# Privacy Preservation with Attribute Reduction in Quantitative Association Rules using PSO and DSR

G.Sudha Sadasivam
Phd, Professor
Department of Computer Science & Engineering
PSG College of Technology
Coimbatore,
Tamil Nadu – 641 004, India

S.Sangeetha
Lecturer
Department of Information Technology
PSG Polytechnic College
Coimbatore,
Tamil Nadu – 641 004, India

K.Sathya Priya
Assistant Professor
Department of Computer Science & Engineering
PSG College of Technology
Coimbatore,
Tamil Nadu – 641 004, India

## ABSTRACT

Data mining aims at extracting hidden information from data. Data mining poses a threat to information privacy. Privacy preserving data mining hides the sensitive rules and prevents the data from being disclosed to the public. Attribute reduction techniques reduce the dimensionality of dataset. Rough sets are used for attribute reduction to yield reduced sets. An attribute reduct is a subset of attributes formed using rough sets. This paper proposes two approaches to hide sensitive fuzzy association rules namely, decreasing support value of item in RHS of association rule and Particle Swarm Optimization (PSO). The proposed approach is implemented using map reduce paradigm. Experimental results demonstrate the performance of the proposed approach.

## General Terms

Data mining, privacy preservation in data mining, DSR based data hiding,rough set approach to attribute reduction and PSO based data hiding.

## Keywords

Rough sets; attribute reduction; map reduce; discernibility matrix; PSO; privacy preserving data mining; fuzzification ; DSR ; quantitative association rule ; lost rule ; ghost rule.

## 1. INTRODUCTION

Feature selection or attribute reduction is essential, as the dataset of a data mining model frequently contains more information than is needed to build the model. For example, a dataset may contain 500 columns that describe characteristics of customers, but perhaps only 50 are useful. Maintaining redundant attributes impacts on cpu and memory efficiency during training and testing process. Redundant features degrade the quality of patterns mined.Noise makes it more difficult to discover meaningful patterns from the data. Redundant or irrelevant data may lead to poor results when using the dataset in knowledge discovery.

Feature selection is a process which chooses a subset of the original features present in a given dataset which provides the most useful information. Thus feature selection maintains the attributes that characterises the data set. Data set quality increases through feature selection of dataset [8] Classification accuracy can be increased as a result of feature selection through the removal of noisy, irrelevant, or redundant features. Execution time of learning algorithms can also be improved significantly. When there are fewer attributes, identification of trends and correlations within the data becomes easier [4]. A good feature selection algorithm removes unnecessary attributes that affect both rule comprehension and rule prediction performance.

Data mining, deals with discovering new patterns from large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics and database systems. The goal of data mining is to extract knowledge from a data set in a human-understandable structure. The problem of privacy-preserving data mining has become important in recent years because of the increasing sophistication of data mining algorithms to leverage personal information from datasets. Data mining has been viewed as a threat to privacy because of the widespread proliferation of electronic data maintained by corporations. This has lead 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 so as to preserve privacy. The aim of privacy preserving data mining is to design methods which continue to be effective, without compromising security.

Hadoop is a large-scale reliable software infrastructure that supports data intensive distributed applications. It implements a computational paradigm called, Map-Reduce which allows the user to create different applications that can be executed in hadoop infrastructure. It provides a framework for processing huge datasets on distributed infrastructure. The phases involved in map reduce are:

- "Map": The master node takes the input, partitions it up into smaller sub-problems, and distributes it to worker nodes. The worker node processes it and passes the result to the reducer.
- "Reduce": The reducer takes the answers of all the sub-problems from mappers and combines them to get the output.

The proposed approach uses roughsets to reduce dataset dimensionality. Fuzzification is used to mine fuzzified association rules.Then Decreasing the support of right hand side (DSR) and Particle Swarm Optimization (PSO) based metaheuristic approach is used to hide sensitive association rules from the data. A mapreduce model is used to foster parallelism. This combined effect of feature selection followed by data hiding produces good results.

## 2. LITERATURE SURVEY

Attribute reduction (also called feature selection ) is a key problem in the field of the machine learning [10]. Feature selection is the process of selecting relevant attributes. As it reduces the dimensionality of the data, it enables the learning algorithms to operate more effectively and rapidly. Rough Set based attribute reduction (RSAR) reads in a dataset and eliminates redundant attributes that do not characterize the dataset. As most real world datasets have large amount of redundancy, RSAR can be used as a preprocessor to speed up training using other artificial intelligence systems[13]. The rough set theory based attribute reduction can roughly be divided into the following categories: discernibility matrix-based method [11], information entropy-based methods[13],positive region-based methods[5] and other evolutionary methods [2]. Our work uses a mapredce version of discernability matrix-based method.

The process of construction of a core [10],[6] is described as follows. Consider a decision table T=(U,C,D) with a universe U={u1,u2,…un} of 'n' attributes and C represents conditional attributes while D represents decision attributes. C ∩ D = Φ. Such that For the information system, S=(U, C U D) for every subset B of A indiscernability relation IND(B)is defined as{(x,y) ε U X U: a(x)=a(y) for every aεB}.Given $X \subseteq U$ is a set of objects and $B \subseteq A$ is a selected set of attributes. The lower approximation of X with respect to B is $B_*(X) = \{x \in U : [x]_B \subseteq X\}$.The upper approximation of X with respect to B is $B^*(X) = \{x \in U : [x]_B \cap X\} \neq \phi$. The positive region of decision d ε D with respect to B is POSB(d)= U{B*(X) :XεU/IND(d)}

The positive region of decision attribute with respect to B represents approximate quality of B. Not all attributes are necessary while preserving approximate quality of original information system. Reduct is the minimal set of attribute that preserve approximate quality.A reduct of B is a set of attributes B' $\subseteq$ B such that POSB(d)= POSB'(d), and there is no C $\subseteq$ B' such that POSB(d)= POSC(d). The intersection of all reducts is called core of the information system.

Discernibility matrix of T, denoted M(T), is a n×n matrix defined as:

$$m_{ij} = \begin{cases} \{c \in C: c(u_i) \neq c(u_j)\} & if\ \exists\, d \in D[d(u_i) \neq d(u_j)] \\ \lambda & if\ \forall\, d \in D[d(u_i) = d(u_j)] \end{cases} \quad (1)$$

for i, j=1,2…n

mij is the set of all the condition attributes that classify objects ui and uj into different classes.A discernibility function is constructed from a discernibility matrix by or-ing all attributes in cij and then and-ing all of them together. After simplifying the discernibility function using absorption law, the set of all prime implicants determines the set of all reducts of the information system.

With the development of data analysis and processing technique, organizations, and governments are increasingly publishing microdata. Microdata contains unaggregated information about individuals for data mining purposes. While the released data sets provide valuable information to researchers, they also contain sensitive information about individuals whose privacy may be at risk. Providing security to sensitive data against unauthorized access has been a long term goal for the database security research community and for the government statistical agencies.

Privacy Preserving Data Ming (PPDM) algorithms are mainly used on the tasks of classification, association rule and clustering[15],[7]. According to privacy protection technologies privacy preserving association rule mining algorithms commonly can be divided into two broad categories [12],[16].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. Blocking based technique is characterized by introducing uncertainty without distorting the database. In our approach we propose to use distortion based technique.

The basic idea in quantitative data mining is to map the categorical attribute values into corresponding binary attribute values. The rule hiding techniques based on fuzzy methods requires membership function to be specified by an expert. These algorithms either use Increasing the Support of Left Hand Side (ISL) or DSR approaches to hide sensitive association rules. However, only two work has been done in the field of hiding fuzzy association rule in quantitative data. One is by T.Bergeroglu [3] who proposed an algorithm to hide fuzzy association rule in quantitative data[3]. The basic idea of this algorithm was to decrease the confidence of a rule by increasing support of L.H.S. of rule. However the works require the membership function to be predefined, thus are usually built by human experts or experienced users. If the experts are not available, then the membership functions cannot be accurately defined, or the fuzzy systems developed may not perform well[14].

A method to hide sensitive fuzzy association rule [1] in which the data is fuzzified, then modified using APRIORI algorithm is used to extract rules and identify sensitive rules. The sensitive rules are hidden by DSR approach. A learning method is also proposed for automatic derivation of fuzzy membership function.

A novel particle swarm optimization trained auto associative neural network for privacy preservation is used then decision tree and logistic regression are invoked for data mining [9].

Our proposed approach uses MapReduce version of rough sets for attribute set reduction, to identiy the independent attributes. It then uses fuzzification and MapReduce version of DSR for hiding sensitive rules. We have also proposed usage of PSO for hiding sensitive rules

## 3. PROPOSED APPROACH

The workflow of the proposed approach is depicted in fig1.

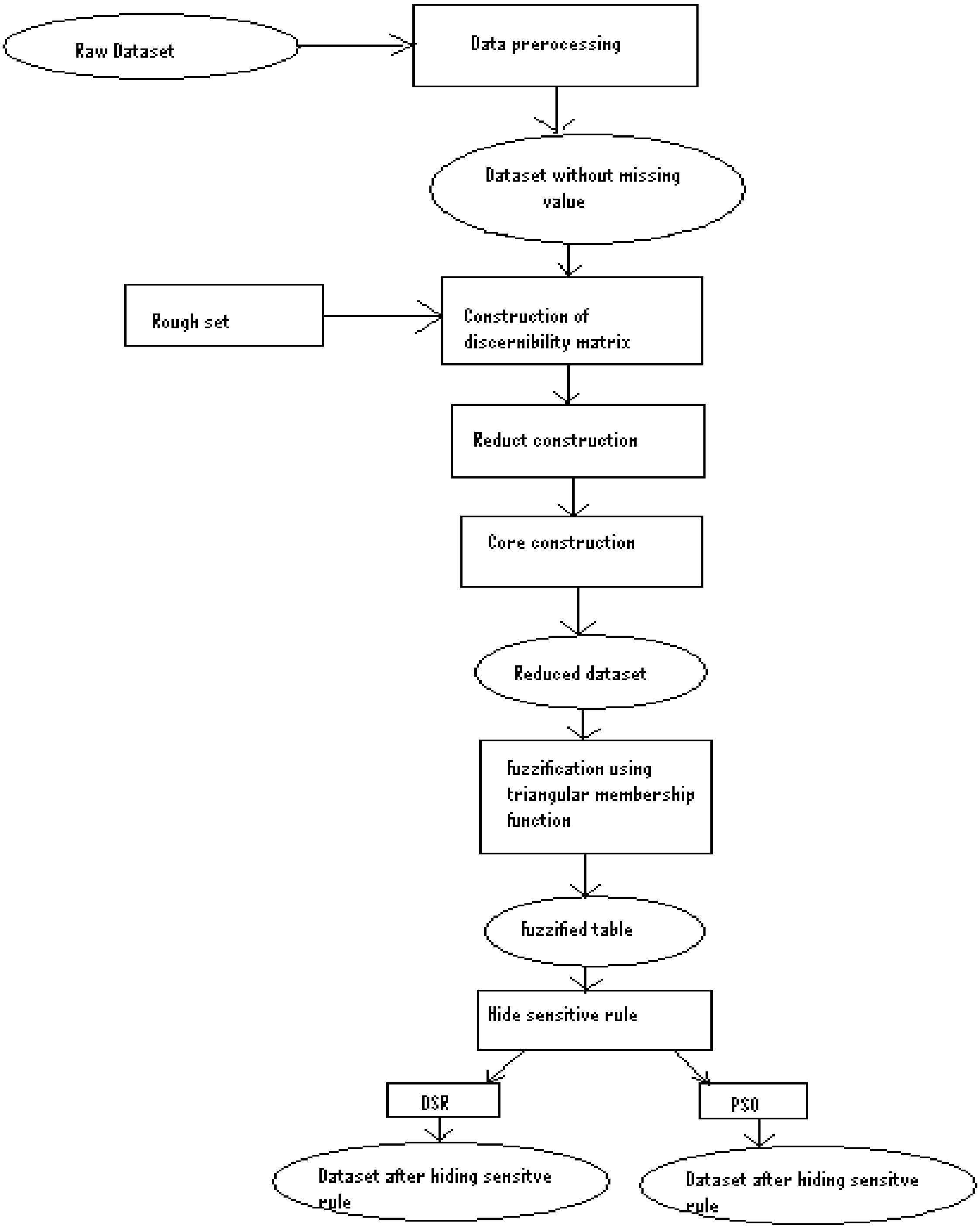
Raw Dataset
Data prerocessing
Dataset without missing value
Rough set
Construction of discernibility matrix
Reduct construction
Core construction
Reduced dataset
fuzzification using triangular membership function
fuzzified table
Hide sensitive rule
DSR
PSO
Dataset after hiding sensitve rule
Dataset after hiding sensitve rule

**Fig1. Workflow of the proposed Approach**

The various steps in the proposed approach is described as follows.

## 3.1. Data Collection

The dataset is collected from UCI machine learning repository. The data contains attributes like sample code number, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses, and class.

## 3.2. Data Preprocessing

The missing value of attribute is identified and replaced by the maximal occurring value in the corresponding column. A map reduce process for the same consists of two phases

MAP: Read each transaction into a mapper and collect values using attribute or column number as key.

REDUCE: Read in attribute values grouped by column number. Store the values, identify and replace the missing values by maximum occurring value in the column and write the processed output.

Figure 2 illustrates the workflow of preprocessing using map reduce.

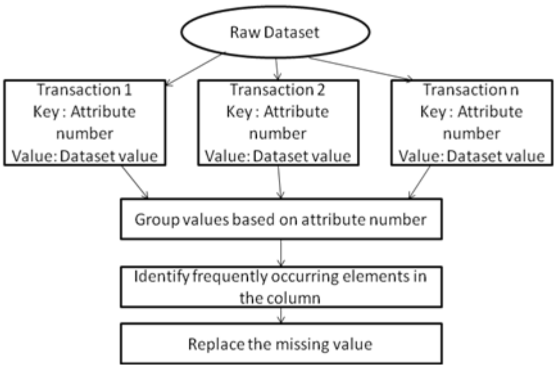

**Fig2. Preprocessing map reduce**

## 3.3. Attribute Reduction

Discernibility matrix is constructed and the core of final reduct set is identified.

MAP: Read in one transaction in a mapper and compare it with transactions to create one row of discernibility matrix.

REDUCE:Read in portions of discernibility matrix from mapper and construct reduct core. Get the output from the output collector which is sorted according to the number. Store the values, perform intersection between the reducts to form core and get the exact reduct attributes of the dataset.

Figure 3 depicts attribute reduction using map reduce.

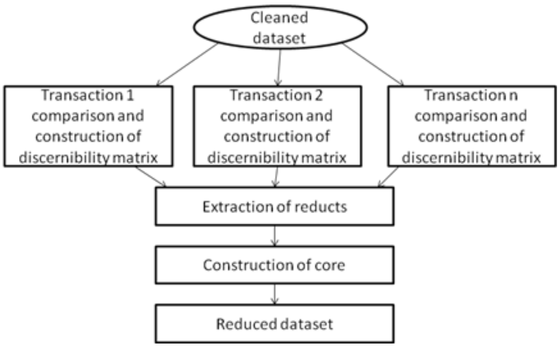
Cleaned dataset
Transaction 1
comparison and construction of discernibility matrix
Transaction 2
comparison and construction of discernibility matrix
Transaction n
comparison and construction of discernibility matrix
Extraction of reducts
Construction of core
Reduced dataset

**Fig 3. Attribute Reduction map reduce**

## 3.4. Fuzzification and Sensitive Rule Hiding using DSR

Sensitive rules are hidden and confidence value before and after hiding are obtained correspondingly.

The map reduce procedure for the same is described as follows

MAP: Read in one transaction (from reduced dataset) and fuzzify the data using triangular membership function. Produce the fuzzified value of a transaction as intermediate output to the reducer

REDUCE: The fuzzified values are grouped according to the column number and rules are mined. Sensitive rules are identified and support values of right hand side attribute is reduced. Confidence value before and after hiding is displayed to the user.

Figure 4 depicts sensitive rule hiding using map reduce

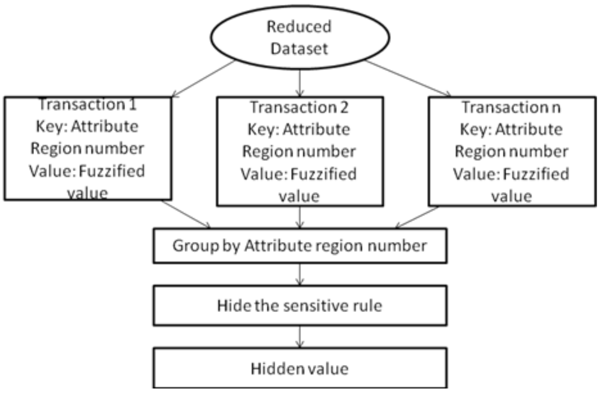

**Fig 4. Sensitive Rule Hiding DSR map reduce**

## 3.5. Sensitive Rule Hiding using PSO

PSO is based on the sociological behaviour associated with bird flocking. The algorithm maintains a population of particles where each particle represents a potential solution to an optimization problem. Let s be the size of the swarm. Each particle ‘i’ can be represented as an object with several characteristics. These characteristics are assigned the following symbols:

$X_i$ : The current position of the particle
$V_i$ : The current velocity of the particle
$Y_i$ : The personal best position of the particle

The personal best position associated with the particle i is the best position that the particle has visited, yielding the lowest fitness value for minimisation problem. The symbol f is used to denote the objective function that is being minimised. The update equation for the personal best position is denoted as follows

$$Yi(t+1) \geq \begin{cases} Yi(t) & if\ f\left(Xi(t+1) \geq f\left(Yi(t)\right)\right) \\ Xi(t+1) & if\ f\left(Xi(t+1)\right) < f\left(Yi(t)\right) \end{cases} \tag{2}$$

Two versions of PSO exists called the lbest and gbest models. The difference between the two algorithms ia based on the set of particles with which a given particle will interact directly, where the symbol y will be used to represent this interaction. The definition of used in the $\hat{y}$ gbest model is as follows:

$$\bar{Y}(\bar{t}) \in \{Y_0(t), Y_1(t), \ldots, Y_s(t)\} \mid f\left(\bar{y}(t)\right)$$
$$= \min\left(f\left(Y_0(t)\right), f\left(Y_1(t), \ldots, f\left(Y_s(t)\right)\right)\right) \tag{3}$$

This definition states that Y is the best position discovered by any of the particle so far. The algorithm makes use of two independent random sequences , r1 ~ U(0,1) and r2 ~ U(0,1). The values of r1 and r2 is scaled by constants 0<c1, c2<=2. These constants are called the acceleration coefficients, and they influence the maximum size of the step that particle can take in a single iteration. The velocity update step is specified separately for each dimensionj 1,….n, so that Vi,j denotes the jth dimension of the velocity vector associated with the ith particle. The velocity update equation is then

$$Vi,j(t+1) = Vi,j(t) + c1r1,\underline{j(t)}\,[Yi,j(t) - \cancel{X}i,j(t)] + c2r2,j(t)\left[\hat{Y}j(t) - Xi,j(t)\right] \quad (4)$$

## 4. ALGORITHMS USED

This section details on feature selection procedure using rough set and hiding fuzzy rules using DSR.

### 4.1. Feature selection using rough set

The procedure is as follows.

1. Obtain the reduced dataset

2. Construct discernibility matrix based on the indiscernible and discernible relationship

$$mij = \{\{c \in C : c(ui) \neq c(uj)\} if\ \exists d \in D[d(ui) \neq d(uj)]\} \qquad (5)$$

3. Perform adsorption operation on the matrix to remove supersets. This identifies discernible elements.

4. Perform union on the elements of subset and intersection between the subsets to construct core.

5. The attributes without redundancy is retained and the database is updated.

An illustration of the working of the proposed algorithm is as follows

STEP 1: Cleaning

The database as in Table 1 is cleaned by substituting the maximum occurring element in the column. After cleaning the dataset Table 2 is obtained. During cleaning maximum occurring element in the attribute is replaced for missing element.

**Table 1: Dataset with missing values**

| | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| T2 | 5 | 4 | 4 | 5 | 7 | 1 | 3 | 2 | 1 | 2 |
| T3 | 3 | 1 | 1 | 1 | 2 | ? | 3 | 1 | 1 | 2 |
| T4 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 2 |
| T5 | 4 | 1 | 1 | 3 | 2 | ? | 3 | 1 | 1 | 2 |

**Table 2: Cleaned Dataset**

| | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 2 |
| T2 | 5 | 4 | 4 | 5 | 7 | 1 | 3 | 2 | 1 | 2 |
| T3 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 2 |
| T4 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 2 |
| T5 | 4 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 2 |

STEP 2:Construction of Discernibility Matrix

Discernibility matrix is constructed as given in table3. As shown in table3, between T1 and T2 attributes A1,A2, A3,A4 and A7 are discernible.

STEP 3: Adsorption

The supersets {A0,A1,A2,A3,A4,A5,A7}, {A0,A1,A2,A4,A5,A7} and {A0,A3,A5} are removed. The subsets obtained are {A1,A2,A3,A4,A7} , {A0,A5} AND {A0,A3}.

STEP 4: Construction of core

Union and intersection of the subsets obtained from the previous step is performed as follows

{A1vA2vA3vA4vA7}^{A0vA5}^{A0vA3}

This yields core elements and the final reduced set of attributes are {A0,A1,A2,A3,A4,A5,A7}.Hence the reduced set of attributes are given in table4

**Table 3: Discernibility matrix**

| | T1 | T2 | T3 | T4 |
|---|---|---|---|---|
| T2 | A1,A2,A3,A4,A7 | | | |
| T3 | A0,A5 | A0,A1,A2,A3,A4,A5,A7 | | |
| T4 | A0,A1,A2,A4,A5,A7 | A0,A1,A2,A3,A4,A5,A7 | A0,A1,A2,A4,A5,A7 | |
| T5 | A0,A3 | A0,A1,A2,A3,A4,A7 | A0,A3,A5 | A0,A1,A2,A3,A4,A5,A7 |

**Table 4: Reduced attributes**

| | A0 | A1 | A2 | A3 | A4 | A5 | A7 |
|---|---|---|---|---|---|---|---|
| T1 | 5 | 1 | 1 | 1 | 2 | 1 | 1 |
| T2 | 5 | 4 | 4 | 5 | 7 | 1 | 2 |
| T3 | 3 | 1 | 1 | 1 | 2 | 2 | 1 |
| T4 | 6 | 8 | 8 | 1 | 3 | 4 | 7 |
| T5 | 4 | 1 | 1 | 3 | 2 | 1 | 1 |

## 4.2 Hiding sensitive fuzzy association rules using DSR

In a quantitative database, if a critical rule $X \rightarrow Y$ needs to be hidden, its confidence value is decreased to a value smaller than the minimum confidence value. One way of decreasing confidence value is decreasing the support value of an item Y at RHS, and the other way is increasing the support value of item X at LHS. Our approach decreases confidence value of a rule, by decreasing the support value of RHS item. If the value of item in RHS is greater than 0.5 and value of item in LHS then its value is subtracted from 1.

Abbreviations used in the proposed algorithm are given as follows:
C : Reduced dataset with ‘n’ transactions
F : Fuzzified database
X : A set of predicting items
TL : Transactions belong to a LHS item
TR :Ttransactions belong to a RHS item
U : Rule
Rh : sensitive rule

**Input:**
(1) Reduced dataset
(2) Minimum support value (min_support),
(3) Minimum confidence value (min_confidence).

**Output:**
A transformed database D’ so that useful fuzzy association rules cannot be mined.

**Algorithm DSR:**

```
1. Reduced dataset
2. Fuzzification of the cleaned database, C → F;
3. In fuzzified database F, calculate every item’s support value where f→F;
4. IF all f (support) < min_support THEN EXIT; // there isn’t any rule
6. Find large 2-itemsets from F;
7. FOR EACH X’s large 2-itemset //find all rules
          Find R = {Rules from itemset X};
          //for X= {i1, i2}, rules are i1 → i2, i2 → i1.
          Compute confidence of the rule U;
          IF confidence (U) > min_confidence and sensitive THEN
             Add the rule U to Rh;
          end//if
     end//end of FOR EACH
//Hides all rules in Rh
8. FOR EACH U in Rh {//until no more rule can be hidden
          FOR EACH TR of rule{
            if TR >0.5 and TR > TL
           TR = 1 - TR
            end // if
          end // FOR EACH.
Re-calculate confidence value of rule U
          if rule U(confidence) > min_confidence
          FOR EACH TR of the rule
            if TR = 1.0
            TR = 0.0
            end// if
          end // FOR EACH
          else go to step 9
          end //if
9. Transform the updated database F to D’  and output updated D’
10. end
```

An illustration of the working of the proposed algorithm is as follows

STEP 1: Cleaning

The database after attribute reduction is taken as input, as in Table 5

**Table 5: Reduced dataset**

| | A0 | A1 | A2 | A3 | A4 | A5 | A7 |
|---|---|---|---|---|---|---|---|
| **T1** | 5 | 1 | 1 | 1 | 2 | 1 | 1 |
| **T2** | 5 | 4 | 4 | 5 | 7 | 1 | 2 |
| **T3** | 3 | 1 | 1 | 1 | 2 | 2 | 1 |
| **T4** | 6 | 8 | 8 | 1 | 3 | 4 | 7 |
| **T5** | 4 | 1 | 1 | 3 | 2 | 1 | 1 |

STEP 2: Fuzzification

The database as shown in table 5 is fuzzified using triangular membership function given in equation (1) into 3 regions Z, O, B as shown in fig5.The fuzzified details shown in table 6 .

$$\mu=\mathrm{Max}\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \qquad (6)$$

Where a is the left end of the triangle, b is the peak of the triangle and c is the right end of the triangle (values are the corresponding x axis values)

STEP 3:

Calculate the support count of each attribute region, *R* on the transactions data by summing up the fuzzy values of all the transactions in the fuzzified transaction data as in table 6.

**Table 6: Fuzzified dataset**

| **T** | **A0** | | | **A1** | | | **A2** | | | **A3** | | | **A4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **N** | $A_0z$ | $A_0o$ | $A_0b$ | $A_1z$ | $A_1o$ | $A_1b$ | $A_2z$ | $A_2o$ | $A_2b$ | $A_3z$ | $A_3o$ | $A_3b$ | $A_4z$ | $A_4o$ | $A_4b$ |
| **T1** | 1.0 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.4 | 0.0 | 0.0 |
| **T2** | 1.0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.0 | 0.8 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.6 | 0.4 | 0.0 |
| **T3** | 0.6 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.4 | 0.0 | 0.0 |
| **T4** | 0.8 | 0.2 | 0.0 | 0.4 | 0.6 | 0.0 | 0.4 | 0.6 | 0.0 | 0.2 | 0.0 | 0.0 | 0.6 | 0.0 | 0.0 |
| **T5** | 0.8 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.0 | 0.0 | 0.6 | 0.0 | 0.0 | 0.4 | 0.0 | 0.0 |
| **Count** | 4.2 | 0.2 | 0.0 | 1.8 | 0.6 | 0.0 | 1.8 | 0.6 | 0.0 | 2.2 | 0.0 | 0.0 | 2.4 | 0.4 | 0.0 |

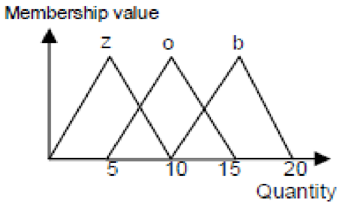

**Fig 5: Triangular Membership Function**

STEP 4:

Check whether count of each attribute is greater than or equal to the predefined minimum support value. If an attribute satisfies the above condition, put it in the set of large-2 itemsets (L2). Consider the minimum support is set to 2.0 and minimum confidence to 75%. The regions A0z, A3z and A4z have their support value greater than minimum support, so are considered in forming the rules and finding the corresponding confidence value. The rules can be $A_0z \rightarrow A_4z$, $A_0z \rightarrow A_3z$, $A_4z \rightarrow A_3z$, $A_4z \rightarrow A_0z$, $A_3z \rightarrow A_0z$, $A_3z \rightarrow A_4z$. Consider $A_6z \rightarrow A_0z$ is a critical rule to be hidden and the support of the rule is calculated as Support($A_3z \rightarrow A_0z$) = min($A_3z$, $A_0z$) as shown in table 7.

**Table 7: Fuzzy values of $A_0z$ and $A_3z$**

| | $A_3z$ | $A_0z$ | Support |
|---|---|---|---|
| T1 | 0.2 | 1.0 | 0.2 |
| T2 | 1.0 | 1.0 | 1.0 |
| T3 | 0.2 | 0.6 | 0.2 |
| T4 | 0.2 | 0.8 | 0.2 |
| T5 | 0.6 | 0.8 | 0.6 |
| Count | 2.2 | | 2.2 |

STEP 5:

For each two large itemsets, based on user specified minimum confidence value, rules are extracted. Confidence value of A→B rule is computed as follows:

$$Confidence(A \rightarrow B) = \frac{Support(AB)}{Support(A)} \qquad (7)$$

The confidence value is calculated for the rule $A_6z \rightarrow A_0z$

Confidence($A_3z$->$A_0z$) = 2.2 / 2.2 = 100%

STEP 6:

To hide a critical rule, its confidence value is decreased by decreasing support(AB). In order to hide the rule $A_3z \rightarrow A_0z$, the support ($A_3zA_0z$) is reduced by subtracting the transaction value of A0z from 1 when the value of Co is greater than 0.5 and corresponding $A_6z$'s value. Using this procedure the support values of transaction T3, T4 and T5 are reduced as shown in table 8

**Table 8: Modified T3, T4 and T5**

|  | $A_3z$ | $A_0z$ | Support |
|---|---|---|---|
| T1 | 0.2 | 1.0 | 0.2 |
| T2 | 1.0 | 1.0 | 1.0 |
| T3 | 0.2 | 0.4 | 0.2 |
| T4 | 0.2 | 0.2 | 0.2 |
| T5 | 0.6 | 0.2 | 0.2 |
| Count | 2.2 |  | 1.8 |

Now Confidence($A_3z \rightarrow A_0z$) =1.8/2.2=81%

Since the confidence is still greater than minimum confidence, in those transactions that have A6z and A0z value as 1, A0z is replaced with 0 as shown in table 9.

**Table 9: Modified T2**

|  | $A_3z$ | $A_0z$ | Support |
|---|---|---|---|
| T1 | 0.2 | 1.0 | 0.2 |
| T2 | 1.0 | 0.0 | 0.0 |
| T3 | 0.2 | 0.4 | 0.2 |
| T4 | 0.2 | 0.2 | 0.2 |
| T5 | 0.6 | 0.2 | 0.2 |
| Count | 2.2 |  | 0.8 |

Now Confidence(A3z →A0z) =0.8/2.2=36%

## 4.3. Algorithm for hiding sensitive fuzzy association rules using PSO

$X_i$ : The current position of the particle

$V_i$ : The current velocity of the particle

$Y_i$ : The personal best position of the particle

```
Create and initialize an n-dimensional PSO: S
Repeat:
        for each particle i ∈ [1,……S] :
                If f(S.Xi) < f(S.Yi)
Then S.Yi = S.Xi
                If f(S.Yi) < f(S.$\hat{Y}$)
                        Then S.$\hat{Y}$ = S.Yi
        End for
Perform PSO updates on S using equations 3 and 4
Until stopping condition is true
```

Work flow of the PSO algorithm is depicted in fig.6.

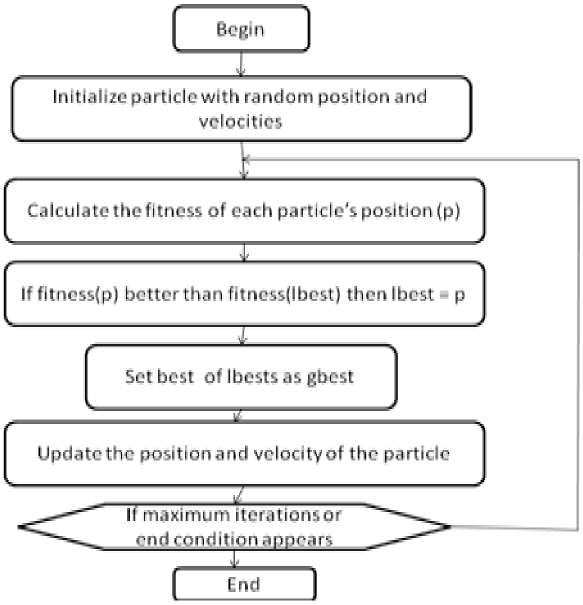
Begin
Initialize particle with random position and velocities
Calculate the fitness of each particle's position (p)
If fitness(p) better than fitness(lbest) then lbest = p
Set best of lbests as gbest
Update the position and velocity of the particle
If maximum iterations or end condition appears
End

**Fig 6: PSO algorithm**

*FITNESS FUNCTION*

$$f(T_i) = \sum_{j=1}^{m} \left( \frac{T_i(a_j)}{S(a_j)} \right) \qquad (8)$$

Where $$S(a_j) = \sum_{i=1}^{n} T_i\,(a_j) \qquad (9)$$

m – No. of attributes , n – No. of transactions, f($T_i$) – Fitness for a transaction $T_i$ and S($a_j$) – Support of attribute $a_j$

An illustration of the working of the proposed algorithm is as follows

STEP1: Input

Sensitive rules are identified and the corresponding attributes involved in the rules are given as input as shown in table10.

**Table10. Input particle**

| Transactions | A0z | A3z | A4z |
|---|---|---|---|
| **T1** | 1.0 | 0.2 | 0.4 |
| **T2** | 1.0 | 1.0 | 0.6 |
| **T3** | 0.6 | 0.2 | 0.4 |
| **T4** | 0.8 | 0.2 | 0.6 |
| **T5** | 0.8 | 0.6 | 0.4 |
| **Count** | 4.2 | 2.2 | 2.4 |

STEP2: Fitness function

Fitness function is calculated by considering each transaction as a particle, the value is divided by the sum of the corresponding attribute as given in table 11.

**Table11. Fitness of all transactions**

| Transactions | A0z | A3z | A4z | Fitness |
|---|---|---|---|---|
| **T1** | 1.0 | 0.2 | 0.4 | 0.49 |
| **T2** | 1.0 | 1.0 | 0.6 | 0.94 |
| **T3** | 0.6 | 0.2 | 0.4 | 0.4 |
| **T4** | 0.8 | 0.2 | 0.6 | 0.53 |
| **T5** | 0.8 | 0.6 | 0.4 | 0.62 |
| **Count** | 4.2 | 2.2 | 2.4 | |

Since hiding is a minimization problem the particle with minimum fitness survives to the next generation.

STEP3: gbest and lbest

For first iteration Xi = Yi which is the local best. For further iteration current fitness is compared with the previous fitness and the particle with minimum fitness is selected as lbest. Gbest is the global best position from all the iterations.

STEP4: Velocity

Current velocity of the particle is calculated using the equation 4

STEP5: New Xi value

After calculating vi value the new generation xi value is updated based on equation 5

STEP6:

Above steps are continued until stopping condition is true. Number of generations taken for the convergence of the proposed algorithm is 25.

## 5. EXPERIMENTAL RESULTS

We conducted experiments based on the breast cancer dataset and the results are analyzed. The original dataset with missing values are cleaned and reduced using rough set theory. The original dataset contains 10 attributes among which 2 attributes are identified as irrelevant and redundant based on rough set approach and it is removed from the dataset.

After dimensionality reduction the sensitive rules are hidden using DSR approach and PSO approach.Six different experiments are performed to compare the performance of the proposed algorithm with and without attribute reduction (rough set). The first experiment finds the relationship between number of total and hidden rules, and number of transactions. In this experiment, the minimum confidence value is set at 70% and minimum support values are taken as 56, 88, 103 and 176 for 300, 500, 600 and 699 transactions respectively. The results are depicted in Fig. 7. With roughest noisy or irrelevant information’s are removed and only the important rules are retained.

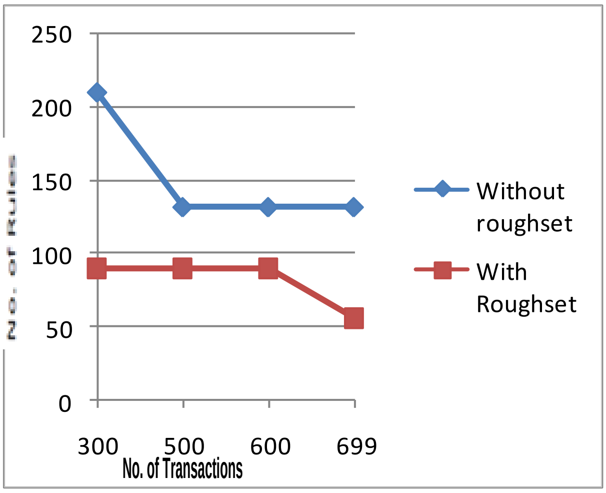

**Fig7: Number of Total and Hidden Rules**

The second experiment finds the number of total and hidden rules for different values of minimum support for 699 transactions with and without using rough set. The minimum confidence value is set at 70%.The results are depicted in Fig.8.Number of rules are reduced by using the roughest approach. The main reason is that only independent attributes are maintained after dimensionality reduction. So the number of rules for the same support is also reduced.

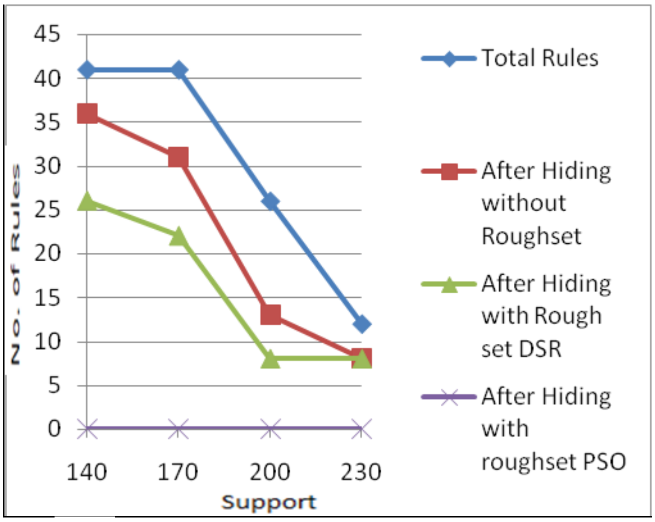

**Fig8: Number of Rules under different values of minimum support**

The third experiment finds the number of total and hidden rules for different values of minimum support for 699 transactions with and without attribute reduction. The minimum support value is set at 170.The results are depicted in Fig.9. Again rough set with PSO performs well.

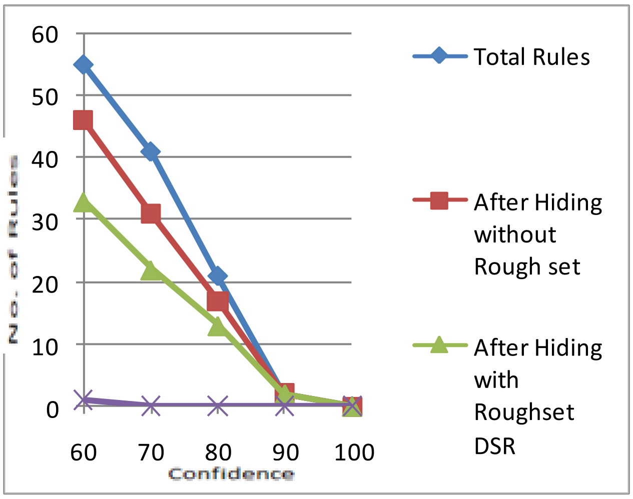

**Fig 9: Number of Rules under different values of minimum confidence**

The fourth experiment finds the relationship between number of lost rules with different values of confidence .The minimum support value is set at 170.The results are depicted in Fig.10.Here more rules are lost in PSO approach.

The fifth experiment finds the relationship between the number of new rules generated as side effect by hiding process. The results are depicted in Fig. 11. No ghost rules are generated.

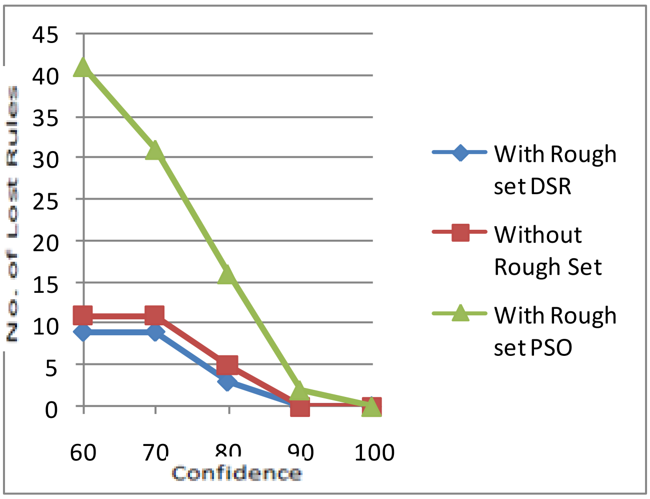

**Fig10: Rules lost after hiding a set of four rules under different values of confidence**

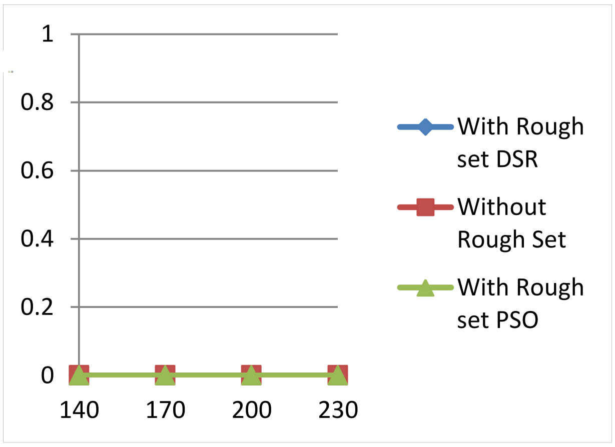

**Fig11. New Rules generated for hiding a set of four rules**

The final experiment finds the number of modifications that are made in the dataset for hiding four sensitive rules. The modifications are analyzed by varying the number of transactions. Number of modifications is higher using PSO approach. The results are depicted in Fig. 12. PSO approach needs more number of modifications hence DSR with roughset proves to be an efficient approach.PSO can be enhanced to minimize the number of modifications.

## 6. ANALYSIS OF COMPUTATIONAL COST

### 6.1. Computational Cost for Preprocessing

- Complexity without map reduce

$$O(m * n)$$

- Complexity with map reduce

$$O\left(\frac{m * n}{N}\right)$$

$$O(m) \qquad \text{if } n = N$$

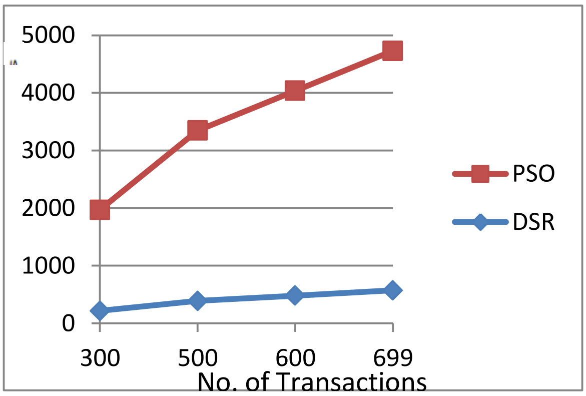

**Fig12. Number of modifications**

Where

m – No. of attributes, n – No. of transactions
N – No. of mappers

## 6.2. Computational Cost for Attribute Reduction

- Computational cost without map reduce

$$O\left(\frac{n^2*m}{2}\right)$$

- Computational cost with map reduce

$$O\left(\frac{n^2*m}{N*2}\right)$$

$$O\left(\frac{n*m}{2}\right) \qquad \text{if } n = N$$

Where

n – No. of transactions, m – No. of attributes,
N – No. of mappers

## 6.3. Computational Cost for Rule Hiding

- Computational cost without map reduce

$$O(m*n)$$

- Computational cost with map reduce

$$O\left(\frac{m*n}{N}\right)$$

$$O(m) \qquad \text{if } n = N$$

Where

n – No. of transactions, m – No. of regions ,
N – No. of mappers

## 7. CONCLUSION

Attribute subset reduction can be used as a preprocessing step in data mining to produce accurate results. Noisy data is removed from the data set which improves the performance of classification, time efficiency and rule mining. Data hiding prevents extraction of useful association rules from quantitative data by decreasing the support of the RHS of the rule. Unlike previous approaches which mainly deals with association rules in binary database, the proposed approach deals with hiding the association rules in quantitative database. The proposed technique uses attribute reduction hence reduces computational cost by removing redundant attribute and also reduces the number of lost rules. Rule hiding performed using DSR approach produces less number of lost rules and also less number of modifications when compared to PSO. Hence DSR proves to be an efficient approach over PSO. The main advantage of the proposed system is that it does not produce ghost rules. Experimental results of the proposed approach demonstrate efficient information hiding with less side effects. PSO can be enhanced to reduce the number of lost rules and the number of modifications The Map Reduce version parallelizes the task and proves to be an efficient approach.

## 8. REFERENCES

[1]. Author, (2011) “A New method for preserving privacy in Quantitative Association Rules Using DSR approach with automated generation of membership function”, *Information and communication technologies (WICT).* 2011 world congress on 11 -14 Dec 2011, Mumbai, India, pp 148 – 153

[2]. M. Banerjee; S.Mitra and A.An(2006). Feature Selection Using Rough Sets.In: *Multi-Objective Machine Learning*, Ed. Yaochu Jin, Series on *Studies in Computational Intelligence* 16 (Springer-Verlag, Berlin), pp.3-20.

[3]. T. Berberoglu and M. Kaya,(2008) “Hiding Fuzzy Association Rules in Quantitative Data”,The 3rd InternationalConference on Grid and Pervasive Computing Workshops, pp. 387-392.

[4]. E.P.M. de Sousa; C. Traina; A.J.M. Traina; L. Wu and C. Faloutsos,(2007), “A Fast and Effective Method to Find Correlations among Attributes in Databases,” Data Mining and Knowledge Discovery, vol. 14, pp. 367-407.

[5]. J.Zh.Dong; N.Zhong and S.Ohsuga. Using rough sets with heuristics for feature selection. *N.Zhong (Eds.):RSFDGrC’99*, Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing Pages178-187Springer-Verlag London, UK ©1999.

[6]. R.Jenshen and Q.Shen.(2004b) Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches.*IEEE Transactions on Knowledge and Data Engineering*,2004, Vol.16, Issue:12, pp. 1457-1471.

[7]. Jitendra kumar and Binit Kumar Sinha(2010), "privacy preserving clustering in Data Mining" , B. Tech Thesis, NIT, Rourkela, India.

[8]. Neil Mac Parthala´in; Qiang Shen and Richard Jensen,(2010) ,“ A Distance Measure Approach to Exploring the Rough Set Boundary Region for Attribute Reduction”. IEEE transaction on Knowledge and data engineering, Vol.22, No.3, pp. 305-317.

[9]. Paramjeet; V.Ravi; Naveen Nekuri; Chillarige Raghavendra Rao; (2012), “Privacy preserving data mining using particle swarm optimization trained auto-

associative neural network: an application to bankruptcy prediction in banks", International journal of data mining, modeling and management Vol.4, No.1, pp. 39-56

[10]. Q. Shen and R. Jensen,(2004a) "Selecting Informative Features with Fuzzy-Rough Sets and Its Application for Complex Systems Monitoring," Pattern Recognition, vol. 37, no. 7, pp. 1351-1363

[11]. A.Skowron and C.Rauszer.(1992) The Discernibility Matrices and Functions in Information Systems. Handbook of applications and advances of the Rough set theory,*Intelligent Decision Support*,331-362.

[12]. Vassilios S. Verykios; Elisa Bertino; et al.,(2004) "State-of-the-art in Privacy Preserving Data Mining," SIGMOD Record, Vol. 33, No. 1, pp.50-57.

[13]. G.Y.Wang. "*Rough Set Theory and Data Mining"*. Xi'an Jiaotong University Press,Xi'an, 2001.

[14]. G.Y.Wang; J. Zhao; J.J.An, et al.(2004) Theoretical study on attribute reduction of rough set theory: comparison of algebra and information views. *In: Proceedings of the Third IEEE International Conference on Cognitive Informatics*,pp-148 – 155.

[15]. Wu Xiaodan; Chu Chao-Hsien; Wang Yunfeng; Liu Fengli; Yue Dianmin,(2007) Privacy Preserving Data Mining Research: Current Status and Key Issues, Computional Science- ICCS ,pp:762-772.

[16]. Yongcheng Luo; Yan Zhao; Jiajin Le, I, (2009)"A Survey on the Privacy Preserving Algorithm of Association Rule Mining",IEEE Explore electronic commerce and security, vol.1, pp.241-245 .