Indian Journal of Science and Technology, *Vol 7(6), 826–832, June 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Hiding Sensitive Association Rules by Elimination Selective Item among R.H.S Items for each Selective Transaction

Atefe Ramezani*, Mohammad Naderi Dehkordi and Faramarz Safi Esfahani

Department of Computer Engineering, Islamic Azad University–Najafabad Branch, Iran;
ati.ram66@gmail.com, naderi@iaun.ac.ir, fsafi@iaun.ac.ir

Abstract

This paper focuses on hiding sensitive association rule which is an important research problem in privacy preserving data mining. For this, we present an algorithm that decreases confidence of sensitive rules to below minimum threshold by removing selective item among items of consequent sensitive rule (R.H.S) for each selective transaction. Finally, we qualitatively compare the efficiency of the proposed algorithm with that of already published algorithms in hiding association rules.

Keywords: Association Rule Minig, Datamining, Privacy Preserving

1. Introduction

Lately, the significant advances in data collection, data storage technologies and also the widespread use of the World Wide Web, has led to a huge volume of data. Therefore, Data mining has itself becomes a technique for automatically and intelligently extracting information or knowledge from a large amount of data. Despite the fact that it can assist data owners in strategic planning and decision making, it may also lead to reveal sensitive information. Therefore, in parallel development of data mining, a variety of questions can be raised including whether the data sources are used for other than the main goal. So, the new thread in data mining was introduced that should be designed a data mining system with privacy which can be faster with high volume of data storage and also able to prevent the disclosure of sensitive information. For this purpose, privacy preserving data mining has been extensively studied by researchers[4].

Privacy preserving in association rule mining is one of the important and significant researched techniques of data mining. It achieves to extract and reveal hidden relations and interesting association structures among large sets of data items in the transaction databases. Nowadays many organizations and companies keep their data in transaction data sets for processing and extracting knowledge using association rule mining[5,13].

In this paper, we focus on privacy preserving association rule mining. In doing so we assume that a certain subset of association rule, which is extracted from specific datasets, is considered as sensitive rules. Our goal then is modification of original data source in such a way that it would be impossible for the adversary to mine the sensitive rules from the modified data set and on the other hand, to minimize the side effects created by the hiding process as the sanitizing process can influence the original set of rules by

I. hiding and eliminating not sensitive rules that before of sanitizing process these rules extracting (lost rules)
II. extracting and disclosing unreal rules in the mining of the modified database, which were not supported by the original database (ghost rules)[9].

*Author for correspondence

2. Background and Related Work

Few papers entitled Privacy Preserving Data Mining (PPDM) appeared in 2000. While they introduced similar problem, the concepts of privacy were completely different.

2.1 Secure Multiparty Computation

Secure multiparty computation to encrypt data values[7], ensuring that no party acquires anything about another's data values. The goal of Secure Multiparty Computation (SMC) is that the parties involved infer nothing but the results[14].

2.2 Obscuring Data

Another approach relying on data obscuration, modifying the data values so real values are not revealed[1]. 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[5].

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[3,12].

This paper is organized as follows; First, The general problem formulation and the basic definitions of association rule mining are discussed. Then, the proposed algorithm for sensitive association rules is given. Therefore, gives the experimental results of the proposed technique. The last section provides the conclusion and future work.

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 transactions[1].

3.2 The Basics of Association Rules

Formally, association rules are defined as follows: Let $I = \{i_1,...,i_n\}$ 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 $X \cap Y = \varnothing$. 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[3,10].

3.3 Side Effects

The data loss (undesirable side effects) is defined, which 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[9,12,13].

3.4 Proposed Algorithm

The proposed heuristic algorithm tries for least complexity and the adverse effects of lead hide sensitive association rules to a minimum.

As, the proposed algorithm each time hiding a sensitive rule, preprocessing on transactions of original dataset and among the whole transactions, finds only transactions that fully supports sensitive rule. Then, given priority to each transaction that obtained in this way,

- Determine number of (sensitive, non sensitive and of negative-border that can be extracted under the influence of deletion operation) association rules that supported by transaction provided by at least one common item in right hand side of (sensitive, non sensitive)rules and also one common item in left hand side of negative border rules with one item in R.H.S of current sensitive rule(because , for us, only rules are important that with current sensitive rule have common and be affected by elimination of item)
- Determine sum of confidence of common (sensitive, non sensitive and of negative-border) association rules. So that, whatever in this sum of confidence for sensitive common rules are less, meaning earlier affected by deletion item and sooner hidden, and whatever the sum of confidence of non sensitive rules are more, meaning later affected by deletion item and also whatever the sum of confidence for negative-border rules are less, meaning later affected by deletion item

By obtaining each of these amounts and replace in formula of transaction priority, each transaction can be determined. Then, transactions sorted with highest priority. Therefore, among the items on the R.H.S (right hand side) of current sensitive rule, select the item that has the highest priority. Because, each item can repeat the different (sensitive, non sensitive and of negative-border) association rules with different confidence, we can have logic of given priority to transactions which is also used for items. So that, there may be different selection of items for each transaction and this causes a sudden support of only one item that does not reduce. And that, every time to reduce confidence of sensitive rule, an item is selected to remove, causes fewer side effects (lost rules, ghost rules). This process continues until confidence of current sensitive rule reduces below MCT threshold. Thus, this algorithm has three main stages:

1. The stage of selection of the appropriate transaction
2. The stage of selection of the appropriate item
3. The stage of removing selected item from selected transaction

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

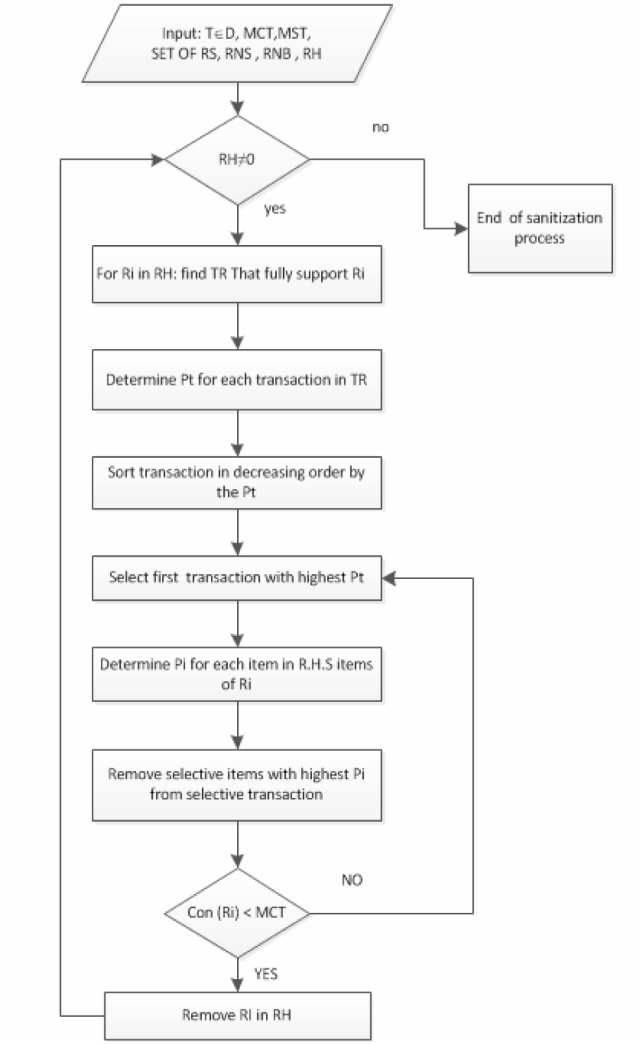

Figure 1. Flowchart of proposed algorithm.

The proposed algorithm has the following steps:

Input: transactions T∈D, non-sensitive rules set R_{NS}, Negative-border rules set NBRS, Rules to hide set R_H, Threshold MCT, MST

Output: modified database D_M

Step1.for each $R_i \in R_H$:

1-1. Find T_R For R_i {t in D /t full supports R_i}

1-2. For each transaction in T_R:

1-2-1. determine (f^+) the number of sensitive rules in R_H with at least one common items with I_R that supported by t

1-2-2. determine (f_1^-) the number of non-sensitive rules in R_{NS} with at least one common items with I_R that supported by t

1-2-3. determine (f_2^-) the number of negative-border rules in NBR_S with at least one common items with I_L that supported by t

1-2-4.determine the ratio:

If $f_1^- + f_2^- > 0$:

$$p_t = \frac{2f^+(\sum con_{RS-Common})^{-1}}{f_1^-(\sum con_{RNS-Common})^{-1} + f_2^-(\sum con_{RNB-Common})}$$

Else:

$$p_t = 2f^+(\sum con_{RS-Common})^{-1}$$

1-3. sort $t \in T_R$ in descending order of their of p_t

Step2. While (conf (R_i) < MST or SUP (R_i) < MST)

2-1. Select the first t with the highest p_t

2-2. for each R.H.S items from R_i:

2-2-1. determine (f'^+) the number of i in R.H.S R_H

2-2-2. determine ($f_1'^-$) the number of i in R.H.S R_{NS}

2-2-3. determine ($f_2'^-$) the number of i in L.H.S NBR_S

2-2-4. Determine the ratio:

If $f_1'^- + f_2'^- > 0$:

$$p_i' = \frac{2f'^+(\sum con_{RS-Common})^{-1}}{f_1'^-(\sum con_{RNS-Common})^{-1} + f_2'^-(\sum con_{RNB-Common})}$$

Else:

$$p_i' = 2f'^+(\sum con_{RS-Common})^{-1}$$

2-3. Select victim item=max p_i' {p_i': i∈I_R}

2-4. Remove victim item from t

2-5. Remove t from T_R

Step3. Re compute support and confidence of all rules in RH , RNS , NBRS that affected by remove victim

Step4. Update, such that:

4-1. if (conf(r) < MCT && SUP(r) < MST)

4-1-1. Remove r in R_H

4-1-2. if (conf(r) < MCT && SUP(r) ≥ MST) or if (conf(r) ≥ MCT && SUP(r) < MST)

4-1-2-1. Add r in NBR_S

4-2. if (conf(r) < MCT && SUP(r) < MST)

4-2-1.remove r in RNS

4-2-2. if (conf(r) < MCT && SUP(r) ≥ MST) or if (conf(r) ≥ MCT && SUP(r) < MST)

4-2-3. Add r in NBR_S

4-3. if (conf(r) ≥ MCT && SUP(r) ≥ MST)

4-3-1. Remove r in NBR_S

4-3-2.add r in R_{NS}

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 vista operating system at 2.10 GHz with 2 GB RAM. We used two datasets that these datasets are available through FIMI[15] 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 algorithms[6,11] for rule hiding that we also implemented. The first algorithm is called 1.b[6] and the second algorithm is called RRLR[11].

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:

1. The first category includes tests to hide the 3, 5, 7 sensitive association rule on dense dataset (Chess) and

Table 1. Properties of Datasets

Dataset	Number of transaction	Number of item	Avg. Items.
Mushrooms	8124	119	24
Chess	3196	76	37

Table 2. Result of mining on datasets

Dataset	MST	MCT	Association rules before hiding process.
Mushrooms	40%	70%	2495
Chess	90%	94%	5027

sparse dataset (mushrooms) with evaluation criteria: 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%[13].

As, Figures 2 and 3 show result of experiments of these algorithms. These figures indicate that these algorithms don't have hiding failure.

2. The second category includes tests to hide the 3, 5, 7 sensitive association rule on dense dataset (Chess) and 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, $\tilde{R}_P(D)$ corresponds the set of all non-sensitive rules in the original database D and $\tilde{R}_P(D')$ is the set of all non-sensitive rules in the sanitized database D'.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 is expected to miss[13].

In Figures 4 and 5, we see, the proposed algorithm performs better than algorithm1.b and algorithm RRLR.

3. The third category includes tests to hide the 3, 5, 7 sensitive association rule on dense dataset (Chess) and sparse dataset (mushrooms) with evaluation criteria:

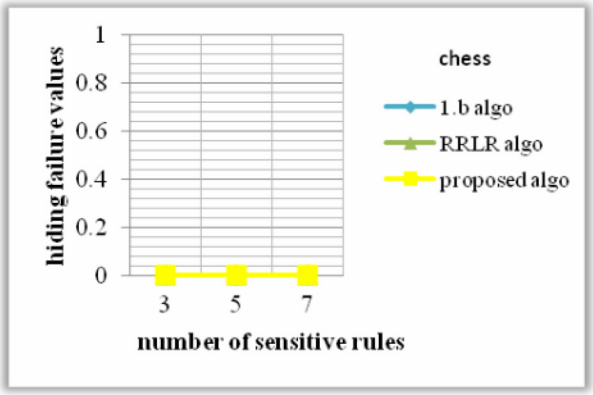

Figure 2. Failure hiding after the hiding process.

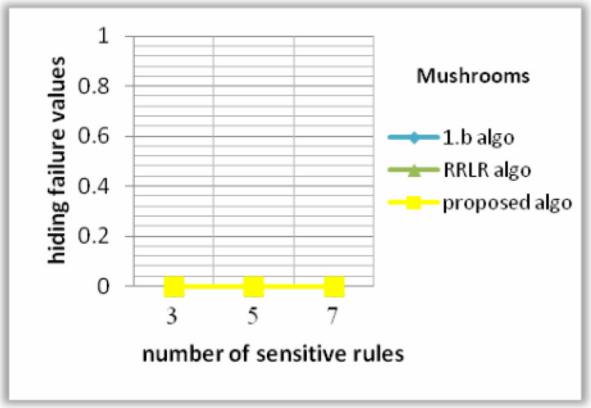

Figure 3. Failure hiding after the hiding process.

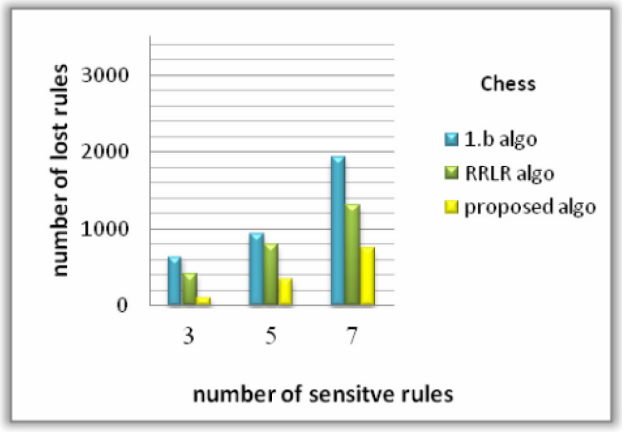

Figure 4. Rules lost after the hiding process.

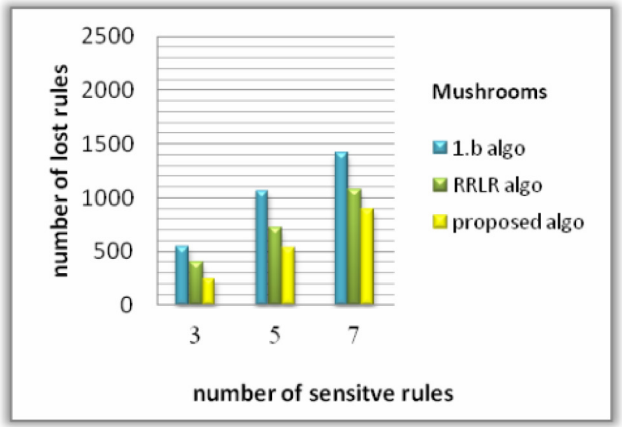

Figure 5. Rules lost after the hiding process.

Artifactual Patterns (AP), this measure quantifies the percentage of the discovered patterns that are artifacts. It is computed as follows:

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

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'[13].

Figures 6 and 7 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 1.b.

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 right hand side of sensitive rules for each transaction that fully support sensitive ruled and sorted these transaction according to priority formula, 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 1.b and algorithm RRLR by Evaluation criterions: hiding failure (HF), misses cost (MC), artifactual patterns (AP). The results obtained indicated that proposed algorithm is better than the other algorithms.

As future work, we plan to test the above techniques in real datasets that differ in the dependency of their item sets. In addition, we plan to construct a new algorithm that has a better run time.

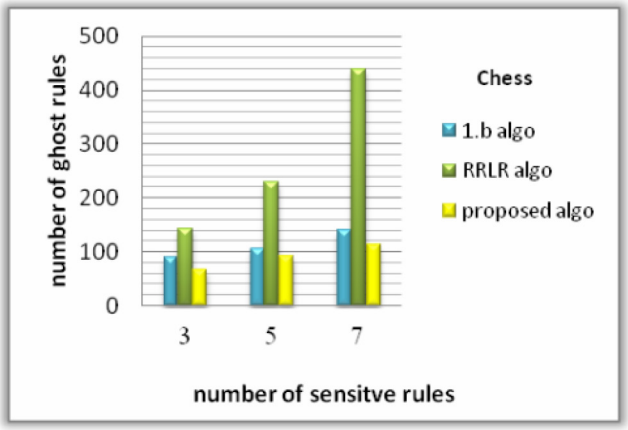

Figure 6. Creat rules after the hiding process.

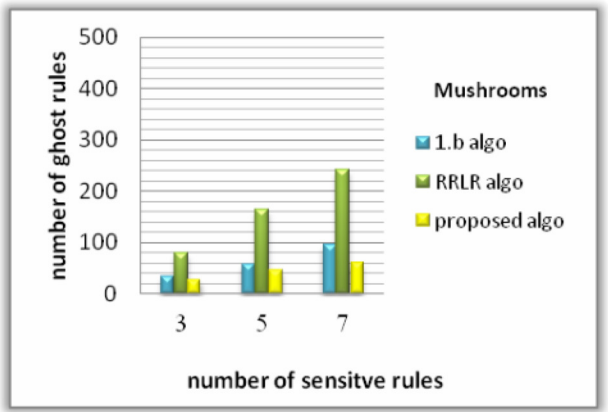

Figure 7. Creat rules after the hiding process.

6. References

1. Agrawal R, Srikant R. Privacy-preserving data mining. Proceedings of the 2000 ACM SIGMOD Conference on Management of Data; 2000. p. 439–50.
2. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD'93; 1993. p. 207–16.
3. Atallah M, Bertino E, Elmagarmid A, Ibrahim A, Verykios VS. Disclosure limitation of sensitive rules. Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99); 1999. p. 45–52.
4. Bertino E, Fovino IN, Povenza LP. A framework for evaluating privacy preserving data mining algorithms. Data Min Knowl Discov. 2005; 121–54.
5. Clifton C, Marks D. Security and privacy implications of data mining. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96); 1996. p. 15–19.
6. Dasseni E, Verykios VS, Elmagarmid AK, Bertino E. Hiding association rules by using confidence and support. Proceedings of the 4th International Workshop on Information Hiding; 2001. 369–83.
7. Bertino E, Lin D, Jiang W. A survey of quantification of privacy preserving data mining algorithms. Privacy-Preserving Data Mining Advances in Database Systems. 2008; 183–205.
8. Lindell Y, Pinkas B. Privacy preserving data mining. Advances in Cryptology. CRYPTO. 2000; 36–54.

9. Pontikakis ED, Tsitsonis AA, Verykios VS. An experimental study of distortion-based techniques for association rule hiding. Proceedings of the 18th Conference on Database Security (DBSEC 2004). 2004; 325–39.
10. Saygin Y, Verykios VS, Clifton C. Using unknowns to prevent discovery of association rules. SIGMOD Record. 2001; 30(4):45–54
11. Shah K, Thakkar A, Ganatra A. Association rule hiding by heuristic approach to reduce side effects & hide multiple r.h.s. items. International Journal of Computer Applications. 2012; 45(1):0975–8887.
12. Verykios VS, Bertino E, Fovino IN, Provenza LP, Saygin Y, Theodoridis Y. State-of-the-art in privacy preserving data mining. ACM SIGMOD Record. 2004; 33(1):50–57.
13. Verykios S, Gkoulalas-Divani A. A Survey of Association Rule Hiding Methods for Privacy. Springer. 2010; 34.
14. Yao AC. How to generate and exchange secrets. Proceedings of the 27th IEEE Symposium on Foundations of Computer Science; 1998; IEEE. p. 162–67
15. Available from: http://fimi.cs.helsinki.fi/data/