ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 5, No.11 , September 2014
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

A Method for Hiding Association rules with Minimum Changes in Database

Zahra Sheykhinezhad[1], Mohammad Naderi dehkordi[2] and Hamid Rastegari[3]

[1] Department of Computer Engineering, Islamic Azad University–Najafabad Branch, Isfahan, Iran

S_sheykhinezhad@yahoo.com

[2] Department of Computer Engineering, Islamic Azad University–Najafabad Branch, Isfahan, Iran

Naderi@iaun.ac.ir

[3] Department of Computer Engineering, Islamic Azad University–Najafabad Branch, Isfahan, Iran

Rastegari@iaun.ac.ir

Abstract

Privacy preserving data mining is a continues way for to use data mining, without disclosing private information. To prevent disclosure of sensitive information by data mining techniques, it is necessary to make changes to the data base. Association rules are important and efficient data mining technique. In order to achieve this algorithm is proposed, that as well as hiding sensitive association rules, having the lowest side effects on the original data set. Proposed algorithm by removing selective item, among items of antecedent sensitive rule (L.H.S.), causes to decrease confidence of sensitive rule below less them threshold and hide the sensitive rule. Also keeps sensitive rules until the end of securing process is reduce the failure hiding, and because the internal clustering, hiding sensitive rules performed synchronic takes insensitive rules to reduce the loss. This algorithm is compared with basic algorithm, on dense and sparse data base. The results with criteria of hiding failure, is indicates 41.6% improvement in dense data base and 28% in made with software data base. With criteria of lost rule, is indicates 70%, 57.1% and 83.3% improvement over the base algorithm. Which indicates the proposed algorithm is efficient.

Keywords: *Privacy Preserving Data Mining, Association Rules, Hiding Sensitive Rules, The security data base.*

1. Introduction

Recently, significant improvements in data collection, data Storage technology and the widespread use of The World Wide Web have led to huge volumes of data. Therefore, data mining method in their to extract information automatically and intelligently or knowledge from large amounts of data. Despite the fact that it can be the owners of data in strategic planning, and decision-making, it also may lead to the disclosure of sensitive Information. Thus, the parallel development Data mining, including the types of questions can be raised Are data sources used for other than the main aim. So, new topic in the data mining Tell that to design a data-mining system with privacy, which can be faster, high-volume the data storage and the ability to prevent disclosure Sensitive information. For this reason, privacy is maintained Data mining has been widely studied by researchers [1].

Privacy in Association rule mining of considerable research in data mining. To extract and reveal hidden relationships and structures, interesting relationships between large sets of data in a database transaction. Today, many organizations and companies protect their data collection and transaction processing, data mining, knowledge mining relationship [2],[3]. In this paper, we present a privacy preserving mining law relationship focus. In doing so we assume that some subset of Rules, which are extracted from a specific data set, Rules are considered as sensitive. In this paper, we focus on privacy preserving association mining rules. In doing so we assume that some the following set of rules, which are extracted from certain Data set, considered as sensitive rules[4]. Our goal then is The original data source is modified so that it Would be impossible for the enemy to mine sensitive Terms of improved data collection and the Hand, in order to minimize side

effects created by the hiding The process of sanitizing a process can affect the by a set of rules

I. sensitive rules are hidden or removed before the process of sanitizing the mining laws (lost rules)

II. II. Mining and mining disclosure rules Unreal Database changes that were not supported by Original database (ghost rules)[5].

2. Background and Related Work

Approach relying on data obscuration, modifying the data Values so real values are not revealed1. As, A major feature of PPDM techniques is entail modifications to the data in order to sanitize them from sensitive information (both private data items and complex data correlations) or anonymity them with some uncertainty level. Therefore, in evaluating a PPDM algorithm it is important to determine the quality of the transformed data. To do so, we need methodologies for the estimation of the quality of data, intended as the state of the individual items in the database resulting from the application of a privacy preserving technique, and also the quality of the Information that is exposed and extracted from the modified data by using a given data mining method[2]. Verykios et al. categorized PPDM techniques as
Five different dimensions: (1) data distribution; (2) data Modification; (3) the data mining algorithm which the Privacy preservation technique is proposed and designed
For; (4) the data type (single data items or complex data Correlations) that needs to be protected from reveal; (5)
Preserving privacy approach (heuristic, reconstruction or cryptography-based approaches). Clearly, it does not include all the possible PPDM algorithms. However, it gives the algorithms that have been designed and proposed so far, centralizing on their main features. Data Mining discovers inferences that are interesting, but do not always hold. Methods and ways have been proposed
To alter and modify data to bring the support or confidence of specific rules below a threshold [6], [7].
The remainder of this paper is as follows: First, the basic definitions of main issue research and data mining association rules are discussed. The proposed algorithm for hiding sensitive rules has been presented. Finally the results of the proposed approach and the future work are provided.

3. Problem Formulation

3.1 Transactional Databases

A transactional database is a relation consisting of transactions in which each transaction t is determined by an ordered pair, defined as $t = <TID, list\ of\ elements>$,
Where *TID* is a unique transaction identifier number and list of items expresses a list of items composing the

3.2 The Basics of Association Rules

Formally, association rules are defined as follows: Let $I = \{i1,..., in\}$ be a set of literals, called items. Let D be a database of transactions, where each transaction t is an item set such that $t \subseteq I$. A unique identifier, called *TID*, is associated with each transaction. A transaction t supports X, a set of items in I, if $X \subset t$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y=\emptyset$.
Thus, we say that a rule $X \Rightarrow Y$ holds in the database D with *confidence (MCT)* if $\frac{|X \cup Y|}{|X|} \geq$MCT where $|X|$ is the Number of occurrences of the set of items X in the set of transactions D. Similarly, we say that a rule $X \Rightarrow Y$
Hold in the database D with *support (MST)* if $\frac{|X \cup Y|}{|D|} \geq$MST where D is number of transactions in database D.
Association rule mining algorithms depend on support And confidence and mainly have two major phases:
I. depending on a support (MST) set by the user and Data owners, frequent item sets are given through consecutive scans of database;
II. Strong association rules are extracted from the frequent item sets and limited by a minimum confidence (MCT) also set by user and data owners[5],[8].

3.3 Side Effects

The data loss (undesirable side effects) is defined,
This results after the hiding process, by using four statements below:

1. If a rule R before the hiding process has $conf\ (R) > MCT$ and after the sanitized process has $conf\ (R) < MCT$ then this rule has been lost and hidden.

2. If a rule R before the hiding process has $conf\ (R) < MCT$ and after the sanitized process has $conf\ (R) < MCT$
Then this rule has been created and discovered(Ghost rule).
Clearly, one of the aims for an association rule hiding

Technique would be the limitation of lost rules (among the non-sensitive ones) and ghost rules, as far as possible [4], [6], [3]

Figure 1. Indicates flowchart of process of the proposed algorithm.

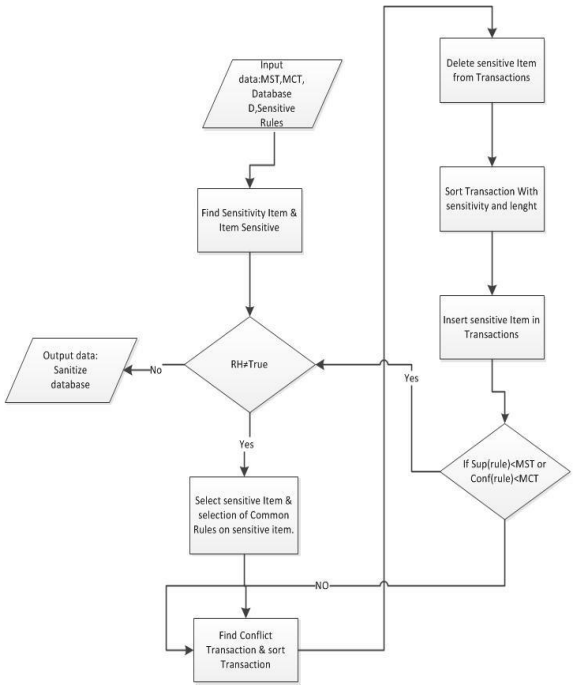

Figure 1. Flowchart of proposed algorithm

3.4 Proposed Algorithm

The proposed algorithm is a heuristic technique and trying to hide sensitive rules based approach to confidence minimum adverse effects, including the failure to hide the rules is lost. LHS support to enhance the sensitive rules based approach to confidence they are trying to hide. The proposed algorithm is based on the rules of their confidence in ascending order and then selects the rule and then selects the items on the left. If the selected item to the left is another rule, that rule can select the degree of to hide to act according to the rules of the selected transactions to hide calculated. Sensitive rule is hidden until remove selected items from the transaction and Insertion it in another transaction. Remove items from the transaction before the transaction based on the degree of sensitivity to length and Insertion them regularly and before transactions are arranged according to the length sensitivity. Sort of action will have the lowest missing rule.

The conflict degree is the transaction, the number of rules that will be fully involved in the transaction.

After each change, confidence and support the updated MST and MCT if reduces below threshold, then sensitive rule is hiding, and the situation is True. To keep situation sensitive rules, control will be failure in to hide. In this case, a sensitive rule is hidden when its status is True, but the rule to hide again the rule is extracted its status, will be false. Since the condition of to hide, is true the situation all the rules, rules again extracted to be chosen again for to hide.

So the algorithm has five basic steps are:
1 calculated the Sensitivity per item
2- Calculate the degree of conflict
3-sorting
4- Remove the item (s) on the left side of
5- Insert the item (s) in the transaction (s) selected

The proposed algorithm has the following steps:
Input: transactions T∈D, non-sensitive rules,
Rules to hide set RH,
Threshold MCT, MST
Output: modified database DM
Step1.for each R*i*∈RH:

1. Find sensitivity of each item ∈ RH set IS
2. Find conflict T∈D set TS
3. Sort RH by ascending order of their confidence
4. Hiding
5. While all the sensitive rules are not hidden

5.1 Select LHS Item RH[i]
5.2 If victim item there are other LHS on the Rules, then
5.2.1 Add Index other Rule in CR
5.3 Find conflict T∈D set TS
5.3.1While RH[i] is not true,
5.3.2 Sort TS by conflict decreasing, Length ascending, Sensitivity ascending
5.3.3 Remove victim item from first transaction in TS
5.3.4 Sort TS by Sensitivity ascending, Length ascending
5.3.5 Insert victim item from first transaction in TS
5.3.6 Start Update support& confidence

4. Performance Evaluation

We have performed extensive experiments in order to Compare the effectiveness of the algorithm presented in Above. We run this algorithm in windows 7 operating Systems at 2.10 GHz with 6 GB RAM. We used three

Datasets that these datasets are available through FIMI15 And their properties are summarized in Table 1. And also Table 2 present the result of mining of these databases. We will compare the proposed algorithm with published algorithm for rule hiding that we also implemented. The algorithm is called RRLR [9].

In order to, Experiments were carried out on these algorithms can be divided into the following in general categories and results obtained from each one separately investigated:

Experiments conducted on the proposed algorithm and the base algorithm in the form of a table RRLR completely prepared. The failure of the proposed algorithm in comparison with algorithms RRLR rate of lost rules for all three dataset chess, mushroom and synthetic is better.

To compare and evaluate the proposed method and algorithm testing RRLR the MST, MCT dataset listed differently on purpose. It is defined as the fraction of the sensitive association rules that appear in the sanitized database divided by the ones that appeared in the original dataset.

Table 1. Properties of Datasets

Dataset	Number of transaction	Number of item	Avg. Items.
Chess	3196	75	37
mushroom	8124	119	23
synthetic	100	151	49

Table 2. Result of mining on datasets

Dataset	Association rules before Hiding process.	MST	MCT
Chess	320	88	90
Chess	62	89	90
Chess	860	86	80
Mushroom	69	80	85
Mushroom	714	50	70
Synthetic	1918	35	60
Synthetic	32	45	60
Dataset	254	40	70

The following tests were carried out on the chess database with: MST=89, MCT=90

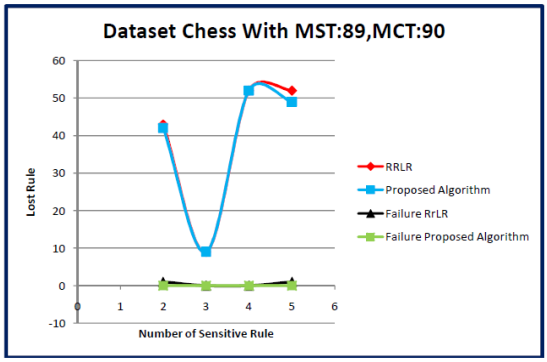

Figure 2. Rules lost after the hiding process.

The following tests were carried out on the mushroom database with: MST=80, MCT=85

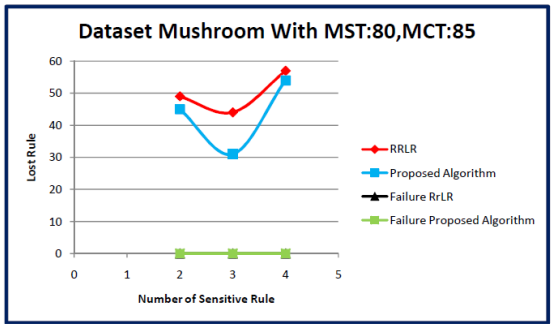

Figure 3. Rules lost after the hiding process.

The following tests were carried out on the synthetic database with: MST=80, MCT=85

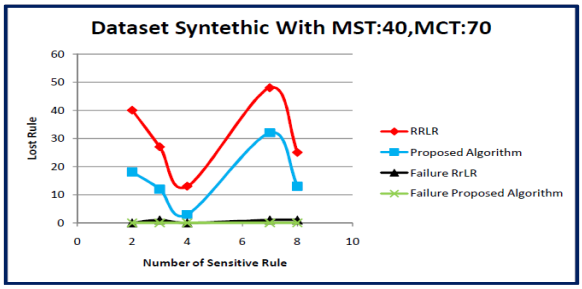

Figure 4. Rules lost and failure hiding after the hiding process.

The following tests were carried out on the mushroom database with: MST=80, MCT=85

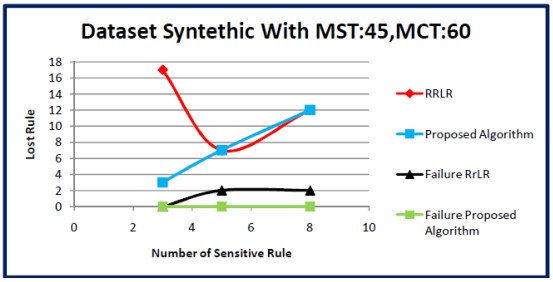

Figure 5. Rules lost and failure hiding after the hiding process.

According to experiments performed on the proposed algorithm compared to the base algorithm, the failure hiding is zero and Lost rules than RRLR algorithm in the worst case is equal, and the best case is dropped.

The first category includes tests to hide the 1, 2, 3, 4, 5 sensitive association rule and 3 different value for MCT and MST on dense dataset (Chess) and hide failure (HF), this measure quantifies the percentage of the sensitive patterns that remain disclosed in the sanitized dataset. It is defined as the fraction of the sensitive association rules that appear in the sanitized database divided by the ones that appeared in the original dataset. Formally,

$$HF = \frac{|R_{P(D')}|}{|R_{P(D)}|} \quad (1)$$

where, R_P (D') equals to the sensitive rules disclosed in the sanitized dataset D'. R_P (D) to the sensitive rules appearing in the original dataset D and |X| is the size of set X. Ideally, the hiding failure should be 0%[3].
As, Figures 2 show result of experiments of these algorithms. These figures indicate that proposed algorithm don't have hiding failure.

The second category includes tests to hide the 2, 3, 4, 5, 6, 8 sensitive association rule and 3 different value for MCT and MST on dense dataset sparse dataset (Mushrooms) with evaluation criteria: misses cost (MC), this measure quantifies the percentage of the non sensitive patterns that are hidden as a side-effect of the sanitization process. It is computed as follows:

$$MC = \frac{|\tilde{R}_P(D)| - |\tilde{R}_P(D')|}{|\tilde{R}_P(D)|} \quad (2)$$

where, R~ P(D) corresponds the set of all non-sensitive rules in the original database D and R~ P(D') is the set of all non-sensitive rules in the sanitized database D' [10].As one can notice, there exists a agreement between the misses cost and the hiding failure, since the more sensitive association rules one needs to hide, the more Association rules are expected to miss [3]. In Figures 3, we see, the proposed algorithm performs Better than algorithm RRLR.
The third category includes tests to hide from 1 to 8 sensitive association rule and 8 different value for MCT and MST on synthetic dataset with evaluation criteria: Artifact Patterns (AP), this measure quantifies the Percentage of the discovered patterns that are artifacts.
It is computed as follows:

$$AP = \frac{|P'| - |P \cap P'|}{|P'|} \quad (3)$$

where, P is the set of association rules exposed in the original database D and P ´ is the set of association rules exposed in D ´ [3],[9].
Figures 4 present the number of ghost rules that are created after hiding process. These figures show that algorithm RRLR extracted more ghost rules. The proposed algorithm performs slightly better than algorithm RRLR. Of course, all of the factors presented in the database have been evaluated. Results of tests are presented in Table 3, 4, 5.

Table 3. Implementation results of the based algorithm and the proposed algorithm on chess database

Number Test	Dataset name: chess	MST:88 , MCT=90	Sensitive Rule
1	RRLR	Lost: 281 Ghost: 0 Failure: 0	9→40 29→52 60→58
	Proposed Algorithm	Lost:277 Ghost: 0 Failure: 0	
2	RRLR	Lost: 6 Ghost: 0 Failure: 0	40→60 60→40,9
	Proposed Algorithm	Lost:6 Ghost: 0 Failure: 0	
3	RRLR	Lost: 127 Ghost: 0 Failure: 0	40→52,9
	Proposed Algorithm	Lost: 127 Ghost: 0 Failure: 0	
4	RRLR	Lost: 43 Ghost: 0 Failure: 0	52→29,58,60,9
	Proposed Algorithm	Lost: 43 Ghost: 0 Failure: 0	
5	RRLR	Lost: 262 Ghost: 0 Failure: 0	58→9 58→60 60→29,9 40→29,58
	Proposed Algorithm	Lost: 254 Ghost: 0 Failure: 0	
6	RRLR	Lost: 278 Ghost: 0 **Failure:1**	29→9 9→40 9→29,40 60→29,9 58→60,9
	Proposed Algorithm	Lost: 243 Ghost: 0 Failure: 0	

	Dataset:Chess	MST=86 , MCT=80	
7	RRLR	Lost: 101 Ghost: 0 Failure: 0	3→7 7→5 7→9 9→7 7→29
	Proposed Algorithm	Lost: 37 Ghost: 0 Failure: 0	
8	RRLR	Lost: 625 Ghost: 0 **Failure:1**	29→40 36→52 52→29,36 40→52,58,60
	Proposed Algorithm	Lost: 504 Ghost: 0 Failure: 0	
9	RRLR	Lost: 650 Ghost: 0 Failure: 0	7→5 52→60 58→56 29→40,52,9
	Proposed Algorithm	Lost: 566 Ghost:2 Failure: 0	
10	RRLR	Lost: 519 Ghost: 0 Failure: 0	7→9 58→7 40→29
	Proposed Algorithm	Lost: 475 Ghost:2 Failure: 0	
	Dataset: Chess	MST=89 , MCT=90	
11	RRLR	Lost: 43 Ghost: 0 **Failure:1**	29→9 9→40
	Proposed Algorithm	Lost: 42 Ghost:8 Failure: 0	
12	RRLR	Lost: 52 Ghost: 0 Failure: 0	5→7 9→29 9→58 52→58
	Proposed Algorithm	Lost: 52 Ghost: 0 Failure: 0	
13	RRLR	Lost: 9 Ghost: 0 Failure: 0	5→7 7→5 40→9
	Proposed Algorithm	Lost: 9 Ghost: 0 Failure: 0	
14	RRLR	Lost: 52 Ghost: 0 **Failure:1**	9→52 58→9 58→29 58→40 9→29,52
	Proposed Algorithm	Lost: 49 Ghost: 0 Failure: 0	

Table 4. Implementation results of the based algorithm and the proposed algorithm on mushroom database

	Dataset Mushroom	MST=80, MCT=85	
1	RRLR	Lost: 53 Ghost: 0 Failure: 0	35→33 33→80 35→84
	Proposed Algorithm	Lost: 53 Ghost: 0 Failure: 0	
2	RRLR	Lost:57 Ghost: 0 Failure: 0	33→80 80→33 87→84 80→33,84
	Proposed Algorithm	Lost: 54 Ghost: 0 Failure: 0	
3	RRLR	Lost: 44 Ghost: 0 Failure: 0	35→80,84 80→84,87
	Proposed Algorithm	Lost: 31 Ghost:3 Failure: 0	87→33,80,84
4	RRLR	Lost: 49 Ghost: 0 Failure: 0	33→84 35→33,84
	Proposed Algorithm	Lost: 45 Ghost:3 Failure: 0	
	Dataset Mushroom	MST=50, MCT=70	
5	RRLR	Lost: 535 Ghost:19 Failure: 0	56→33 20→33 38→35 0→80 20→87 33→35,84
	Proposed Algorithm	Lost: 498 Ghost: 0 Failure: 0	
6	RRLR	Lost: 16 Ghost: 0 Failure: 0	69→33 69→80 69→84
	Proposed Algorithm	Lost: 16 Ghost: 0 Failure: 0	
7	RRLR	Lost: 458 Ghost:11 Failure: 0	80→35,87
	Proposed Algorithm	Lost: 458 Ghost:18 Failure: 0	
8	RRLR	Lost: 170 Ghost: 0 Failure: 0	0→87 20→33 78→33 60→80
	Proposed Algorithm	Lost: 170 Ghost: 0 Failure: 0	

Table 5. Implementation results of the based algorithm and the proposed algorithm on synthetic database

	Dataset Synthetic	MST:35, MCT:60	
1	RRLR	Lost: 202 Ghost:1 Failure: 0	21→36,37 37→21,36 11→21,37 6→10,36 13→12,6
	Proposed Algorithm	Lost: 82 Ghost:237 Failure: 0	
2	RRLR	Lost: 575 Ghost:15 Failure: 0	1→3 7→1 9→1 34→3 17→7 37→9 9→6 15→9
	Proposed Algorithm	Lost: 120 Ghost:1277 Failure: 0	
3	RRLR	Lost: 142 Ghost: 0 Failure: 0	23→21,36 23→11,21 21→11,17,34
	Proposed Algorithm	Lost: 73 Ghost:43 Failure: 0	
	Dataset Synthetic	MST:45 , MCT:60	
4	RRLR	Lost: 12 Ghost: 0 **Failure:2**	17→34 36→17 28→34 34→32 34→6 13→34 36→34 11→34
	Proposed Algorithm	Lost: 12 Ghost:20 Failure: 0	
5	RRLR	Lost: 17 Ghost: 0 Failure: 0	17→34 17→36 34→32

	Proposed Algorithm	Lost: 3 Ghost:4 Failure: 0	
6	RRLR	Lost: 7 Ghost: 0 **Failure:2**	17→34 36→34 34→11 20→15 21→11
	Proposed Algorithm	Lost: 7 Ghost:18 Failure: 0	
	Dataset Synthetic	MST:40, MCT: 70	
7	RRLR	Lost: 25 Ghost: 0 Failure: 0	7→22 7→34 6→2 2→34
	Proposed Algorithm	Lost: 24 Ghost:25 Failure: 0	
8	RRLR	Lost: 48 Ghost: 3 **Failure: 1**	22→34 15→8 13→12 13→36 21→13 20→21 2→11
	Proposed Algorithm	Lost: 32 Ghost:61 Failure: 0	
9	RRLR	Lost: 27 Ghost: 0 **Failure:1**	12→20 12→21 21→11,34
	Proposed Algorithm	Lost: 12 Ghost:11 Failure: 0	
10	RRLR	Lost: 25 Ghost: 0 **Failure:1**	1→32 13→1 9→8 14→17 17→37 18→34 20→32 20→15
	Proposed Algorithm	Lost: 13 Ghost:57 Failure: 0	
11	RRLR	Lost: 13 Ghost:2 Failure: 0	1—32 15→7 9→8 28→17
	Proposed Algorithm	Lost: 3 Ghost:36 Failure: 0	
12	RRLR	Lost: 40 Ghost:1 Failure: 0	21→36 11→21
	Proposed Algorithm	Lost: 18 Ghost:15 Failure: 0	

5. Conclusion and future work

Association rule hiding methods can be very helpful when databases must be shared without the revealing of sensitive information. Accordingly, we had tried to present the algorithm that after the sensitive association rules have been removed, the database can still be mined for extraction of useful information. This algorithm with elimination selective item among items of left hand side of sensitive rules for each transaction that fully support sensitive ruled and sorted these transactions according to Sensitive them, cause to reduce confidence of sensitive rules below minimum threshold to hide sensitive rule with the least possible side effects each time. Finally, this algorithm was compared with algorithm RRLR by Evaluation criterions: hiding failure (HF), misses cost (MC), art factual patterns (AP). The results obtained indicated that proposed algorithm is better than the other algorithms. As future work, The proposed algorithm can be used to improve the time to sort of insert and delete items from the transaction, eliminated and only once do the sorting operation Can also determine the number of changes required To delete a rule, delete and insert operations to needed at once did.

References

[1] I. N. F. E. Bertino, L.P. Provenza, "A Framework for Evaluating Privacy Preserving Data Mining Algorithms," pp.121-154, 2005.

[2] C. W. C. a. D. Marks, "Security and privacy implications of data mining," *In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pp. 15–19, Feb. 1996 1996.

[3] V. S. V. a. A. Gkoulalas-Divani, "A Survey of Association Rule Hiding Methods for Privacy," vol. 34, springer 2010.

[4] M. Kantardzic, "Data Mining: Concepts, Models, Methods, and Algorithms," *Wiley-IEEE Press*, 2011.

[5] A. A. T. E. D. Pontikakis, and V. S. Verykios, "An experimental study of distortion–based techniques for association rule hiding," *In Proceedings of the 18th Conference on Database Security (DBSEC)*, pp. 325–339, 2004.

[6] E. B. M. Atallah, A. Elmagarmid, M. Ibrahim, V. S. Verykios, "Disclosure limitation of sensitive rules," *In Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, pp. 45–52, 1999.

[7] E. B. V. S. Verykios, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State–of–the–art in privacy preserving data mining," *ACM SIGMOD Record*, vol.pp. 50-57, 2004.

[8] V. S. V. Y. Saygin, and C. W. Clifton, "Using unknowns to prevent discovery of association rules," ibid.pp. 45–54, 2001.

[9] A. T. K. Shah, 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, May 2012.

[10] H. W. S. Wu1, "Research On The Privacy Preserving Algorithm Of Association Rule Mining In Centralized Database," *Proc. of the International Symposiums on Information*, 2008.

Zahra Sheykhinezhad Master student at the Computer Engineering- Najafabad branch Islamic Azad University.

Mohammad Naderi Dehkordi Degree: ph.D- field: Software Engineering- University: SRBIAU- Year:2009
Experiences Title: Vice-Chairman for Education- Organization: Computer Engineering Department-IAUN from 2009 to 2014 Assistant Professor Faculty of Computer Engineering- Najafabad branch Islamic Azad University.

Hamid Rastegari degree: ph.D- field: Computer Science - Soft Computing- University: UTM- country: Malaysia- year: 2011
Experiences **1-**Title: Head of Computer Department- Organization: University of Applied Science and Technology from 2002 to 2007
2-Title: Head of Postgraduate Department- Organization: IAUN from 2013 to Pres.
3-Title: Coordinator of 1st National Conference on Computer Science ament- Organization: IAUN from 2013 to 2013.
Assistant Professor Faculty of Computer Engineering- Najafabad branch Islamic Azad University.