Privacy Preserving in Association Rule Mining

Zahra Kiani Abari[1] , Mohammad Naderi Dehkordi[2]

[1]Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
Zahrakiani@sco.iaun.ac.ir
[2]Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
Naderi@iaun.ac.ir

Abstract

Association rule mining is one of the most important techniques of data mining that are used to extract the association patterns from large databases. Association rules are one of the most important assets of any organization that can be used for business development and profitability increase. Association rules contain sensitive information that threatens the privacy of its publication and they should be hidden before publishing the database. The aim of hiding association rules is to delete sensitive association rules from the published database so that possible side effects are reduced. In this paper, we present a heuristic algorithm DCR to hide sensitive association rules. In the proposed algorithm, two clustering operations are performed on the sensitive association rules and finally, a bunch of smaller clusters is chosen to hide. A selection of a smaller bunch of clusters reduces the changes in the database and side effects. The results of performing experiments on real databases, shows the impact of the proposed algorithm on missing rules reduction.

Keywords: *Data Mining, Association Rules, Frequent Item-sets, Privacy Preserving Data Mining, Clustering.*

1. Introduction

The vast amount of data produced by organizations; nevertheless, most of these organizations are faced with poverty of knowledge. By using data mining tools, hidden knowledge in the data can be extracted[1]. Nowadays, data mining has wide applications in various fields such as marketing, medical analysis, and business[2]. Extracted data with data mining tools assist individuals and organizations in taking better decisions and improvement of business processes[3]. Association rule mining is one of the most widely used data mining tools which extracts the dependency patterns from large databases extracted. An association rule presents the links between items in the database. Association rule mining consists of two stages: in the first stage, frequent item sets, by using association rule mining algorithms such as Apriori Algorithm [1], are extracted from the large volumes of data, then in the second stage, association rules are extracted from the set of frequent items. Consider $I = \{i_1, i_2, \dots, i_n\}$ as set of items, D as Database of transactions and t as each transaction which $t \subset I$.An association rule will be represented as $X \rightarrow Y$,so that $X \subseteq I$, $\mathrm{Y} \subseteq \mathrm{I}$, and $\mathrm{X} \cap \mathrm{Y} = \varphi$ [1]. For instance, a rule with the support 70% shows that 70% of the customers at a supermarket to buy cheese, will also buy bread. The support of a rule is calculated by using the formula 1:

$$Support(X, Y) = \frac{|X \cup Y|}{|D|} \qquad (1)$$

$|X \cup Y|$ shows the number of transactions that consists X and Y and $|D|$ is the number of transaction in the database. The rule confidence is 100%. That means 70% of transactions includes cheese and bread. Confident measurement is calculated as follows:

$$Confidence(X, Y) \frac{|X \cup Y|}{|X|} \qquad (2)$$

$|X|$ is the number of transactions that consist X. Association rules extracted from a database are divided into two groups of weak and strong association rules[4]. If the confidence of an association rule is below the confidence threshold, it will be called as a weak association rule, whilst the strong association rule confidence is equal or above the confidence threshold which has been defined by the user. The strong association rules will be classified in two categories of sensitive and non-sensitive. Sensitive association rules consist important information and patterns which disclosure of those could jeopardize the owners of information[4]. So, the sensitive association rules should be hidden before sharing them. Hiding sensitive association plays a vital important role in protecting sensitive knowledge in sharing. The aim of hiding association rules is to delete sensitive ones in published database. There are two strategies in hiding sensitive association rules:

- LHS support increase
- RHS support decrease

Association rules hiding algorithms can be divided into three main approaches border-based, exact, and heuristic[5]. In both border base and exact approach, in order to reduce the side effects of hiding process, positive border of frequent items is reformed. Although these two approaches in hiding sensitive association items are effective, in some extend, it does not operate pragmatically in hiding some association rules. Heuristic algorithms to find the optimal solution is not guaranteed, but basically, a close solution to the best solution is presented in the shortest time. Heuristic algorithms use distortion and blocking to hide sensitive information[6]. In distortion, to reduce the support or the confidence of sensitive rules under the threshold, the appropriate items are added, or removed from the appropriate transaction[7]. In blocking technique, some items are replaced by unknown or "?", so the support or confidence of sensitive rules will be decreased under the threshold [8]. In recent years, many heuristic algorithms have been introduced to hide association rules. These algorithms would change the original database to reduce the support and confidence of sensitive rules below the threshold. All these algorithms suffer from side effects such as Hiding failure (lack of success in hiding some sensitive association rules), Misses cost (hiding non-sensitive association rules), and artifactual rules (Generated new association rules that are not supported by the database)[5]. The side effects have an important role in motivating of the proposed algorithm. In this paper, DCR (Dual Clustering Rules), a heuristic algorithm for hiding sensitive association rules, is proposed. DCR clusters sensitive rules based on similarity in the RHS and LHS of the rules and then the smallest cluster will be selected for deleting. So that, based on the smallest cluster, deleting or addition operation will be performed in order to hide sensitive information in databases. Selection of the smallest cluster, reduces the changes in the database and reduce the number of rules may be missing.

This paper follows as: in section 2, related works will be examined. In section 3 proposed algorithm will be introduced. In section 4 the proposed algorithm would be compared to DSRRC [9], ADSRRC [10], and MDSRRC [11]. Above all, the outcomes will be evaluated and the result will be provided in section 5.

2. Related works

Information sharing is often beneficial for database owners, however, in some cases, it may disclose personal information. Privacy preserving techniques in data mining, prevent unauthorized access to information. In this paper our focus is to hide sensitive association rules. In this section, algorithms to hide the association rules that have been introduced in recent years, will be evaluated.

In the year of 2001, Saygm et al, proposed two algorithms to hide sensitive association rules. The first one focuses on hiding the rules by reducing the minimum support of the item-sets that generated these rules. The second one focuses on reducing the minimum confidence of the rules[12].

In the year 2002, Oliveiraet et al, proposed four algorithm called Naïve, MinFIA, MaxFIA and IGA to hide sensitive association rule. Each algorithm selects the sensitive transactions to sanitize based on degree of conflict. Naïve Algorithm removes all items of selected transaction except for the item with the highest frequency in the database. The MinFIA algorithm selects item with the smallest support in the pattern as a victim item and it removes the victim item from the sensitive transactions. Unlike the MinFIA, algorithm MaxFIA selects the item with the maximum support in the restrictive pattern as a victim item. Algorithm IGA groups restricted patterns in groups of patterns sharing the same item- sets so that all sensitive patterns in the group will be hidden in one step[13].

In the year 2004, Verykios et al, presented three algorithms 1.a, 1.b and 2.a for hiding sensitive association rules. Algorithm 1.a hides association rules by increasing the support of the rule's antecedent until the rule confidence decreases below the minimum confidence threshold. Algorithm 1.b hides sensitive rules by decreasing the frequency of the consequent until either the confidence or the support of the rule is below the threshold. Algorithm 2.a decreases the support of the sensitive rules until either their confidence is below the minimum confidence threshold or their support is below the minimum support threshold. In 1.a algorithm large number of new frequent item-sets is introduced and, therefore, an increasing number of new rules are generated. Algorithm 1.b and 2.a affects number of non sensitive rules in database due to removal of items from transaction[14].

In the year of 2005, Wang et al, proposed ISL and DSR algorithm to hide sensitive association rules. ISL with increasing support of rules' LHS, reduces confidence under the threshold, so the sensitive association rules will be hidden. DSR decreases the whole rule's support and confidence below the threshold to hide sensitive association rules. Hiding the sensitive items and the arrangement of database transactions affects the result in both algorithm operations. DSR has no hiding failure; notwithstanding, ISL will fail if there is no suitable transaction to add[15].

In the year of 2007, Wang et al, proposed two algorithms, DCIS (Decrease Confidence by Decrease Support) and DCDS (Decrease Confidence by Decrease Support) to automatically hide collaborative recommendation association rules without pre-mining and selection of hidden rules. The DCIS algorithm try to increase the support of left hand side of the rule and algorithms DCDS try to decrease the support of the right hand side of the rule[16].

In the year of 2008, Weng et al, proposed FHSAR (Fast Hiding Sensitive Association Rules)to hide association rules for fast hiding sensitive association rules. The algorithm can completely hide given sensitive association rule by scanning database only once, which significantly reduced the execution time. In this algorithm correlations between the sensitive association rules and each transaction in the original database are analyzed which can effectively select the proper item to modify[17].

In the year of 2010, Modi et al, introduced DSRRC. In this algorithm sensitive rules are clustered based on similar RHS and then hiding operation will be performed. Hiding Association rules collectively by using clusters instead of single rules reduces both amounts of changes in the database and the side effects. DSRRC algorithm after each change, sorts the database which increases hiding process time. This algorithm depends on the database orientation and the result of the outcome will be vary by any modification in database[9].

In the year of 2010, Kumar Jain et al, proposed a heuristic algorithm for hiding association rules that are based on ISL and DSR. It operates based on both ISL and DSR techniques which not only does increase the LHS support, but also the total support will be decreased. Although, this algorithm has no failure in hiding, the database will be changed a lot due to simultaneous reduction of rules' support and confidence [18].

In the year of 2012, Komal Shah et al, proposed improved algorithms called ADSRRC and RRLR to reform DSRRC limitations. ADSRRC, the same as DSSRC, tries to cluster sensitive rules based on similar RHS. In this algorithm at first the sensitivity of the transactions is calculated, then they will be sorted in descending order. For this reason, arrangements of transactions have no effect on algorithm result. RRLR has been designed to hide various association rules with different RHS. In this algorithm by reducing the confidence of sensitive rules below the threshold, the process of concealment done. Since these two algorithms do two sorting operations, they perform quicker in term of runtime than DSRRC [10].

In the year of 2013, Domadiya et al, proposed MDSRRC to hide association rules. MDSRRC can hide rules with multiple RHS and LHS. At first, sensitivity of items in rules' RHS calculated and then the most sensitive item will be selected to delete. MDSRRC, in comparison with DSRRC, reduces database modification and side effects with deleting the effective candidate item [11].

In the year of 2012, Jain et al introduced an algorithm in which hides sensitive association rules without altering the support of frequent item-sets. In this algorithm has been tried to use a new concept named Representative rule, in which by help of the Representative rule and without any access to the main database, all sensitive rules can be inferred. This algorithm changes the position of items, instead of removing any items in transactions, to hide association rules. So that causes no modification in frequent item-sets' support, size of database, and finally with less change in database it hides the maximum number of sensitive association rules. This is due to the existence of suitable transactions in the database to alter the position of sensitive items; otherwise hiding process will be failed[3].

3. Proposed algorithm

In this paper, we proposed DCR (Dual Clustering Rules) to hide association rules. DCR use clustering to minimize side effects such as hiding failure and misses cost. Clustering sensitive rules and hiding clusters, instead of hiding rules individually, reduces the changes in the database in which it minimizes the side effects. In process of clustering it should be noticed that the sensitive rules structure remarkably influences the number of generated clusters. For instance, consider these sensitive rules as $c \to b$, $d \to a$, $c \to a$ and $b \to a$. If these rules are clustered based on similar RHS, the clusters will be at Table 1. So that, two clusters have been generated that in fact by deleting items of "a" and "b" as enough, these four sensitive rules will be hidden. Now consider $b \to a$, $b \to c$, $b \to d$ and $c \to e$. If they are clustered based on similar RHS, the clusters will be at Table 2. So, four clusters will be generated which it is necessary to delete "a", "c", "d", and "e" from the database as enough in order to hide these sensitive rules; this is clear that clustering based on similar RHS generates four clusters, while clustering based on similar LHS generated only two clusters (Table 3).In the proposed algorithm, two processes of clustering will be done. That means it clusters based on both similar RHS and LHS and then the minimum cluster will be selected. If the numbers of two clusters are equal, for a decrease of misses cost, clusters based on similar LHS will be selected. By performing two clustering processes literally, structural sensitive rules' effects in clustering and hiding operation have been decreased.

Table 1: Clustering based on RHS

Cluster_RHS	***Rules***
a	$b \rightarrow a, c \rightarrow a, d \rightarrow a$
b	$c \rightarrow b$

Table 2: Clustering based on RHS

Cluster_RHS	***Rules***
a	$b \rightarrow a$
c	$b \rightarrow c$
d	$b \rightarrow d$
e	$c \rightarrow e$

Table 3: Clustering based on LHS

Cluster_LHS	***Rules***
b	$b \rightarrow a, b \rightarrow c, b \rightarrow d$
c	$c \rightarrow e$

3.1 DCR framework

Some important concept used in proposed algorithm are as follows:

- **Sensitive item**: If there is an item in sensitive rules is called a sensitive item.
- **Item weight**: Number of iterations of any sensitive item
- **Transaction weight**: The total weight of items in a transaction.
- **Heavy transaction**: Heavy transaction is the one that is greater than zero.
- **Light transaction**: Light transaction is the one that is equal to zero.

3.2 DCR algorithm

Input: Original database D, Minimum Support Threshold (MST), and Minimum Confidence Threshold (MCT).
Output: Sanitized database D.
Initialize prerequisites

1. Measurement of sensitive items (Number of iterations of any sensitive item will be calculated).

2. Heavy transaction will be sorted based on their weight in descending order (In condition of weight equivalence, they will be sorted based on their length in ascending order).

3. Light transaction will be selected and based on their length they will be sorted

4. Sensitive rules will be clustered based on similar RHS and then the set of RHS will be generated. The set of RHS consists sensitive rules' RHS.

5. Sensitive rules will be clustered based on similar LHS and then the set of LHS will be generated. The set of LHS consists sensitive rules' LHS.

6. If RHS set is smaller than LHS set (with less numbers), RHS will be selected for hiding; otherwise LHS set will be selected.

7. If RHS set has been selected for hiding, sensitive rules' support will be decreased as follow:

    7.1. While all sensitive rules are not hidden, it deletes RHS item-sets from heavy transactions.

8. If LHS set has been selected for hiding, sensitive rules' confidence with the increasing LHS support of sensitive rules, will be decreased as follows:

    8.1. While all sensitive rules are not hidden, it adds LHS item-sets in light transactions.

    8.2. If all sensitive rules are not hidden (it might happen that there would be insufficient light transaction to add LHS item-sets) it shifts to step 4 After clustering unhidden sensitive association rules based on similar RHS it keeps on the process from step 7.

4. Illustrative example

In this section for further understanding proposed algorithm will be described as follows. In Table 4 the original database is shown. In this example MST=50, MCT=50, and sensitive rules are$c \longrightarrow e$, $c \longrightarrow a$,$d \longrightarrow h$. As in table 5 it is presented, at first, items' weight are calculated. Then it tries to specify light and heavy transactions of database. As it follows in Table 6, heavy transactions are sorted by their weighs in descending order and then sorted by their length in ascending order respectively. Similarly, in Table 7 light transactions based on their length are sorted in ascending order. Ascending sort by length of transaction ultimately makes changes smaller on transactions. Modification of smaller transactions reduces misses cost. For the next step, sensitive rules after clustering sensitive rules based on

RHS and LHS, their relative sets will be generated. As it is illustrated in Table 8, RHS set include 3 items or in another term clustering sensitive rules based on similar RHS caused 3 clusters. In Table 9, LHS sets are shown, in which clustering sensitive rules based on similar LHS generates 2 clusters and LHS set includes two items. So in here, because LHS set is smaller than RHS set, LHS set will be selected for hiding and the algorithm decreases sensitive rule's confidence by adding LHS item-sets in light transactions. In table 10, sanitized database specified.

Table 4: Original database

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

Table 5: Sensitive item

Sensitive Item	Weight
a	1
c	2
e	1
h	1

Table 6: Heavy transaction

TID	***Weight***	***Lenght***
13	6	6
3	6	7
2	5	4
7	5	5
12	5	5
15	5	5
1	4	3
5	4	4
10	4	4
6	3	3
8	3	4
9	3	5
4	2	5
13	6	6
3	6	7

Table 7: Light transaction

TID	***Weight***	***Item***
14	0	b g
11	0	b f g

Table 8: Clustering based on similar RHS

Cluster	***rules***
A	$c \longrightarrow a$
E	$c \longrightarrow e$
H	$d \longrightarrow h$
RHS={a, e, c}	

Table 9: Clustering based on similar LHS

Cluster	***rules***
C	$c \longrightarrow a, c \longrightarrow e$
D	$d \longrightarrow h$
LHS={c,d}	

Table 10: Sanitized database

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

5. Evaluation of proposed algorithm

In this paper, it has been trying to use DSRRC [9], ADSRRC[10] , and MDSRRC [11] to evaluate the proposed algorithm due to similar operation in hiding association rules. All four algorithms have been examined on PC with Core i3 CPU, 4 GB Ram, and Windows 7 operating system. The selective database for testing these algorithms are Chess and Mushroom. Properties of two databases are as follows:

Table 11: Database properties

Database Name	*Number of Transaction*	*Number of Item*	*Status*
Chess	3196	75	Dense
Mushroom	8124	119	Sparse

In this evaluation three factors (Hiding failure, Misses cost, and artifactual rules) have been selected. In each test, support, confidence, the number of sensitive rules and their types have been defined differently in order to evaluate algorithms' effectiveness in different circumstances.

In the first stage, tests have been operated on the Mushroom database. In each algorithm, hiding failure, Misses cost, artifactual rules, and number of modifications has been investigated separately. In each test the value of support, confidence, and number of sensitive rules are varied and any time it has been repeated from 1 to 10 rules. The average result of algorithms is presented in figure 1; in which in special situations, ADSRRC and DSRRC suffer from hiding failure. However, both DCR algorithm and MDSRRC are devoid of any hiding failure. Number of missing rules of the DCR algorithm in comparison of all is remarkably minimized. In all four algorithms, number of artifactual rules are low, but his factor in the DCR algorithm in both selection of LHS for hiding and addition LHS of sensitive rules, will increase. In the DCR algorithm, hiding process is based on less changes and modifications of algorithms, so, it has a noticeable effect on misses cost reduction.

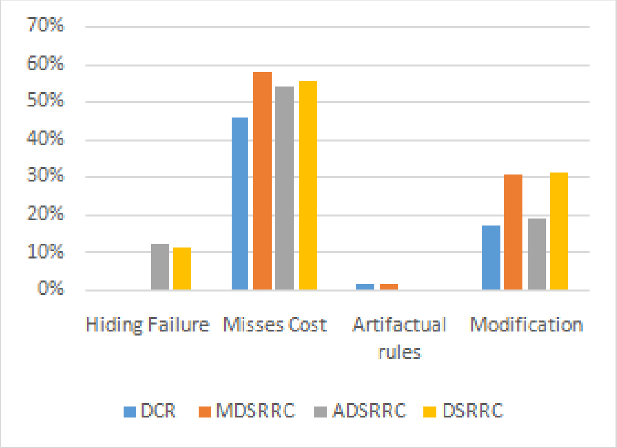

Fig. 1 Examination result of Mushroom database

In the second stage, the test has been done on Chess database. These examinations have been repeated with various support, confidence, and number of sensitive rules. The average result of four algorithms is presented in figure 2 in which hiding failure in all is zero. Similar to pervious test, number of missed rules and modifications in DCR algorithm are less than other three. Number of artifactual rules in the DCR algorithm due to adding LHS still is more than other three algorithms.

It can be concluded that, the DCR algorithm in dense and sparse databases has no hiding failure. And due to dual clustering process and selection of smallest cluster, it operates more efficiently than DSRRC, ADSRRC, and MDSRRC in reduction of misses cost and modifications in the database.

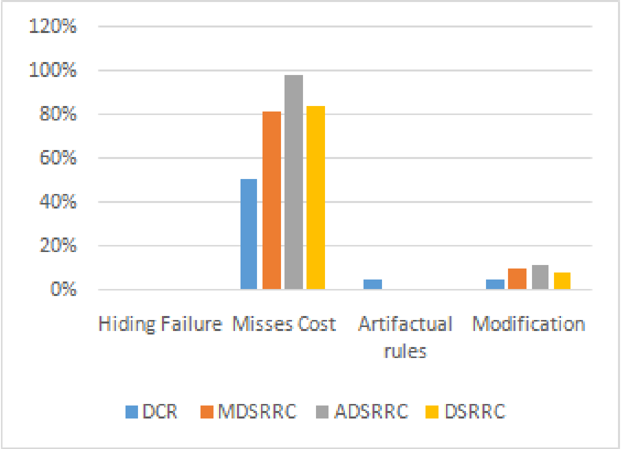

Fig. 2 Examination result of chess database

6. Conclusion

In this paper DCR algorithm has been proposed. The aim of this algorithm is to reduce hiding process side effects, especially hiding failure and Misses cost. Clustering is the method that is used for hiding association rules in this algorithm. Sensitive rules will be hidden in clusters instead of separately, which reduces misses cost. In this algorithm in order to enhance the outcome of clustering and eliminate the influences of the sensitive rules on clusters, two clustering procedures are performed on the sensitive rules based on similar RHS and LHS. After selection of smaller cluster hiding procedure will be done. If clusters based on similar RHS are selected, support of sensitive rules by removing RHS items from heavy transactions below the threshold, will be decreased and the sensitive rules will be hidden. Else cluster based on the LHS are selected, Confidence of sensitive rules by inserting LHS items in light transactions will be decreased below the threshold and they will be hidden. It has been tried to decrease the number of missing rules and changes by performing dual clustering and selection of the most appropriate clusters. The examination results, shows DCR performance in dense and sparse databases. In future works, it is possible to improve algorithm rules with multiplex RHS and LHS. In addition, it is likely to decrease the number of artifactual rules by improvement in insertion techniques.

References

[1] R. Agrawal, T. Imieliński, A. Swami, T. Imieli'nski, and Y. Guo, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, 1993, vol. 22, no. 2, pp. 207–216.

[2] Y. Guo, "Reconstruction-based association rule hiding," in *Proceedings of SIGMOD2007 Ph. D. Workshop on Innovative Database Research*, 2007, vol. 2007, pp. 51–56.

[3] D. Jain, P. Khatri, R. Soni, and B. B. K. Chaurasia, "Hiding Sensitive Association Rules without Altering the Support of Sensitive Item (s)," in *Advances in Computer Science and Information Technology. Networks and Communications*, vol. 3, no. 2, Springer, 2012, pp. 500–509.

[4] K. Shah, A. Thakkar, and A. Ganatra, "A Study on Association Rule Hiding Approaches," *doaj.org*, no. 3, pp. 72–76, 2012.

[5] S. Oliveira and O. Zaiane, "Algorithms for balancing privacy and knowledge discovery in association rule mining," *Database Engineering and …*, 2003.

[6] V. S. Verykios, E. D. Pontikakis, Y. Theodoridis, and L. Chang, "Efficient algorithms for distortion and blocking techniques in association rule hiding," *Distributed and Parallel Databases*, vol. 22, no. 1, pp. 85–104, Jul. 2007.

[7] Y. K. Jain, V. K. Yadav, and G. S. Panday, "An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining," *International Journal on Computer Science and Engineering*, vol. 3, no. 7, pp. 2792–2798, 2011.

[8] V. Yadav and R. Jindal, "Security Information Hiding in Data Mining on the bases of Privacy Preserving Technique," *International Journal of Computer Applications*, vol. 1, no. 15, pp. 49–52, 2010.

[9] C. N. Modi, U. P. Rao, and D. R. Patel, "Maintaining privacy and data quality in privacy preserving association rule mining," in *Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on*, 2010, pp. 1–6.

[10] R. S. Effects, H. M. R. H. S. Items, K. Shah, A. Thakkar, and A. Ganatra, "Association Rule Hiding by Heuristic Approach to Reduce Side Effects and Hide Multiple R. H. S. Items," *International Journal of Computer Applications*, vol. 45, no. 1, 2012.

[11] N. H. Domadiya and U. P. Rao, "Hiding sensitive association rules to maintain privacy and data quality in database," in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 2013, pp. 1306–1310.

[12] Y. Saygm, V. S. Verykios, C. Clifton, and Y. Saygin, "Using unknowns to prevent discovery of association rules," *ACM SIGMOD Record*, vol. 30, no. 4, pp. 45–54, 2001.

[13] S. R. M. Oliveira and O. R. Zaiane, "Privacy preserving frequent itemset mining," in *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, 2002, pp. 43–54.

[14] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, “Association rule hiding,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 4, pp. 434–447, 2004.

[15] S.-L. Wang and A. Jafari, “Hiding sensitive predictive association rules,” in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2005, vol. 1, pp. 164–169.

[16] S. Wang, D. Patel, A. Jafari, and T.-P. Hong, “Hiding collaborative recommendation association rules,” *Applied Intelligence*, vol. 27, no. 1, pp. 67–77, 2007.

[17] C.-C. Weng, S.-T. Chen, and H.-C. Lo, “A Novel Algorithm for Completely Hiding Sensitive Association Rules,” in *Intelligent Systems Design and Applications, 2008. ISDA’08. Eighth International Conference on*, 2008, vol. 3, pp. 202–208.

[18] 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, p. 1494, 2010.