An efficient modifying scheme for Hiding Sensitive Association Rules (HSAR)

POORNIMA.M [#1], SUMATHI.A[#2] , KALAICHELVI.V[#3] , MANIMOZHI. K[#4], MEENAKSHI. P[#5]
Assistant Professor, Department of CSE
SRC, SASTRA UNIVERSITY, KUMBAKONAM, TN, INDIA
[1] visha_ms@src.sastra.edu

***Abstract* - Large number of database contains sensitive information that must be secured against unauthorized accesses. Recent advances in data mining techniques have increased the disclosure risks that one may encounter when releasing data to outside parties. This work focused on a new approach that strategically modifies a few transactions in the transaction database to decrease the supports or confidences of sensitive rules without producing the side effects. Since the correlation among rules can make it impossible to achieve this goal. This paper proposed heuristic methods for increasing the number of hidden sensitive rules and reducing the number of modified entries. Undesired side effects are avoided in the rule hiding process. All the sensitive rules are hidden without spurious rules falsely generated.**

Keyword- Apriori algorithm, Association rule mining, Sensitive rule hiding

I. INTRODUCTION

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information [3]. A variety of data mining problems have been studied to help people get an insight into the huge amount of data. One of them is association rule mining, proposed by Agrawal et al in 1993. It is an important data mining model studied extensively by the database and data mining community. An association rule mining is to detect relationships or associations between specific values of categorical variables in large data sets. Association Rules are very good in finding out better relationships between items. But business not only needs relations also it requires better security[2].

Association rule mining[5] is usually implemented in two phases: (a) frequent item sets are identified based on the support *s* set by the database owner, and (b) association rules with confidence *c* set also by the user are derived from the frequent item sets. A **frequent itemset** is an itemset whose support is greater than some user-specified minimum support (denoted L_k, where *k* is the size of the itemset). A **candidate itemset** is a potentially frequent itemset (denoted C_k, where *k* is the size of the itemset) for generating frequent itemset we have a well known algorithm called Apriori Algorithm[4]. Apriori algorithm, find out the 1-itemset, 2-itemset and so on to derive frequent grouping of data. After applying the steps of Apriori algorithm, we get the desired frequent grouping and then we apply association rules to find out desired rules that help us in developing marketing strategies. The count of Itemset I (denoted as Ci) is the number of transactions containing I in D and the database size (denoted as Ds) is the number of transactions in D. Frequent Itemset is given as input for generating association rules[5]. Get the Confidence Threshold to calculate the confidence value for each rule mapped from frequent itemset.

Conf(X=>Y) = Support(XUY) / Support(X)

If the confidence value greater than confidence threshold then it is selected as association rule. For two itemsets X and Y, X→Y holds in D (Strong Rule) if both the following conditions hold[6].

1. Support of I = Ci/Ds > MST
2. Confidence of X→Y = Support(XUY)/Support(X) > MCT

Where, MST – Minimum Support Threshold,

MCT – Minimum Confidence Threshold

In the existing method the hiding strategies need to modify the original database with large number of entries, it can be produce more number of false rules and also generates new rules in the original database D[7]. To address this identified major issue this paper highlights the new hidden sensitive rules with effective modifying database techniques. It minimizes the number of changes in the original database entries, such that the information loss incurred by the hiding process is minimal.

II. APRIORI ALGORITHM

For generating frequent item set we have a well known algorithm called **Apriori Algorithm[2].**

It includes the following steps:

Step 1: Generate unique itemset from transactional database

Step 2: Evaluate subsets for unique itemset.

Step 3: Calculate support value for itemset

$$\text{Support } (I_1) = \frac{\text{Total no. of occurrence for } I_1}{\text{Total no. of transactions in the database}}$$

Step 4: Check support value with support threshold. If support value >support threshold then include it in frequent itemset. Eliminate candidates that are infrequent, leaving only those that are frequent.

Step 5: Frequent itemset is given as input for generating association rules. Get the Confidence Threshold, Calculate the confidence value for each rule mapped from frequent itemset.

Conf(X=>Y) = Support (XUY) / Support(X)

Step 6: If the confidence value greater than confidence threshold then it is selected as association rule.

C) Illustration Of Apriori Algorithm

Transactional Database - Set of Transactions

Transactional Database: Support Threshold = 30%

T_1 : {i_1, i_2}
T_2 : {i_3, i_4}
T_3 : {i_2, i_4}
T_4 : {i_1, i_2 , i_4}

Step 1: Unique Itemset = {i_1, i_2 , i_3 , i_4}

Step 2: Subset = {{i_1}, {i_2 }, {i_3 }, {i_4}, **(one itemset)**

{i_1, i_2 }, {i_1, i_3}, {i_1, i_4}, {i_2, i_3}, {i_2, i_4}, {i_3, i_4},**(two itemset)**
{i_1, i_2 , i_3}, {i_1, i_2 , i_4}, {i_1, i_3 , i_4}, {i_2 , i_3 , i_4}, **(three itemset)**
{i_1, i_2, i_3, i_4}} **(four itemset)**
{i_1}, {i_2 }, {i_3 }, {i_4}

{i_1}, {i_2 }, {i_3 }, {i_4}

Step 3: Support({i_1}) = 2/4*100 = 50% Accepted
Support({i_2}) = 3/4*100 = 75% Accepted
Support({i_3}) = 1/4*100 = 25% Rejected
Support({i_4}) = 3/4*100 = 75% Accepted

{i_1, i_2 }, {i_1, i_3}, {i_1, i_4}, {i_2, i_3}, {i_2, i_4}, {i_3, i_4}

Eliminate candidates that have infrequent item sets. In above i3 has minimum support value so it should be consider as infrequent.

Step 4: Support ({i_1, i_2}) = 2/4*100 = 50% Accepted
Support ({i_1, i_4}) = 1/4*100 = 25% Rejected
Support ({i_2, i_4}) = 2/4*100 = 50% Accepted

{i_1, i_2 , i_3}, {i_1, i_2 , i_4}, {i_1, i_3 , i_4}, {i_2 , i_3 , i_4}, {i_1, i_2, i_3, i_4}

In the above all the candidates are eliminated due to minimum support value.

Step 5:

Frequent Itemset :

{{i_1},{i_2},{i_4},{ i_1, i_2},{ i_2, i_4}}

Confidence Threshold =60%

$\{i_1\}$ => $\{i_2\}$	= Support ($\{i_1,i_2\}$) / support($\{i_1\}$)	= 100%	Accepted
$\{i_1\}$ => $\{i_4\}$	= Support ($\{i_1,i_4\}$) / support($\{i_1\}$)	= 50%	Rejected
$\{i_1\}$ => $\{i_2, i_4\}$	= Support ($\{i_1,i_2, i_4\}$) / support($\{i_1\}$)	= 50%	Rejected
$\{i_2\}$ => $\{i_1\}$	= Support ($\{i_2, i_1\}$) / support($\{i_2\}$)	= 66%	Accepted
$\{i_2\}$ => $\{i_4\}$	= Support ($\{i_2, i_4\}$) / support($\{i_2\}$)	= 66%	Accepted
$\{i_4\}$ => $\{i_1\}$	= Support ($\{i_4,i_1\}$) / support($\{i_4\}$)	= 33%	Rejected
$\{i_4\}$ => $\{i_2\}$	= Support ($\{i_4,i_2\}$) / support($\{i_4\}$)	= 66%	Accepted
$\{i_4\}$ => $\{i_1,i_2\}$	= Support ($\{i_1,i_2,i_4\}$) / support($\{i_4\}$)	= 33%	Rejected
$\{i_1,i_2\}$ => $\{i_4\}$	= Support ($\{i_1,i_2,i_4\}$) / support($\{i_1,i_2\}$)	= 50%	Rejected
$\{i_2,i_4\}$ => $\{i_1\}$	= Support ($\{i_1,i_2,i_4\}$) / support($\{i_2,i_4\}$)	= 50%	Rejected

Step 6: Association Rules:

$\{i_1\} => \{i_2\}$

$\{i_2\} => \{i_1\}$

$\{i_2\} => \{i_4\}$

$\{i_4\} => \{i_2\}$

B) Database Modification Schemes

In this work sensitive association rules are protected by modifying its transactional database.

C) *Association rule hiding*

Let D be the database after applying a sequence of modifications to D. A strong rule X → Y in D will be hidden in D' if one of the following conditions holds in D'[8]:

1. $Sup_{XUY} < MST$
2. $Conf_{X \to Y} < MCT$

As per the definition if we want to hide the rule we have to modify the database accordingly.

III. PROPOSED SYSTEM

HIDING SENSITIVE RULES

Steps to be followed for hiding sensitive rules:

Step 1: Hide only rules that are supported by disjoint large item sets.

Step 2: Hide association rules by decreasing either their support or their confidence.

Step 3: Select to decrease either the support or the confidence based on the side effects on the information that is not sensitive.

Step 4: One rule to be hiding at a time.

Step 5: Decrease either the support or the confidence, one unit at a time. If an item in XUY is deleted from a transaction containing XUY, Sup_{XUY} and $Conf_{XUY}$ will be decreased.

ILLUSTRATION OF HIDING SENSITIVE RULE APPROACH

Transactional Database: Support Threshold = 30%

T_1: $\{i_1, i_2\}$
T_2: $\{i_3, i_4\}$
T_3: $\{i_2, i_4\}$
T_4: $\{i_1, i_4\}$

Step 1: Unique Itemset = $\{i_1, i_2, i_3, i_4\}$

Step 2: Subset = $\{\{i_1\}, \{i_2\}, \{i_3\}, \{i_4\}$, **(one itemset)**
$\{i_1, i_2\}, \{i_1, i_3\}, \{i_1, i_4\}, \{i_2, i_3\}, \{i_2, i_4\}, \{i_3, i_4\}$,**(two itemset)**
$\{i_1, i_2, i_3\}, \{i_1, i_2, i_4\}, \{i_1, i_3, i_4\}, \{i_2, i_3, i_4\}$, **(three itemset)**
$\{i_1, i_2, i_3, i_4\}\}$ **(four itemset)**
$\{i_1\}, \{i_2\}, \{i_3\}, \{i_4\}$

$\{i_1\}, \{i_2\}, \{i_3\}, \{i_4\}$

Step 3: Support($\{i_1\}$) = 2/4*100 = 50% Accepted
Support($\{i_2\}$) = 2/4*100 = 50% Accepted
Support($\{i_3\}$) = 1/4*100 = 25% Rejected

Support($\{i_4\}$) = 3/4*100 = 75% Accepted

Step 4: Support($\{i_1, i_2\}$) = 1/4*100 = 25% Rejected

Support($\{i_1, i_4\}$) = 1/4*100 = 25% Rejected

Support($\{i_2, i_4\}$) = 1/4*100 = 25% Rejected

Step 5:

Frequent Itemset:

$\{\{i_1\},\{i_2\},\{i_4\}\}$

Confidence Threshold=60%

$\{i_1\}$	=> $\{i_2\}$	= Support ($\{i_1,i_2\}$) / support($\{i_1\}$)	= 50% Rejected
$\{i_1\}$	=> $\{i_4\}$	= Support ($\{i_1,i_4\}$) / support($\{i_1\}$)	= 50% Rejected
$\{i_2\}$	=> $\{i_1\}$	= Support ($\{i_2,i_1\}$) / support($\{i_2\}$)	= 50% Rejected
$\{i_2\}$	=> $\{i_4\}$	= Support ($\{i_2,i_4\}$) / support($\{i_2\}$)	= 50% Rejected
$\{i_4\}$	=> $\{i_1\}$	= Support ($\{i_4,i_1\}$) / support($\{i_4\}$)	= 33% Rejected
$\{i_4\}$	=> $\{i_2\}$	= Support ($\{i_4,i_2\}$) / support($\{i_4\}$)	= 50% Rejected

Sensitive association rules[5] are rules that contain sensitive knowledge showing strategic patterns and trends. Consider the original database has first rule $\{I_1\} \rightarrow \{I_2\}$. For hiding this rule, Support ($\{I_1\} \cup \{I_2\}$) is decreased by deleting an element in the transaction that contains ($\{I_1\}U\{I_2\}$). Transaction 4 holds ($\{I_1\}U\{I_2\}$) and ($\{I_2\}U\{I_4\}$).Any one of the items I_1 or I_2 or I_4 can be deleted. Here in case, if item I_2 is deleted then all the sensitive rules will be hidden, since I_2 exists in more than one rule and also the modified entries in original database may be reduced. But it should be opposed for choosing the smallest transaction in size; therefore it will probably choose mostly average size transactions which will have small side effects to the confidence of the other rules.

Suppose that we would like to decrease the support of a rule $I_1 => I_2$. The smallest possible transaction that supports this rule is $\{I_1 => I_2\}$, and suppose that such a transaction exists. Removing I_1 from that transaction will cause the confidence of the rules (other than $I_1 => I_2$) that contain I_1 in their antecedent to increase which will cause the introduction of new rules observing the minimum confidence requirement. However for average size transactions this will probably not be the case since they will contain both the antecedent and the consequent of the rules and the confidence of these rules will decrease upon the removal of I_1. Similarly each rule is mapped with the database and necessary modifications are made to get a new database D'.

In existing work the hiding strategies need to modify the original database with changing large number of entries it can be produce more number of false rules and also generates new rules. This proposed work minimized to modify large number of entries and also reduce false rate generation.

IV. CONCLUSION

Usually the database which we generated hides the sensitive rules with side effects. Side effects cause generation of false rule and loss of rules. This work developed for reducing number of modified entries in original database and improves the efficiency and time complexity with limited side effects and gives assurance for hiding all sensitive association rules with modified original databases. Also it can be created model for centralized environment and also it is going to apply it for distributed environment. Another thing side effects created during hiding will be eliminated to possible extent.

REFERENCES

[1] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi, Improving Efficiency of Apriori Algorithm Using Transaction Reduction, International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013 ISSN 2250-3153.

[2] Rui Chang; Zhiyi Liu; , "An improved apriori algorithm," Electronics and Optoelectronics (ICEOE), 2011 International Conference on , vol.1, no., pp.V1-476-V1-478, 29-31 July 2011

[3] Han J,Kamber M.Data Mining:Concepts and Techniques.Higher Education Press,2001.

[4] Kong Fang , Qian Xue-zhong,Research of improved apriori algorithm in mining association rules,Computer Engineering and Design, 2008,v29, n16, p4220-4223.

[5] Nabil R. Adam and John C. Wortmann. Security-Control Methods for Statistical Databases: A Comparison Study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[6] Yi-Hung Wu,Chia-Ming Chiang and Arbee L.P. Chen,Hiding Sensitive Association Rules with Limited Side Effects, Knowledge and Data Engineering,IEEE Computer Society,January 2007 (vol. 19 no. 1)pp. 29-42.

[7] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure Limitation Of Sensitive Rules. *Proceedings of Knowledge and Data Exchange Workshop*,1999.