International Journal of Computing and Network Technology
ISSN 2210-1519
Int. J. Com. Net. Tech. 3, No. 3 (Sept. 2015)

# An Enhanced Algorithm for Hiding Sensitive Association Rules Based on ISL and DSR Algorithms

**Alaa K. Jumaah[1], Sufyan Al-Janabi[2] and Nazar Abedlqader Ali[3]**

[1]*Technical College of Informatics, Sulaimani Polytechnic University, Kurdistan Region of Iraq, Iraq*
[2]*College of Science and Technology, University of Human Developmen , Kurdistan Region of Iraq, Iraq*
[3]*College of Administration, University of Sulaimani, Kurdistan Region of Iraq, Iraq*

*Received: 18 May 2015, Revised: 10 August 2015, Accepted: 20 August 2015, Published: 1 (September) 2015*

**Abstract:** Many privacy preserving data mining algorithms attempt to hide what database owners consider as sensitive. Specifically, in the association rules domain, many of these algorithms are based on item restriction methods; that is, removing items from some transactions in order to hide sensitive frequent item sets. There are two known algorithms for that purpose, ISL (Increase Support of Left) and DSR (Decrease Support of Right). Both of them make use of user specified values for minimum support threshold and minimum confidence threshold as input. Since ISL and DSR techniques aim at hiding all sensitive rules, they cannot avoid the undesired side effects. In this paper a new algorithm for hiding sensitive rules is proposed based on ISL and DSR. It depends on decreasing the confidence of the sensitive rule by dealing with the both Left Hand Side (LHS) and Right Hand Side (RHS) according to the ratio between them. This technique introduces introduce a reasonable side effect (lost and new rules) as compared with ISL and DSR algorithms which introduces a high number of new rules and/or lost rules. Experimental results have shown that the proposed algorithm effectively reduces the side effects which occur due to the hide operation when it compared with the ISL and DSR algorithms and that it gives a good hiding ratio for the sensitive rules.

**Keywords:** privacy preserving data mining; association rule; minimum support threshold; minimum confidence threshold; Left Hand Side; Right Hand Side.

## 1. INTRODUCTION

Sensitive rule hiding is a subfield of Privacy Preserving Data Mining (PPDM) which can be divided into two categories. One is the preserving of data privacy, which considers all or parts of the data to be sensitive. Its goal is to blur the sensitive data but keep the summary information unchanged. The other is the preserving of information privacy, assuming that only the summary information is sensitive. Its goal is to hide the sensitive information but retain most of the original data. Sensitive rule hiding belongs to the second category. The various approaches proposed by researchers hide sensitive information efficiently and accurately but also face the problem of side effects. The side effects occur due to correlation which exists between items in the database. The side effects may decrease the informational accuracy to the users because the property of correlation in association rules may possess spurious or wrong information. Some other side effects may appear such as hiding non-sensitive rules unnecessarily and accidentally disclosing some sensitive rules. So the challenging task is how to protect sensitive rules from users without affecting informational accuracy of the users. Thus, side effects have to be avoided as far as possible [1].

Also, a number of techniques like perturbation and anonymization have been developed to hide association rules from being discovered from published data. In practically for a single data set, given specific rules or patterns to be hidden, many data altering techniques for hiding association rules have been proposed. They can be categorized into three basic approaches. The first approach hides one rule at a time. It first selects transactions that contain the items in a given rule. It then tries to modify items, transaction by transaction, until the confidence or support of the rule falls below minimum confidence or minimum support. The modification is done by either removing items from the transaction or inserting new items to the transactions. The second approach deals with groups of restricted patterns or sensitive association rules at a time. It first selects the transactions that contain the intersecting patterns of a group of restricted patterns. Depending on the disclosure threshold given by users, it sanitizes a percentage of the selected transactions in order to hide the restricted patterns. The third approach deals with hiding certain constrained classes of association rules [2].

Recent development in PPDM has proposed many efficient and practical techniques for hiding sensitive patterns or information from been discovered by data mining algorithms.

*E-mail address: alaa_alhadithy@yahoo.com, sufyan.aljanabi@uhd.edu.iq,* nzar@mail.com

E. Dasseni et al. generalized the problem in the sense that they considered the hiding sensitive rules. The proposed three single rule heuristic hiding approaches which are based on the reduction of either the support or the confidence of the sensitive rules. In all the three approaches, the goal was to hide the sensitive rules while minimally affecting the support of the non-sensitive rules. Moreover, since this work aimed at hiding all the sensitive knowledge appearing in the dataset, it failed to avoid undesired side-effects such as lost and false rules [3]. V. S. Verykios et al. extended the work of Dasseni et al. They presented two fundamental approaches in order to protect sensitive rules from disclosure. The first approach prevented rules from being generated by hiding the frequent sets from which they are derived. The second approach reduced the importance of the rules by setting their confidence below a user-specified threshold. They developed five algorithms that hide sensitive association rules based on these two approaches but they generated high side effects and required multiple database scans [4]. S. Wang et al. proposed two algorithms, ISL (Increase Support of Left Hand Side (LHS)) and DSR (Decrease Support of Right Hand Side (RHS)), to automatically hide informative association rule sets without pre-mining and selecting of hidden rules. The first algorithm tries to increase the support of left hand side of the rule (modifies transaction that partially supports the sensitive rules) until the support or confidence for this rule becomes less than minimum support threshold and or minimum confidence threshold. The second algorithm tries to decrease the support of the right hand side of the rule (modifies transaction that fully support sensitive rule) until the support or confidence for this rule becomes less than minimum support threshold and or minimum confidence threshold. Both algorithms exhibit side effects like hide failure, loss rules, and appearance of new rule [5]. I. Chandrakar et al. proposed an algorithm to hide a sensitive rule. It can hide rules according to the location of the sensitive item that supports these rules. If the sensitive item appears in LHS of rule, the ISL algorithm is used to hide this rule, and if the sensitive item appears in RHS of rule, DSR algorithm is used to hide this rule. This proposed algorithm prunes more sensitive rules compared to ISL algorithms [6]. G. Deepti et al. proposed an approach to modify ISL and DSR algorithms. The modification depended on the transactions that support the sensitive rules. This can be done by evaluating the priority for each transaction according to a number of rules which can be supported by this transaction. Now if ISL algorithm is used to hide sensitive rules, the extracted transaction will be sorted in ascending order according to their priority, and if DSR algorithm is used to hide sensitive rules, the extracted transaction will be sorted in descending order according to their priority. This approach supports the output for hiding sensitive rule with limited side effects [7]. Finally, W. T. Chembian at el. proposed Weight Based Sorting Distortion (WBSD) algorithm. It distorts certain data which satisfies a particular sensitive rule, then hides those transactions which support a sensitive rule and assigns them a priority and sorts them in ascending order according to the priority value of each rule. This method reduces loss of data and minimizes the undesirable side effects, but it is still complex and needs more time because it performs large number of scan operation [8].

## 2. PROBLEM DESCRIPTION

For the association rule mining, Let I= {i_1, 〖 i〗 _2,….i_m} be a set of literals, called items. Given a set of transactions D, where each transaction T in D is a set of items such that $T \subseteq I$ , an association rule is an expression $X \Rightarrow Y$ where $X \subseteq I$ ,$Y \subseteq I$ ,and $X \cap Y = \emptyset$ . Strength of a rule whether it is strong or not is measured by two parameters called support and confidence of the rule. These two parameters help in deciding the interestingness of a rule.

For a given rule $X \Rightarrow Y$

Support is the percentage of transaction that contains both X and Y ($X \cup Y$) or is the proportion of transactions jointly covered by the LHS and RHS and is calculated as:

$$\text{Support} = (X \cup Y)/N$$

Where, N is the number of transactions.

Confidence is the percentage for a transaction that contains also contains or is the proportion of transactions covered by the LHS that are also covered by the RHS and is calculated as [9]:-

$$\text{Confidence} = (X \cup Y)/|X|$$

As an example, for a given database in Table (1), for a minimum support of 33% and a minimum confidence of 70%, nine association rules can be found as follows:

$Y \Rightarrow X$ (66%, 100%), $Z \Rightarrow X$ (66%, 100%),

$Y \Rightarrow Z$ (50%, 75%), $Z \Rightarrow Y$ (50%, 75%),

$XY \Rightarrow Z$ (50%, 75%), $AZ \Rightarrow Y$ (50%, 75%),

$YZ \Rightarrow X$ (50%, 100%), $Z \Rightarrow XY$ (50%, 75%),

$Y \Rightarrow XZ$ (50%, 75%),

Where, the percentages inside the parentheses are supports and confidences respectively.

**TABLE I. DATA SET EXAMPLE [9]**

| TID | Items |
|---|---|
| T1 | XYZ |
| T2 | XYZ |
| T3 | XYZ |
| T4 | XY |
| T5 | X |
| T6 | XZ |

The problem here is to hide sensitive rules (sensitive rules are those rules that contain sensitive item) and minimizes the loss items. This can be done by modifying the data base transaction, so that, the confidence of the rules can be reduced and become less than minimum confidence threshold [9].

## 3. PROPOSED HIDING ALGORITHM

The proposed algorithm for hiding sensitive rules is based on algorithms ISL and DSR. It depends on decreasing the confidence of the rule. Algorithm ISL hides the sensitive rules by increasing the support of the rule in LHS until the rule confidence decreases below the Min_Conf threshold. Algorithm DSR hides sensitive rules by reducing the support of each selected rule and the reduction is done by decreasing the frequency of the RHS through transactions that support the rule until the rule confidence is decreased below the Min_Conf threshold. The proposed algorithm hides sensitive rules by dealing with two approaches: increasing the support of the rule in LHS and decreasing the frequency of the RHS. This can be done according to ratio between the frequent of LHS and the frequent of all rule's items. This technique will introduce a reasonable side effect (lost and new rules) rather than algorithms ISL and DSR which introduce a high number of new rules and lost rules. However, the sensitive rules should contain the sensitive item in LHS. The proposed algorithm also calculates how many transactions (support each sensitive rule) need to be modified. Not only the item that has maximum support is removed from these transactions, but the numbers of modifications are distributed for all items in frequent itemset according to the support ratio for each one in database. This will reduce the side effect of the original database because it will be increased the number of modified transactions with the low weight (transactions that contain minimum number of items).

The proposed hiding algorithm may not hide all the sensitive rules because there is a large number of sensitive rules in this case and a large number of transactions need to be modified. When the proposed algorithm tries to hide non-hidden rules again, it needs more time; therefore the proposed algorithm can be programmed to ask the user about the number of iteration needed to be used for hiding rules. Increased number of iterations will increase the accuracy of hiding rules until all the rules will be hidden, but it needs more execution time.

Also the proposed hiding algorithm supports new techniques for hiding rules. It allows inputting more than one sensitive item and hiding all the rules that can be supported by these items together. This means that if the input is two items like "12" and "15", the proposed algorithm will hide all rules generated by these two items together like "12 15 ⇒3" or "2 12 15 ⇒10".

**The following steps are required in the proposed hiding algorithm:**

1. *Input sensitive item x.*
2. *Generate all rules that contain x in LHS*
3. *For each sensitive rule do {*
4. *Extract all transactions that fully support sensitive rule ($T_r$).*
5. *Extract all transactions that partially support sensitive rule ($T_l$).*
6. *If $\frac{|T_r|}{|T_l|} <$ min_conf, then go to 26 (end loop).*
7. *Evaluate the number of transaction ($|T_{Mr}|$) needed to be modified only with RHS by*
   $$|T_{Mr}| = |T_r| - (min\_conf * |T_l| \quad (1)$$
8. *Evaluate the number of transaction ($|T_{Ml}|$) needed to be modified only in LHS by*
   $$|T_{Ml}| = \frac{|T_r|}{min\_conf} - |T_l \quad (2)$$
9. *Evaluate the ratio for RHS ($R_r$) by*
   $$R_r = \frac{|T_l|}{(|T_r|+|T_l|)} \quad (3)$$
10. *Evaluate the ratio for LHS ($R_l$) by*
    $$R_l = \frac{|T_r|}{(|T_r|+|T_l|)} \quad (4)$$
11. *Evaluate the number of transaction ($|T_{Mr}|$) needed to be modified in RHS according to the ratio by*
    $$|T_{MR}| = |T_{Mr}| * R_r \quad (5)$$
12. *Evaluate the number of transaction ($|T_{Ml}|$) needed to be modified in LHS according to the ratio by*
    $$|T_{ML}| = |T_{Ml}| * R_l \quad (6)$$

*// **Add items to LHS***

13. *For each item i in LHS {*
14. *Count the number of the transactions that support rule's LHS (maximum items), but not support both item i and rule's RHS ($ID_{Ni}$). where*

    *$|ID_{Ni}|$ = support for LHS without both item i and RHS items in DB,*

    *$|ID_{tNi}|$ = summation for support items;*

    *where $|ID_{tNi}| = \sum_{i=1}^{n} ID_{Ni}$*
15. *If $|T_{ML}| > |ID_{tNi}|$( That mean there are no enough transactions can hide sensitive rule) Then*

    $|T_{ML}| = |ID_{tNi}|$ (7)

    $|T_{MR}| = |T_r| - (min\_conf * (|T_l| + |ID_{tNi}|))$. (8)
16. *Evaluate the number to be removed from each item ($Iri$) in rule's antecedent by*

    $|Iri| = |ID_{Ni}| / |ID_{tNi}| * |T_{ML}|$ (9)

    *Where |Iri| = number of item i needed to remove,*

    *(Note: - If there are just one item in LHS*

    $|Ir| = |T_{ML}|$
17. *Extract transaction ($T_{lNi}$) that support rule's LHS and not support item i and RHS items.*
18. *Sort these transactions in ascending order to minimize the impact in database.*
19. *Add item i to these transaction by setting the value for this item to "1" instead of "0".*
20. *} end of add loop*

*// **Remove items from RHS***

21. *For each item i rule's consequent {*
22. *Evaluate the support of each item in rule's consequent need to be removed from ($T_r$) by using*

    $|Iri| = |IDi| / |ID_{tr}| * |T_{MR}|$ (10)

    *Where*

    *|Iri| = number of item i needed to be removed,*

    *|IDi| = count for item i in database,*

    *$|ID_{tr}|$ = summation for all items count in database;*

    *Where $|ID_{tr}| = \sum_{i=1}^{n} ID_i$*

    *(Note: - If there is just one item in RHS, then*

    $|Ir| = |T_{MR}|$ )
23. *Sort ($T_r$) in ascending order to minimize the impact in database.*
24. *Sort RHS items in descending order according to the |Ir|. This also will minimize the side effects that can happen when modifying the database.*
25. *Remove items from ($T_r$) according to the above sorting process by setting the value of this item to "0" instead of "1".*
26. *} end of remove loop*
27. *} // end hiding rule*
28. *If all rules are hidden then go to 30*
29. *Else go to 3*
30. *END*

The pseudo code for the proposed algorithm is shown in Figure 1

**The Proposed Hiding Algorithm**

*Input: a source database D, min_support, min_confidence, set of sensitive items X, and number of iteration*

*Output: a transformed database D', where rules containing X on LHS will be hidden.*

*For each iteration*

*{1. For each item in x ∈ X*

*{2. Generate all rules that contain x in LHS*

*3. For each rule r do*

*{ 4. Extract $T_r$= {t ∈ D / t fully support r }*

*5. Extract $T_l$ = {t ∈ D / t partially support r}*

*6. If $\frac{|T_r|}{|T_l|} <$ min_conf , then go to 26 ( end loop).*

*7. Calculate $|T_{Mr}| = |T_r| - (min\_conf * |T_l|)$. // RHS.*

*8. Calculate $|T_{Ml}| = \frac{|T_r|}{min\_conf} - |T_l|$. // LHS*

*9. Calculate $R_r = \frac{|T_l|}{(|T_r|+|T_l|)}$*

*10. Calculate $R_l = \frac{|T_r|}{(|T_r|+|T_l|)}$*

*11. Calculate $|T_{MR}| = |T_{Mr}| * R_r$*

*12. Calculate $|T_{ML}| = |T_{Ml}| * R_l$*

*// **Add items to (LHS)***

*13. For each item i in **LHS** {*

*14. Count $|ID_{Ni}|$ // support for LHS without item i and RHS items in DB*

*15. $|ID_{tNi}| = \sum_{i=1}^{n} ID_{Ni}$ // summation for support items;*

*16. If $|T_{ML}| > |ID_{tNi}|$ // no enough transactions can hide sensitive rule)*

*{ $|T_{ML}| = |ID_{tNi}|$*

*$|T_{MR}| = |T_r| - (min\_conf * (|T_l| + |ID_{tNi}|))$. }*

*17. Calculate $|Iri| = |ID_{Ni}| / |ID_{tNi}| * |T_{ML}|$ // number of item i needed to remove,*

*18. Extract ($T_{lNi}$) {t ∈ D / t partially support r and not support i}.*

*19. Sort ($T_{lNi}$) // in ascending order.*

*20. Set_to_one (t.values_of_items i, $T_{lNi}$)*

*21. } // end for add loop*

*22. } // end of loop x rule*

*23. } // end of iteration*

*// **Remove items from (RHS)***

24. *For each item i in RHS {*
25. *Count |IDi| // support of item i in database,*
26. *Calculate* $|ID_{tr}|=\sum_{i=1}^{n} ID_i$ *// summation for all (RHS) items count in database;*
27. *Sort* ($T_r$) *// in ascending order according to number of items in transaction*
28. *Sort (Ir) // in descending order according to (|Ir|).*
29. *Set_to_zero (t.values_of_items i,* $T_r$)
30. *} // end of remove loop*
31. *} // end hiding rule*

**Figure 1. The pseudo code for the proposed algorithm.**

## 4. RESULTS AND PERFORMANCE EVALUATION

To assess the performance of the proposed hiding algorithm compared with the performance of algorithms ISL and DSR. These three algorithms have been used to hide all sensitive rules that include specific or sensitive item in LHS. For each data set (30000, 60000, and 90000 transactions), all association rules that have minimum support threshold and minimum confidence threshold are generated and stored in an appropriate file. After completion of the hiding process for all the specific rules, the released database is mined and the new association rules are extracted, and then the generated rules are compared with the previous file to evaluate the side effects. The experiments here used minimum support threshold 6% and the range for minimum confidence given is 40-50%. The experimental results are obtained by averaging from 4 independent trials for each size of transaction with different sensitive rules. The following figures explain the average of the experimental results. The experiments have been performed on a notebook with 2G MHz processor and 2 GB memory, under Windows XP operating system. The sequence database generated for the experiments can be generated by using a Sequence Database Generator "SeqDBGen" [10] that works like IBM data generator [11].

Figures 2, 3, and 4 describe the side effects for the proposed algorithm, ISL, and DSR algorithms respectively. For new rules side effect, it can be observed that algorithm ISL has the highest ratio for new rules (about 52%) if compared to proposed and DSR algorithms because it depends on adding new items to the transactions in database. Algorithm DSR has a minimum ratio for new rules (about 1.5%) because it does not add any item to transactions in database but the majority for these new rules contains the sensitive item in LHS. The proposed algorithm has an acceptable ratio for new rules side effect minority for these rules sensitive item in LHS (about 10%). About lost rule side effect, it is observed that algorithm DSR has the highest ratio for lost rules (about 28.5%) if compared to the proposed and ISL algorithms. Algorithm ISL has a minimum ratio for lost rules (about 0.2%) because this algorithm does not remove any items from database transactions. The proposed algorithm also has an acceptable ratio for lost rules (about 10%). About hiding failures side effect, it can be observed that there is a very small ratio with algorithm DSR because all the sensitive rules are related to the specific items; this will reduce the overlapping between these rules and lead to minimizing the ratio for hiding failures. Algorithm LSL fails to hide all sensitive rules because, for each rule it needs to modify a large number of database transactions and it also suffers from problem when no enough transactions are available for hiding process in each rule; therefore, it has a high ratio for hiding failures (about 57%). The proposed hiding algorithm has hiding failures ratio (about 9%) when it used just for one iteration of hiding rules. It reduces failures that can happen in LHS by solving the problem which occurs as a result of insufficient transactions for hiding process. This problem can be solved by converting the reminder transactions from LHS process to RHS process. The proposed algorithm can reduce the hiding failures to less than 9%, and also it can give no hiding failure when it use more than one iteration for hiding process, but it needs more execution time. Figures 5, 6, and 7 present hiding ratios for the Proposed, ISL, and DSR algorithms respectively. This ratio represents the number for sensitive rules to the total number of association rules in database (All algorithms must be use with the same hiding ratio). Figures 8, 9, and 10 describe the time measurement results for the proposed, ISL, and DSR algorithms respectively. Note that algorithm DSR needs less time than ISL and the proposed algorithms.

From the above results, it is concluded that, algorithm ISL has minimum ratio for lost rules side effect, but it has the highest ratio for new rules and hiding failures side effect. It needs more execution time. Algorithm DSR has minimum ratio for new rules side effect, and smallest ratio for hiding failures, but it suffers from highest lost rules side effect. It needs less execution time than other algorithms. The proposed algorithm has an acceptable ratio for new and lost rules side effect; it reduces the ratio for lost rules side effect in DSR algorithm about 60% and reduces the ratio for new rules side effect in ISL about 80%. When the proposed algorithm use one iteration for hiding process, it can hide about 91% from the sensitive rules. If it use more than one iteration, hiding failures will be decreased, but it needs more time. The drawback for this algorithm is that it needs more execution time than the other two algorithms because it tries to minimize the hiding process side effects. It is observed that the side effects and execution time for the algorithms are high because a large number of sensitive rules need to hidden

(about 150 rules) that have a high confidence. This large number of rules with high confidence causes more side effects and takes more execution time.

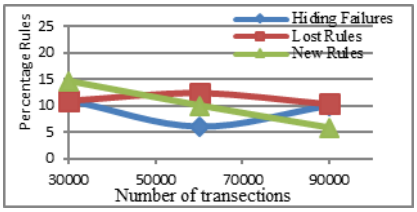


**Figure 2. Side effects for proposed algorithm**

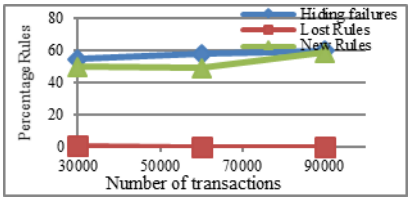


**Figure 3. Side effects for ISL algorithm**

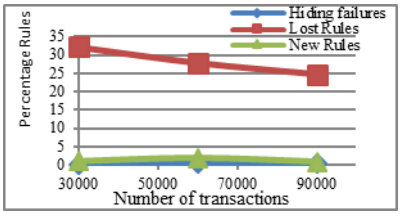


**Figure 4. Side effects for DSR algorithm**

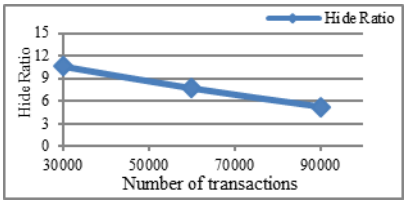


**Figure 5. Hide ratio for proposed algorithm**

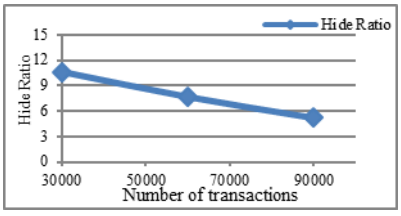


**Figure 6. Hide ratio for ISL algorithm**

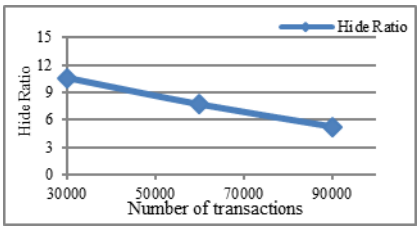


**Figure 7. Hide ratio for DSR algorithm**

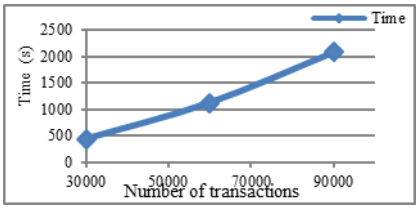


**Figure 8. Required time for proposed algorithm**

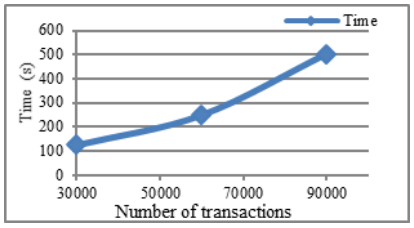


**Figure 9. Required time for ISL algorithm**

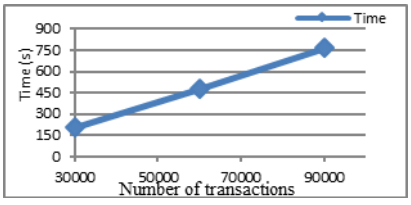


**Figure 10. Required time for DSR algorithm**

## 5. Conclusion and Future Work

The Extraction of knowledge form large amount of data is an important issue in data mining systems. One of most important activities in data mining is association rule mining and the new head for data mining research area is privacy of mining. In this paper, we have proposed a new algorithm for hiding sensitive rules based on ISL and DSR algorithms. The proposed algorithm deals with the LHS and RHS together according to the ratio between them and it selects the transactions with the lowest weight (less impact in database) for modifying original database. During hiding process, it can be observed that side effects (new and lost rules) depend on the nature of sensitive rule; if the rule has a high confidence, the database will have more impact and the side effects will also be increased. In addition, when there is overlapping between the sensitive rules, the side effects will increase too. Also the required time for

hiding process linearly grows with the size of the database and the number of sensitive rules. According to the obtained results, the proposed algorithm reduces the side effects that happen during hiding process in ISL and DSR algorithms and also it hides about 90% for the sensitive rules, but it needs an extra time. In a future work, the proposed algorithm can be developed by identifying some new techniques to build new structures for database transactions. These structures allow reducing the number of database modifications during hiding process, which can reduce the side effects in the database. Further research can also be done to enhance the proposed hiding algorithms so as to reduce the required time for hiding process by supporting techniques for indexing data and using fast sorting algorithms.

## ACKNOWLEDGMENT

This research is supported by University of Sulaimani, College of Science, Computer Science Department and Sulaimani Polytechnic University, Computer Science Institute.

## REFERENCES

[1] N. M. Lakshmi and K S. Rani, "An Improved Algorithm for Hiding Sensitive Rules Using Exact Approach", International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol. 2, No. 1, (2012), pp 97-104.

[2] A. K. Juma'a, S. T. Faraj, N. A. Ali, "Hiding Sensitive Frequent Itemsets over Privacy Preserving Distributed Data Mining", Rafdeen. J. of Comp. & Math's. University of Mosul, Vol. 10, No. 1, (2013), pp. 91-105.

[3] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino, "Hiding Association Rules by Using Confidence and Support", Proceedings of the 4th International Workshop on Information Hiding, , (2001), pp. 369–383.

[4] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association Rule Hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, (2004), pp. 434-447.

[5] S. Wang, B. Parikh, and A. Jafari, "Hiding informative association rule sets", Journal of Expert Systems with Applications, Vol. 33, (2007), pp.316–323.

[6] I. Chandrakar, Y. U. Rani, M. Manasa and K. Renuka, "Hybrid Algorithm for Privacy Preserving Association Rule Mining", Journal of Computer Science,Vol. 6, No. 12, (2010), pp. 1494-1498.

[7] G. Deepti, S S. Sane, and J. Manoj, "Privacy Preserving using Association Rule Mining with Limited Side Effect", International Journal of Computer Application, Vol. 49, No.20, (2012), pp. 975-987.

[8] W. T. Chembian, and J. Jane, "A Study and Enhancement of Association Rule Hiding for Privacy Preserving Data Mining", European Journal of Scientific Research, Vol.77, No.2, (2012), pp. 265-274.

[9] D. Jain, A. Sinhal, N. Gupta, P. Narwariya, D. Saraswat, and A. Pandey, "Hiding Sensitive Association Rules without Altering the Support of Sensitive Items", In Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012, CCSIT 2012, Part I, LNICST 84,(2012) pp. 500–509.

[10] Sequence Database Generator "SeqDBGen", 2013 [Online] Available: http://www.philippe-fournier-viger.com/seqdbgen.

[11] IBM data generator, 2013 [Online] Available: http://www.ibmquestdatagen.sourceforge.net.

## Biographical notes

Dr. Alaa K. Jumaah obtained his BSc (1997) and MSc (2004) degrees in Computer Engineering from the University of Technology, Baghdad, Iraq. He received his PhD in Database Systems from the University of Sulaimani, Kurdistan Region of Iraq, Iraq in 2013. He was a faculty member in Computer Science Institute, Sulaimani Polytechnic University. From 2013 he is the Head of Database Technology Department at the Technical College of Informatics, Sulaimani Polytechnic University. His research interests include Database technology, information security and network security

Prof. Dr. Sufyan Al-Janabi was born in Haditha, Iraq (1971). He obtained his B.Sc. (1992), M.Sc. (1995), and Ph.D. (1999) in Electronic and Communications Engineering from the College of Engineering, Nahrain University in Baghdad. He was a faculty member in Computer Engineering Dept., University of Baghdad in 1999 and the Head of that department in 2001. During May 2004- June 2006 he was the Dean of College of Information Technology, Nahrain University. He served as Dean of the College of Computer, University of Anbar, Ramadi during July 2006- May 2010. He is currently on sabbatical leave to University of Human Development (UHD), Sulaimaniya, KRG-Iraq. His research interest includes internet protocols, information security, and quantum cryptography. Prof. Al-Janabi is the winner of the 1st Award for the Best Research Paper in Information Security from the Association of Arab Universities (AARU), Jordan, 2003.

Dr. Nazar Abedlqader Ali received the BSc and MSc degrees in Statistics from Salahaddin University, Erbil, Iraq in 1986 and 1990 respectively. Then, he received his PhD in Database Systems from the University of Sulaimani, Kurdistan Region of Iraq in 2008. During 2010-2014 he was the Dean of College of Administration, University of Sulaimani. He is now a faculty member at the College of Administration, University of Sulaimani. His research interest includes Database and Data mining technologies.