Association rule hiding using integer linear programming

Suma B., Shobha G.
Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Sep 5, 2020
Revised Dec 12, 2020
Accepted Jan 19, 2021

Keywords:

Association rule hiding
Data sanitization
Integer linear program
Privacy preserving data mining
Sensitive rules

ABSTRACT

Privacy preserving data mining has become the focus of attention of government statistical agencies and database security research community who are concerned with preventing privacy disclosure during data mining. Repositories of large datasets include sensitive rules that need to be concealed from unauthorized access. Hence, association rule hiding emerged as one of the powerful techniques for hiding sensitive knowledge that exists in data before it is published. In this paper, we present a constraint-based optimization approach for hiding a set of sensitive association rules, using a well-structured integer linear program formulation. The proposed approach reduces the database sanitization problem to an instance of the integer linear programming problem. The solution of the integer linear program determines the transactions that need to be sanitized in order to conceal the sensitive rules while minimizing the impact of sanitization on the non-sensitive rules. We also present a heuristic sanitization algorithm that performs hiding by reducing the support or the confidence of the sensitive rules. The results of the experimental evaluation of the proposed approach on real-life datasets indicate the promising performance of the approach in terms of side effects on the original database.

Corresponding Author:

Suma B.
Department of Computer Science and Engineering
RV College of Engineering
Mysore Road, Bengaluru, 560059, India
Email: sumab_rao@rvce.edu.in

1. INTRODUCTION

Data mining aims to explore and analyze the huge volumes of data and data mining systems are categorized depending upon the types of knowledge they discover. However, the knowledge discovered through various data mining algorithms may contain sensitive information about an individual or business. Disclosure of such sensitive information may cause a threat to security. Henceforth, comprehensive sanitization of the database is essential when data is shared with a third party.

Privacy preserving data mining (PPDM) [1, 2] has evolved as an interesting problem in database security applications due to the diverse conflicting requirements of data sharing, proprietary data disclosure, privacy concern and knowledge discovery. The objective of PPDM is to develop data sanitation algorithms which modify the data such that even after applying mining algorithms the sensitive knowledge remains intact. Verykios *et al.* [3] analysed the state-of-the-art, presented classification hierarchy and clustering of different privacy preserving data mining techniques. Bertino *et al.* [4] presented an approach for evaluating different attributes of a privacy preserving algorithm. Knowledge hiding, a subfield of PPDM, can be achieved by a process known as data sanitization [5]. The process of knowledge hiding modifies the sensitive data before delivering it to the third party [6] in order to ensure data privacy. In this paper, we focus on the knowledge hiding process in the context of association rule mining (ARM). The sensitive rule hiding problem is common in collaborative ARM applications where only a part of the information found in the data must be revealed by the organization, strategic knowledge inferred by the sensitive rules must be concealed. Hence, sensitive patterns should no longer be extracted from the database at the same time database utility be well maintained. The technique that we propose is formulated based on the mathematical optimization problem called integer linear programming (ILP). The objective of the proposed ILP formulation is to minimize the total weight of sensitive transactions while achieving zero hiding failure. The solution obtained from the ILP determines the transactions that need to be sanitized in order to hide the sensitive rules.

The paper is organized as follows: Section 2 presents a brief overview of previous works on sensitive association rule hiding. Section 3 provides a formalization of the problem and the proposed methodology is described in section 4. Section 5 discusses performance results of the proposed methodology. The final section of this paper presents concluding remarks and future extensions.

2. RELATED WORK

Data sanitization approaches are categorized into heuristic, border, exact and evolutionary algorithms. The heuristic approach uses blocking or distortion technique to determine suitable sensitive items and transactions for modification. Two fundamental heuristic approaches were presented in [7] to prevent sensitive rules from disclosure. The first method hides the frequent itemsets from which sensitive rules are derived thereby preventing sensitive rules from being generated. The second method reduces the relevance of sensitive rules by bringing its confidence below the given minimum threshold. Oliveira *et al.* [8, 9] implemented an index schema and the transaction retrieval engine to speed up the process of sanitization. Aggregate, Hybrid and Disaggregate algorithms presented in [10] perform better than the SWA algorithm [9] in terms of data utility but suffers from computational complexity. Item grouping algorithm (IGA) presented in [8] is improved in [11] to decrease the number of modifications. Hong *et al.* in [12] proposed a greedy SIF-IDF algorithm that uses TF-IDF measure from information retrieval to compute the correspondence between sensitive itemsets and transactions. Cheng *et al.* [13] proposed an algorithm that reduces data distortion degree by modifying the least number of transactions in order to conceal a sensitive rule. Le *et al.* [14] proposed a distortion-based approach that modifies the minimum number of transactions during the hiding process. Pang *et al.* [15] devised a sensitive association rule hiding algorithm on outsourced data uploaded from multiple data owners in a twin cloud architecture using homomorphic cryptosystem. Shaoxin *et al.* [16] proposed a database reconstruction-based technique for hiding frequent itemsets achieves a high degree of privacy and reasonable data utility of the synthetic database. The main drawback of the heuristic approach is that in the majority of cases, it fails to deliver an optimal solution to the sanitization problem.

The border approach focuses on reducing side effects on non-sensitive itemsets during the process of database sanitization. The sanitization process utilizes border theory [17] to reduce the impact of the sanitization process on low support non-sensitive itemsets. The border approach presented by Sun and Yu [18] uses the positive border to keep track of the impact of transaction sanitization. Telikani *et al.* [19] devised the DCR algorithm using the combination of heuristic and border-based approaches in order to minimize the impact on non-sensitive rules while hiding sensitive rules. Greedy algorithm presented in [20] uses border theory to provide an optimal solution for hiding sensitive frequent itemsets.

The exact approach formulated the sanitization problem as a constraint satisfaction problem (CSP). Menon *et al.* [21] utilized ILP to formulate a CSP that determines the least number of transaction sanitizations in order to conceal sensitive itemsets. Divanis and Verykios in [22] defined a CSP to select candidate itemsets for modifications. The sanitization algorithm determines frequent itemsets that belong to positive and negative borders. The first phase of the sanitization process terminates when all sensitive itemsets are concealed with zero side effects. Otherwise, the second phase is executed until the feasible solution to the CSP is found. CSP based approaches efficiently maintain data accuracy but require high computation time. Evolutionary algorithms encode the sanitization problem into a population of binary solutions. Cuckoo Optimization method proposed in [23] conceals sensitive association rules, while it minimizes the number of cycles and access. GA-based algorithms proposed in [24] and PSO-based algorithms devised in [25] are deletion-based approaches, compute righteousness of chromosome to determine side-effects of sanitization by defining fitness function. Each solution consists of a transaction set which is used for the chromosome encoding. Wu *et al.* [26] presented an algorithm ACS2DT based on ant colony system to reduce side effects. Genetic algorithm approach proposed in [27] formulates an objective function that computes the side effect on non-sensitive rules. ABC4ARH rule hiding algorithm presented in [28] selects sensitive transactions by using an improved discrete binary artificial bee colony algorithm. Genetic algorithm-based approaches provide strategies only for identifying transactions to be removed from or to be added into the database.

3. PROBLEM STATEMENT

Let $I = \{\gamma_1, \gamma_2, \dots, \gamma_m\}$ be the finite set of m items. An itemset is a nonempty subset I_k where $Ik \subseteq I$ and k-Itemset is an itemset containing k items. Let $D = \{T_1, T_2, \dots, T_n\}$ with $\forall i, 1 \le i \le n: Ti \subseteq I$ be a tuple of transactions over I. This tuple is called the transaction of the database D. A transaction $T_i \epsilon D$ supports an itemset α iff $\alpha \subseteq T_i$. The support count of α is the number of transactions containing α, denoted as |α|. The itemset α is frequent if its support is greater than or equal to the given minimum support threshold.

An association rule is defined as an implication expression $\alpha \to \beta$ where $\alpha, \beta \subseteq I$ and $\alpha \cap \beta = \Phi$. A rule α→ β is said to hold a support σ in the database, where σ is the fraction of transactions that covers both α and β. A rule α→β is said to have confidence δ in the set of transactions where δ measures how frequently itemset β appears in the transactions that covers itemset α. The support σ and confidence δ are mathematically formulated by (1) and (2).

$$\text{support}\ (\alpha \to \beta) = \frac{|\alpha \cup \beta|}{|D|} \tag{1}$$

$$\text{confidence}\ (\alpha \to \beta) = \frac{|\alpha \cup \beta|}{|\alpha|} \tag{2}$$

Let σ_{min} and δ_{min} be the user specified minimum support threshold and the minimum confidence threshold. A rule is strong if it satisfies both support and confidence thresholds. The ARM algorithm finds all strong association rules. To determine the strong rules, the rule mining algorithm first finds all the itemsets in D that are frequently enough to be considered important i.e. support ≥ σ_{min} (frequent itemsets) and subsequently derives rules that are strong enough to be considered interesting. Sensitive rules are strong association rules that the data owner wants to hide. The association rule hiding problem aims to restrict theses sensitive rules from being disclosed.

The sensitive association rule hiding problem addressed in this paper is stated as follows: Given a transactional database D, minimum support threshold σ_{min}, minimum confidence threshold δ_{min}, a set of strong rules R mined from D and a set of sensitive rules S⊆R to be hidden, modify the original D into a transformed database D′ to hide sensitive rules S from being disclosed, while minimally influencing non-sensitive rules in the set R-S. In this paper, we propose an approach to reduce the support or confidence of the sensitive rules below the user specified minimum threshold by sanitizing selected transactions of D such that no sensitive rule is discovered from D′. The proposed approach conceals sensitive association rules while maintaining data utility.

4. PROPOSED SOLUTION

This section presents the proposed ILP based strategy for hiding sensitive rules. A sensitive rule $\alpha \to \beta$ can be is hidden using one of the following methods:

Method 1: removing an item jϵα or β from the selected transactions until support $(\alpha \to \beta) < \sigma_{min}$.

Method 2: adding all items jϵα to the selected transactions until confidence $(\alpha \to \beta) < \delta_{min}$.

Method 3: removing an item jϵα from the selected transactions until support $(\alpha \to \beta) < \sigma_{min}$or confidence $(\alpha \to \beta) < \delta_{min}$.

The insertion or deletion of any item may lead to side effects including, ghost rules and lost rules: i) Ghost rules are new non-sensitive rules discovered from the transformed database D′ but not present in the input database D; and ii) Lost rules are non-sensitive rules which are discovered from the input database D but lost in the transformed database D′ during hiding process. The solution to the hiding process is split down into three phases: Pre-processing, ILP formulation and hiding process.

4.1. Pre-processing

To find the solution to the hiding problem, we employ the item deletion strategy of method 3, as it has more utility in hiding sensitive association rules. One of the key issues that need to be resolved for the sanitization is identifying suitable transactions in the database for modifications. If an item that belongs to the consequent itemset of a sensitive rule is removed from the selected supporting transaction, it reduces both the support of the inducing itemset and the confidence of the sensitive rule, but the support of the antecedent part remains unaffected. In contrast, if an item of antecedent itemset of sensitive rule from a supporting transaction is removed, it reduces union support of antecedent and generating itemset. This technique decreases the confidence slowly as compared to the former techniques. To optimize and speed up the hiding process, a pre-processing phase is implemented to find database D_1 with all sensitive transactions that completely supports one or more sensitive rules. The pre-processing phase also finds non-sensitive rules that contain no items of any sensitive rules and deletes from the set of non-sensitive rules S′ because database sanitization has no impact on such non-sensitive rules.

4.2. ILP formulation

The solution to the rule hiding problem is modelled with the ILP shown in (3), (4) and (5).

$$\min \sum_{\forall i: T_i \in D_1} c_{\,i} v_i \tag{3}$$

$$\text{subject to} \sum_{\forall i,j:\, S_j \subseteq T_i \in D_1} v_i \geq n_{min}, \quad \forall S_j \in \; S \tag{4}$$

$$v_i \;\in \{0,1\}, \forall i: T_i \in D_1 \tag{5}$$

Each variable v_i, coefficient c_i corresponds to a transaction T_i in the pre-processed database D_1 and each constraint corresponds to a sensitive association rule S_j in S. A constraint contains a variable if the corresponding sensitive rule is supported by the transaction T_i.

The linear system has a variable for each sensitive transaction and |S| constraints. The objective function of the ILP shown in (3) aims to minimize database side effects while achieving zero hiding failure. In order to conceal the sensitive rule S_j the conditional constraints given in (4) ensure that at least n_{min} transactions that support sensitive rule S_j are selected for sanitization. In (5) represents the selection or rejection of a transaction T_i and enforces each variable v_i be zero or one. The solution generated by the linear program indicates the set of transactions that need to be selected for sanitization. The coefficient assigned to each sensitive transaction can have a significant effect on the collection of transactions identified for each modification and hence on the quality of transformed database D′. In order to compute the minimum number of transactions n_{min}, the following properties are used.

Property 1: Let T_s be the transaction set supporting the sensitive rule α→β. To reduce the confidence of the sensitive rule below δ_{min}, the least number of transactions to be sanitized in T_s is n_1=⌈|α→β|-|α|*δ_{min}⌉ +1. Proof: If an item of the consequent itemset of a sensitive rule S_j is deleted from a sensitive transaction T_s, then support of S_j decreases by 1. Assume n_1 is the least number of transactions that are forced to be sanitized in T_s to decrease the rule's confidence below δ_{min}, then we have (|α→β|-n_1)/|α|<δ_{min}. Therefore n_1>|α→β|-|α|*δ_{min}. Since n_1 is is the least integer, we can derive $n_{1=}$⌈|α→ β|-|α|*δ_{min}⌉+1.

Property 2: Let T_s be the transaction set supporting the sensitive rule α→ β. To reduce the support below σ_{min}, the least number of transactions to be sanitized in T_s is n_2=⌈|α→β|-σ_{min}*|D|⌉+1. Proof: If an item from consequent itemset of a sensitive rule S_j is deleted from a sensitive transaction T_s, then support of S_j decreases by 1. Assume that n_2 is the least number of transactions from T_s that requires sanitization to reduce the support of S_j below σ_{min}, then we have (|α→β|- n_2)/|D|<σ_{min}. Therefore n_2>|α→β|-|D|*σ_{min}. Since n_2 is the least integer, we can derive n_2=⌈|α→β|-σ_{min}*|D|⌉+1.

From properties 1 and 2, it can be deduced that the least number of transactions that require sanitization is n_{min}=min($n_{1,}$ n_2) to suppress the sensitive rule α→β. Since decreasing the support of some sensitive rule A→B may have an impact on the support of antecedent itemset of the sensitive rule α→β, n_1 cannot be calculated in advance. Therefore n_{min}=n_2=⌈|α→β|-σ_{min}*|D|⌉+1.

The coefficient for each transaction that is included in the constraint matrix is computed using the coefficient computing algorithm shown in Figure 1. The constraint matrix is created by considering transactions that support one or more sensitive rules. The objective function is devised such that the binary variables that indicate the selection or rejection of a transaction are multiplied by the pre-calculated coefficients that reflect its vulnerability of being affected by the sanitization. We assign the weight for each item present in the consequent itemset the sensitive rules based on its presence in the number of antecedent and consequent itemset of the sensitive rules.

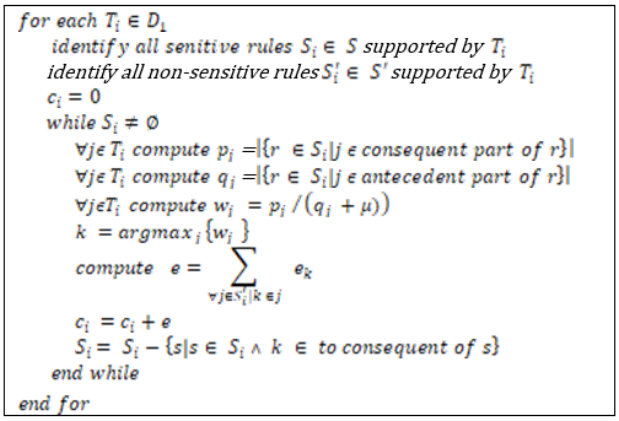
Furthermore, a small constant μ is added to the denominator to prevent the possibility of division by zero. The impact of deleting the maximum weight item on non-sensitive transactions is used as the coefficient of a sensitive transaction. The value of the transaction coefficient indicates the risk of over concealing non-sensitive rules on selecting the transaction for sanitization. A sensitive transaction containing less non-sensitive rules with large support yields a low coefficient. On the other hand, a transaction that contains more non-sensitive rules with support closer to σ_{min} is less likely to get selected for sanitization. The impact, e_k of deleting the item ′k′ on a non-sensitive rule S'_j is calculated using (6).

$$e_k = 1 - \left(\left|S'_j\right|_1 - \left(\left|S'_j\right| - \sigma_{min} \times |D|\right)\right)/|D_1| \tag{6}$$

where $|S'_j|_1$, $|S'_j|$ are support of non-sensitive rule S'_j in D_1 and D respectively. Deleting an item from the two different transactions T_i and T_j with the same number of non-sensitive rules does not ensure that they are vulnerable to introducing side effects to the same degree.

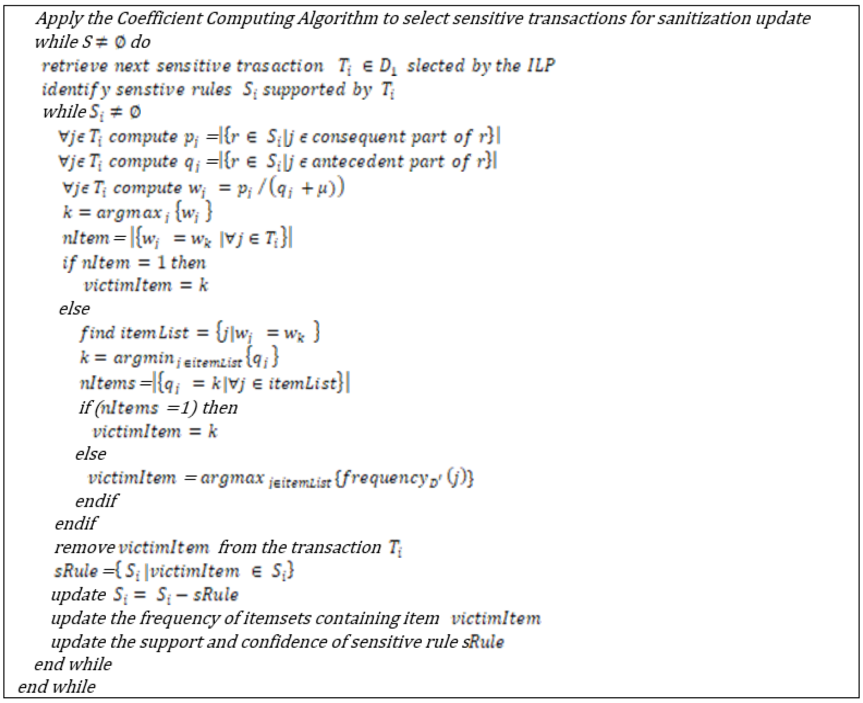
4.3. Hiding process

The sanitization algorithm ILPARH shown in Figure 2 hides each sensitive rule α→β by deleting an item from consequent itemset β until its support is below σ_{min} or its confidence is below δ_{min}. The number of item deletions required for a sensitive rule α→β is given by the equation $n_{min}=n_2$.The algorithm computes weight w_j for each item j, where $j\epsilon T_i$ as described in Coefficient Computing Algorithm and an item with the maximum weight is selected as victim item for deletion. If two or more items have the same maximum weight, then an item contained in the fewest antecedent of sensitive rules is selected. If a tie arises another time, then an item with highest support is selected for deletion. If two or more items have the same maximum support, then the sanitization algorithm picks the victim item randomly. The sensitive rules containing the victim item is also removed from S_i. The support of affected frequent itemsets, sensitive rules, and confidence of the affected sensitive rules are updated. This procedure is repeated until S_i is left empty.



```
for each $T_i \in D_1$
    identify all senitive rules $S_i \in S$ supported by $T_i$
    identify all non-sensitive rules $S'_i \in S'$ supported by $T_i$
    $c_i = 0$
    while $S_i \neq \emptyset$
        $\forall j \epsilon T_i$ compute $p_j = |\{r \in S_i | j \epsilon$ consequent part of $r\}|$
        $\forall j \epsilon T_i$ compute $q_j = |\{r \in S_i | j \epsilon$ antecedent part of $r\}|$
        $\forall j \epsilon T_i$ compute $w_j = p_j / (q_j + \mu))$
        $k = argmax_j \{w_j\}$
        compute $e = \sum_{\forall j \in S'_i | k \in j} e_k$
        $c_i = c_i + e$
        $S_i = S_i - \{s | s \in S_i \wedge k \in$ to consequent of $s\}$
    end while
end for
```

Figure 1. Coefficient computing algorithm



```
Apply the Coefficient Computing Algorithm to select sensitive transactions for sanitization update
while $S \neq \emptyset$ do
  retrieve next sensitive trasaction $T_i \in D_1$ slected by the ILP
  identify senstive rules $S_i$ supported by $T_i$
  while $S_i \neq \emptyset$
    $\forall j \epsilon T_i$ compute $p_j = |\{r \in S_i | j \epsilon$ consequent part of $r\}|$
    $\forall j \epsilon T_i$ compute $q_j = |\{r \in S_i | j \epsilon$ antecedent part of $r\}|$
    $\forall j \epsilon T_i$ compute $w_j = p_j / (q_j + \mu))$
    $k = argmax_j \{w_j\}$
    $nItem = |\{w_j = w_k | \forall j \in T_i\}|$
    if $nItem = 1$ then
      $victimItem = k$
    else
      find $itemList = \{j | w_j = w_k\}$
      $k = argmin_{j \epsilon itemList} \{q_j\}$
      $nItems = |\{q_j = k | \forall j \in itemList\}|$
      if $(nItems = 1)$ then
        $victimItem = k$
      else
        $victimItem = argmax_{j \epsilon itemList} \{frequency_{D'}(j)\}$
      endif
    endif
    remove $victimItem$ from the transaction $T_i$
    $sRule = \{S_i | victimItem \in S_i\}$
    update $S_i = S_i - sRule$
    update the frequency of itemsets containing item $victimItem$
    update the support and confidence of sensitive rule sRule
  end while
end while
```

Figure 2. ILPARH algorithm

5. PERFORMANCE EVALUATION AND RESULTS DISCUSSION

This section presents the results of experimental evaluations carried out on different real-world datasets. We evaluated our proposed algorithm and compared the results with the results of the DCR algorithm [19]. A set of experiments are conducted to measure the performance of the algorithms in terms of side-effects and execution time. The ILPARH and DCR algorithms were implemented in R and were executed in an Intel Pentium 4 using the Windows 10 Operating System at 2.50 GHz with 4 GB of RAM.

5.1. Datasets

We examined the proposed algorithm using three different transaction datasets that are publicly accessible through the FIMI repository: mushroom, chess and BMS-1. These datasets exhibit different characteristics with regard to the maximum size of an itemset, number of transactions and average transaction size. The configurations of the overall datasets depicted in Table 1 where |I|, |D| and AvgSize respectively indicate the maximum size of an itemset, the number of transactions and the average size of transactions. The parameters σ_{min} and δ_{min} were set to confirm that ARM algorithm results in adequate number of strong associations rules.

5.2. Experimental results

In order to demonstrate the efficiency of the proposed algorithm, several experiments were conducted on real-life datasets. At first, using the association rule mining algorithm, frequent itemsets are generated with the threshold parameter σ_{min}. Then, association rules are discovered with the threshold parameter δ_{min}. The Table 2 depicts the number of association rules that are generated for datasets Chess, Mushroom and BMS-1. Some these rules are selected as sensitive association rules S.

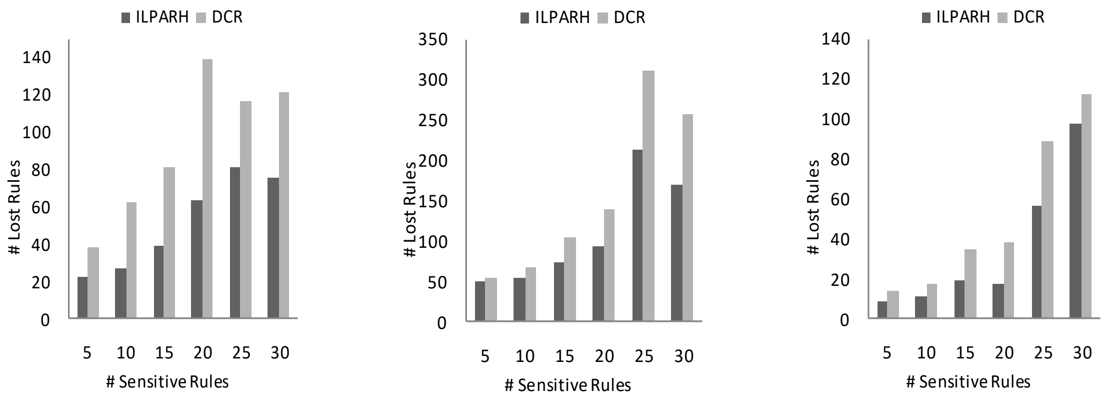
Table 1. Characteristics of datasets

Dataset	|I|	|D|	AvgSize
Chess	75	3,196	37.0
Mushroom	119	8,124	23.0
BMS-1	497	59,601	2.42

Table 2. Values set for ARM algorithm threshold parameters

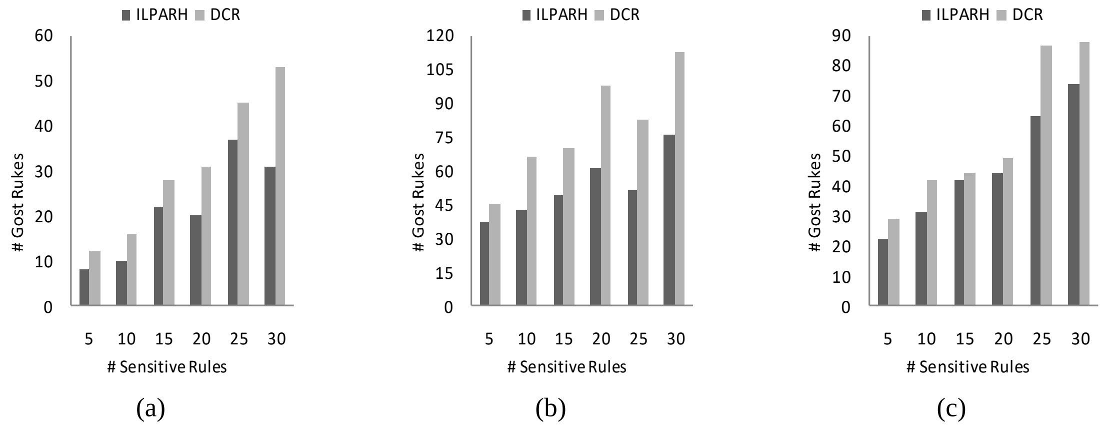
Dataset Name	σ_{min}	δ_{min}	# Association Rules
Chess	0.95	0.98	303
Mushroom	0.40	0.70	3828
BMS-1	0.001	0.70	2224

The major performance criterion of the sanitization algorithm is the side effects it incurs on the data. We measure the side effects by summing up the number of lost rules and the number of ghost rules introduced. Figure 3 depicts the relationship between number of lost rules and number of sensitive rules. Figure 4 depicts the relationship between number of new rules generated and number of sensitive rules. The results show that the proposed method generates fewer side effects in comparison with the DCR algorithm. The reason is that the ILPARH algorithm utilizes ILP to determine the candidate transactions for modifications and thereby lead to a higher quality data. Figure 5 shows the relationship running time of the algorithm and number sensitive rules. As illustrated in Figure 5 there is an increase in the running time when compared to the DCR algorithm. The reason is that our algorithm requires additional computation to calculate the coefficients of each sensitive transaction in the ILP formulation.



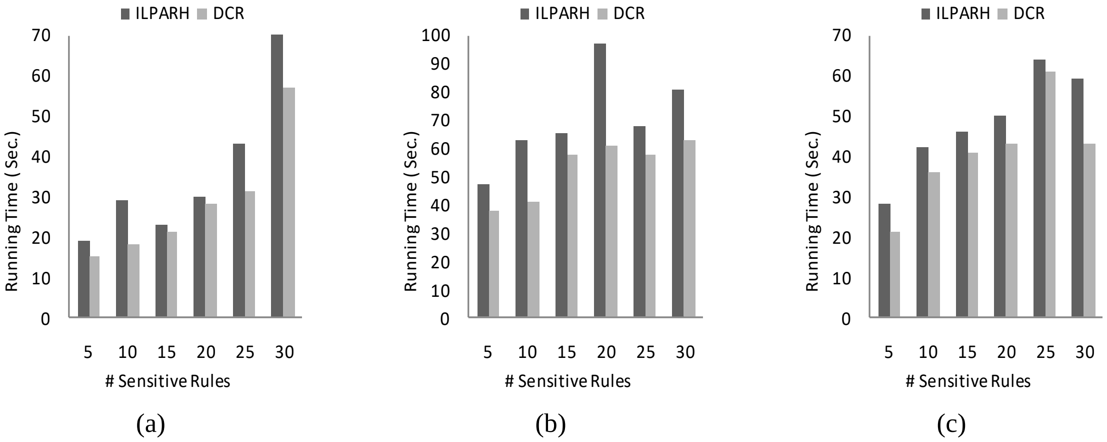
(a) (b) (c)

Figure 3. Side-effects in terms of number of lost rules for different number of sensitive rules



(a) (b) (c)

Figure 4. Side-effects in terms of number of ghost rules for different number of sensitive rules



(a) (b) (c)

Figure 5. Runtime under different number of sensitive rules

6. CONCLUSION

In this paper, we presented a privacy-preserving algorithm ILPARH to protect sensitive association rules. The degree of the side effects on non-sensitive rules is used as coefficients of sensitive transactions in the ILP formulation. We exploit the characteristics of objective function to utilize the partial results of the CSP and deriving the solution for hiding sensitive rules. The results of experiments show that our approach minimizes the number of concealed non-sensitive rules and also discovery of ghost rules. In our future work, we intend to employ the evolutionary based framework to identify the candidate transactions for modifications during the sanitation process. Also, the evolutionary approach in conjunction with our victim item determining technique can be adopted to reduce side effects with the improved algorithmic efficiency.

REFERENCES

[1] A. Rakesh, and R. Srikant, "Privacy-preserving data mining," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 439-450.

[2] L. Yehuda, and B. Pinkas, "Privacy preserving data mining," *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 00)*, vol. 1880, 2000, pp. 36-54.

[3] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., and Theodoridis, Y., "State-of-the-art in privacy preserving data mining," *ACM Sigmod Record*, vol. 33, no. 1, pp. 50–57, 2014.

[4] Bertino, Elisa, I. N. Fovino, and L. P. Provenza, "A Framework for Evaluating Privacy Preserving Data Mining Algorithms," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp.121-154, 2005.

[5] Askari, Mina, R., Safavi-Naini, and Ken Barker, "An information theoretic privacy and utility measure for data sanitization mechanisms," *Proc. of the 2nd ACM Conf. on Data and App. Security and Privacy*, 2012, pp. 283-229.

[6] A. Evfimievski, "Randomization in privacy preserving data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 43-48, 2002.

[7] Verykios, V. S., Elmagarmid, A. K., Bertino, E., Saygin, Y., and Dasseni, E., “Association rule hiding,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 434–44, 2004.

[8] Oliveira, Stanley R. M., and Osmar R. Zaiane., “Privacy preserving frequent itemset mining,” *Proceedings of IEEE international conference on privacy, security and data mining*, 2002, pp. 43-54.

[9] Oliveira, Stanley RM, and Osmar R. Zaiane., “Protecting sensitive knowledge by data sanitization,” *Third IEEE International conference on data mining,* 2003, pp. 613-616.

[10] Amiri, A., “Dare to share: Protecting sensitive knowledge with data sanitization,” *Decision Support Systems*, vol. 43, no. 1, pp. 181-191, 2007.

[11] Li, Yu-Chiang, Jieh-Shan Yeh, and Chin-Chen Chang, “An effective sanitization algorithm for hiding sensitive patterns on data mining,” *Advanced Engineering Informatics*, vol. 21, no. 3, pp. 269-280, 2007.

[12] Hong *et al*., “Using TF-IDF to hide sensitive itemsets,” *Applied Intelligence*, vol. 38, no. 4, pp. 502–510, 2013.

[13] Cheng, P., Roddick, J. F., Chu, S. C., and Lin, C. W., “Privacy preservation through a greedy, distortion-based rule-hiding method,” *Applied Intelligence*, vol. 44, no. 2, pp. 295–306, 2016.

[14] Le B., Le, Bac, Lien Kieu, and Dat Tran., “Distortion-Based Heuristic Method for Sensitive Association Rule Hiding,” *Journal of Computer Science and Cybernetics*, vol. 35, no. 4, pp. 337-354, 2019.

[15] P. Hongping, and B. Wang, “Privacy-Preserving Association Rule Mining Using Homomorphic Encryption in a Multikey Environment,” *IEEE Systems Journal*, pp. 1-11, 2020.

[16] Li, S., Mu, N., Le, J., and Liao, X., “Privacy preserving frequent itemset mining: Maximizing data utility based on database reconstruction,” *Computers and Security,* vol. 84, pp. 17-34, 2019.

[17] M. Heikki, and H. Toivonen, “Level wise search and borders of theories in knowledge discovery,” *Data Mining and Knowledge Discovery,* vol. 1, no. 3, pp. 241-258, 1997.

[18] S. Xingzhi, and P. S. Yu, “A border-based approach for hiding sensitive frequent itemsets,” *Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005, pp. 426-433.

[19] T., Akbar, and A. Shahbahrami, “Optimizing association rule hiding using combination of border and heuristic approaches,” *Applied Intelligence,* vol. 47, no. 2, pp. 544-557, 2017.

[20] Suma, B., and G. Shobha, “A Greedy Approach to Hide Sensitive Frequent Itemsets with Reduced Side Effects,” *Proceedings of International Conference on Remote Engineering and Virtual Instrumentation*, 2019, pp. 849-858.

[21] Menon, Syam, S. Sarkar, and S. Mukherjee, “Maximizing accuracy of shared databases when concealing sensitive patterns,” *Information System Research*, vol. 16, no. 3, pp. 256-270, 2005.

[22] Gkoulalas-Divanis, Aris, and Vassilios S. Verykios., “An integer programming approach for frequent itemset hiding,” *Proceedings of the 15th ACM conference on information and knowledge management*, 2016, pp. 748-757.

[23] Afshari, Mahtab Hossein, M. Naderi Dehkordi, and Mehdi Akbari., “Association rule hiding using cuckoo optimization algorithm,” *Expert Systems with Applications,* vol. 64, pp. 340–351, 2016.

[24] Lin, C. W., Hong, T. P., Yang, K. T., and Wang, S. L., “The GA-based algorithms for optimizing hiding sensitive itemsets through transaction deletion,” *Applied Intelligence*, vol. 42, no. 2, pp. 210-230, 2014.

[25] Lin, J. C. W., Liu, Q., Fournier-Viger, P., Hong, T. P. *et al*., “A sanitization approach for hiding sensitive itemsets based on particle swarm optimization,” *Engineering Applications of Artificial Intelligence,* vol. 53, pp. 1-18, 2016.

[26] Wu, Jimmy Ming-Tai, Justin Zhan, and J. Chun-Wei Lin, “Ant colony system sanitization approach to hiding sensitive itemsets,” *IEEE Access,* vol. 5, pp. 10024-10039, 2017.

[27] Khuda Bux, N., Lu, M., Wang, J., Hussain, S., and Aljeroudi, Y., “Efficient Association Rules Hiding Using Genetic Algorithms,” *Symmetry,* vol. 10, no. 11, pp. 576-594, 2018.

[28] Telikani, A., Gandomi, A. H., Shahbahrami, A., and Dehkordi, M. N., “Privacy-preserving in association rule mining using an improved discrete binary artificial bee colony,” *Expert Systems with Applications,* vol. 144, 2020.

BIOGRAPHIES OF AUTHORS

Suma B. is working as Assistant Professor at Computer Science and Engineering Department, RV College of Engineering, Bengaluru, since 2010. She has received her master’s degree from NITK, Surthkal and pursuing her Ph.D. (CSE) from VTU. Her research areas of interest are Algorithm Design, Data Mining.

Shobha G. is Professor in Computer Science and Engineering Department and associated R.V. College of Engineering, Bengaluru, since 1995. She has received her master’s degree from BITS, Pilani and Ph.D (CSE) from Mangalore University. Her research areas of interest are database management systems, data mining, data warehousing, image processing and information and network security.