International Journal of Intelligent Computing and Information Science

A Novel Approach for Hiding Sensitive Association Rules using Data Perturbation and Query Restriction Strategy in Recommendation Systems

Reham Kamal
Ain Shams UniversityCairo, Egypt
reham.m.kamal@gmail.com

Wedad Hussein
Ain Shams University Cairo, Egypt
wedad.hussein@fcis.asu.edu.eg

Rasha Ismail
Ain Shams University Cairo, Egypt
rashaismail@cis.asu.edu.eg

Abstract

Mining association rules is considered to be core topic of data mining. Discovering these associations is beneficial and is highly needed to the correct and appropriate decision made by decision makers in the different fields. Association rule Mining imposes threats to data sharing, since it may disclose patterns and various kinds of sensitive knowledge that are difficult to find. Such information must be protected against unauthorized access. In this paper, we used DPQR strategy (data perturbation and query restriction) to hide the sensitive patterns. Experimental results showed that our proposed system can hide sensitive rules with multiple items in consequent (right hand side (R.H.S)) and antecedent (left hand side (L.H.S)) with efficient and faster performance compared to MDSRRC (Modified Decrease Support of R.H.S. items of Rule Cluster) with average improvement 96.22 % as well as generating accurate recommendations without revealing sensitive information.

Keywords *Data mining, recommender system, privacy.*

1. Introduction

A huge amount of data is produced by organizations; nevertheless, most of these organizations are faced with poverty of knowledge. Thanks to data mining tools, hidden knowledge in the data can be extracted. Nowadays, data mining has wide applications in various fields such as marketing, medical analysis, and business. Knowledge extracted with data mining tools assist individuals and organizations in taking better decisions and improve business processes. Association rule mining is one of the most widely used data mining tools which extracts dependency patterns from large databases. An association rule represent the links between items in the database [1].

Lately, more researches in data mining emphasize the seriousness of the problems about privacy. Privacy issues in data mining cannot simply be addressed by restricting data collection or even by restricting the use of information technology. The main problem faced is the need to balance the confidentiality of the disclosed data to authorized users. Privacy preserving data mining (PPDM) comes up with the idea of protecting sensitive data or knowledge to preserve privacy while data mining techniques can still be applied efficiently [2] There have been two types of privacy concerning data mining [3], [4]: data privacy, and information privacy. In data privacy, the database is modified in order to protect sensitive data of individuals. While in information privacy (e.g. Clustering or association rule), the modification is done to preserve the sensitivity of knowledge that can be mined from the database. To express it in another way, data privacy relates to input privacy while information privacy relates to output privacy. To address this confidentiality and privacy preserving issue, the original database is sanitized in such a way that sensitive patterns are hidden. Association rule mining enables us to find the relationship between the items. Many companies disclose their database for the mutual benefit, but before doing so they got to ensure their private data is hidden.

Association rule mining consists of two stages: in the first stage, frequent item sets, by using association rule mining algorithms such as Apriori Algorithm, 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= {i1, i2, i3…in} as set of items, D as the Database of transactions and t as each transaction where t ⊂I. An association rule will be represented as $X \rightarrow Y$, so that $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \Phi$. We say the rule $X \rightarrow Y$ holds in the database D with confidence C if $|XUY|/|X| \geq C$. It can also be said that the rule $X \rightarrow Y$ has support S if $|XUY|/|D| \geq S$. Note while the support is a measure of the frequency of a rule, the confidence is a measure of the strength of the relation between sets of items.

Association rule hiding problem aims to prevent some of these rules, which is referred as "sensitive rules", from being mined. Given a data set D to be released, a set of rules R mined from D, and a set of Sensitive Rules SR⊆R to be hided, how can we get a new data set D‟, such that the rules in SR cannot be mined from D‟, while the rules in R- HSR can still be mined as many as possible.

Our proposed framework is based on hiding sensitive association rules by using DPQR strategy to hide sensitive association rules by applying the DPQR strategy only to those items contained in the sensitive rules in one iteration which obviously reduced the run time and generated accurate recommendation results.

The remainder of the paper is organized as follows: the related work is presented in section 2. Our proposed framework is explained in section 3. In section 4, experimental data and results are shown. Conclusion and future work are discussed in section 5.

2 Related work:

Association rule hiding approaches can be categorized into: heuristic, border, exact, or reconstruction (reform) based algorithms, hybrid approaches.

Heuristic approaches use trials for modifications in the database. These techniques are efficient, scalable and fast, however they do not give optimal solution and also are CPU-intensive and require various scans depending on the number of association rules to be hidden [4].
Border based approaches track the border of the non-sensitive frequent item sets and greedily apply data modification that may have minimal impact on the quality of the border to accommodate the hiding sensitive rules. These approaches outperform the heuristic one and cause substantially less distortion to the original database to facilitate the hiding of the sensitive knowledge. Yet, in many cases these approaches are unable to identify optimal hiding solutions, although such solutions may exist for the problem at hand [5].
Exact approaches are considered as non-heuristic algorithms which consider the hiding process as a constraint satisfaction problem that may be solved using linear programming. These approaches provide a better solution compared to other approaches and can provide optimal hiding solution with ideally no side effects, but they suffer from high degree of difficulty and complexity [5].
Finally, **reconstruction based approaches** conceal the sensitive rules by sanitizing the itemset lattice rather than sanitizing the original dataset. Compared with the original dataset, itemset lattice is a medium production that is closer to association rules. These types of approaches generate less side effects in database than heuristic approaches. Despite its benefits, sanitization of the new database from scratch becomes impractical and this should be avoided.

As for the heuristic approaches, The **ADSRCC** (Advanced Decrease Support of R.H.S. items of Rule Cluster) and **RRLR** (Remove and Reinsert L.H.S. of Rule) algorithms, used for hiding sensitive rules, were introduced by Komal Shah et al in [6]. Both algorithms are considered to be heuristic and were presented to overcome drawbacks of existing rule hiding algorithm DSRRC (Decrease Support of R.H.S. items of Rule Cluster) like: (i) it does not maintain data quality (ii) execution time is increased due to database sorting after each modification. Algorithm ADSRRC overcomes limitation of multiple sorting in database as it selects transaction to be modified based on different criteria than DSRRC algorithm. Algorithm RRLR overcomes limitation of hiding rules having multiple R.H.S. items. RRLR algorithm performs more efficiently than ADSRCC.
Damandiya et al [7] attempted to overcome the drawbacks of ADSRCC and developed a novel heuristic technique named **MDSRRC** (Modified Decrease Support of R.H.S. items of Rule Cluster) algorithm that depend on the support reduction technique to hide the sensitive association rules with multiple items in (R.H.S) and (L.H.S). At first, sensitivity of items in rules' RHS calculated and then the most sensitive item will be selected to delete. MDSRRC, in comparison with ADSRRC, reduces database modification and side effects with deleting the effective candidate item. But with limitations in time efficiency as it sort database transaction after each modification and side effects in lost and ghost rules.

All the above related work needs multiple database scans to hide the sensitive rules which causes complex run time and heavy calculations.

A for the border-based approach, two algorithms in [8] rely on the max-min criterion for hiding of sensitive items. Both algorithms apply the idea of the max-min criterion in order to minimize the impact of

the hiding process to the revised positive border which is produced by removing the sensitive item sets and their super item sets from the lattice of frequent item sets, by restricting the impact on the border. However, border based approaches were usually useful to hide the sensitive items in frequent itemset only

Regarding exact approaches, authors in [9] proposed the first exact methodology to perform sensitive frequent itemset hiding based on the notion of a hybrid database generation.

Menon et al. [10] introduced a scheme that consists of exact and heuristic parts for the hiding of sensitive frequent patterns. The exact part formulated the problem as a constraint satisfaction problem (CSP) with a goal function of identifying the minimum number of transactions that need to be sanitized to complete the hiding process of all sensitive patterns. To reduce the NP-hardness problem, the authors reduce the problem size considering only the sensitive itemsets, requesting that their support remains under minimum support threshold. Moreover the constraints imposed in the CSP formulation capture the number of transactions that need to be sanitized for the hiding of each itemset. An integer programming solver is then applied to find the best solution to the CSP and derive the objective. In turn this objective is provided as an input to a heuristic algorithm to identify the actual number of transactions and perform their sanitization.

Unfortunately, all the proposed work did not break the high time complexity due to integer programming complex calculations.

In reconstruction approaches, the work in [11] proposed a coarse Constraint-based Inverse Item set Lattice Mining procedure (CIILM) for hiding sensitive frequent item sets.

Authors of [12] introduced a hybrid system to hide sensitive association rules through hybridization between border and heuristic approaches. They applied the advantage of both border and heuristic methods to hide sensitive items. The border based solution was enhanced by two heuristic techniques. The first heuristic used max-min solutions to select victim items to hide, then the second heuristic removed victim items from transactions.

In the work of Narges Jamshidian in [13] a hybrid system based on two algorithms **ISSDD** (Intelligent Sanitize Selection in Dense Database) and **ISSSD** (Intelligent Sanitize Selection in Sparse Database) is proposed, the distortion techniques (with the approaches based on reduction of rule confidence (Confidence-based) and reduction of rule support (Support-based) were used. The work in [14] depends on the reduction of support and confidence of sensitive rules but without editing or disturbing the given database of transactions directly (as it is generally done in previous works) rather performing the same task indirectly by modifying the some new introduced terms(**Hiding counter**) associated with database transactions and association rules.

Bux,N,K et al. [15] suggested a Genetic Algorithm (GA) based scheme for hiding sensitive items that minimize side effects and iteration. The objective functions were performed recursively to reach enhanced time cost and control the side effects. The decrease the confidence rule was chosen to select the victim rules.

The following table summarize the hiding approaches:

Table 1 hiding approaches comparison

Approach		Advantages	Limitations
Heuristic Based [6][7][10]	Distortion technique	Efficiency, scalability and quick responses.	1. Produce undesirable side effects in new database (i.e. Lost rules and new rules). 2. Difficult to revert the changes made in database.
	Blocking technique	1. It maintains accuracy of database, since instead of inserting false value it just blocks original value. 2. Minimizes side effects.	1. Suffer from various side effects like ghost rule, lost rule etc. 2. Difficult to reproduce original dataset.
Border based [8]		1. Maintains data quality by greedily selecting the modification with minimal side effects. 2. Improvement over pure heuristic approach.	1. Unable to identify optimal hiding solution since Theory of border difficult to understand Based on heuristic approach.
Exact based [9][10]		Guarantees quality for hiding sensitive information than other approaches.	High complexity due to linear integer programming
Reconstruction based [11]		1. Create privacy aware database by exacting sensitive characteristic from the original database. 2. Has less side effects in database than heuristic approach.	1. Number of transaction is restricted in new released database.
Hybrid techniques [12][13][14]		Can provide better data private protection or better measures.	High complexity due to combining of two or more different techniques.
Genetic algorithm [15]		provide recursive computation to reduce the time	Hide one rule every run.

3 Methods:

Our proposed framework is divided into five modules as shown in Figure 0.1. The modules are: normalizing data and transforming to Boolean format, applying Apriori algorithms to generate all rules association rules, then applying perturbation strategy on sensitive rules, analyzing the database performance, then finally generating recommendation to users. Each module will be explained in the next sections.

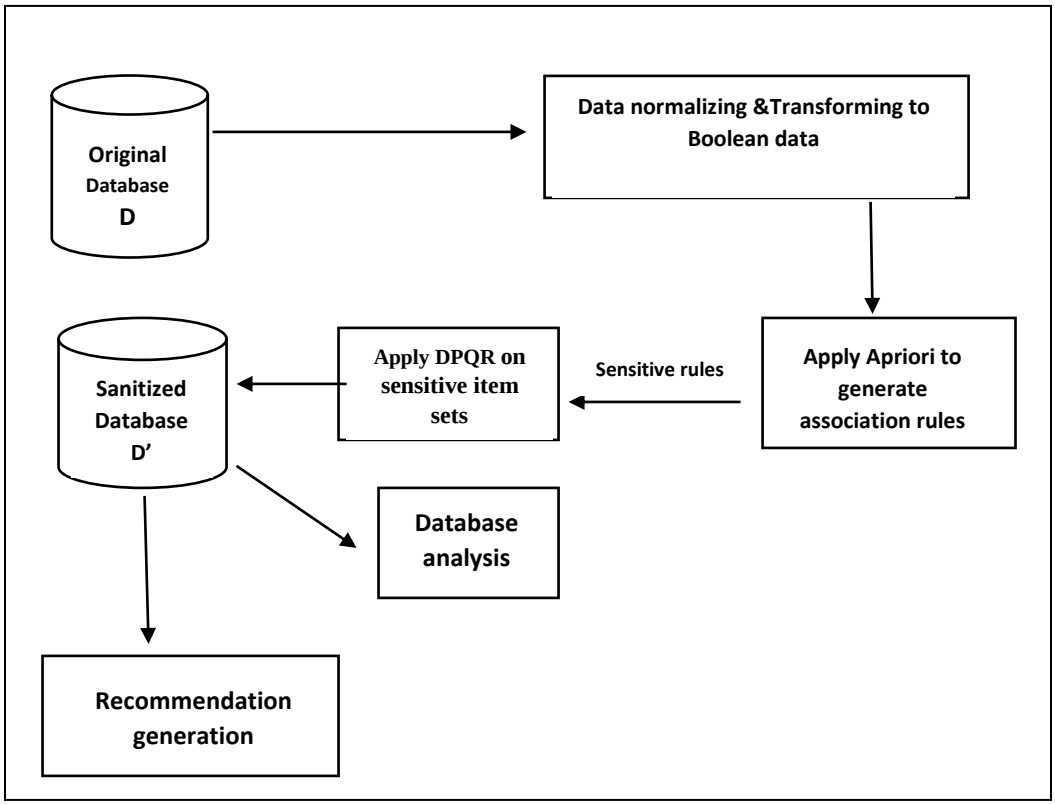

Figure 0.1 Proposed framework

3.1 Data normalizing and transforming to Boolean format:

The original database transactions are stored as a set of transactions each consisting of: UserId, MovieId, and Rating. We ignored the rating as it is not used in our approach and then normalized transforming the transactions to Boolean format as shown in

, where '1' shows the existence of the item in the transaction and '0' shows the non-existence.

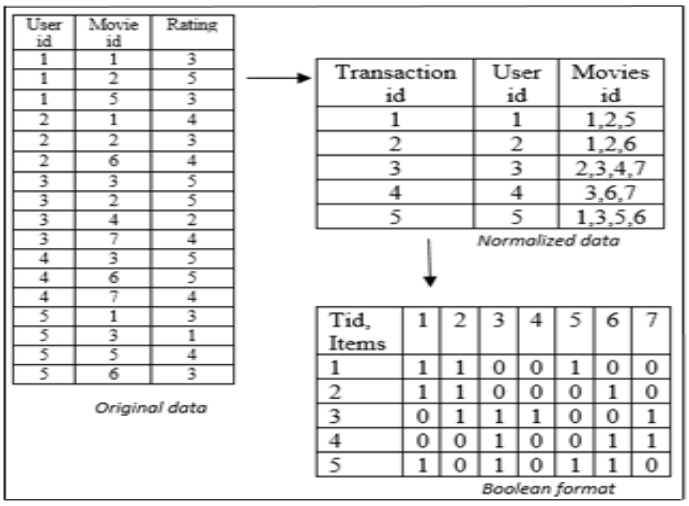

Figure 3 Data normalization

3.2 Apply Apriori to generate all association rules

In this module Apriori algorithm is applied on the normalized data to generate all possible association rules given Minimum support threshold (MST) and Minimum confidence threshold (MCT). For our example let us assume MST=2 and MCT= 0.6, the following 5 association rules will be generated:

1→2 - 1→5 - 3→7 - 1→6 - 3→6

Let's assume that the database owner specified the following rules to be sensitive rules:

1→6 - 3→6 - 3→7

3.3 Apply Data perturbation and query restriction strategy:

This module is the main contribution of this work, where the perturbation strategy is applied to obtain the sanitized database D' such that all sensitive rules are hidden and can be used to generate recommendations safely without privacy constraints.

In the proposed framework we used the algorithm in our previous work [16] that integrated two strategies (data perturbation and query restriction) to improve privacy-preserving degree. The algorithm uses three random parameters p1, p2 and p3 such that$0<p1, p2, p3<1$ and $p1+p2+p3=1$. Where p1 is the probability that data item kept the same, while p2 is the probability that the data item reversed, and p3 is the probability that the item is 0. The data perturbation strategy is realized by using parameters p1 and p2. When p3 is adopted, the data item is hidden and algorithm achieves the query restriction strategy. The pseudocode for the algorithm of data perturbation is shown below in fig 4. As can be seen from the figure we only apply DPQR on the sensitive item sets, in our example {1, 3, 6, and 7}. And here we have two different settings for the perturbation parameters, in case of LHS itemsets {1,3} we use high values for P1 as we desire to increase support of LHS, while in case of RHS {6,7} itemset we use higher value for P3 as we need to decrease the support of RHS.

```
Input: Sensitive itemset, p1p2, p3
Output: Sanitized database D'
Steps:
        For each transaction t ∈ D
        {
                For each item in sensitive itemset
                {
                Generate random number θ1;
                If (θ1≤p1)
                                i=i;
                Else if (p1≤θ1≤p1+p2)
                                i=1-i;
                Else if (p1+p2<θ1<p1+p2+p3)
                                i=0;
                }
        }
```

Figure 4 Pseudocode for data perturbation

3.4 Database analysis:

The performance of association rule hiding algorithm is commonly measured by three measures explained in [14] and given by equations 1, 2 and 3:

i. <u>Hiding Failure (HF):</u>

Measures the percentage of sensitive rules that that the system failed to hide and can be mined from the sanitized dataset

$$\text{Hiding failure} = \frac{|AR|sen(D')}{|AR|sen(D)} \qquad (1)$$

ii. <u>Artificial rules (AR):</u>

Measures the artificial association rules (ghost rules) that cannot be extracted from the original dataset but it can be extracted from sanitized dataset, which is created during the sanitization process due to the addition of noise in the data.

$$\text{Artificial rules} = \frac{|AR|(D') - |AR\ (D) \cap AR\ (D')|}{|AR|(D')} \qquad (2)$$

iii. <u>Lost Rules (LR):</u>

Measures the amount of non-sensitive association rules (lost rules) that have been hidden by the system after sanitization.

$$\text{Lost rules} = \frac{|AR\ Non{-}sen(D)| - |AR\ Non{-}sen(D')|}{|AR\ Non{-}sen(D)|} \qquad (3)$$

Where |AR|(D') and |AR| (D) are the number of association rules in sanitized database and original database respectively.

3.5 Recommendation generation

This phase consists of two modules: re-estimating the support of itemsets then measuring the cosine similarity between items to generate recommendations.

Re-estimating the support of itemsets: Since data set was normalized to Boolean format. The real dataset can be shown as a Boolean matrix T, and D is the disturbed dataset matrix by using the three parameters p1, p2 and p3. The number of '1' and '0' included in the ith column of T is defined as C_1^T, C_0^T respectively. While C_1^D and C_0^D have the similar definition. Let M be the transformation matrix,

$$C_D = M\, C_T \qquad (4)$$

Where $C_D = \begin{pmatrix} C_1^D \\ C_0^D \end{pmatrix}$, $C_T = \begin{pmatrix} C_1^T \\ C_0^T \end{pmatrix}$ and $M = \begin{bmatrix} p1 & p2 \\ p2+p3 & p1+p3 \end{bmatrix}$.

Given the following equation:

$$C_T = M^{-1}\, C_D \qquad (5)$$

The support of 1-item set C_1^T can be obtained by the following formula:

$$C_1^T = \frac{C_1^D - p2(C_1^D + C_0^D)}{p1-p2} \qquad (6)$$

As all items are disturbed in the same way, we can extend the formula to estimate the n-itemset support, the only difference is that M is $2^n * 2^n$ Matrix, C^T and C^D are both the $2^n * 1$ matrixes as follows:

$$C^T = \begin{pmatrix} c_{2^n-1}^T \\ \vdots \\ c_1^T \\ c_0^T \end{pmatrix}, \quad C^D = \begin{pmatrix} c_{2^n-1}^D \\ \vdots \\ c_1^D \\ c_0^D \end{pmatrix} \qquad (7)$$

M_{2^n}= [mij] is a matrix of order 2n, When matrix is invertible, $M_{2^n}^{-1}$ = [bij]. From equation (5), $c_{2^n-1}^T$ the support of n-item set can be calculated as follows:

$$c_{2^n-1}^T = b_{0,2^n-1} C_0^D + b_{0,2^n-2} C_1^D + \ldots\ldots + b_{0,0} c_{2^n-1}^D \qquad (8)$$

The calculation of matrix invert is optimized using recursive relations and the counting overhead of item sets is decreased using set theory as shown in [17], the complete Pseudocode of mining sanitized data is shown in fig 5.

Input: the sanitized database D, the parameters p1, p1, p3 and min sup.
Output: the frequent itemsets after reconstruction

```
        C1 = find _ candidate _ itemset (D);
Reconstruct (C1,min_ sup);              // estimate the real support of the 1-itemset
        L1 = {c ∈C1 | c.sup ≥□min_ sup};
 For (k=2; kn-1 ≠0; k++)
       {
                Ck=Apriori_gen (Lk-1)              // generate candidate itemset
               Reconstruct (Ck, min_ sup);          // reconstruct real support of the
          itemset
                 If (c.sup ≥□min_ sup)
                {
                  C ∈ Lk
                hashtab.add (c.count);
                 }

             Return L =∪k Lk
           }

Reconstruct (Ck, min_ sup)
       {
       if (k =1)
```

$$M_{2^1}^{-1} = \frac{1}{p1 - p2}\begin{pmatrix} 1-p2 & -p2 \\ p1-1 & p1 \end{pmatrix}$$

```
          Else
```

$$M_{2^k}^{-1} = \frac{1}{p1 - p2}\begin{pmatrix} 1-p2\ M_{2^{k-1}}^{-1} & -p2\ M_{2^{k-1}}^{-1} \\ p1-1\ M_{2^{k-1}}^{-1} & p1\ M_{2^{k-1}}^{-1} \end{pmatrix}$$

```
       Foreach ci in Ck
```

$$\{\ Ci.supp = \sum_{j=0} M[0][j] * CtransDB\ ;\}$$

```
       }
```

Figure 5 the complete Pseudocode of mining sanitized data

Measuring cosine similarity*:* Cosine similarity is measured between items to determine which items are most similar to each other according to the following equation:

$$Cos_{sim(A,B)} = \frac{supp\_count(A,B)}{\sqrt{Supp\_count(A)} * \sqrt{supp\_count(B)}} \qquad (9)$$

Choosing top-N items according to user history: Similar-items are arranged by descending order according to similarity (sim) count and support (sup) count. The result will be in the following format: Item: Item1, sim, sup; Item2, sim, sup; Item3, sim, sup…etc. Then for each user choose top-N items that are similar to his 1-items sets according to his history.

4 Results:

In this section, we test the performance of our framework by two steps, first we measure the effectiveness of the DPQR in hiding the sensitive rules using the measures mentioned above (equations1-3). Second we test the recommendation accuracy using Mean absolute error (MAE) as will be explained below.

For the evaluation MovieLens database is used. In this data set there are 2000 transaction across 50 movie.

4.1 Measure performance of the hiding algorithm:

Several experiments for the random parameters were performed to reach Hiding failure 0% which are shown below in Table 2, the best settings for perturbation parameters were found to be p1= 0.9 and 0.3, p2=0.0 and p3=0.1and 0.7.

Table 2 Performance measures

Number of transactions	Random parameter for RHS (p1=0.3, p2=0.0,p3=0.7)	Random parameter for LHS (p1=0.9,p2=0.0, p3=0.1)	Hiding failure	Artificial rule	Lost rule
50	1	0.4	0 %	0.12 %	28%
100	0.8	0.3	0%	0.25%	40%
1000	0.7	0.2	0%	0.30%	45%
2000	0.7	0.3	0%	0.50%	55%
Average for DPQR			0%	0.29 %	42 %
Average for MDSRRC			0 %	0 %	26.66 %

As can be seen from the above table that the average artificial rule is 0.29% and the average lost rule is 42% this is because we depend on random parameters to perturb our data that's why we conducted several experiments to reach 0% hiding failure which is our main target to hide all sensitive rules.

We compared our results with MDSRRC algorithm [18] and we found that our system run time is better that MDSRRC which is the main contribution of this paper as shown in Figure 0.2 and table 3. This huge reduction in runtime is because the MDSRRC algorithm needs multiple data base scan to hide all sensitive rules unlike our algorithm that needs to scan the data base only once.

Table 3 DPQR and MDSRRC runtime comparison

Number of transactions	DPQR run time in S	MDSRRC run time in S	Runtime improvement %
10	0.022	2.503	99.1 %
50	0.15	12.515	98.8%
100	0.4	25.03	98.4%
1000	6.5	70.08	90.7%
2000	10.5	180.3	94.1%
Average	3.5144	58.085	96.22 %

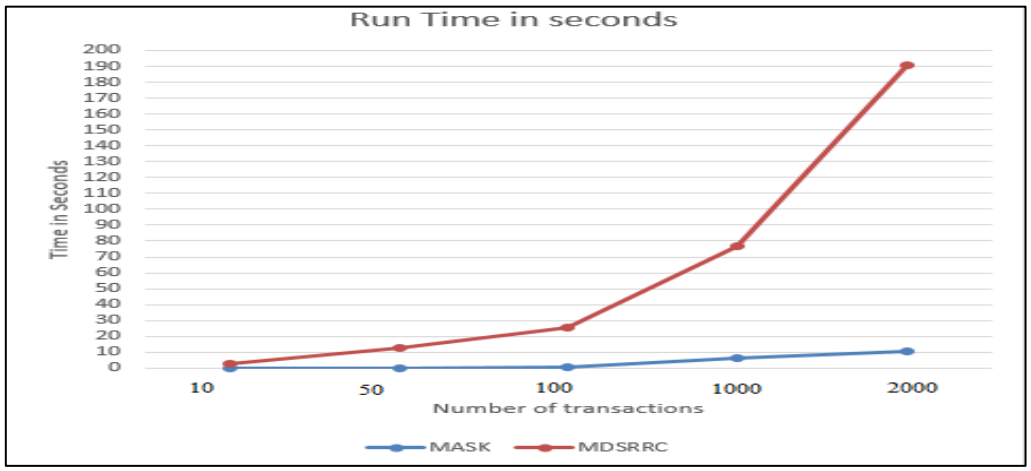

Figure 0.2 Run Time comparison

4.2 Testing the recommendation accuracy:

In this section we will measure the recommendation precision. For each user, counts the number of common itemsets generated from recommendation system before and after the distortion process. Measuring formula for precision is as (10) where A, B respectively represents privacy preserving recommendation system and traditional recommendation system.

$$Recommendation\ precision = \frac{same\_num(A,B)}{Total\ Number\ of\ recommended\ items} x100 \quad (10)$$

The values of recommendation precision for different number of users and compared to our previous work [16] is shown below in Figure 0.3

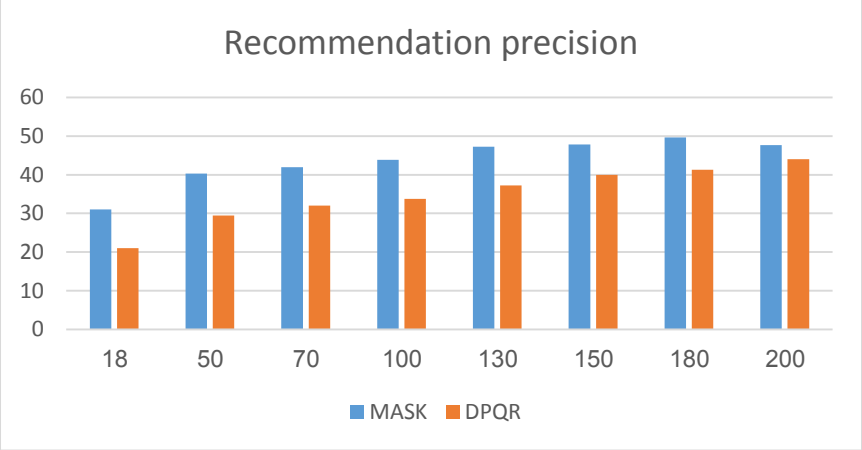

Figure 0.3 recommendation precision

As our artificial rule (AR) and lost rule (LR) measures are approximately 0.25% of the original rules this causes the accuracy of recommendations to decrease than our previous work, but with guarantee that no sensitive rules are revealed since our hiding failure (HF) is 0%.

5 Conclusion and future work:

In this paper, we used a novel approach to hide sensitive association rules by applying DPQR strategy using three random perturbation parameters (p1, p2, p3) to perturb original data then with the help of probability statistical method mining results are obtained. The performance of our system is evaluated using hiding failure, artificial rule and lost rule measures and showed efficient results. We compared our algorithm run time with MDSRRC run time and it was proven that our system is better in run time. Also our system is able to generate reliable recommendation results. The main drawback of this system is that the lost rule and artificial rule measures need more improvement which will be noticed in the future work.

6 Acknowledgments

I deeply thank God for providing me the strength and patience all the way and inspiring me from the beginning of my work. I want to express my gratitude to my professors for guiding me through every step in our work. Finally, I thank my husband and family. I wouldn't accomplish anything without their support and love. They are the bless that pushes me throughout every stage in my life.

References

[1] Zahra Kiani Abari1 , Mohammad Naderi Dehkordi, " Privacy Preserving in Association Rule Mining" *ACSIJ Advances in Computer Science: an International Journal,,* vol. Vol. 4, no. 1, No.13 , January 2015.

[2] Mohnish Patel, Aasif Hasan , Sushil Kumar, "A survey Preventing discovering association rules for large data base.," *International Journal of Scientific Research in Computer Science and Engineering, ,* vol. vol. 1, no. issue 3, pp. pp. 35-38, 2013.

[3] Sarra Gacem, Djamila Mokeddem and Hafida Belbachir, "Privacy preserving data mining: Case of association rule" *International Journal of Computer Science Issues, ,* Vols. vol. 10,, no. issue 3, no. 1,, pp. 91-96 , May 2013.

[4] K. Sathiyapriya, G. Sadasivam, " A survey on privacy preserving association rule mining", *International Journal of Data Mining & Knowledge Management Process,,* Vols. vol. 3, , no. no. 2,, pp. 119-131, 2013.

[5] Mohamed Refaat Abdellah, Khalid Shafee Badran, M. Badr Senousy, "Privacy Preserving Association Rule Hiding Techniques: Current Research Challenges," *International Journal of Computer Applications (0975 – 8887),* Vols. Volume 136 – No.6, , February 2016.

[6] Komal Shah, Amit Thakkar, Amit Ganatra, " Association Rule Hiding by Heuristic Approach to Reduce Side Effects & Hide Multiple R.H.S", International Journal of Computer Applications, 2012.

[7] Nikunj Domadiya, Udai Pratap Rao, "Hiding sensitive association rules to maintain privacy and data quality in database", 3rd IEEE International Advance Computing Conference (IACC),2013.

[8] V. M. a. V. S. Verykios, ". A max min approach for hiding frequent item sets.," *In Workshops Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), pages 502–506, 2006.*

[9] George V. Moustakides, Vassilios S. Verykios ". A max min approach for hiding frequent item sets.," *In Workshops Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), pages 502–506, 2006.*

[10] Syam Menon, Sumit Sarkar, Shibnath Mukherjee, "Maximizing accuaracy of shared databases when concealing sensitive patterns" in *" Information systems reseach 16(3), pp. 256-270, 2005.*

[11] Yuhong Guo, " Reconstruction-Based Association Rule Hiding", *Proceedings of SIGMOD2007 Ph.D Workshop on Innovative Database Research 2007(IDAR2007), Beijing, China, 2007.*

[12] Telikani, A. & Shahbahrami, A, "Optimizing association rule hiding using combination of border and heuristic approaches," Vols. Volume 47, Issue 2, pp 544–557., 2017.

[13] Narges Jamshidian Ghalehsefidi, Mohammad Naderi Dehkordi, ""A Hybrid Algorithm based on Heuristic

Method to Preserve Privacy in Association Rule Minin", 2016.

[14] Ramesh Chandra Belwal , Jitendra Varshney , Sohel Ahmed Khan , Anand Sharma , Mahua Bhattacharya," Hiding sensitive association rules efficiently by introducing new variable hiding counter", 2013.

[15] Bux,N,K., Lu,M., Wang,J., Hussain,S, Aljeroudi,Y "Efficient Association Rules Hiding Using Genetic Algorithms.," *Symmetry ,* Vols. 10(11), pp.576., 2018.

[16] Reham M.Kamal, Wedad Hussien, Rasha Ismail "Privacy preserving recommender system based on improved MASK and query restriction," *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, pp. 310-314, 2017.*

[17] *Haoliang Lou ; Yunlong Ma ; Feng Zhang ; Min Liu ; Weiming Shen*, "Data Mining for Privacy Preserving Association Rules Based on Improved MASK Algorithm,," in *Proceedings of the IEEE 18th International Conference on Computer Supported Cooperative Work in Design*, 2014.

[18] Dipak Gaikwad, Ajay Mahadik, Pralhad Jadhav, Sangram Kakade "Hiding Of Sensitive Association Rules With MDSRRC Algorithm for Preserving Privacy In Database," *International Journal of Networks and Systems,* pp. Volume 4, No.3, April - May 2015.

[19] G. Lee and Y. C. Chen "Protecting sensitive knowledge in association patterns mining,", *Data Mining and Knowledge Discovery,* Vols. vol. 2, pp. 60- 68, , no. issue 1, Jan.-Feb. 2012.