International Journal of Applied Information Systems (IJAIS) ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12 - No. 8, November 2017 - www.ijais.org

Privacy Preserving Informative Association Rule Mining

Kshitij Pathak
Department of Computer Science and Engineering
Rajiv Gandhi Technical University, Bhopal (M.P.), India

Sanjay Silakari
Professor & Dean
Department of Computer Science and Engineering
Rajiv Gandhi Technical University, Bhopal (M.P.), India

Narendra S. Chaudhari
Dean, Research and Development & Professor
Department of Computer Science & Engineering
Indian Institute of Technology, Indore (M.P.) India

ABSTRACT

Privacy preserving data mining has two major directions: one is the protection of private data, i.e., data hiding in the database whereas another one is the protection of sensitive rule (Knowledge) contained in data known as knowledge hiding in the database. This research work focuses on protection of sensitive association rule. Corporation individual & other may get mutual benefit by sharing their data, but at the same time, they would like to be sure that their sensitive data remains private or not disclosed, i.e., hiding sensitive association rules. Approaches need to be given sensitive association rule in advance to hide them, i.e., mining is repaired. However, for some application pre-process of these sensitive association rules is combined with hiding process when predictive items are given, i.e., hiding informative association rule set. In this work, we propose two algorithms ISLFASTPREDICTIVE, DSRFASTPREDICTIVE to hide informative association rule with n-items. Earlier work hided 2-item association rules. Algorithms proposed in the paper execute faster than ISL & DSR algorithms prepared earlier as well as a side effect have been reduced. ISLFASTPREDICTIVE and DSRFASTPREDICTIVE algorithms work better as database scans are reduced since transaction list of elements is used in algorithms, i.e., a list of the transaction which supports itemsets and selection of transactions are done on the basis of presence of frequent itemsets.

Keywords

Informative Association Rules, Knowledge Hiding in Database, Frequent Itemset, Privacy-Preserving Data Mining, Sensitive Association Rules

1. INTRODUCTION

Data Sharing can bring many benefits for business collaboration as well as research. However, owners like to hide their sensitive data/Information before sharing their database for mining.[13, 19, 34] reflects the requirement of preserving the privacy with shared databases. Benefits of data sharing come from the business world. For hiding sensitive data, various transformation methods have been discussed in [1, 2, 3, 4, 5, 7, 23]. Hiding sensitive knowledge, i.e., association rules were first discussed in [8, 10]. Approaches for hiding sensitive association rules falls into categories like data distortion [35, 24, 25, 29, 20, 36], Data Blocking [31], Border-Based [32, 12], Data Reconstruction [14, 9, 40, 15] and cryptography approaches [22, 42]. Performance is a major concern with hiding association rules [21, 26, 28, 27, 44]. [42] presents a novel approach to hide sensitive rules with limited side effects. [33] throws light on the development of techniques which are under the knowledge-hiding that relates to the association rule-mining task. [11] extends the work to spatial data. [41] describes the measures that can be used with association rule hiding. [43] hides rules by transactions adding or removing. [38] works on multiple tables.

2. INFORMATIVE ASSOCIATION RULE SETS

Association Rule Mining was introduced earlier in 1993. In association rule mining with market basket data, a set of items is defined as I= $\{I_1, I_2, I_3,..., I_n\}$. The itemsets of size one from these sets are called 1-itemsets, itemsets of size two are called 2-itemsets, and similarly, itemsets of size k is called k-itemsets. In market basket data, the database contains transactions where each transaction represents a set of items purchased in a particular transaction. An association rule is represented as A $\rightarrow$ B where A and B are itemsets which is a subset of I having support and confidence greater than user-specified support and user-specified confidence. For example, Pen$\rightarrow$Paper with support 80% and confidence 90% implies that Pen and Paper both are present in 80% of total transactions and 90% is the case wherever Pen is present, Paper is also present. So,

$$Support(A \rightarrow B) = \frac{Support(A \cap B)}{|D|} \tag{1}$$

$$Confidence(A \rightarrow B) = \frac{Support(A \cap B)}{Support(A)} \tag{2}$$

Support is used for removing an uninteresting rule as low support rule occur just by chance and confidence provides the reliability of an association rule. Consider Database D_1 shown in Table 1 with user-defined support threshold 55% and user-defined confidence threshold is 80%, eight association rules get generated as shown in Table 2.

Table 1. : Sample database D_1

Transaction_Id	Items
1	U,V,W,X
2	U,V,W
3	U,V,W
4	U,V,W,X
5	W
6	V
7	U,V,X,Y,Z

For hiding of sensitive association rules, first sensitive rules are selected from a list of rules generated and then applied to association rule hiding algorithm whereas in informative rule sets all

Table 2. : Association rules for sample databases D_1

S_No	LHS		RHS	Support	Confidence	Lift
1	{W}	→	{U}	0.57	0.8	1.12
2	{U}	→	{W}	0.57	0.8	1.12
3	{W}	→	{V}	0.57	0.8	0.93
4	{U}	→	{V}	0.71	1	1.16
5	{V}	→	{U}	0.71	0.83	1.16
6	{U,W}	→	{V}	0.57	1	1.16
7	{V,W}	→	{U}	0.57	1	1.4
8	{U,V}	→	{W}	0.57	0.8	1.12

association rules are not generated. Here only those association rules are hided which contains predicting items on left-hand side of the rule. So while hiding the rules, they are mined from the database containing predicting items on LHS. Let suppose predicting item is V then association rules having predicting item V on the LHS are $\{V\} \rightarrow \{U\}$, $\{V,W\} \rightarrow \{U\}$, $\{U,V\} \rightarrow \{W\}$. These three rules need to be hided. So the problem of hiding sensitive information association rule sets is defined as follows:
Given a transactional database 'D' with minimum support Threshold "MST" and minimum confidence threshold "MCT", sets of association rules and predicting item sets PI, then all the sensitive association rules are identified as X→ Y where X $\subseteq$ PI, and non sensitive rules are Z $\rightarrow$ Y where Z $\subsetneq$ PI, so sensitive rules need to be hided and non sensitive rules does not be affected as much as possible.
In [39], two algorithms are proposed to hide information rule sets but side effects can be reduced by selecting the candidate transaction as the one having least number of frequent itemsets belong to it as well as performance can be enhanced by using Transaction ID List of items.
In this paper, two algorithms are presented which are the enhancement of work done in [39, 37] to improve the performance as well as to reduce the side-effect.

3. PROPOSED ALGORITHMS

A sensitive association rule can be hidden by

—Reducing the support of rule by either decreasing the support of LHS or decreasing the support of RHS

—Reducing the Confidence of Rule by either increasing the support of LHS or decreasing the support of the rule.

This work presents two algorithms, viz. ISLFASTPREDICTIVE, DSRFASTPREDICTIVE. In ISLFASTPREDICTIVE and DSRFASTPREDICTIVE algorithms, the TIDList of items has been used which greatly improves the performance of the algorithm. The logic behind the ISLFASTPREDICTIVE algorithm is to hide sensitive association rules by reducing the support of rule containing predicting item by increasing the support of LHS of the rule. In DSRFASTPREDICTIVE algorithm, the logic is to hide sensitive association by reducing the support of RHS of rule till support of confidence of the rule falls below a threshold. Algorithms select candidate transactions to be modified on the basis of a number of frequent itemsets present in it in increasing order. The algorithms are shown in algorithm 1 and algorithm 2.

4. EXAMPLES

This section presents two examples which step by step represents the action of algorithms as well as highlights the benefit of the proposed approach. Examples shown below also examine the output of the algorithms presented in [39].
Example 1:
Consider the database D_1 and select the predicting item V and

```
input  : Database, Set of Predicting items
output : Modified database to hide Informative association
         rules
Find all frequent Itemsets (F_sets);
foreach Predicting item I ε Predictingitem do
    foreach item Y ⊆ I do
        if Y ⊈ F_sets then
            Predictingitem = Predictingitem- {I};
            Break;
        end
    end
end
foreach X ε Predictingitem do
    Compute confidence of rule AR where Conf(AR) ≥
    MinConf and AR is of form X → Y i.e. Predicting item
    is on L.H.S. ;
    foreach rule AR having Conf(AR) ≥ MinConf do
        LhsList = GenerateList(X) ;
        // List of transactions containing
           itemset x
        RhsList = GenerateList(Y) ;
        // List of transactions containing
           itemset y
        Rule = LhsList ∩ RhsList;
        NoofModificationRequired = (|Rule| * 100) / (MCT - |LhsList|) ;
        CandidateTransactionToBeModified = (T - RhsList)
        ∩ (T - LhsList);
        // List of transactions which does not
           support RHS and partially or no
           support for LHS
        if |CandidateTransactionToBeModified| <
        NoofModificationRequired then
            Print("ISLFASTPREDICTIVE will not work
            for hiding this rule");
        end
        else
            Sort(CandidateTransactionToBeModified,by =
            presence no of frequent item sets);
            for k ← 1 to NoofModificationRequired do
                Pick a Transaction T from
                CandidateTransactionToBeModified;
                Pick a item i from X with least presence in
                number of frequent item sets ;
                // Pick item from left-hand side
                   of sensitive association rule
                SetToOne(T,i);
            end
        end
    end
end
```

Algorithm 1: ISLFASTPREDICTIVE Algorithm

also the maximum size of an itemset is taken as two just to compare the result with the algorithms presented in [39]. DSR algorithm [39] hides the sensitive association rule V $\rightarrow$ U successfully but hides five nonsensitive rules as shown in Table 3. DSRFASTPREDICTIVE algorithm proposed in the paper successfully hides the sensitive rule V $\rightarrow$ U and also no ghost rule generated and no rule lost. ISL algorithm [39] and ISLFASTPREDICTIVE algorithm successfully hide the sensitive rule without any side-effects.
Example 2:
Consider the database D_2 shown in Table 4 with MST=55% and MCT=80%. Let the predicting item be "B" and considering maximum 2-itemset then sensitive rule B $\rightarrow$ A needs to be hided.

input : Database, Set of Predicting items
output: Modified database to hide Informative association rules

Find all frequent Itemsets (F_{sets});
foreach *Predicting item* $I \varepsilon$ Predictingitem **do**
  foreach *item* $Y \subseteq I$ **do**
    if $Y \nsubseteq F_{sets}$ **then**
      Predictingitem = Predictingitem- $\{I\}$;
      Break;
    end
  end
end
foreach $X \varepsilon$ Predictingitem **do**
  Compute confidence of rule AR where Conf$(AR) \geq$ MinConf and AR is of form X $\rightarrow$ Y i.e. Predicting item is on L.H.S. ;
  foreach *rule* AR *having* Conf$(AR) \geq MinConf$ **do**
    LhsList = GenerateList(X) ;
    // List of transactions containing itemset x
    RhsList = GenerateList(Y) ;
    // List of transactions containing itemset y
    Rule = LhsList $\cap$ RhsList;
    NoofModificationRequired = Min($|\text{Rule}| - \frac{\text{TotalNumberofTransaction} * MST}{100}$, $|\text{Rule}| - \frac{|\text{LhsList}| * MCT}{100}$) ;
    CandidateTransactionToBeModified = RhsList $\cap$ LhsList;
    // List of transactions which fully support RHS and LHS
    Sort(CandidateTransactionToBeModified,*by = presence no of frequent item sets*);
    for $k \leftarrow 1$ **to** NoofModificationRequired **do**
      Pick a Transaction T from CandidateTransactionToBeModified;
      Pick a item i from Y with least presence in number of frequent item sets ;
      // Pick item from right-hand side of sensitive association rule
      SetToZero(*T,i*);
    end
  end
end

Algorithm 2: DSRFASTPREDICTIVE Algorithm

Table 3. : Rule lost after the application of DSR algorithm on D_1

S_No	LHS		RHS	Support	Confidence	Lift
1	{W}	$\rightarrow$	{U}	0.57	0.8	1.12
2	{U}	$\rightarrow$	{W}	0.57	0.8	1.12
3	{V}	$\rightarrow$	{U}	0.71	0.83	1.16
4	{U,W}	$\rightarrow$	{V}	0.57	1	1.16
5	{V,W}	$\rightarrow$	{U}	0.57	1	1.4
6	{U,B}	$\rightarrow$	{W}	0.57	0.8	1.12

ISL algorithm [39] hides the sensitive association rule with one lost rule, i.e., one nonsensitive rule gets hided whereas proposed ISLFASTPREDICTIVE algorithm successfully hides sensitive association rule without any side-effect. DSR [39] and DSR-FASTPREDICTIVE algorithm both hides sensitive rule with five lost rules.
So, it is evident from both examples that it is better sometimes to select candidate transaction as the one which contains the least number of frequent itemsets. Approach gets performance improved by using Transaction ID List. In ISL and DSR algorithms lot of database scans are required to execute the algorithm, but in proposed approach, no database scans are needed in hiding part of algorithm since while mining the association rules, transactions list of frequent itemsets has been generated and will be used in later part of the algorithm.

Table 4. : Database D_2

Transaction_Id	Items
1	ABCD
2	ABC
3	ABC
4	ABCD
5	C
6	B
7	BCDEF

5. EXPERIMENTAL VALIDATION

We have performed performance evaluation experiments on a PC with a core-i3 processor with 3 GB RAM running on Ubuntu-16.04 Operating System and the language used for implementation is R Language [30] and package used for working in R is ”arules” [16, 18, 17]. The datasets used in the evaluation trials are generated using IBM synthetic data generator [6]. The database size employed in the data set range from 10K to 100K with average transaction length, ATL = 5, and a total number of items is 50 and number of predicting item is 2. The minimum support threshold picked is 4% & minimum confidence threshold picked is 20%. To evaluate the performance of the algorithms following effects are considered:

a) Time Effects.
b) Side Effects.

For Time Effects, we are considering the CPU time/ running time to run ISLFASTPREDICTIVE, DSRFASTPREDICTIVE to hide sensitive association rules (AR_H) selected from the set of Association Rules generated (AR) containing the predicting item on LHS. For Side Effects, we measured Rule Hiding Failure, Rule Falsely Generated (Ghost Rules) and Rules Falsely Hidden (Lost Rules). The Rule Hiding Failure Side Effect counts the number of sensitive association rules; algorithm fails to hide. Rule Falsely Generated (Ghost Rules) side effect counts the number of rules that were not available with the original dataset, but after the modifications performed by the algorithm, the Rule appears. The Rules Falsely hidden (Lost Rules) side effect counts the number of nonsensitive rules hided because of the data distortion process. ISLFASTPREDICTIVE algorithm is compared with ISL [39] concerning running time of the algorithm and various side effects. DSR algorithm [39] is compared with DSR-FASTPREDICTIVE concerning running time of the algorithm and various side effects. All the graphs plotted to represent the average of 10 iterations of experiments.
Fig. 1, 2, 3 and 4 accounts for the Hiding Failure, Lost Rule, Ghost Rule and CPU Time of ISLFASTPREDICTIVE and ISL against various database sizes ranging from 10K to 100K respectively. Fig. 1 represents the number of rules algorithms ISL and ISLFASTPREDICTIVE fail to hide. The graph is shown using percentage because since we performed experiments with different database size for ten iterations and every time two arbitrary predicting items are selected for which informative rules to be hided. So, the count of a number of a sensitive association rule is different each time and graph represent the average of 10 iterations. It is deceptive from the Fig. 1 that ISLFASTPREDICTIVE algorithm perform better in comparison to ISL concerning hiding failure side-effect. These algorithms sometimes fail to hide

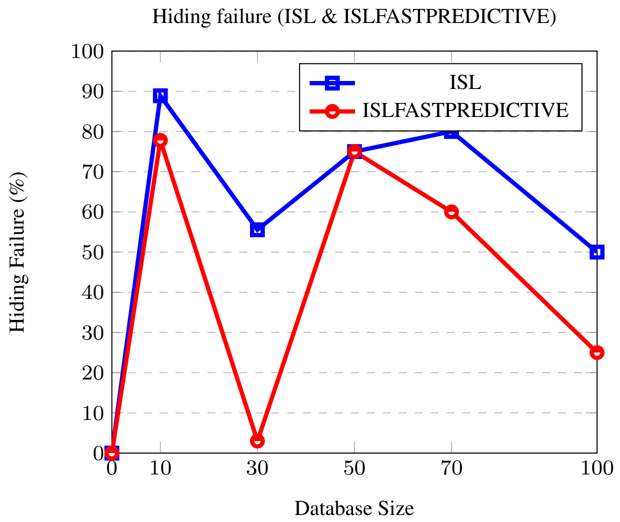

Fig. 1: Hiding failure (ISL & ISLFASTPREDICTIVE)

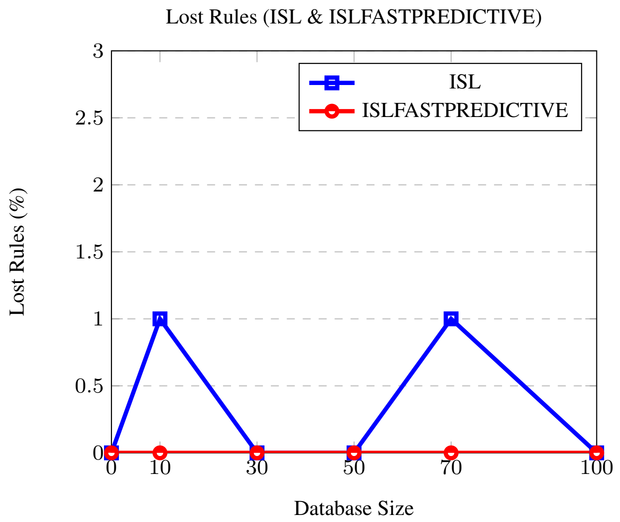

Fig. 2: Lost Rules (ISL & ISLFASTPREDICTIVE)

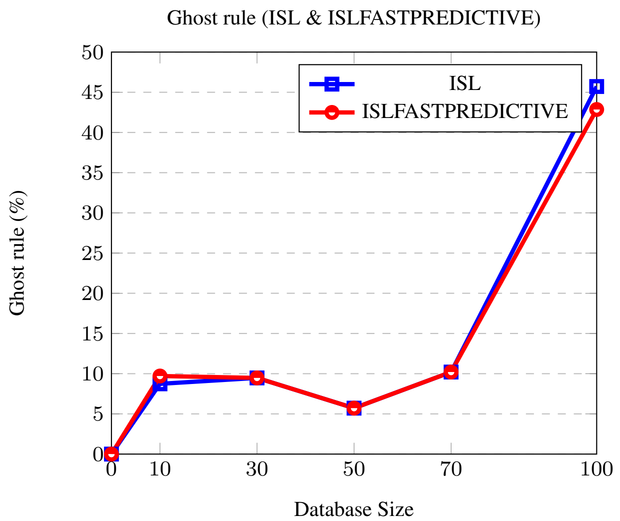

Fig. 3: Ghost rule (ISL & ISLFASTPREDICTIVE)

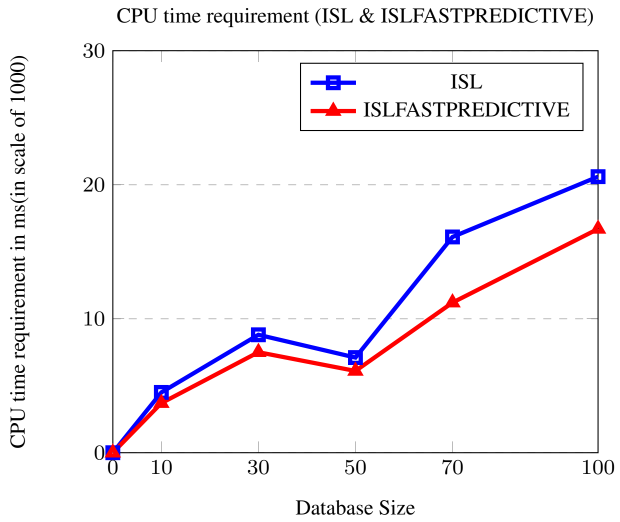

Fig. 4: CPU time requirement (ISL & ISLFASTPREDICTIVE)

all sensitive association rules. Fig. 2 represents the percentage of lost rules generated by taking an average of 10 iterations of experiments. From Fig. 2, it is evident that lost rules count is reduced since transactions selected for modifications are the ones which contain the least number of frequent itemsets, so lost rule count is reduced. Fig. 3 represents the percentage of ghost rules generated by taking an average of 10 iterations of experiments. As of Fig. 3, it is clear that the number of ghost rules is reduced by a small fraction, but no appreciable difference is identified. In general, it can be said both algorithms almost perform similarly for ghost rule side effect, results are better with hiding failure and lost rules count. Also, results with proposed approach can be better when the maximum transactions to be modified are of all the same length, and there are such transactions where a number of frequent itemsets presence is very less. Fig. 4 shows the comparison of running time with ISL and ISLFASTPREDICTIVE algorithm. As it is very clear from the Fig. 4, that ISLFASTPREDICTIVE algorithm takes less time as compared to ISL. ISL scans database multiples times which increases the time requirement of the algorithm, and it becomes too high as the database size increases. ISLFASTPREDICTIVE algorithm is based on the Transaction Id list of the itemsets which already gets generated during the mining of association rules, so there are no multiple scans of the database in ISLFASTPREDICTIVE algorithm. Hence, the performance is far better with ISLFASTPREDICTIVE algorithm.
DSRFASTPREDICTIVE algorithm is compared with DSR [39] concerning running time of the algorithm and various side effects. All the graphs plotted to represent the average of 10 iterations of experiments. Fig. 5, 6, 7 and 8 accounts for the Hiding Failure, Lost Rule, Ghost Rule and CPU Time of DSRFASTPREDICTIVE and DSR against various database sizes ranging from 10K to 100K respectively. Hiding Failure is 0 with both DSR and DSRFASTPREDICTIVE algorithm. Graphs suggest that DSRFASTPREDICTIVE algorithm performs better with both ghost rule and lost rule side-effect in comparison to DSR and the best result is with lost rule and running time of the algorithm.

6. SUMMARY OF COMPARISON BETWEEN ISL, DSR, ISLFASTPREDICTIVE & DSRFASTPREDICTIVE ALGORITHMS

(1) ISL algorithm start hiding process before checking the feasibility of approaches so many times if algorithm fails to hide certain rules, a lot of computation gets wasted, and the

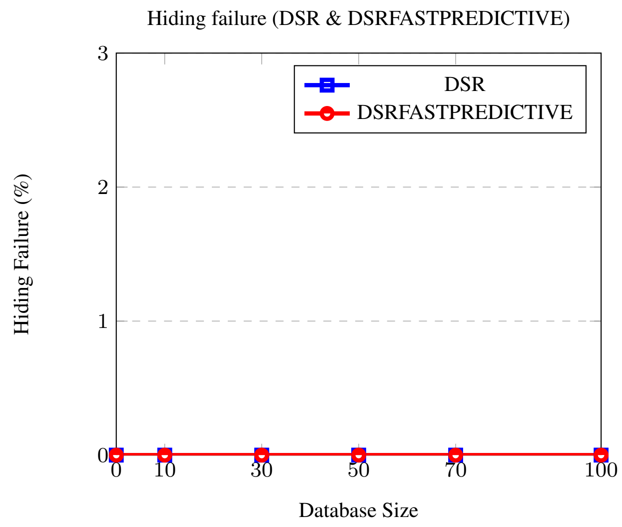

Fig. 5: Hiding failure (DSR & DSRFASTPREDICTIVE)

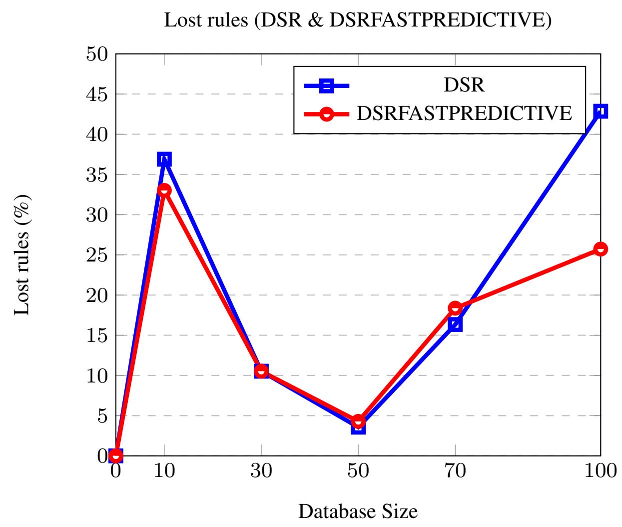

Fig. 6: Lost rules (DSR & DSRFASTPREDICTIVE)

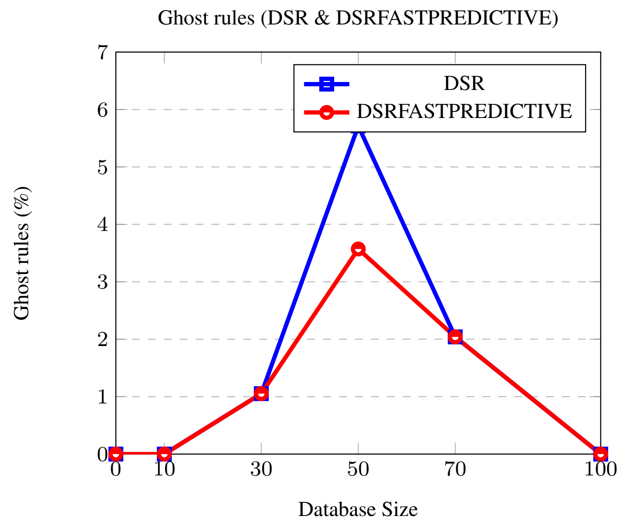

Fig. 7: Ghost rules (DSR & DSRFASTPREDICTIVE)

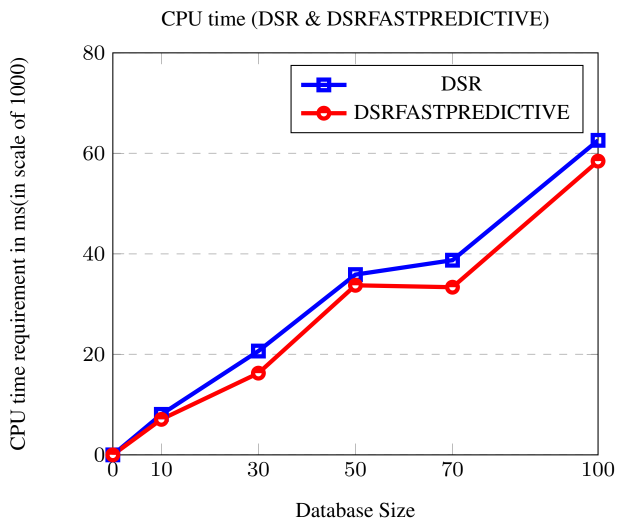

Fig. 8: CPU time requirement (DSR & DSRFASTPREDICTIVE)

database is modified and being rolled back as a side-effect. In proposed approach algorithm first checks whether the approach can hide a particular rule, i.e., the feasibility of the approach is verified first which helps in reducing the computation and rollback of modifications.

(2) ISL and DSR algorithms are designed in such a way that it can be used for hiding informative rules where length is two, but proposed approach takes any number of length of informative association rules.

(3) ISL and DSR algorithms give priority to transactions on the basis of length of transactions whereas the ISLFASTPREDICTIVE and DSRFASTPREDICTIVE algorithm give priority to transactions on the basis of a number of frequent itemsets present in it which helps in reducing the side effect. ISL and DSR algorithm running time is high because of multiple scans of database whereas ISLFASTPREDICTIVE and DSRFASTPREDICTIVE perform better since transactions list of frequent itemsets is utilized in hiding process.

7. CONCLUSION

This paper proposes two new algorithms based on transaction list of frequent itemsets prepared while mining of association rules containing predictive items. The experimental result shows the fruitfulness of the approach. Experiments suggest that proposed approach enhances ISL and DSR algorithm regarding running time but at the same time side effects have been reduced. After experimenting with a wide range of standard datasets as well as real datasets we have come to the conclusion that approaches performed much better when there are lots of transaction of the same length since previous approaches select transactions to be modified on the basis of length whereas proposed approaches modify on the basis of the count of frequent itemsets. This can be further optimized to generate much better results.

8. REFERENCES

[1] Nabil R Adam and John C Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.

[2] Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment, 2005.

[3] Charu C Aggarwal and Philip S Yu. On privacy-preservation of text and sparse binary data with sketches.

In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 57–67. SIAM, 2007.

[4] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255. ACM, 2001.

[5] Rakesh Agrawal, Roberto Bayardo, Christos Faloutsos, Gerald George Kiernan, Ralf Rantzau, and Ramakrishnan Srikant. Auditing compliance with a hippocratic database, October 5 2010. US Patent 7,810,142.

[6] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[7] Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. Privacy preserving olap. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 251–262. ACM, 2005.

[8] Mike Atallah, Elisa Bertino, Ahmed Elmagarmid, Mohamed Ibrahim, and Vassilios Verykios. Disclosure limitation of sensitive rules. In *Knowledge and Data Engineering Exchange, 1999.(KDEX'99) Proceedings. 1999 Workshop on*, pages 45–52. IEEE, 1999.

[9] Xia Chen, Maria Orlowska, and Xue Li. A new framework of privacy preserving data sharing. In *Proc. of the 4th IEEE ICDM Workshop: Privacy and Security Aspects of Data Mining. IEEE Computer Society*, pages 47–56, 2004.

[10] Chris Clifton and Don Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19, 1996.

[11] Qin Ding, Qiang Ding, and William Perrizo. Parman efficient algorithm to mine association rules from spatial data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(6):1513–1524, 2008.

[12] Aris Gkoulalas-Divanis and Vassilios S Verykios. Exact knowledge hiding through database extension. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):699–713, 2009.

[13] Michael Grean and Michael Shaw. Supply-chain partnership between p&g and wal-mart. *E-Business Management*, pages 155–171, 2002.

[14] Yuhong Guo. Reconstruction-based association rule hiding. In *Proceedings of SIGMOD2007 Ph. D. Workshop on Innovative Database Research*, volume 2007, pages 51–56, 2007.

[15] Yuhong Guo, Yuhai Tong, Shiwei Tang, and Dongqing Yang. A fp-tree-based method for inverse frequent set mining. In *British National Conference on Databases*, pages 152–163. Springer, 2006.

[16] Michael Hahsler, Christian Buchta, Bettina Gruen, and Kurt Hornik. *arules: Mining Association Rules and Frequent Itemsets*, 2017. R package version 1.5-2.

[17] Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta. The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets. *Journal of Machine Learning Research*, 12:1977–1981, 2011.

[18] Michael Hahsler, Bettina Gruen, and Kurt Hornik. arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, October 2005.

[19] Christopher T Heun. When to share: Wal-mart and other companies reassess their data-sharing strategies. *Information Week*, 2001.

[20] Dhyanendra Jain, Pallavi Khatri, Rishi Soni, and Brijesh Kumar Chaurasia. Hiding sensitive association rules without altering the support of sensitive item (s). *Advances in Computer Science and Information Technology. Networks and Communications*, pages 500–509, 2012.

[21] Dejiang Jin and Sotirios G Ziavras. A super-programming approach for mining association rules in parallel on pc clusters. *IEEE Transactions on Parallel and Distributed Systems*, 15(9):783–794, 2004.

[22] Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE transactions on knowledge and data engineering*, 16(9):1026–1037, 2004.

[23] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[24] Stanley RM Oliveira and Osmar R Zaiane. Privacy preserving frequent itemset mining. In *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, pages 43–54. Australian Computer Society, Inc., 2002.

[25] Stanley RM Oliveira and Osmar R Zaïane. Protecting sensitive knowledge by data sanitization. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 613–616. IEEE, 2003.

[26] Kshitij Pathak, Narendra S Chaudhari, and Aruna Tiwari. Privacy preserving association rule mining by introducing concept of impact factor. In *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, pages 1458–1461. IEEE, 2012.

[27] Kshitij Pathak, Aruna Tiwari, and Narendra S Chaudhari. Computational complexity of association rule hiding algorithms. *Evolution in Networks and Computer Communications*, (1).

[28] Kshitij Pathak, Aruna Tiwari, and Narendra S Chaudhari. A reduction of 3-sat problem from optimal sanitization in association rule hiding. In *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, pages 43–46. IEEE, 2011.

[29] Emmanuel Pontikakis, Achilleas Tsitsonis, and Vassilios Verykios. An experimental study of distortion-based techniques for association rule hiding. *Research Directions in Data and Applications Security XVIII*, pages 325–339, 2004.

[30] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[31] Yücel Saygin, Vassilios S Verykios, and Chris Clifton. Using unknowns to prevent discovery of association rules. *Acm Sigmod Record*, 30(4):45–54, 2001.

[32] Xingzhi Sun and Philip S Yu. A border-based approach for hiding sensitive frequent itemsets. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[33] Vassilios S Verykios. Association rule hiding methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):28–36, 2013.

[34] Vassilios S Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *ACM Sigmod Record*, 33(1):50–57, 2004.

[35] Vassilios S Verykios, Ahmed K Elmagarmid, Elisa Bertino, Yücel Saygin, and Elena Dasseni. Association rule hiding.

IEEE Transactions on knowledge and data engineering, 16(4):434–447, 2004.

[36] En Tzu Wang, Guanling Lee, and Yu Tzu Lin. A novel method for protecting sensitive knowledge in association rules mining. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 1, pages 511–516. IEEE, 2005.

[37] Shyue-Liang Wang. Maintenance of sanitizing informative association rules. *Expert Systems with Applications*, 36(2):4006–4012, 2009.

[38] Shyue-Liang Wang, Tzung-Pei Hong, Yu-Chuan Tsai, and Hung-Yu Kao. Multi-table association rules hiding. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pages 1298–1302. IEEE, 2010.

[39] Shyue-Liang Wang, Bhavesh Parikh, and Ayat Jafari. Hiding informative association rule sets. *Expert Systems with Applications*, 33(2):316–323, 2007.

[40] Yongge Wang and Xintao Wu. Approximate inverse frequent itemset mining: Privacy, complexity, and approximation. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[41] Junjie Wu, Shiwei Zhu, Hui Xiong, Jian Chen, and Jianming Zhu. Adapting the right measures for pattern discovery: A unified view. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1203–1214, 2012.

[42] Yi-Hung Wu, Chia-Ming Chiang, and Arbee LP Chen. Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data engineering*, 19(1), 2007.

[43] Xiaoming Zhang. Knowledge hiding in data mining by transaction adding and removing. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, volume 1, pages 233–240. IEEE, 2007.

[44] Cheng Zheng. An incremental updating technique for mining indirect association rules. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 1, pages 217–221. IEEE, 2008.