE STUDY

In this section, a database with 10 transactions and 9 items is considered to show the efficacy and performance of the proposed algorithm. Look at table (2) presented in [7]. In this example, 6 sensitive rules are selected for hiding. Table (3) shows the input of the algorithm.

Table 1. Sample database.

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

Table 2. Input Algorithm.

Sensitive Rules	e→f, d→f, f→d, g->d, h→f, ch→df
MST	30%
MCT	60%

At the first stage, the transactions which include at least one sensitive rule are selected. Table (4) shows the sensitive transactions.

Table 3. Sensitive transaction.

TID	Items
1	h, g, f, d, c, b, a
3	h, g, f, d, c, b
4	h, f, c, b, a
5	i, g, e, d, c
7	g, f, e, d, c, b
8	i, h, f, d, c
9	i, f, e, d, a
10	h, f, e, c, a

At the second and third stages the sensitive items are identified and sorted based on their levels of sensitivity, MRC and support in a descending way. Table (5) shows sensitive items.

Table 4. Sensitivity of each Sensitive items.

Items	Sensitivity	MRC	Support
f	5	8	2
d	4	7	2

At the fourth stage, the first item is selected as the victim item and sensitive rules are clustered based on that. In this example, the rules which have item f on their right side are selected for sanitization. Therefore, the cluster includes rules e→f, d→f, h→f, ch→df.

At the fifth stage, the SIF value of each transaction and the conflict degree of transactions are calculated based on equations (5) and (6). For example, for calculating $SIF_{e\rightarrow f}$ in transaction 1, the number of sensitive rules of rule e→f in the transaction equals 1 and the number of items of transaction equals 7; so $SIF_{e\rightarrow f}$ equals 2/7. Transaction 1 completely supports rules d→f ،h→and ch→df ; so conflict equals 3/4. Table (6) shows the calculation results of each transaction.

Table 5. The SIF values of sensitive Rule in each transaction.

TID	Items	$SIF_{e\rightarrow f}$	$SIF_{d\rightarrow f}$	$SIF_{h\rightarrow f}$	$SIF_{ch\rightarrow df}$	Conflict
1	h, g, f, d, c, b, a	0.143	0.286	0.286	0.571	0.75
2	h, g, f, d, c, b	0.167	0.333	0.333	0.667	0.75
3	h, f, c, b, a	0.2	0.2	0.4	0.6	0.25
4	i, g, e, d, c	0.2	0.2	0	0.4	0
5	g, f, e, d, c, b	0.333	0.333	0.167	0.5	0.5
6	i, h, f, d, c	0.2	0.4	0.4	0.8	0.75
7	i, f, e, d, a	0.4	0.4	0.2	0.4	0.5
8	h, f, e, c, a	0.4	0.2	0.4	0.6	0.5

At the sixth stage, the IDF value of each item in the database and the RC and MRC values of each sensitive rule are calculated. For example, for calculating $RC_{e\rightarrow f}$, item f is in the form of (3-(0.3*8))+1. The sensitive rule frequency is 3, MST value is 0.3 and the number of sensitive transactions is 8. The maximum RC for each item is regarded as MRC value. Table (7) shows the MRC value of each item in the database.

Table 6. The MRC value of each item in Database.

Items	$RC_{e\rightarrow f}$	$RC_{d\rightarrow f}$	$RC_{h\rightarrow f}$	$RC_{ch\rightarrow df}$	MRC
h	0	0	4	2	4
g	0	0	0	0	0
f	2	4	4	2	4
d	0	4	0	2	4
c	0	0	0	2	2
b	0	0	0	0	0
a	0	0	0	0	0
e	2	0	0	0	2
i	0	0	0	0	0

The IDF value of each item in the database is calculated by equation (9). Table (8) shows the IDF values of the items in the database.

Table 7. The IDF value of each items.

Items	Count	MRC	IDF
h	5	4	0.903
g	4	0	0.301
f	7	4	0.426
d	6	4	0.602
c	7	2	0.204
b	4	0	0.301
a	4	0	0.301
e	4	2	0.602
i	3	0	0.426

At the seventh stage, the IDF and IDF-SIF values of transactions are calculated. For calculating IDF value of the transaction, the sum of the sensitive items present in the transactions is considered. For example, calculating the SIF-IDF of transaction 1 is performed as follows. Table (9) shows the SIF-IDF values of the transactions.

$$\text{SIF-IDF}=[(IDF_{e\rightarrow f} \times SIF_{e\rightarrow f})+(IDF_{d\rightarrow f} \times SIF_{d\rightarrow f})+(IDF_{h\rightarrow f} \times SIF_{h\rightarrow f}) +(IDF_{ch\rightarrow df} \times SIF_{ch\rightarrow df})] \times \text{Conflict}$$

Table 8. The IDF and SIF_IDF value for all the transaction.

TID	$IDF_{e\rightarrow f}$	$IDF_{d\rightarrow f}$	$IDF_{h\rightarrow f}$	$IDF_{ch\rightarrow df}$	$SIF_{e\rightarrow f}$	$SIF_{d\rightarrow f}$	$SIF_{h\rightarrow f}$	$SIF_{ch\rightarrow df}$	SIF-IDF
1	0.426	1.028	1.329	2.135	0.143	0.286	0.286	0.571	1.465
2	0.426	10.28	1.329	2.135	0.167	0.333	0.333	0.667	1.71
3	0.426	0.426	1.329	1.533	0.2	0.2	0.4	0.6	0.405
4	0.602	0.602	0	0.806	0.2	0.2	0	0.4	0
5	1.028	1.028	0.426	1.232	0.333	0.333	0.167	0.5	0.685
6	0.426	10.28	1.329	2.135	0.2	0.4	0.4	0.8	2.051
7	1.028	10.28	0.426	1.028	0.4	0.4	0.2	0.4	0.659
8	1.028	0.426	1.329	1.533	0.4	0.2	0.4	0.6	0.973

At the eighth stage, the transactions are sorted based on SIF-IDF values in a descending way. At the ninth stage, the victim item is deleted from the first transaction and this continues until at least one of the sensitive rules becomes hidden. Table (10) shows the sorted transactions.

Table 9. The sorted transactions according to the SIF-IDF values.

TID	Items	SIF-IDF
6	i, h, f, d, c	2.051
2	h, g, f, d, c, b	1.71
1	h, g, f, d, c, b, a	1.465
8	h, f, e, c, a	0.973
5	g, f, e, d, c, b	0.685
7	i, f, e, d, a	0.659
3	h, f, c, b, a	0.405
4	i, g, e, d, c	0

Table (11) shows the final sanitized database. The percentage of lost rules in the proposed algorithm and three basic algorithms is shown in table (12). As you see, there are less lost rules in the proposed algorithm than in other three algorithms.

Table 10. The final sanitized database.

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

Table 11. Percentage of Lost rule in sanitized database by 4 algorithm.

Algorithm name	Lost rule
Proposed	73.71
1.b	86.38
MDSRRC	83.1
SIF-IDF	89.67

ASSESSMENT

The proposed algorithm and three SIF-IDF, 1.b and MDSRRC algorithms, are assessed in a Core i7 and RAM 6G system. For doing the tests, two real databases, Mushroom and Chess, were chosen. The features of the databases are shown in table (13).

Table 12. The parameters of the Chess and Mushroom database.

Database Name	Number of items	Average transactions size	Number of transactions
Chess	74	37	3196
Mushroom	119	23	8124

Different tests with different MST and MCT values were done in both databases. Figure (2) shows the percentage of lost rules in Chess database. 1.b algorithm loses the most insensitive rules and the proposed algorithm has the least lost rules. The more the number of sensitive rules and commonality of sensitive rules are, the better the results of the proposed items will be.

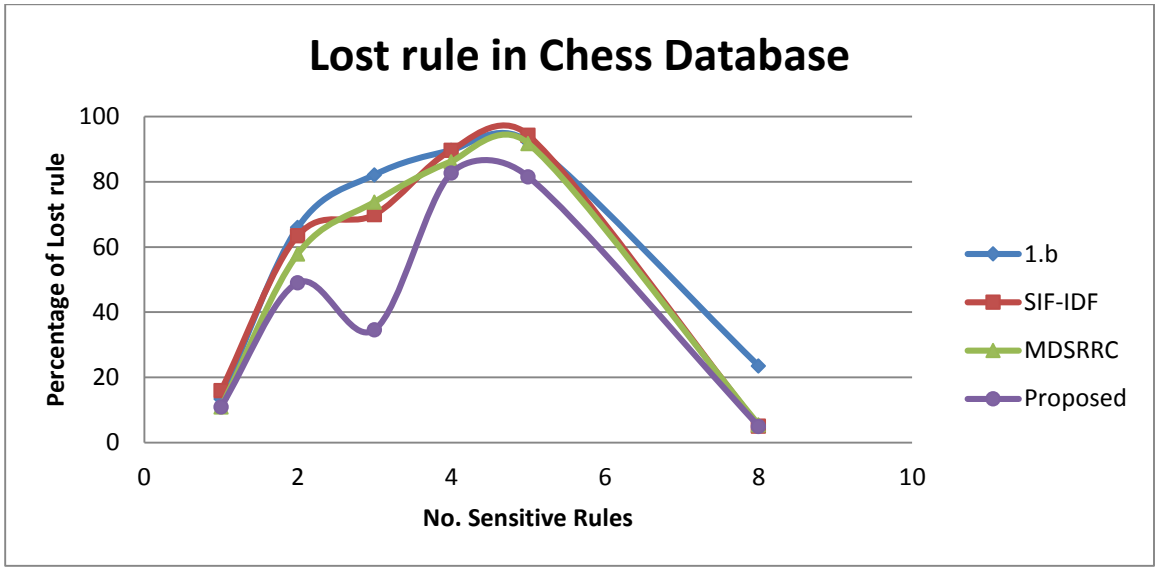

Figure 2. Percentage of lost rules in Chess database.

Figure (3) shows percentage of lost rules in Mushroom database. In Mushroom database, since the transactions are short and not dense, the percentage of lost rules in the proposed database has less difference than SIF-IDF and MDSRRC in tests 2 and 4 of sensitive rules. But when the number of sensitive rules increases, the results of the proposed algorithm improves.

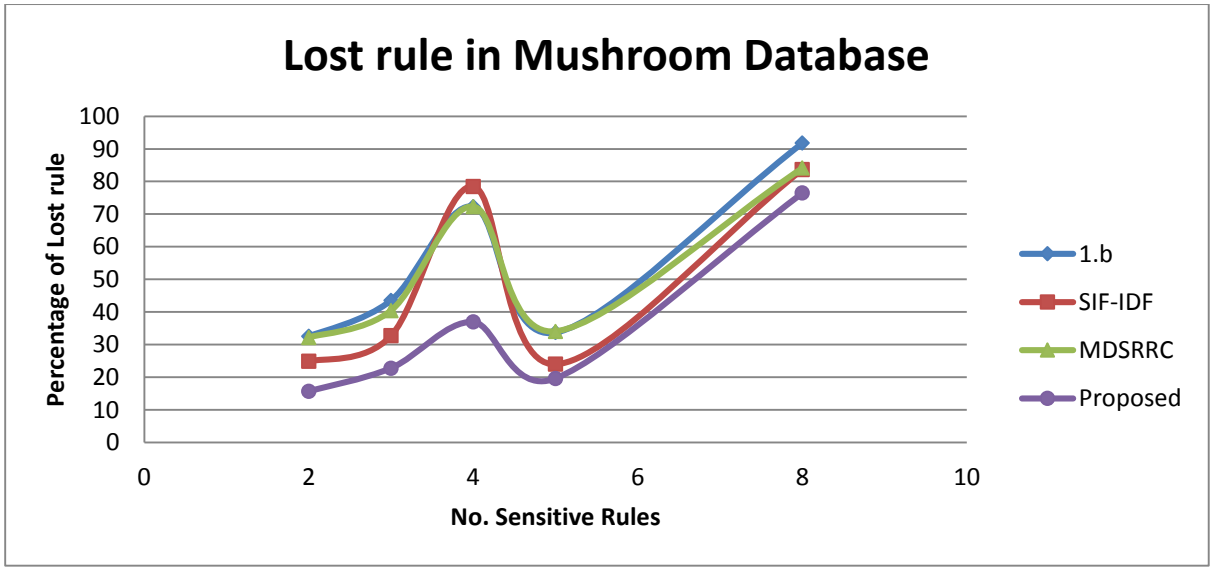

Figure 3. Percentage of lost rules in Mushroom database.

The runtime of the proposed algorithm is better than the runtime in MDSRRC and SIF-IDF algorithms and has little difference from the runtime in 1.b algorithm. Figure (4) shows the runtime of the proposed algorithm in Chess database and Figure (5) and (6) show the runtime in Mushroom database (in seconds).

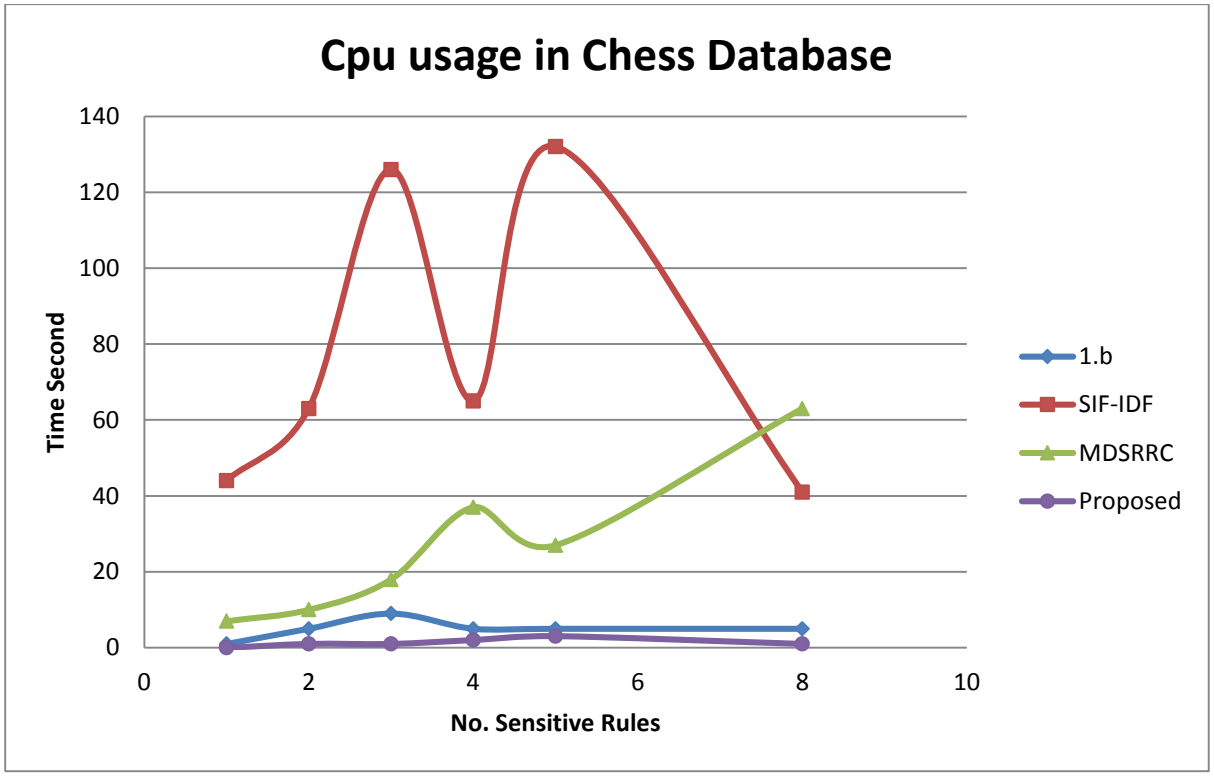

Figure 4. CPU usages in Chess database.

Since the runtime is long in SIF-IDF algorithm within Mushroom database and to clearly show the runtime of other algorithms in this database, the runtime is shown in two figures.

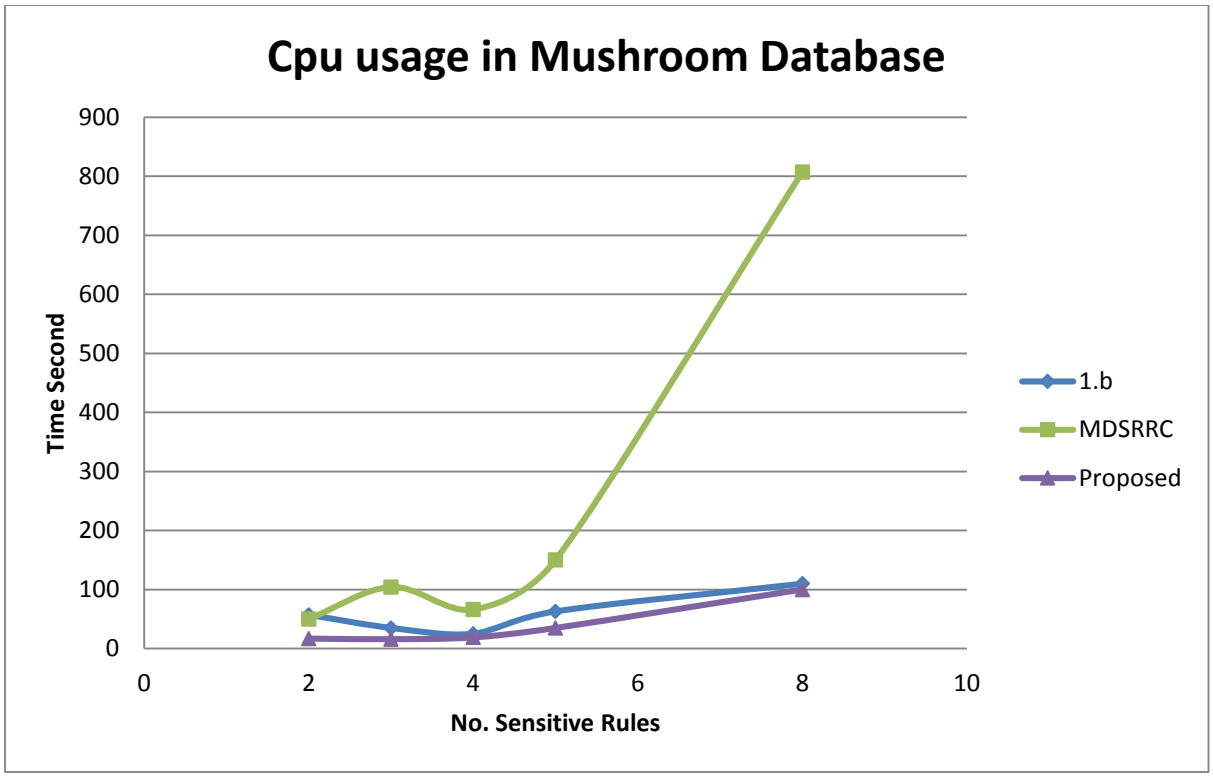

Figure 5. CPU usages in Mushroom database.

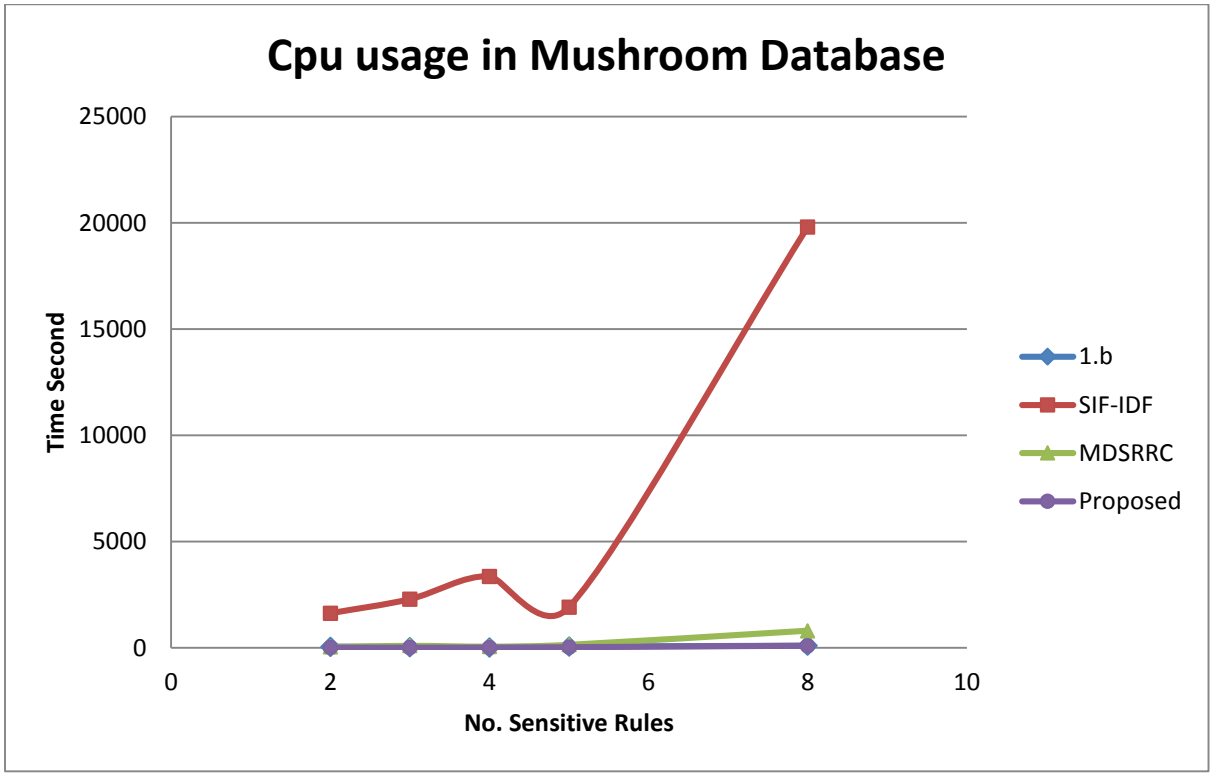

Figure 6. CPU usages in Mushroom database.

In addition to runtime and lost rules, the hiding failure in Chess and Mushroom databases is assessed too. Figure (7) shows the average hiding failure in the databases.

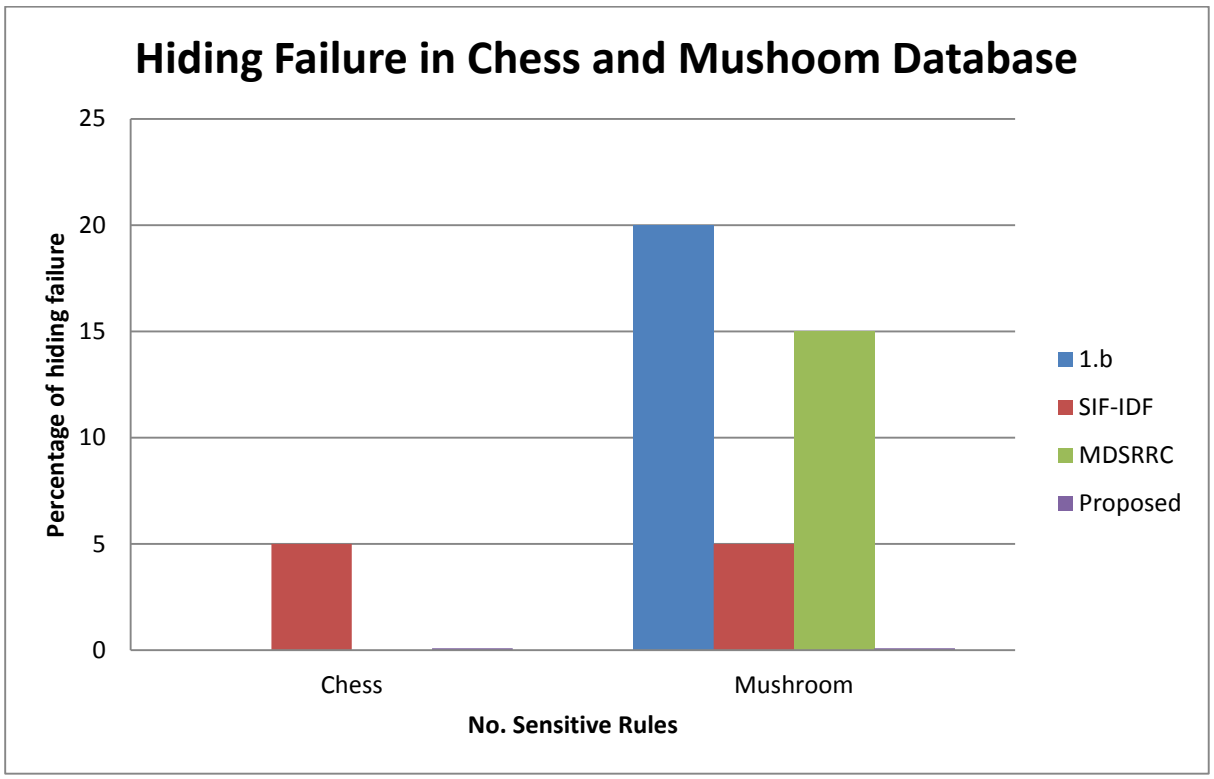

Figure 7. Average of Hiding failures in Chess and Mushroom database.

CONCLUSION

In this paper, a new algorithm based on SIF-IDF algorithm was proposed to hide sensitive rules. The aims of the proposed algorithm were reducing hiding failure, runtime and lost rules as the side effects of database sanitization. The proposed algorithm hides sensitive rules using heuristic technique and support-based approach. In this algorithm, in addition to selecting the best transaction, it was tried to select the best victim item to hide sensitive rules. Hiding failure in the proposed was 0, in SIF-IDF within Chess and Mushroom databases was 5 percent, in MDSRRC algorithm was 15 percent and in 1.b algorithm was 20 percent within Mushroom database. The proposed algorithm protects more insensitive rules. Losing insensitive rules is more 1.b algorithm than in other algorithms. Also, the proposed algorithm needs a shorter runtime for sanitization. In the present research, the conflict degree of the transaction was used for selecting the suitable transaction. In future works, insensitive items can be used to select the best transactions. Also, in selecting the victim item, several victim items can be selected and deleted simultaneously to reduce the runtime.

REFERENCES

[1] V.S. Verykios, E. Bertino, I.N. Fovino, “State-of-the-art in privacy preserving data mining”, SIGMOD Record, Vol. 33, No. 1, March 2004, pp. 50–57.

[2] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios, “Disclosure limitation of sensitive rules”, Knowledge and Data Engineering Exchange, November 1999, pp. 45-52.

[3] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, E. Bertino, "Hiding association rules by using confidence and support", IHW '01 Proceedings of the 4th International Workshop on Information Hiding, 2001, pp. 369–383.

[4] V.S. Yerykios, E.D. Pontikakis, Y. Theodoridis, L. Chang, ” Efficient algorithms for distortion and blocking techniques in association rule hiding”, Distributed and Parallel Databases, Vol.22, No.1, 2007, pp.85-104;

[5] K. Shah, A. Thakkar, A. Ganatra, “Association rule hiding by heuristic approach to reduce side effects & hide multiple R.H.S. items”, International Journal of Computer Applications, Vol. 45, No. 1, May 2012, pp. 1–7.

[6] N.H. Domadiya and U.P. Rao. “Hiding sensitive association rules to maintain privacy and data quality in database”, Advance Computing Conference, February 2012, pp. 1306–1310.

[7] T-P Hong, C-W Lin, K-T Yang, “Using TF_IDF to hide sensitive itemset”, Applied Intelligence, Vol.38, 2013, pp.502-510.

[8] C-W Lin, T-P Hong, H-C Hsu, “Reducing Side Effects of Hiding Sensitive Itemsets in Privacy Preserving Data Mining”, The ScientificWorld Journal, Vol. 2014, April 2014, pp.1-5

[9] N.J. Ghalehsefidi, M.N. Dehkordi, “A hybrid approach to privacy preserving in association rules mining”, Advances in Computer Science: an International Journal, Vol.3, No.12, November 2014, pp.69-72.

[10] Han, J., M Kamber, “Data mining:concept and techniques”, 2end ed., Diane Cerra, CA: San Francisco, 2006, pp.203-233.