Hiding Sensitive Fuzzy Association Rules Using Weighted Item Grouping and Rank Based Correlated Rule Hiding Algorithm

K. SATHIYAPRIYA[1], G. SUDHASADASIVAM[2], C. J. P. SUGANYA[3]
Department of Computer Science and Engineering
PSG College of Technology
Coimbatore
INDIA
[1]sathya_jambai@yahoo.com, [2]sudhasadhasivam@yahoo.com, [3]cjpsuganya@gmail.com

Abstract

Extracting knowledge from large amount of data while preserving the sensitive information is an important issue in data mining. Providing security to sensitive data against unauthorized access has been a long term goal for the database security research community. Almost all the research in privacy preservation is limited to binary dataset. Business and scientific data contain both quantitative and categorical attributes. The technique used for privacy preservation must ensure security of the database while maintaining the utility and certainty of the mined rules at highest level. This paper presents two techniques to hide quantitative sensitive fuzzy association rules - Weighted Item Grouping Algorithm and Rank based Correlated Rule Hiding Algorithm. Then the performance of the two techniques is evaluated based on the number of lost rules and ghost rules generated and how effectively the sensitive rules are hidden. The experimental results shows that the Rank based correlated rule hiding provides better performance than weighted item grouping in terms of side effects and number of modifications.

Key-Words: - Data perturbation, Fuzzy, Correlation analysis, Sensitive Association rules, Item Grouping, Rule Hiding, Quantitative data, weighted, privacy preservation, data security.

1 Introduction

One of the most popular activities in data mining is association rule mining. Rule mining is used for finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. It involves analyzing and presenting strong rules discovered in databases using different interest measures.

An association rule is defined as an implication $X \rightarrow Y$, where both X and Y are defined as sets of attributes (interchangeably called items) . Here X is called as the body (LHS) of the rule and Y is called as the head (RHS) of the rule. It is interpreted as follows: “for a specified fraction of the existing transactions, a particular value of an attribute set X determines the value of attribute set Y as another particular value under a certain confidence”. For instance, an association rule in a supermarket basket data may be stated as, “In 20% of the transactions, 75% of the people buying butter also buy milk in the same transaction”; 20% and 75% represent the support and the confidence, respectively. The significance of an association rule is measured by its support and confidence. Support is the percentage of transactions that contain both X and Y, while confidence is the ratio of the support of X UY to the support of X. Business and scientific data have richer attribute types which can be quantitative or categorical. The traditional methods of finding frequent items and performing level wise search cannot be applied for quantitative rule mining.

One way of mining quantitative rules is to treat them like categorical attributes and generate rules for all possible values. But, in most cases, a given numeric value will not appear frequently. So the domain of each quantitative attribute is divided into intervals and rules are formulated. This is called discretization. Choosing intervals for numeric attributes is quite sensitive to support and confidence measures. Intervals cannot be generated randomly because the data set may be skewed. It was shown that if the range of the attribute is divided into equal intervals it leads to two problems of minsupport and minconfidence[1]. When the number of intervals found for a single quantitative

attribute is high, the support of one of these intervals becomes low. This is called "Minsupport" issue. Building larger intervals in order to cope with the first problem, raises another challenge. If the number of intervals is less, more information is lost and rules mined are different from that in original data. This is called "Minconfidence" issue.

A tradeoff has to be found to discretize “correctly” numeric attributes with respect to MinSupport and MinConfidence. Another problem with discretization is sharp boundary problem.

For example, consider the rule, if(years_employed>=2)and(income>=50000) then credit = approved. If a customer has a job for two years and a credit income of $ 49,000 then the application for credit approval may be rejected. Such a precise cut-off seems unfair. So, in order to avoid the sharp boundary problem the data is fuzzified.

In this example, the income can be discretized into categories like low, medium, high and fuzzy logic can be applied to allow fuzzy threshold or boundaries to be defined for each category. Unlike the notion of traditional crisp sets where an element either belongs to a set S or its complement, in fuzzy set theory, elements can belong to more than one fuzzy set. In this example, the income $ 49,000 belongs to both medium and high fuzzy sets, but to different degrees.

The problem of privacy-preserving data mining has become important in recent years because of the increasing sophistication of data mining algorithms that can extract personal information from datasets. A rule is characterized as sensitive if its disclosure risk is above a certain confidence value. Sensitive rules should not be disclosed to the public, as they can be used to infer sensitive data and provide an advantage for the business competitors. Using Data Mining methods it is possible to extract association rules that violate personal privacy. This leads to increased concerns about the privacy of the underlying data. So, a number of techniques have been proposed for modifying or transforming the data in such a way to preserve privacy. The aim of privacy preserving data mining is to design methods which continue to be effective, without compromising security.

The rest of this paper is organized as follows. A review of the literature is provided in Section 2. Section 3 defines the problem. The proposed algorithms Weighted Item grouping and Rank-based correlated item hiding, to hide the sensitive fuzzy association rules is discussed in section 4. The complexity of the proposed approaches were analyzed in section 5. Experimental results are given in Section 6. Section 7 includes the conclusion.

2 Related Work

Techniques for hiding sensitive association rules can be classified into two broad categories[2] namely, distortion based technique and blocking based technique. In distortion based technique, the data is distorted such that the support and confidence of sensitive association rules is reduced below threshold. Here threshold refers to minimum value of support and confidence below which the association rule becomes uninteresting. This technique has side effects of ‘Lost Rules’ and ‘Ghost Rules’. Lost Rules refers to undesirable hiding of items and association rules that are not sensitive. Ghost rules are non genuine association rules which become part of association rules set. Distortion based technique reduces these side effects while maintaining a linear time complexity with the size of dataset. This technique is a serious bottleneck in medical applications where deleting a part of dataset leads to wrong inference. Blocking based technique is characterized by introducing uncertainty without distorting the database. It also suffers from side effects of lost item, lost rule and ghost rule.

Rule hiding techniques proposed by Vassilios et. al.[3] are distortion based algorithms, that are evaluated based on their efficiency and side effects. Side effects of these algorithms were high.

Yuhong Guo et. al.[4][5] presented FP-tree based method for inverse frequent set mining. In this algorithm after extraction and pruning of frequent itemset, FP-tree is constructed, which is later converted into many versions of modified database. The strength of this technique is its efficiency. More than one modified database is released. Number of released databases was characterized by the number of non frequent items chosen. Limitation of this technique is that it focused on hiding sensitive items only and also has side effect of large number of lost rules.

T.P.Hong et al [6], presented an algorithm that integrates fuzzy set concepts and the apriori mining algorithm to find interesting fuzzy association rules in given transaction data sets. This algorithm considers only the fuzzy region that has support more than the minimum support for framing the rules. So it is claimed to have good time complexity.

T. Berberoglu[7], proposed a novel method to hide critical fuzzy association rules from quantitative data. The sensitive rule is hidden by increasing the

support value of LHS of the rule which in turn decrease the confidence of the rule.

Manoj Gupta et al[8], proposed fuzzification with variable fuzzy membership function and decreasing the support can be used for quantitative association rule hiding. But this work requires the membership function to be predefined and are usually built by human experts.

ADSRRC (Advanced Decrease Support of Right Hand Side items of Rule Cluster) and RRLR (Remove and Reinsert Left Hand Side of Rule) proposed by Komal Shah et al[9] overcomes the limitations of multiple sorting and hiding rules with multiple items on the right hand side of the algorithm DSRRC (Decrease Support of Right Hand Side items of Rule Cluster) respectively.

Sonia Hameed et al [10] proposed a scheme for privacy preservation of fuzzy association rules (PPFAR) based on fuzzy correlation analysis. The fuzzy set concept is integrated with fuzzy correlation analysis and Apriori algorithm to mark interesting fuzzy association rules which are considered as sensitive. Experimental results show that PPFAR scheme hides more sensitive rules with minimum number of modifications and maintains quality of the released dataset.

A method to hide sensitive fuzzy association rule [11] mines the fuzzified data using modified APRIORI algorithm. The sensitive rules are hidden by Decreasing the Support of Right Hand side of the Rule(DSR) approach. A learning method is also proposed for automatic derivation of fuzzy membership function.

Frequent Hiding Sensitive Frequent Item and Frequent Hiding Sensitive Association Rule[13][14] are based on support and confidence framework. Transactions in dataset are weighted based on its support for a sensitive rule and are sorted in descending of their weight. Transactions are modified till the confidence of sensitive association rules fall below given threshold. Among antecedent and consequent, random selection was made for pruning.

S. L. Wang et. al.[15][20] introduced two strategies for hiding sensitive association rules. The first strategy, called Increasing the Support of Left Hand Side of the rule(ISL), decreases the confidence of a rule by increasing the support of the itemset in its Left Hand Side(LHS). The second approach, called Decreasing the Support of Right Hand Side of the rule(DSR), reduces the confidence of the rule by decreasing the support of the itemset in its Right Hand Side(RHS). Both algorithms rely on the distortion of a portion of the database transaction to lower the confidence of the association rule. The algorithms required a reduced number of database scans and exhibit an efficient pruning strategy. Moreover, the DSR algorithm seems to be more effective when the sensitive items have high support.

A Genetic algorithm based method for preventing extraction of sensitive association rules from quantitative data[16] hides sensitive rules by decreasing the support of the RHS of the rule. Genetic Algorithm maximizes the number of non sensitive rules that can be mined from the released dataset by minimizing the number of modifications to the data. Unlike previous approaches which mainly deal with association rules in binary database, the proposed approach deals with hiding the association rules in quantitative database.

N V Muthu lakshmi et al[18], proposed a heuristic based methodology to hide the sensitive item sets efficiently. This methodology protects private information by sanitizing the data after analyzing the side effects, so that side effects can be fully avoided or accepting few side effects which will not harm the informational accuracy.

Apriori algorithm generates large number of candidate sets which results in combinatorial explosion. This in turn increases the number of database scans and complexity of calculations. Many methods are proposed to overcome this limitations[21- 23].

3 Problem Statement

Privacy preserving data mining involves getting valid data mining results in addition to hiding sensitive information. Most of the studies proposed concentrated on hiding association rules associated with binary items without giving importance to its quantity. However, many transactions in real world applications have quantitative values. For a diabetes patient, the quantity of the attribute sugar in blood is more important than the presence or absence of sugar.

Consider the case of a health drink reseller who purchase health drink at low price from two companies, A and B. Reseller also grants them access to his customer database. B supplier may misuse the database to mine association rules related to A, inferring facts like “People who buy Milk also buy the product A”. Using this information, B supplier offers a discount coupon on milk with each purchase of B. Hence, sales on A drops rapidly and A supplier cannot offer it at low price as before. This enables product B monopolize the health drink market which results in the hike of health drink prices. As a result, reseller may start losing business

to his competitors. This scenario emphasis need for research on sensitive knowledge hiding in database.
This paper proposes two methods that combines previous researches and take them one step ahead. The first method combines fuzzy association rule mining[8] and item grouping algorithm[12]. Item grouping algorithm was proposed to work with binary dataset in the previous research. In this paper it is modified to work with fuzzy data and the item to be modified first is chosen based on the weight assigned to the items in the sensitive rule groups. The item which supports more sensitive rules and less non sensitive rules is given higher weight. The second method combines fuzzy association rule mining[8] and fuzzy correlation scheme[10]. In this paper the items in the highly correlated rules are ranked based their support to number of sensitive and non sensitive rules. The item with higher rank is chosen for modification first.
The problem can be stated as consisting of two parts, Mining fuzzy association rules and hiding the sensitive association rule by i) Weighted item Grouping and ii) Rank Based Correlated item Hiding

4 Proposed Algorithm

Input: Source database D, Min Support Threshold (MST), Min Confidence Threshold (MCT).
Output: Transformed database D' so that sensitive fuzzy association rules are hidden hence cannot be mined.

4.1 Weighted Item Grouping Algorithm (WIGA)

1. Fuzzification of the database, D→ F
2. In fuzzified database F, calculate every item's support value where f ∈ F.
3. If all f(support)< min_support; EXIT. // there isn't any rule.
4. Find Large 2_itemsets in F.
5. For each X's large 2 itemsets
6. Calculate the support value of the rule U→ min(UL, UR) where UL and UR is the support of the left hand side and right hand side of the rule.
7. Find R = {Rules from itemset X};
8. Select and remove a sensitive rule S_r from R;
9 . Compute confidence of the rule S_r;
10. If confidence (S_r) > min_confidence MCT then
11. Add the rule S_r to S_{RH} ;
12. For each $item_k$ ∈ D do
    For each sensitive association rule s_{ri} ∈ S_{RH} do
    If $item_k$∈ items(s_{ri}) then
    T[$item_k$].weight =T [$item_k$].weight + 1;
    end
    end
13. For each $item_k$ ∈ D do
    For each nonsensitive association rule ns_{ri}∈ R do
    If $item_k$∈ items(ns_{ri}) then
    [$item_k$].NSweight = [$item_k$].NSweight + 1;
    end
    end
14. For each sensitive $item_k$ ∈D do
    Sort T [$item_k$].weight in descending order
    Sort T[$item_k$].NSweight in ascending order
    end
15. Group sensitive rules in a set of groups GP such that ∀ G∈GP, ∀ s_{ri}, s_{rj} ∈ G, s_{ri},and s_{rj} share the same itemset I in LHS || RHS respectively.
16. Order the groups in GP by the size in number of sensitive rules in group.
    ∀ s_r ∈ G_i ∩G_j do
    if size(G_i) ≠size(G_j) then
    remove s_r from smallest (G_i, G_j)
17. For each T[$item_k$] in S_r do
    For each sensitive association rule s_{ri}∈ G_k do
    IF $item_k$ is in RHS of s_{ri}
    Find Tx = {t| t ∈ D such that 1-max (TL, TR) < min (TL, TR)};
WHILE (confidence(s_{ri}) ≥ min_confidence and support (s_{ri}) ≥ min_support and Tx is not empty)
    Choose the first transaction t from Tx;
    IF TR > 0.5 and TL = TR THEN
    TR = 1 − TR;
    ELSE
    max (TL,TR) = 1 − max (TL,TR);
    Remove and save the transaction t from Tx;
    Re-compute support and confidence of rule S_{ri}
    end// WHILE
    end //if
    IF $item_k$ is in LHS of s_{rj}
Find Tx = {t| t ∈ s_{rj} such that 1-min(TL, TR) >TL};
WHILE (confidence(s_{rj}) ≥ min_confidence and support (s_{rj}) ≥ min_support and Tx is not empty)
    Choose the first transaction t from Tx;
    IF T_L < 0.5 THEN
    T_L = 1 – min(T_L, T_R);
    Remove and save the transaction t from Tx;
Re-compute support and confidence of rule S_{rj}
end// WHILE
    end//if
    IF Tx is empty then
    Cannot hide rule S_{rk} and restore F;
    end //if
    end // for
    end // for

4.1.1 Steps of the Weighted Item Grouping Algorithm

In this section the steps of the above algorithm is explained in detail. Fuzzy concepts are used to mine the fuzzy association rules from quantitative data. The sensitive rules in the discovered rule set is then hided using weighted item grouping privacy preservation technique.

Let I={i_1, i_2, i_3...i_m} be the complete item set where each i_j ($1 \leq j \leq m$) is a quantitative attribute and m is the number of items in the database. Given a database D={t_1, t_2,…., t_n} where each t_j is a transaction with attributes I. Let X ={x_1, x_2,…,x_p} and Y = {y_1, y_2,….., y_q} be two large itemsets. Then, the fuzzy association rule is given as follows: A→B where A={ f_1, f_2,…f_p} and B ={g_1, g_2,….. g_q} and $f_i \in$ {the fuzzy regions related to attribute x_i}, $g_j \in$ {the fuzzy regions related to attribute y_j}. X and Y are subsets of I and are disjoint. A and B contain the fuzzy sets associated with the corresponding attributes in X and Y.

In a classical set or crisp set, the objects in a set are called elements or members of the set. An element x belonging to a set A is defined as $x \in A$. A characteristic function or membership function $\mu_A(x)$ is defined as an element in the universe U having a crisp value of 1 or 0 equation 1. For every $x \in U$,

$$\mu_A(x) = \begin{cases} 1 \; for \; X \in A \\ 0 \; for \; X \notin A \end{cases} \quad (1)$$

The membership functions for crisp set can take a value of 1or 0, the membership functions for fuzzy sets can take values in the interval [0,1]. The range between 0 and 1 is referred to as the membership grade or degree of membership [17]. A fuzzy set A is defined as in equation 2.

$$A = \{(X, \mu_A(X)) | X \in A, \; \mu_A(X) \in [0,1]\} \quad (2)$$

Where $\mu_A(X)$ is a membership function belonging to the interval [0,1] [8].

Notations used in this paper are as follows:

n : the total number of transactions data;

m : the total number of attributes (items);

$D^{(i)}$: the *ith* transaction data,$1 \leq i \leq n$;

$I\,j$: the *jth* attribute,$1 \leq j \leq m$;

$|Ij|$: the number of fuzzy regions for $I\,j$;

Rjk : the *k th* fuzzy region of $I\,j$, $1 \leq k \leq |I\,j|$;

$v_j^{(i)}$: the quantitative value of $I\,j$ for $D^{(i)}$;

$f_{jk}^{(i)}$: the membership value of $v_j^{(i)}$ in the region *Rjk* ;

Step 1 Cleaning of the database D→C The database cleaning deals with detecting and removing errors and inconsistencies from the data in order to improve the quality of the data.

Step 2 Generation of fuzzy association rules[8]:

Fuzzification comprises the process of transforming crisp values into grades of membership of fuzzy sets. The cleaned database is fuzzified using triangular membership function into 3 regions z, o, b as shown in the fig. 1

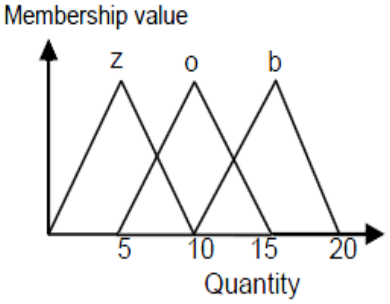

Fig.1 Triangular membership function

The detailed steps of the fuzzy association rules mining algorithm is described as follows.

1. For each transaction data $D^{(i)}$, i=l to n, and for each attribute (item) $I\,j$, j=l to m, transform the quantitative value $v_j^{(i)}$ into a fuzzy set $f_{jk}^{(i)}$ represented as $\left(\frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{j2}^{(i)}}{R_{j2}} + \cdots + \frac{f_{jp}^{(i)}}{R_{jp}}\right)$ using the given membership function for the attribute as shown in fig.1, where p is the number of fuzzy regions for attribute $I\,j$.

2. Calculate the support count of each attribute region *Rjk* on the transactions data as in equation 3.

$$count_{jk} = \sum_{i=1}^{n} f_{jk}^{(i)} \quad (3)$$

3. For each attribute region *Rjk* , $1 \leq j \leq m$ and $1 \leq k \leq |I\,j|$, check whether its *count jk* is greater than or equal to the given minimum support value. If *Rjk* satisfies the above condition, then put it in the set of large 1-itemsets L1. That is:

$$L_1 = \left\{ Rjk \middle| \begin{matrix} count_{jk} \geq \text{minsupport}, 1 \leq j \leq m \; and \\ 1 \leq k \leq |Ij| \end{matrix} \right\}$$

4. Join the large 1-itemsets (L1) to generate the candidate set *C*2 in a way similar to that in apriori algorithm except that two regions belonging to the same attribute (item) cannot simultaneously exist in an itemset in *C*2.

5. For each candidate 2-itemset S with regions (A1 and B1) in *C*2, do the following steps:

(i) Calculate the fuzzy value of each transaction data on itemset S as given equation 4

$$f_s^{(i)} = \min_{j=i \; to \; 2} f_{s_j}^{(i)} \quad (4)$$

(ii) Calculate the fuzzy count of itemset S on the transactions data as in equation 5.

$$count_s = \sum_{i=1}^{n} f_s^{(i)} \quad (5)$$

(iii) If $count_s$ is greater than or equal to the given minimum support value, then put the itemset S in set L2 (Large 2-itemset).

6. For each large 2-itemset, find the interesting useful association rules having confidence value greater than or equal to minimum confidence value. The confidence value of a rule $A \rightarrow B$ is computed as given in equation 6

$$\text{Confidence}(A_1 \rightarrow B_1) = \frac{\text{support}(A_1 \rightarrow B_1)}{\text{support}(A_1)} \quad (6)$$

where support of itemset S with items (A1 and B1) is computed as follows in equation 7

$$\text{Support}(S) = \frac{count_s}{N} \quad (7)$$

Step.3 Weighted Item Grouping Algorithm (WIGA)

The main idea behind this algorithm is to group sensitive rules in groups of rules sharing the same item sets. The algorithm associate variables called weight sensitive(weight) and weight non sensitive (weightNS) with each item. For each occurrence of the given item in the sensitive rule the sensitive weight is incremented. For each occurrence of the item in the non sensitive rule the weightNS is incremented. The items in the sensitive transaction are first sorted in descending order based on the sensitivity weight and in the ascending order based on non sensitive weight. This sorting moves the sensitive item that is supported by least number of non sensitive rules to the top of the list. If two sensitive rules contain same items on the same side, either the LHS or the RHS of the rule , by sanitizing the transactions for this sensitive item, one would take care of hiding these two sensitive rules at once and consequently reduce the impact on the released database.

Step. 4 Hiding sensitive association rules

In order to hide an association rule, A→B , either its support is reduced to be smaller than minimum support value or its confidence is reduced to be smaller than its minimum confidence value. To decrease the confidence of a rule, two strategies can be used. The first one is to decrease the support count of AB , while keeping the support count of A (i.e. LHS. of the rule) constant. The second one is to increase the support count of A (i.e. LHS of the rule), but not support count of AB . This algorithm checks if the item to be hided is in the RHS or LHS of the rule. If the item is in RHS of the rule, the support count of itemset AB is decreased by decreasing the support count of either A or B i.e. item in LHS or RHS of the rule. For this purpose, the value of item in LHS or RHS. is subtracted from one in case the value of item in LHS or RHS is less than the value of item in RHS or LHS respectively. If the item is in LHS of the rule, the support count of item A is increased. The algorithm assumes that an item occurs either in LHS or RHS of the rule if it occurs in more than one sensitive rule, not on the different side of different rules. This again would minimize the impact on the database and reduce the number of lost rules.

4.2 Rank Based Correlated Item Hiding Algorithm(R- CA)

Correlation analysis measures the relationship between two sensitive rules. Assume that there is a random sample (x_1, x_2,,,,,x_n) ∈ X, along with a sequence of paired data { (x_i, $\mu_A(x_i)$, $\mu_B(x_i)$)| i=1...n}, which corresponds to the grades of the membership functions of fuzzy itemsets A and B defined on X. Then, the fuzzy correlation coefficient($r_{A,B}$) between the fuzzy itemsets A and B as defined by Sonia Hameed et al [10] is given in equation 8.

$$r_{A,B} = S_{A,B} / S_A S_B \quad (8)$$

where

$$S_{A,B} = \frac{\sum_{i=1}^{n}(\mu_A(x_i) - \overline{\mu_A}).(\mu_B(x_i) - \overline{\mu_B})}{n-1}$$

$$\overline{\mu_A} = \frac{\sum_{i=1}^{n}\mu_A(x_i)}{n}$$

$$\overline{\mu_B} = \frac{\sum_{i=1}^{n}\mu_B(x_i)}{n}$$

$$s_A^2 = \frac{\sum_{i=1}^{n}(\mu_A(x_i) - \overline{\mu_A})^2}{n-1}$$

$$s_B^2 = \frac{\sum_{i=1}^{n}(\mu_B(x_i) - \overline{\mu_B})^2}{n-1}$$

$$S_A = \sqrt{S_A^2}$$

$$S_B = \sqrt{S_B^2}$$

The value computed from $r_{A,B}$ lies between in [-1, 1]. If $r_{A,B}>0$, then the fuzzy itemsets A and B are positively related. If $r_{A,B} = 0$, then the fuzzy itemsets A and B are negatively related. But, if $r_{A,B}<0$, then the fuzzy itemsets A and B have no relationship at all. The correlation coefficient is calculated among the given sensitive rules. And the two sensitive rules {($SF_x \rightarrow SF_y$) , ($SF_x \rightarrow SF_z$) } are chosen such a way that , the correlation coefficient value between these rules are high. The attributes SF_x,SF_y,SF_z are ordered and ranked. The ranking is done such that the attribute with the highest rank supports more number of sensitive rules and less number of non sensitive rules. The attribute with highest rank is chosen as the victim item. This attribute undergoes data perturbation so that its value goes below the MST and MCT.

Algorithm RANK BASED CORRELATED ITEM HIDING

1 Fuzzification of the database, D→ F
2. In fuzzified database F, calculate every item's support value where f ∈ F
3. If all f(support)< min_support;
EXIT. // There isn't any rule.
4. Find Large 2_itemsets in F.
5. For each X's large 2 itemsets
6. Calculate the support value of the rule↔U min(UL , UR)
7. Find R = {Rules from itemset X};
8. Select and remove a sensitive rule S_r from R;
9 . Compute confidence of the rule S_r;
10. IF confidence (S_r) > min_confidenceMCT then
11. Add the rule S_r to S_{RH} ;
12. FOR EACH rule S_{ri}, $S_{rj} \in S_{RH}$
$R(sr_i, sr_{j)} = corr(sr_i, sr_{j)}$ where Sr_i = Min(fa,fb)
//Calculate the correlation between sr_i and sr_j (Rsr_i, sr_j) using the equation 8.
// *Minimum of fuzzy values of the attributes A,B in S_{ri} represents the fuzzy value for the rule S_{ri}*
End FOR
13. //Ordering the attributes present in the highly correlated rule.
∀Sr ∈ S_{RH}
FOR EACH item $_k$∈ R(sr_i, $sr_{j)}$
Find the sensitive weight and non sensitive weight.
//Rank the attributes higher if they support more sensitive rules ,less non-sensitive rules.
END FOR
14. Data perturbation
∀sensitive rule $sr_i \in S_{RH}$
∀item$_k$ ∈ S_{RH} choose item$_k$ based on their rank
Choose p transactions such that $T_R < T_L$
Compute temp= T_L- T_R
Update $T_R = T_R$ -temp
If(update value doesn't lie within the domain) then T_R=min_domain_value;
end
15. Recalculate and update the support for all the sensitive rules in S_{RH}.
END

4.2.1 Steps of the rank based correlated item hiding algorithm

First two steps in Rank Based correlated item hiding algorithm is same as Weighted Item Grouping algorithm, which includes

Step. 1 Cleaning the dataset

Step. 2 Fuzzification of quantitative dataset.

Step. 3 The Rank Based Correlated Item Hiding

The algorithm finds the correlation among the all possible pairs of sensitive rules. The rules that has higher correlation is chosen. The items in the sensitive rule are assigned sensitive- weight and non sensitive-weight depending on its occurrence in number of sensitive and non sensitive rules. The items in the sensitive transaction are ranked based on the sensitivity weight and non sensitive weight. The item with highest rank is chosen for hiding.

Step. 4 Hiding the sensitive item

To hide the chosen sensitive item, find p transactions such that the value on the right hand side of the rule is less than the value on left hand side of the rule. The value on the right hand side is then replaced with a value obtained by subtracting the RHS value with the difference between LHS and RHS. If this new value is negative then minimum of the domain value is used to replace the RHS value. This decreases the support of right hand side of the rule which in turn reduces the confidence of the rule below minimum confidence.

5 Complexity Analysis

Theorem 5.1: Let a modification process has no invalid modification. Let RI represents the items in the set of sensitive rules to be hided. The upper and lower bound for concealing all restrictive rules are $\sum_{j=1}^{|RI|}(|T_r| + |T_l|)$ and $\max_{r_{j\in RI}}(|T_r|)$, respectively. T_r and T_l are the transactions that has 1-max (T_l, T_r) < min (T_l, T_r) and 1-min(T_l, T_r) >TL values for the LHS and RHS item of the rule .

Proof: If modification of an item from the fuzzy transaction results in decrease of fuzzy support of some restrictive item r_j, where the current support count is greater than $\lfloor \sum_{j=1}^{N} db_{rj} \rfloor$, N is the number of transactions, is called valid item modification. Otherwise it is invalid item modification. A valid item modification reduces the support of at least one restrictive item in the itemset. The value $\sum_{j=1}^{N} db_{rj}$ is equal to the support count of r_j in the source database.

(1) Upper bound: For the weighted Item Grouping algorithm, In the worst case, each valid item modification decreases the support of only one restrictive itemset by 0.1. To hide an arbitrary rule, it is necessary to decrease the support of right hand side of the rule or to increase the support of the item on the left hand side if the rule. The algorithm modifies T_r transactions that has value 1-max (T_l, T_r) < min (T_l, T_r) for the RHS item of the rule. If the rule is not concealed, then the support of the LHS

item of the rule is increased in those transactions that has1-min(T_l, T_r) >T_l value for the LHS item of the rule. The Maximum number of modification for concealing an arbitrary restrictive rule, is $(|T_r| + |T_l|)$, Where Tr represents number of transactions that has 1-max (T_l, T_r) < min (T_l, T_r) value for the RHS item of the rule and T_l represents number of transactions that has is number the transactions that has value value for the LHS item of the rule. Modifications after this is invalid.

For rank based correlated item hiding algorithm, To hide each restrictive item in the restricted rule, the algorithm modifies only those transactions T_r ,which RHS value of the rule less than the value of LHS item of the rule. Therefore the upperbound of modification rate is $\sum_{j=1}^{|RI|}(|T_r|)$.

(2) Lower bound: In the best case, each valid item modification can decrease the support of all restrictive rules. An arbitrary rule requires decreasing its support by $(|T_r|)$. Therefore, the modification process requires at least $\max_{r_{j\in RI}}(|T_r|)$ valid modifications. When the number of deleted items is less than $\max_{r_{j\in RI}}(|T_r|)$, there exists at least one restrictive itemset that has a support value greater than minimum support . Therefore the lower bound for both the algorithms is $\max_{r_{j\in RI}}(|T_r|)$

Definition 5.1: Let ModItem(MI) be the number of items that are modified to completely hide the restrictive ruleset. Then the percentage of actual modifications ModItem(MI) to $\sum_{j=1}^{|RI|}(|T_r| + |T_l|)$ is called the modification rate (MR). MR = $\frac{\text{ModItem(MI)}}{\sum_{j=1}^{|RI|}(|T_r|+|T_l|)*\varphi}$

Theorem 5.2: The running time of Weighted Item Grouping algorithm is O(n_{sr}*n_I+ n_{nsr}*n_I + $2n_{si}^2$ + 2N* n_{si} +n_l+n_r), where n_{sr} is the number of sensitive rules, n_{nsr} is the number of non sensitive rules , n_I is the number of items, n_{si} is the number of unique items in the restrictive ruleset. N is the number of transactions , n_l is number the transactions that has value 1-max (T_l, T_r) < min (T_l, T_r) for the RHS item of the rule and n_r is number the transactions that has 1-min(T_l, T_r) >T_l value for the LHS item of the rule.

Proof: In the step 12 and 13 the each item in the dataset is checked against each nonsensitive rule and each sensitive rule to find the sensitive and non sensitive weight. This results in execution time complexity of nsr*nI+ nnsr*nI . In step 13 and 14 the algorithm sorts the items in descending and ascending order of sensitive and non sensitive weight respectively. This has the complexity of O(nsi^2)+ O(nsi^2) = O(2 nsi^2) .

In step 17, for each item in restrictive rule set, algorithm first scans all the transactions to get those transactions that has a value such that 1-max (Tl, Tr) < min (Tl, Tr), if the item is in RHS of the rule and again scans the transactions to get those transactions with value that satisfies 1-min(Tl, Tr) >Tl if the item is in the LHS of the rule. so the complexity is O(2N* nsi). Finally, nl and nr are the maximum number of transactions that are modified to hide the ruleset. Summing the complexity of these steps, the complexity of the algorithm is O(nsr*nI + nnsr*nI +2nsi^2 + 2N* nsi +nl+nr) ≈ O(n^2)

Theorem 5.3: The running time of Rank Based Correlated Item Hiding Algorithm is O(nsr2 + nsr*nI+ nnsr*nI), where nsr is the number of sensitive rules, nI is the number of items, nSI is the number of sensitive items, N is the number of transactions.

Proof: In step 12 of the algorithm, the correlation among all pair of the sensitive rule is calculated. To find the correlation between two fuzzy rules the minimum of fuzzy value of the items in the rule is taken for each transaction to represent the fuzzy value of the rule. The correlation is found among all pairs of sensitive rules. So the complexity is O(nsr^2).

In step 13 for each item in the sensitive rule the sensitive and non sensitive weight is calculated as in previous algorithm. The complexity for finding the sensitive and non sensitive weight is O(nsr*nI+ nnsr*nI).

In step 14, the transactions whose RHS item value is less than the LHS is modified by scanning the dataset. The modification continues until the support goes below the minimum support. At the worst case, it may result in scanning values of all the transactions. So the complexity is O(N) for hiding one sensitive item. Summing the complexity of all the steps the execution complexity of the algorithm is O(nsr^2 + nsr*nI+ nnsr*nI + N*nSI). Though this algorithm has O(n^2) complexity, the number of database scans is lesser when compared with weighted item grouping algorithm.

6 PERFORMANCE EVALUATION

The performance of the proposed algorithms were measured in terms of number of rules generated before and after hiding the sensitive rules, number of non sensitive rules that were accidentally hidden(lost rules), the number of new rules that were generated as a side effect of hiding sensitive

rule(ghost rule). The effectiveness of the proposed approach is studied based on number of rules mined, number of lost rules and ghost rules generated by setting the following conditions: setting the number of rules to hide as constant, constant minimum support, varying the minimum confidence and varying the number of transactions.

The experiments were conducted using datasets from UCI Machine Learning Repository. The first dataset is breast cancer dataset which consists of one id attribute and nine quantitative attributes with 699 instances. The ID attribute was ignored. The second dataset is Abalone dataset which has seven continuous attributes and 4177 instances**.**

Three different experiments were conducted. The first experiment shows the relationship between the total number of rules mined before hiding and the number of rules mined after applying the hiding algorithm in breast cancer dataset for varying number of transactions. In this experiment, The minimum support value is set as 0.4, the minimum confidence value is set at 0.3%. The result for hiding five sensitive rules is depicted in Fig.1. The results shows that number of rules mined after hiding using Rank-based Correlation analysis is slightly more when compared with hiding using Weighted Item Grouping. The same pattern can be observed when tested with the Abalone dataset as shown in fig.2

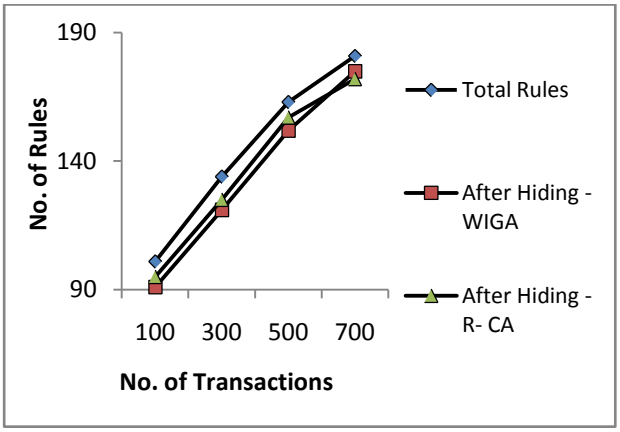

Fig.1 Rules Generation by varying the number of transactions in Breast Cancer Dataset.

The second experiment finds the number of total and hidden rules for different values of minimum confidence with a constant minimum support of 0.5 for 500 transactions using weighted item grouping and Rank-based correlation item hiding algorithm. The results of the experiment for Breast Cancer and Abalone Dataset are shown in Fig.3 and Fig.4 respectively. The results show that the number of rules mined hiding a set of five rules is slightly higher in Rank- based Correlation Item Hiding than in Weighted Item Grouping algorithm.

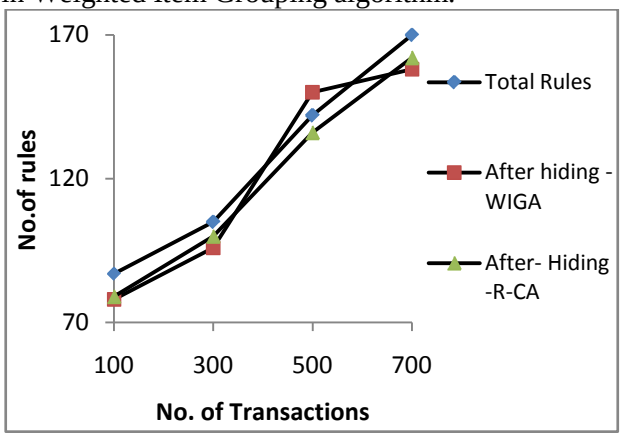

Fig. 2 Rules generation by varying the number of transactions in Abalone Dataset

The third experiment finds the number of lost rules for different number of transactions with constant minimum support of 0.4 and a constant minimum confidence of 0.3 for breast cancer and abalone dataset. The result of this experiment for Weighted item grouping and Rank Based correlation item hiding is depicted in Fig.5 and Fig.6 respectively. The result is compared with the previous work of GA - Based approach(Sathiyapriya,2012). It shows that the number of lost rules in Weighted Item grouping is more when compared with GA based approach and the lost rules in Rank Based Correlation item hiding is lesser when compared with previous work and the Weighted Item grouping algorithm. The Rank Based Correlation Analysis outperforms both the methods.

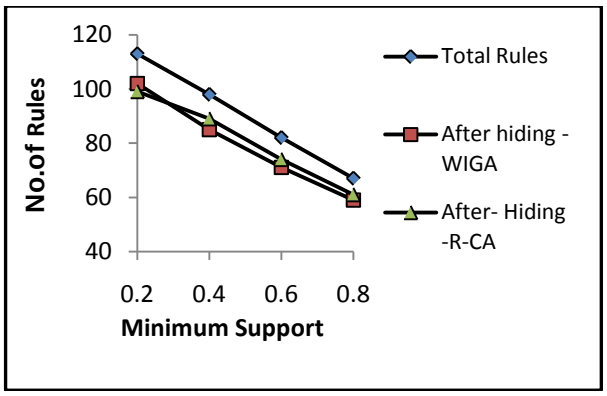

Fig. 3 Rules Generated by varying the minimum confidence in Breast Cancer Dataset

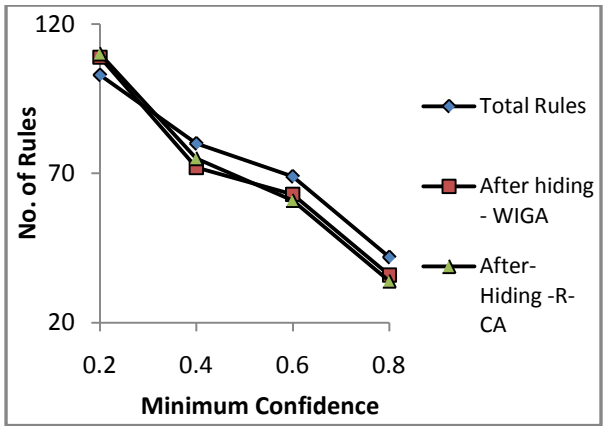

Fig. 4 Rules Generated by varying the minimum confidence in Abalone Dataset

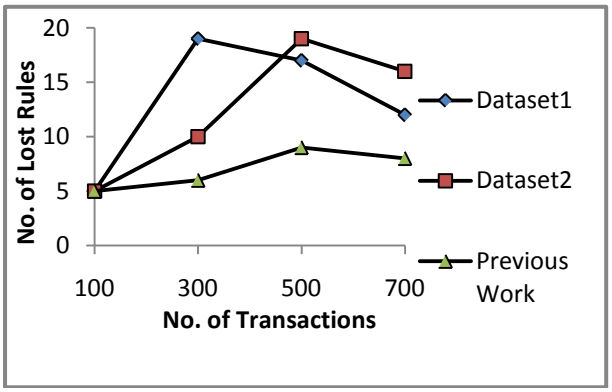

Fig. 5 Number of Rules lost by varying the number of transactions in Weighted Item Grouping Algorithm

The fourth experiment the number of Ghost rules generated for different number of transactions with constant minimum support of 0.4 and a constant minimum confidence of 0.3 for breast cancer and abalone dataset. The result of this experiment for Weighted item grouping and Rank Based correlation item hiding is depicted in Fig.7 and Fig.8 respectively.

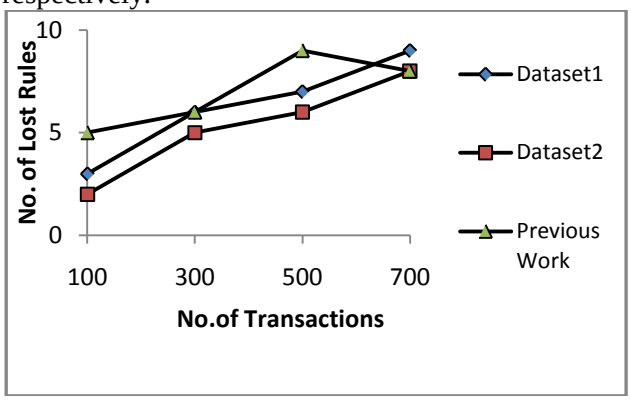

Fig.6 Number of Rules lost by varying the number of transactions in Rank Based correlated item hiding

From the result it can be inferred that the rank based correlated item hiding algorithm generates lesser number of ghost rules when compared with GA-Based approach and Weighted Item Grouping Algorithm. The performance of Weighted Item Grouping and GA - Based approach is almost similar in terms of number of Ghost rules generated.

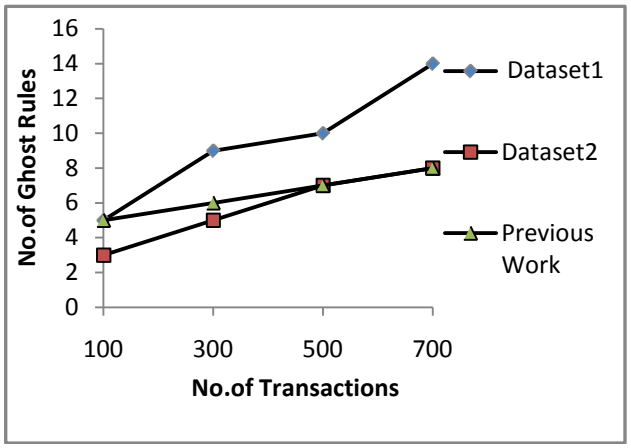

Fig 7. Number of Ghost Rules generated by varying the number of transactions in Weighted Item Grouping Algorithm

A comparison of number of modified entries in weighted item grouping and rank-based Correlated item hiding algorithm with that of the previous work is illustrated in Table 1 and Table 2 respectively. The results shows that the number of entries modified in Weighted Item Grouping is less than the GA Based approach and the number of entries modified in rank based Correlated item hiding is less than those in Weighted Item Grouping.

As the Rank based Correlated Item Hiding algorithm chooses the item that supports lesser non sensitive rules and large number of sensitive rules, among the items in the highly correlated rules, lesser number of modifications in the dataset hides the rule.

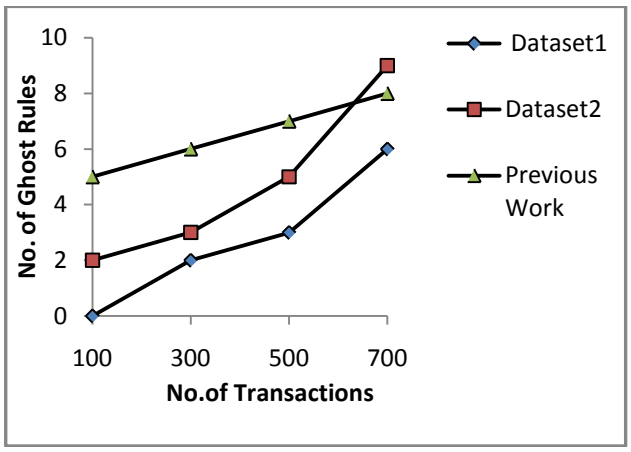

Fig. 8. Number of Ghost Rules generated by varying the number of transactions in Rank Based correlated item hiding

The Weighted Item Grouping algorithm chooses the item that supports lesser non sensitive rules and large number of sensitive rules, among all the rules, the number of modifications is higher. As the GA based approach chooses random transactions the number of modifications is still higher.

Table 1. Modifications in breast Cancer Dataset

No. of Transactions	*No. of Entries*	*Modified Entries*		
		Previous Work	*WIGA*	*R-CA*
100	*800*	*107*	*76*	*61*
300	*2400*	*145*	*95*	*78*
500	*4000*	*234*	*178*	*145*
700	*5600*	*320*	*265*	*182*

Table 2. Modifications in Abalone Dataset

No. of Transactions	*No. of Entries*	*Modified Entries*		
		Previous Work	*WIGA*	*R-CA*
100	*900*	*116*	*96*	*71*
300	*2700*	*351*	*201*	*163*
500	*4500*	*527*	*332*	*275*
700	*6300*	*657*	*479*	*322*

7 CONCLUSION

In this paper, privacy preserving data mining methods for hiding fuzzy association rules is proposed. The goal of the proposed algorithms is to balance the level of privacy and accuracy of dataset as much as possible. The proposed algorithms would allow the parties to share data in a private way with no restrictions. Unlike classical approaches, this method hides the sensitive association rules in quantitative datasets. For this purpose, it employs fuzzy set concept. Experiments conducted on the Breast Cancer and abalone dataset illustrated that the proposed approach produces meaningful results, and limits the side effects than the existing methods. The results of the proposed algorithm are consistent. The number of modifications in the dataset is found to be less in rank based correlation item hiding algorithm than in weighted item grouping algorithm. The drawback of this approach is that the number of lost rules is comparatively high. Since the algorithm clusters the sensitive rules and hides the items in sensitive items, one in a step, there is no hiding failure. Enhancements can be added to increase the effectiveness of the algorithm by minimizing the number of lost rules, number of modifications and the number of database scans.

References

[1] Ramakrishnan Srikant, Rakesh Agrawal," Mining QuantitativeAssociation Rules in Large Relational Tables", Proceedings of the ACM SIGMOD International Conference on Management of Data, 1996.

[2] Yucel Saygin, Vassilios Verykios, and Chris Clifton, "Using Unknowns to Prevent Discovery of Association Rules", SIGMOD Record 30, no.4, 2001, pp.45–54

[3] Vassilios S. Verykios, A.K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association Rule Hiding," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, 2004, pp. 434-447.

[4] Yuhong Guo, Yuhai Tong, Shiwei Tang, Dongging Yang, "A FP-tree-based Method for Inverse Frequent Set Mining," ACM - Proceedings of the 23rd British National Conference on Databases, conference on Databases, 2006, Pp.152-163.

[5]Yuhong Guo, "Reconstruction-Based Association Rule Hiding", Proceedings of SIGMOD2007 Ph.D. Workshop on Innovative Database Research 2007(IDAR2007), Beijing, China, 2007, Pp. 51-56.

[6] T. P. Hong, C. S. Kuo, S. C. Chi, "Mining association rules from quantitative data", Intell. Data Anal. 3 (5), 1999, pp.363–376.

[7] T. Berberoglu and M. Kaya, "Hiding Fuzzy Association Rules in Quantitative Data", The 3rd International Conference on Grid and Pervasive Computing Workshops, 2008, pp. 387-392.

[8] Manoj Gupta and R. C. Joshi, "Privacy Preserving Fuzzy Association Rules Hiding in

Quantitative Data," International Journal of Computer Theory and Engineering, Vol. 1, No. 4, 2009, pp.382-388.

[9] Komal Shah et al, " Association Rule Hiding by Heuristic Approach to Reduce Side Effects & Hide Multiple R.H.S. Items" , International Journal of Computer Applications, Vol.45, No. 1, 2012,Pp.1-7.

[10] Sonia Hameed, Faizal Shahzad,Dr. Sohail Asghar, "A Fuzzy Correlation Scheme For Privacy Preservation In Knowledge Based Systems", Australian Journal of Basic and Applied Sciences. Vol.6 No.9, 2012,pp. 562-571.

[11] K. Sathiyapriya, G. Sudhasadasivam, N. Celin, "A New Method for preserving privacy in Quantitative Association Rules Using DSR Approach With Automated Generation of Membership Function", In the Proceedings of World Congress on Information and Communication Technologies, Mumbai 2011, pp.148-153.

[12] Stanley R.M.Oliveria, Osmar R. Zaiane, "A Unified framework for Protecting Sensitive Association Rules in Business Collaboration", International Journal of Business Intelligence and Data Mining, Vol. 1, No.3, 2006, pp.247 - 287.

[13] Chih-Chia Weng, et.al., "A Novel Algorithm for Completely Hiding Sensitive Frequent Itemset" , Dept. of Information Science, Chung Cheng Institute of Technology, National Defence University, 2007.

[14] Chih-Chia Weng, Shan-Tai Chen, Hung-Che Lo , "A Novel Algorithm for Completely Hiding Sensitive Association Rules" , Eighth International Conference on Intelligent Systems Design and Applications, Vol. 3, 2008,Pp.202-208

[15] S.L. Wang, and A. Jafari, "Using unknowns for hiding sensitive predictive association rules," In Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI 2005), 2005, pp.223–228

[16] K. Sathiyapriya, G. Sudhasadasivam, V.B.Karthikeyan, " A new Method for Preserving Privacy in Quantitative Association Rules using Genetic Algorithm", International Journal of Computer Applications, Vol.60, No.12, Dec 2012.

[17] L.A. Zadeh, "Fuzzy Sets," Information and Control, Vol.8, 1965, pp.338-353.

[18] N V Muthu lakshmi et al, "A Novel Method For Finding Privacy Preserving Association Rule Mining", Indian Journal of Computer Science and Engineering *(IJCSE)*,Vol.3, No.1, 2012,Pp.104-113.

[19] http://mlearn.ics.uci.edu/databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data

[20] S.L.Wang, S. Ito, and A. Jafari, "Using Unknown for Hiding Sensitive Items in Association Rule Mining", WSEAS Transactions on Information Science and Applications, Issue 1, Volume 1, 2004, 489-494.

[21] W.Q. Sun, C.M. Wang, T.Y. Zhang, Y. Zhang ,"Transaction-item association matrix-based frequent pattern network mining algorithm in large-scale transaction database", WSEAS Transactions on Computers, Issue 8, Volume 8, Pp.1327-1336, Aug 2009.

[22] M. Marian, S. Saddys, F.L. Vivian, M.J. Polo, "A method for mining quantitative association Rules", Proc. of 6th WSEAS International Conference on Simulation, Modeling and Optimization, Lisbon, Portugal, September 22- 24, 2006, pp. 173-178.

[23] T.H.S. Alex, I. Maria, S. Bala, "Mining infrequent and interesting rules from transaction records, Proc. of 7th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'08), University of Cambridge, UK, Feb 20-22, 2008, pp. 515-520.