Design of an Augmented and Optimised Hiding Technique and Framework for Privacy Preserving Data Mining

J.Indumathi and G.V.Uma

Abstract— Information era has eye witnessed an implausible explosion of a waft similar to a typhoon of techniques for data garnering, data dissemination, internet technologies and the manifestation of susceptible applications; privacy and security issues in knowledge discovery have reached the pinnacle opening a new sphere of influence of issue connecting to privacy of populace which requires solemn judgment. Hence forth we are pressurized to develop mechanism for altering the unique facts by some means, with the intention that the private data and private knowledge linger private even subsequent to the mining process. There are many mechanisms which have been adopted for privacy preserving data mining. Pinning our attention to the earlier works done for association rule hiding by Aris Gkoulalas et al., is based on the concept of distance flanked by the original database and its sanitized version, where all sensitive rules have been hidden. By quantifying distance, knowledge is gained with minimum modification that needs to be made in the original dataset in order to hide sensitive, while austerely affecting nonsensitive, itemsets.

In this paper, we have endeavored to enhance the existing concealment technique and to develop a conceptual framework with the objective of implementing Privacy Preservation using the masking/concealing/hiding technique. They have used the Apriori algorithm to compute the large itemsets, which is less efficient and doesn't minimize side effect generated by it. We portray an efficient and optimal algorithm. Given a sensitive frequent itemset, for all the transactions containing this itemset, algorithm first identifies the transaction with the shortest length. In such a transaction, the candidate item with the maximal support value is deleted to decrease the support of the sensitive itemset. This sort of an approach hides the frequent sensitive itemsets efficiently and also it hides the non-sensitive itemsets. In this paper, we have tried to overcome the side-effect; along with harnessing the advantages of Frequent Pattern Growth Method which mines the complete set of frequent itemsets without candidate generation. We establish that any kind of Data Mining can be done securely with this algorithm and architecture without sacrificing accuracy. The investigational appraisal shows that this modus operandi yields good results on real world datasets, demonstrating its effectiveness towards solving the problem with good data utility, privacy and performance. Concisely stated it endeavours to disclose self-assurance amid privacy and revelation of information by attempting to minimize the impact on the concealed transactions.

Index Terms — **Association rule mining, Concealing,Frequent Itemset, integer programming, Masking,Privacy preserving data mining, sensitive itemset hiding, optimization.**

Manuscript received December 31, 2007

Indumathi.J received her M.E. from Anna University, Chennai, India in year 1992 and M.B.A from Madurai Kamaraj University, Madurai, India in 1994. She is working for Anna University as a Senior Lecturer. (corresponding author to provide phone: 91-044-22432410; fax: 91-044-22432410; e-mail: indu@cs.annauniv.edu).

G.V.Uma, a Polymath received her M.E. from Bharathidasan University, India in year 1995 and Ph.D. from Anna University, Chennai, India in 2002. She is working for Anna University as a Assistant Professor. (e-mail: gvuma@annauniv.edu).

I. Introduction

In the contemporary information era, data is garnered, collated, bartered and sold. Privacy preserving data mining (PPDM) is a new born discipline whose desire is to authorize delivery transmits of respondent data while preserving respondent privacy. Many techniques have been defined that alter an original dataset into a protected dataset such that, we satisfy two points viz.,analysis on the original and protected datasets should yield similar results (data utility);information in the protected dataset is unlikely to be linkable to the particular respondent it originated from(data safety). Among the two types of Privacy Preserving Data Mining(as in figure 1) the first type of privacy is that the data is altered so that the mining result will preserve certain privacy and the second type of privacy is that the data is manipulated so that the mining result is not affected or minimally affected privacy

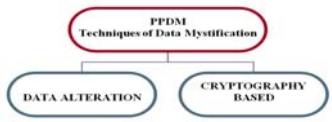

Figure 1. Classification of Privacy preserving data mining (PPDM) Techniques

II. LITERATURE SURVEY

In the terrain of privacy preserving data mining oodles of investigation has been carried out. Clifton et. al. in [6,8,9,3,4,7,14] examined the security and privacy implications of data mining and suggested some data obscuring strategies (aggregation, fuzzyfication, sampling,

augmentation) that can be functioned on the original dataset to proscribe deduction and discovery of sensitive information.

Verykios et. al. in [13,20] presented a categorization of the diverse privacy preserving techniques based on five dimensions: data distribution, data modification, data mining algorithm, data or rule hiding and privacy preservation. Most techniques are heuristic in character, first and foremost in order to hasten up the hiding process. However, heuristics suffer from local minima problems and usually fail to identify the optimal solution. Our approach is generally exact in nature and uses a heuristic only in the case that an optimal solution does not exist.[19]

The problem of hiding frequent itemsets (or association rules) was at the outset studied in [2] by Atallah et al. In this work, finding the optimal sanitization solution to hide sensitive frequent itemsets was proved as a NP-hard problem. Also, a heuristic-based solution was projected to eliminate sensitive frequent itemsets by deleting items from the transactions in the database. [20]

The subsequent work [1,11] extended the sanitization of sensitive frequent itemsets to the sanitization of association rules. The exertion provided some heuristics to prefer the items to be deleted, with the consideration of minimizing the side effect under the assumption that sensitive frequent itemsets were disjoint. The later work [21] further discussed the problem of hiding association rules by changing items to "unknown" instead of deleting them.

Moreover, considerable work [16, 18, 17, 28] has been done in this area by Oliveira and Zaiane. Their work determined on designing a variety of heuristics to minimize the side effect of hiding sensitive frequent itemsets. Particularly, in [6], the Item Grouping Algorithm (IGA) grouped sensitive association rules in clusters of rules sharing the same itemsets. The shared items were removed to reduce the impact on the result database. In [8], a sliding window was applied to scan a group of transactions at a time and sanitized the sensitive rules presented in such transactions. In recent work [9], they considered the attacks against sensitive knowledge and proposed a Downright Sanitizing Algorithm (DSA) to hide sensitive rules while blocking inference channels by selectively sanitizing their supersets and subsets at the same time.

Menon et. al. [27] proposed an integer programming optimization algorithm for hiding sensitive itemsets at the same time as minimizing the number of modified transactions. Finally, in [5], Sun and Yu proposed a greedy border-based approach which is based on the notion of the border constructed by the non-sensitive frequent itemsets in an attempt to track the impact of altering transactions,for hiding sensitive frequent itemsets. Instead of considering each non-sensitive itemset individually, their algorithm focuses on preserving the quality of the resulting border.

The work done by Aris Gkoulalas et al.,[2] for association rule hiding is based on the concept of distance flanked by the original database and its sanitized version, where all sensitive rules have been hidden. By quantifying distance, knowledge is gained with minimum modification that needs to be made in the original dataset in order to hide sensitive, while austerely affecting nonsensitive, itemsets. In this paper, we have endeavored to enhance an existing concealment technique in order to make safe susceptible knowledge from being uncovered in pattern mining. By hiding the sensitive frequent itemsets that lead to the production of the association rules, we are able to secure the sensitive knowledge and minimize the side effect on the quality of the sanitized database so that non-sensitive knowledge can still be mined. They have used the Apriori algorithm to compute the large itemsets, which is less efficient.

In this paper, we have used to harness the advantages of Frequent Pattern Growth Method which mines the complete set of frequent itemsets without candidate generation. The investigational appraisal shows that this modus operandi can yield good results on real world datasets, demonstrating its effectiveness towards solving the problem with good data utility, privacy and performance.

III. PROBLEM DESCRIPTION

A. Problem Statement

The goals of Privacy Preservation using the masking/concealing/hiding technique is to design, develop and implement functionalities like utility, accuracy, privacy Reusability, Portability etc., Specification of our modified method in order to compare and contrast it with the existing technique on a universal arena.

B. Problem Description

Given a database *D* containing transactions *T* and a minimum support threshold m_{sup} set by the owner of the data. A subset SI of the frequent itemsets F, discovered in D, is scrupulous as sensitive and ought to be secluded from being disclosed to unauthorized parties. Our aspiration is to *disinfect* preferred dealings from *D* that will proscribe the fabrication of rules from itemsets in *SI*, as a result generate a clean version D' of the original database, in which all these rules are *hidden*. Moreover, we want to ebb the cleansing impact to any non-sensitive itemsets.

Figure 3.1: A Hiding/Masking/Concealing Based Privacy Preserving Data Mining Systems: High-Level

Due to the finality property of the Apirori algorithm[15], it is unproblematic to discern that hiding the itemsets in *S* will certainly result in hiding all itemsets in *SI*. Moreover, hiding the itemsets in *S* results in hiding all itemsets in *SS*, meaning all sensitive itemsets and their proper supersets. This actuality may at a first glimpse, deceptively, seem superfluous but in veracity it is perfectly acceptable. In this paper, we have tried to overcome the side-effect; along with harnessing the advantages of Frequent Pattern Growth Method which mines the complete set of frequent itemsets without candidate generation. Given a sensitive frequent itemset, for all the transactions containing this itemset, algorithm first identifies the transaction with the shortest length. In such a transaction, the candidate item with the maximal support value is deleted to decrease the support of the sensitive itemset. This approach hides the frequent sensitive itemsets efficiently and also it hides the non-sensitive itemsets.

IV. ARCHITECTURE OF THE PROPOSED WORK

A. *Framework for Masking based PPDM*

Owing to the versatility of the Data Mining tasks, a family of privacy-preserving data mystification (PPDMy) methods for protecting privacy before data are shared can be used to the address privacy preservation in data mining as depicted in the figure 4.1. Here we have focused on the hiding technique for our alteration purposes. The input to this framework is unpreserved data whereas its output is privacy preserved befuddled data which is given as an input to the data mining process.

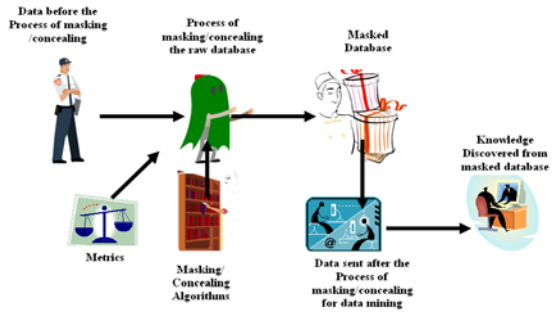

Figure 4.1. Framework for Hiding/Masking/Concealing based PPDM

B. *Block Diagram*

We bring out a diagrammatic schematic representation of the blocks as shown in figure 4.2.for the proposed architecture for concealment Technique for Privacy Preserving Data Mining as shown in figure 3. 1.We also explain the full process in detail.

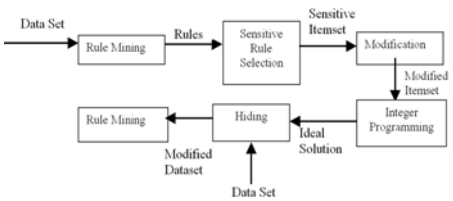

Figure 4.2. Block Diagram

C. *Rule Mining and Sensitive Rule Selection*

We use frequent pattern growth method to find the frequent itemsets, is used in this module from which the association rules can be mined. This method adopts a divide-conquer strategy.

D. *Positive and Negative Border Computation*

Based on the minimum support value (σ_{min}), an initial border can be attained which separates frequent from infrequent itemsets.

E. *Sanitizing the Frequent itemset*

After identifying the negative and positive border for the original database, the sensitive itemsets should be hided. In the hiding procedure we identify F be removing all sensitive itemsets and their supersets from F. We remove these from the list of large itemsets, thus construct a new set F.

F. *Integer Programming*

Following our formalization of the sensitive itemsets hiding modus operandi, the complete quandary can be regarded as a *Constraint Satisfaction Problem* (CSP) [11]. *CSP*s can be solved by using various techniques such as *Linear and Non-linear Programming* [10]. In our context all variables are *binary*; this fact provides us with an important advantage as we will see later on. To solve our *CSP* we use a technique called *Binary Integer Programming* (BIP) [9] that transforms the *CSP* to an optimization problem. Our formulation enables us to solve the sanitization problem in *D* and is capable of identifying the ideal solution (if one exists). In the case of problems where ideal solutions are infeasible we provide a relaxation of this algorithm (using a heuristic targeted for inequalities selection and removal) that allows identification of a good suboptimal solution [2].

V. IMPLEMENTATION

A. *PROPOSED ALGORITHMS*

FP_growth. Mine frequent patterns using an FP-tree by pattern fragment growth.

Input : A transaction database, D; minimum support threshold, min_sup.

Output : The complete set of frequent patterns.

Step 1. The FP-tree is constructed is the following steps

(a) Scan the transaction database D once. Collect the set of frequent items F and their supports. Sort F in support descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as “null”. For each transaction Trans in D do the following.

Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list. Call insert_tree([p|P],T), which is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N’s count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the

same item-name via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.

Step 2. Mining of an FP-tree is performed by calling **FP_growth** (FP_tree, null) , which is implemented as follows.

Procedure FP_growth (Tree, $\propto$)

(1) if Tree contains a single path P then
(2) for each combination (denoted as β) of the nodes in the path P
(3) generate patte4rn $\beta \cup \alpha$ with support = mining support of nodes in β;
(4) else for each a_i in the header of Tree {
(5) generate pattern $\beta = a_i \cup \alpha$ with support = a_i.support;
(6) construct β,s conditional pattern base and then β's conditional FP_tree $Tree_\beta$;
(7) if $Tree_\beta \neq 0$ then
(8) call FP_growth ($Tree_\beta$, β); }

Computation of the Positive Border $B^+(F)$

Procedure PB - COMPUTATION (F)

```
Count {0….| F | } ← 0                    //initialize counters
F_sort = reverse-sort (F)
for each k – item set f ∈ F_sort do
   for all (k – 1) – item sets q ∈ F_sort do
      if q ⊂ f then
         q.count ++
      end if
   end for
end for
for each f ∈ F_sort do
   if f .count = 0 then
      f ∈ B+ (F)              //add item set to B+ (F)
   end if
end for
end procedure
```

Hiding All Sensitive Itemsets and their Supersets

```
Procedure HIDESS (F, F', SI )
   for each s ∈ SI do          //for all sensitive itemsets
      for each f ∈ F do          //for all large itemsets
      if s ⊆ f then          // large itemset is sensitive
         F = F – f           // remove iteset f
      end if
      end for
   end for
end procedure
```

Relaxation Procedure in V

```
Procedure SELECTREMOVE (Constraints C_R , V , D)
   C_R maxlen ← ∪ argmax_i { |R_i| } //C_Ri ↔ V_i
   Cr_msup ← min C_Rmaxlen , i(σ_D ( R_i))    // R_i ∈ V
   for each c ∈ C_R maxlen do
   if σ_D ( R_i ) = cr_msup then
      Remove ( c )    //remove constraint from the CSP
   end if
   end for
end procedure
```

B. COMPARISON OF OUR PROPOSED ALGORITHM WITH THE EXISTING ALGORITHM

The existing algorithm [2] used to hide the sensitive item sets. In this algorithm, selecting a hiding candidate is straightforward. Given a sensitive frequent itemset, for all the transactions containing this itemset, algorithm first identifies the transaction with the shortest length. In such a transaction, the candidate item with the maximal support value is deleted to decrease the support of the sensitive itemset. This approach hides the frequent sensitive itemsets efficiently and also it hides the non-sensitive itemsets. This approach doesn't minimize side effect generated by it. This disadvantage is overcome in the new approach "An Integer Programming Approach for Frequent Itemset Hiding". Because in the new approach the distance between the original database and sanitized database is decreased and the sensitive itemsets are hided.

VI. RESULT AND ANALYSIS

Data Utility is the percentage of similarity between the data mined results from original data and concealed data.

A. Privacy Analysis

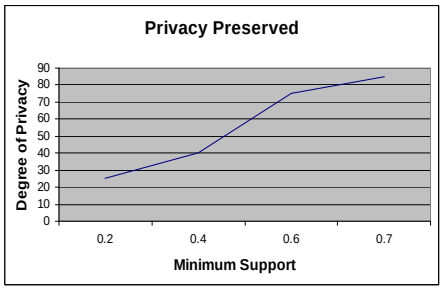

Figure 7.1. Degree of Privacy

Figure 7.1 shows the degree of privacy that can be achieved using this algorithm. As seen from the figure we can note that the degree of privacy can be increased as we increase the minimum support value. Based on the support count value, the number of transaction to be modified is decided.

B. Error Analysis

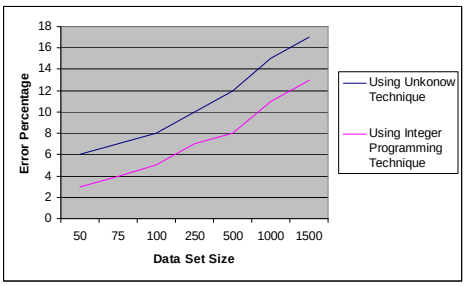

Figure: 7.2. Error Percentage

The above graph(as in Figure 7.2) shows the expected error percentage in comparison with the existing privacy preserving algorithms. In this graph we can see that the error percentage during rule mining using the integer programming technique is much less than the other privacy preserving algorithms, which is an improvement.

C. *Data Utility Analysis*

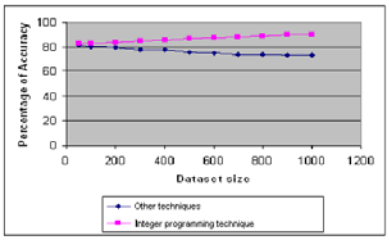

Figure: 7.3. Percentage of Accuracy.

The above graph(as in Figure 7.3) shows the percentage of accuracy that can be achieved using the proposed method compared to the existing method. Here we can see that the percentage of accuracy that can be achieved using the proposed using the integer programming technique is higher than the other methods, which is an enhancement.

D. *Performance Analysis*

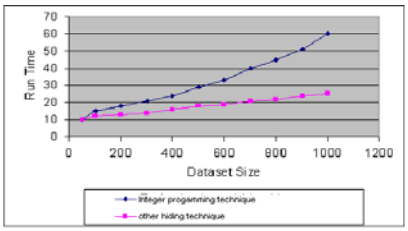

Figure. 7.4. Run time

The above graph(as in Figure 7.4) shows the expected run time of the clustering algorithm increases with an increase in the size of the data set. The proposed algorithm takes relatively extra run time while comparing with other hiding techniques, which requires our attention.

VII. CONCLUSION

In this paper, we have endeavored to enhance an existing concealment technique in order to make safe susceptible knowledge from being uncovered in pattern mining. By hiding the sensitive frequent itemsets that direct to the production of the association rules, we are able to secure the sensitive knowledge and minimize the side effect on the quality of the sanitized database so that non-sensitive knowledge can still be mined. They have used the Apriori algorithm to compute the large itemsets, which is less efficient. In this paper, we have used to harness the advantages of Frequent Pattern Growth Method which mines the total set of frequent itemsets without candidate generation. Given a sensitive frequent itemset, for all the dealings containing this itemset, algorithm first identifies the transaction with the shortest length. In such a transaction, the candidate item with the maximal support value is deleted to dwindle the support of the sensitive itemset. This approach conceals the frequent sensitive itemsets competently and also it hides the non-sensitive itemsets. This approach doesn't minimize side effect generated by it. This disadvantage is overcome in our approach. The investigational appraisal shows that this modus operandi can yield good results on real world datasets, demonstrating its effectiveness towards solving the problem with good data utility, privacy and performance.

VIII. FUTURE WORK

The performance analysis graph (as in Figure 7.4) shows the expected run time of the clustering algorithm increases with an increase in the size of the data set. The proposed algorithm takes relatively extra run time while comparing with other hiding techniques. On analysis it was found that this approach takes more run time to identify the ideal solution, because it forms more number of constraints while the number of sensitive itemset is increased.

As future work, we plan to reduce the number of constraints, and reduce the extra run time which in turn will increase the performance.Future work will also attempt to demonstrate the viability of the architecture through a proof-of-concept prototype. We demonstrate how other techniques can be effectively done using this architecture. In the future, we hope to perk up the efficiency of this approach. As a first direction, we sketch to investigate firmly generating diplomat samples from the database. This would be an orthogonal technique for applications not requiring perfect accuracy.We hope the proposed solution will get hold of new frameworks, techniques, paving way for research track and work well according to the evaluation metrics including hiding effects, data utility, and time performance.

IX REFERENCES

[1] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules", In *Proc. Of the 8th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.

[2] A.Aris Gkoulalas-Divanis and Vassilios S. Verykios ,"An Integer Programming Approach for Frequent Itemset Hiding ", CIKM'06 , November 2006.

[3] Alexandre Evfimievski, "Randomization in Privacy Preserving Data Mining", SIGKDD Explorations, 4(2), Issue 2, 43-48, Dec. 2002.

[4]Alexandre Evfimievski, Johannes Gehrke and Ramakrishnan Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining", PODS 2003, June 9-12, 2003, San Diego, 616, 2003.

[5] CA.B.X. Sun and P. S. Yu., "A border-based approach for hiding sensitive frequent itemsets". *ICDM '05: Proceedings of the 5th IEEE International Conference on Data Mining*, pages 426–433, 2005.

[6] C. Clifton and D. Marks, " Security and privacy implications of data mining", *SIGMOD '96: Proceedings of the 2000 ACM SIGMOD International*

Conference on Management of Data, pages 15–20, 1996.

[7] C. Clifton, "Protecting Against Data Mining Through Samples", in Proceedings of the Thirteenth Annual IFIP WG 11.3 Working Conference on Database Security, 1999.

[8] C. Clifton, "Using Sample Size to Limit Exposure to Data Mining", Journal of Computer Security, 8(4), 2000.

[9] C. Clifton, M. Kantarcioglu, and J. Vaidya. , "Defining privacy for data mining", *WNGDM '02: National Science Foundation Workshop on Next Generation Data Mining*, pages 126–133, 2002.

[10] C. Gueret, C. Prins, and M. Sevaux. , " *Applications of Optimization with Xpress-MP*",. Dash Optimization Ltd., 2002.

[11] E. Dasseni, V. Verykios, A. Elmagarmid and E. Bertino, "Hiding Association Rules by Using Confidence and Support" in Proceedings of 4th Information Hiding Workshop, 369-383, Pittsburgh, PA, 2001.

[12]E. Pontikakis, Y. Theodoridis, A. Tsitsonis, L. Chang, and V. Verykios. , " A quantitative and qualitative analysis of blocking in association rule hiding",. *WPES '04: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, pages 29–30, 2004.

[13] M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios. , "Disclosure limitation .of sensitive rules",. *KDEX '99: Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*, pages 45–52, 1999.

[14]M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data", In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, June 2002.

[15] R. Agrawal and R. Srikant, "Privacy preserving data mining", In *ACM SIGMOD Conference on Management of Data*, pages 439–450, Dallas, Texas, May 2000.

[16] S. Oliveira and O. Zaiane. "Privacy preserving frequent itemset mining". *CRPITS'14: Proceedings of the IEEE International Conference on Privacy, Security, and Data Mining*, pages 43–54, 2002.

[17]S. Oliveira, O. Zaiane, "Protecting Sensitive Knowledge by Data Sanitization", Proceedings of IEEE International Conference on Data Mining, November 2003.

[18] S. Oliveira, O. Zaiane, "Algorithms for Balancing Priacy and Knowledge Discovery in Association Rule Mining", Proceedings of 7th International Database Engineering and Applications Symposium (IDEAS03), Hong Kong, July 2003.

[19] V. Verykios, E. Bertino, I.G. Fovino, L.P. Provenza,Y. Saygin, and Y. Theodoridis, "State-of-the-art in Privacy Preserving Data Mining", SIGMOD Record, Vol. 33, No. 1, 50-57, March 2004.

[20] V. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association Rules Hiding", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, 434-447, April 2004.

[21] Y. Saygin, V. Verykios, and C. Clifton. , "Using unknowns to prevent discovery of association rules",*SIGMOD '01:Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 45–54, 2001.

X APPENDIX

Figure 10.1 shows the screenshot which prompts the user to fill the values of expected maximum item set and threshold value.

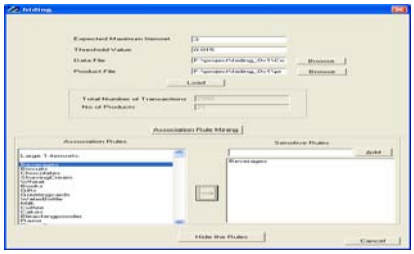

Figure 10.1.Association rule and sensitive itemsets

Figure 10.2 displays the association rules of the original dataset and association rules of modified dataset. It helps the user easily, visualize and compare the association rules of the original and modified dataset.

Figure 10.2. Final output