2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)

Miner for OACCR: Case of Medical Data Analysis in Knowledge Discovery

Samaher Hussein Ali
Department of Software, college of Information Technology
University of Babylon
Hilla, Iraq (00964)
Samaher_hussein@yahoo.com

Abstract—: **Modern scientific data consist of huge datasets which gathered by a very large number of techniques and stored in much diversified and often incompatible data repositories as data of bioinformatics, geoinformatics, astroinformatics and Scientific World Wide Web. At the other hand, lack of reference data is very often responsible for poor performance of learning where one of the key problems in supervised learning is due to the insufficient size of the training dataset. Therefore, we try to suggest a new development a theoretically and practically valid tool for analyzing small of sample data remains a critical and challenging issue for researches. This paper presents a methodology for Obtaining Accurate and Comprehensible Classification Rules (OACCR) of both small and huge datasets with the use of hybrid techniques represented by knowledge discovering. In this article the searching capability of a Genetic Programming Data Construction Method (GPDCM) has been exploited for automatically creating more visual samples from the original small dataset. Add to that, this paper attempts to developing Random Forest data mining algorithm to handle missing value problem. Then database which describes depending on their components were built by Principle Component Analysis (PCA), after that, association rule algorithm to the FP-Growth algorithm (FP-Tree) was used. . At the last, TreeNet classifier determines the class under which each association rules belongs to was used. The proposed methodology provides fast, Accurate and comprehensible classification rules. Also, this methodology can be use to compression dataset in two dimensions (number of features, number of records).**

Keywords- Random Forest; GPDCM; PCA; FP-Growth; Adboosting.

I. INTRODUCTION

As our abilities to collect and store various types of datasets are continually increasing, the demands for advanced techniques and tools to understand and make use of these large data keep growing. No single existing field is capable of satisfying the needs. Data Mining and Knowledge Discovery, which utilizes methods, techniques, and tools from diverse disciplines, emerged in last decade to solve this problem. It brings knowledge and theories from several fields including databases, machine learning, optimization, statistics, and data visualization and has been applied to various real-life applications. Even though data mining has made significant progress during the past fifteen years, most research effort is devoted to developing effective and efficient algorithms that can extract knowledge from data and not enough attention has been paid to the philosophical foundations of data mining.

Knowledge Discovery in Databases (KDD) is the non trivial process of identifying, novel, potentially useful and ultimately understandable patterns in the data [29] .

KDD process is an interactive and iterative multi-step process which uses six steps to extract interesting knowledge according to same specific measures and thresholds, these steps include (data selection, clearing, enrichment, coding, data mining and reporting)[11] . Will these steps are extended by[16] to include (Developing, creating, data cleaning and preprocessing, data reduction, choosing the data mining task, choosing the data mining algorithm(s), data mining, interpreting mined patterns and finally consolidating discovered knowledge).

Data mining is not just a single method or single technique but rather a spectrum of different approaches which searches of patterns and relationships of data [18] .

Data mining is concerned with important aspects related to both database techniques and AI/machine learning mechanisms, and provides an excellent opportunity for exploring the interesting relationship between retrieval and inference/reasoning, a fundamental issue concerning the nature of data mining.

Data mining is becoming more widespread every day, because it empowers companies to uncover profitable patterns and trends from their existing databases. Companies and institutions have been spent millions of dollars to collect megabytes and terabytes of data but are not taking advantages of the valuable and actionable information hidden deep within their data repositories. Companies that do not apply these techniques are in danger of falling behind and losing market shares because their competitors are using data mining and are thereby gaining the competitive edge [11].

In this work, we examine the ability to a Random Forest Data Mining algorithm (RFDM) to pre-process huge database that suffer from missing values problem. Random Forest (RF) [1] is a collection of decision trees grown and combined using the computer code. RF models are often considerably more accurate than a single tree, accuracy achieved is often

competitive with the best alternative methods. Resistance to over training (growing a large number of RF trees does not create a risk of over fitting and each tree is a completely independent random experiment). Speed (trees are grown at high speed because few variables are in use at any one time) also be used to determine how best to filter data before analysis.

While, we propose Genetic Programming Data Construction Method (GPDCM) to pre-process insufficient size of database, by using three different types of genetic programming crossover (node crossover, branch crossover and mixed crossover) apply in parallel way.

After that, reduction diminution of dataset by Principal Component Analysis(PCA)[9] that used to find best features from these available in the database after filtering their(i.e., after performing pre-processing of both types of database "huge" and "insufficient size ").

Then training the FP-Growth algorithm (association data mining algorithm) to generation association rules from the best features, where Fp-growth can be define as powerful computational tools in a generation association rules compare with A priori algorithm. It is base on FP-tree.

After that, perform the equivalence between association rules and their class base on TreeNET data mining algorithm.

II. CURRENT STATUS

Nomura and Miyoshi, 1998 [23] proposed an automatic fuzzy rule extraction method using the hybrid model of the FSOM and the GA with numerical Chromosomes.

McGarry, Tait, Wermter, and MacIntyre, 1999 [13] showed that the weights and cluster centers could be directly interpreted as antecedents in a symbolic IF..THEN type rule.

Mitra S. and Sankar Pal, 2000 [17] described a way of designing a hybrid decision support system in soft computing paradigm for detecting the different stages of cervical cancer.

Sankar Pal, Mitra S. and Mitra P., 2001 [26] presented amethodology that described for evolving Rough-Fuzzy Multi layer perceptron with modular concept using a genetic algorithm to obtain a structured network suitable for both classification and rule extraction.

Isao Okina., 2002[8] examine the approach of using causal network (CN) model for extraction of uncertain knowledge. In particular, we explore a restricted form of causal network model, termed as binary causal network (BCN) model, to reduce the computational complexity in probability calculation and propagation. We provide the definition and the related algorithm forBCN, discuss the complexity and feasibility of this model, and compare the algorithm with other CN algorithms.

Ken McGarry, 2002 [KEN 02] presented the results of ranking and the analysis of rules extracted from RBF neural networks using both objective and subjective measures. The interestingness of a rule can be assessed by a data driven approach.

Hussein K.,2002 [7] suggest algorithm to discover terminated item sets and explain how you can maintenance to terminated item sets to extract dynamic rule miner.

Mutthew G. and Larry B., 2003 [19] describe a way of feature construction and selection using the traditional genetic programming, genetic algorithm and stander c4.5 on number of databases to improve the classification performance of the well-known induction algorithm c4.5.

Nian Yan., 2004[24] introduced the classification of data mining approaches and focused on back propagation neural network and its enhanced applications. The multilayer neural network has been applied in building the classification model. There are two data sets used for the study: HIV data set from bioinformatics research and credit card data set for the risk management. A pre-training process is designed to construct the neural network classifier fast also proposed a new ensemble method: performance weighted ensemble based on the p value and has proved the strongpoint of it compared to the traditional ensemble method.

Malone, McGarry K., and Bowerman C., 2004 [20] demonstrated the use of ANFIS to optimize expert's opinions. The ANFIS model offers the advantage of enabling use of initially approximate data in an effective manner whilst, following training, allowing fuzzy rules to be extracted which represent the optimized fuzzy membership functions.

Malone J., McGarry K., Bowerman C. and Wermter S., 2005 [21] have proposed a technique for the automatic extraction of rules from trained SOMs.

Mahdi A.,2005 [22] suggest a techniques which can discovery knowledge from any database via soft computing techniques which involve fuzzy set, neural network and genetic algorithm by build DBRule Extractor system.

Georgios K, Eleftherios K and Vassili L., 2006 [5] Propese a web search algorithm aims to distinguish irrelevent information and to enhance the amount of the relevent information in respect to a user's query. The proposed algorithm is based on the Ant colony optimization algorithm (ACO)

Jiaxiong Pi., 2007[10] explore the relationship by examining time series data indexed through R*-trees, and study the issues of (1) retrieval of data similar to a given query (which is a plain data retrieval task), and (2) clustering of the data based on similarity (which is a data mining task). Along the way of examination of our central theme, we also report new algorithms and new results related to these two issues. We have developed a software package consisting of a similarity analysis tool and two implemented clustering algorithms: K-Means-R and Hierarchy-R.

Ulf Johansson., 2007[30] two novel algorithms based on Genetic Programming are suggested. The first algorithm (GEMS) is used for ensemble creation, and the second (G-REX) is used for rule extraction from opaque models. The main property of GEMS is the ability to combine smaller ensembles and individual models in an almost arbitrary way. Moreover, GEMS can use base models of any kind and the optimization function is very flexible, easily permitting

inclusion of, for instance, diversity measures. In the experimentation, GEMS obtained accuracies higher than both straightforward design choices and published results for Random Forests and AdaBoost. The key quality of G-REX is the inherent ability to explicitly control the accuracy vs. comprehensibility trade-off. Compared to the standard tree inducers C5.0 and CART, and some well-known rule extraction algorithms, rules extracted by G-REX are significantly more accurate and compact. Most importantly, G-REX is thoroughly evaluated and found to meet all relevant evaluation criteria for rule extraction algorithms, thus establishing G-REX as the algorithm to benchmark against.

Samaher H. and Mahdi A., 2007[27] present a methodology for discovering classification rules in data mining using FCM and GP to classify five well known database. ANDing and ORing form are used in formulation the classification rules.

Samaher H., 2008 [28] present a method to design a programming system using hybrid techniques represented by soft computing and data mining to discovery knowledge in databases. This research proposes a fuzzy c-mean model for attributes clustering. The genetic algorithm is used to determine which such features are most predictive after that, radial basis functions are embedded in two-layer neural topology is used in real applications to classify the database records.

Erica Craig and Falk Huettmann., 2009[3] describe the use of fast, data-mining algorithms such as TreeNet and Random Forests to identify ecologically meaningful patterns and relationships in subsets of data that carry various degrees of outliers and uncertainty.

Wen Xiong and Cony Wang., 2009[31] proposed a hybrid approach based on improved self adaptive ant colony optimization (ACO) and random forest(RF) to select feature from microarray gene expression data. It can capture important feature by preselection and attain near-optimum or optimum by confining the size of ant's solution to accelerate convergence of ant colony. Finally, it constructs optimum by restricted sequential forward selection applied to near optimum.

III. The Still Open Problem

Intelligent Data Analysis provides learning tools of finding data patterns based on artificial intelligence. KDD has a long history of applications to data analysis in business, military, and social economic activities. While the aim of KDD is to discover the pattern of a data set, the size of the data set closely related to one of analysis methods. The classic approach is using certain statistical techniques to deal with data sets of more than 30 samples and by dimension reduction to reveal the pattern. With the rapid increase of Internet development and usage, the amount of data has been enormous. The term "Data Warehouse" has been used to describe such quantities of data and the corresponding methodologies for analysis are under the title of "data mining." *In contrast* to the huge amount of data sets, there is another type of data set which is small (less than 30), but still is significant in terms of socioeconomic cost. Consider severe earthquakes, random terrorist attacks, and nuclear plant explosions; the occurrences of such events are relatively few, the conventional statistic assumptions cannot be verified and thus the methods fail to apply. The ability to predict such kinds of events remains a challenge for the researchers. This leads to the necessity of recovering a pattern by constructing data. It is an art and science for intelligent data analysis.

IV. The Development System

The proposed work handles the different challenges posed by data mining. The main constituents of KDD, at this juncture, include GPDCM, PCA, Ad-boost and two of the fast data mining algorithms (RF, FP-Growth). Each of them contributes a distinct a methodology to addressing problems in its domain. This is done in a cooperative, rather than a competitive, manner where this work uses four types of processing. The block diagram of the proposed system shows in figure (1).

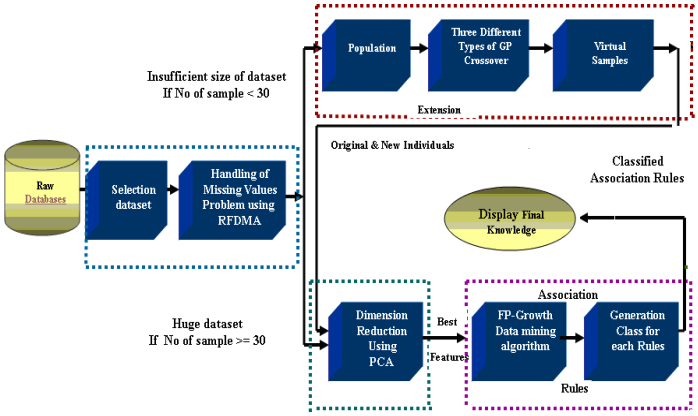

Figure 1. The Block Diagram of the Proposed System

A. *Handling the Missing values using Random Forest*

All dataset suffer from three types of problems:

1) *Missing values Problem*

This problem is considering one of the still open problems, where some of solutions suggest solving this problem as explain follow:-

0.5	?	0.2	0.3	?	0.1

First Solution: - The simplest solution for this problem is the reduction of the data set and the elimination of all samples with missing values. That is possible when large data sets are available, and missing values occur only in a small percentage of samples.

Second Solution:- a data miner and domain expert, can examine manually samples that have no values and enter a reasonable, probable, or expected value, based on a domain

experience. This method is straightforward for small numbers of missing values and relatively with the small data sets. But, if there is no obvious or plausible value for each case, the miner is introducing noise into the data set by manually generating a value.

Third Solution:- automatic replacement of missing values with some constants

- Replace all missing values with a single global constant
- Replace a missing value with its feature mean
- Replace a missing value with its feature mean for the given class
- Replace a missing value with the nearest neighborhood from top or bottom

2) Outliers Problem

This problem occurs when one value is different in types of all other values. This problem solve using RF in [3]

3) Uncertainty Problem

This problem occurs when one value is out of the attribute range. This problem can be solves by RF in [3] .

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. RFs introduce a new principle to random splitting. It alters the tree growing process by narrowing its focus during split selection. For example, if the database contains 100 columns usable for prediction, RFs would begin by randomly selection 7 variables and then select the splitter from the list of 7 predictors. Once the data has been split into two subsets the process can be repeated by splitting each of the two subsets partly at random. To split one of the two subsets, RFs can select a different set of 7 predictors at random and can use the best of these affect a further split. Although in RFs, the t trees grow we split data using the best splitter available, we first randomly limit the list of available splitters to a small number. This means that the selected splitter is best only in the limited sense of being best in the randomly selected list. Each tree is constructed using the following algorithm:

Step1: Let the number of training cases be *N*, and the number of variables in the classifier be *M*.

Step2: We are take the number *m* of input variables be used to determine the decision at a node of the tree; *m* should be less than *M*.

Step3: Choose training set for this tree by choosing *N* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). The rest of the cases need to estimate the error of the tree, by predicting their classes.

Step4: For each node of the tree, we choose randomly *m* variables based on the decision node. Calculate the best split based on these m variables in the training set.

Step5: Each tree become fully grown and not pruned (as can be done in constructing a normal tree classifier).

All the prior models tend to combine more effectively into high performance aggregate models. While RFs is different in that it introduces an entirely new way of generating individual component models and do not combine more effectively into high performance aggregate models. There are two reasons for this: ***First***, combining or averaging the models does not accomplish significant accuracy improvement, if the individual models are too similar to each other. the RFs have two-part randomization process makes the individual models quite different from each other. ***Second*** the other models is slow learning while in RFs because the trees are grown independently of each other with no single tree depending on any other tree for its generation, adding trees does not create a risk of overfitting. RFs can be expanded indefinitely without a loss in performance. If an important pattern genuinely exists in the data portions, it will be uncovered by different trees and genuine patterns will be detected repeatedly by different trees. As a result, the work add the following idea***: "mapping among all the weight values of the record has the missing values with all features of the record has a weight values similar of that record, the mapping perform at the down level of splitter to reduce the time and number of comparing among the weight of features. This method can find the actual values of the missing values, and handle the uncertainty and outliers problems".***

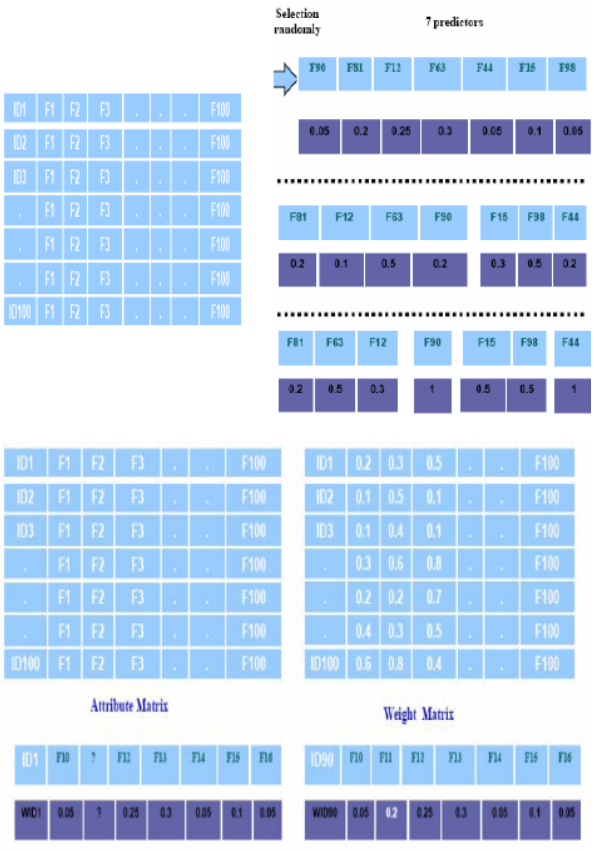

Figure2..Example of Handling Missing Values using RF

V. GENETIC PROGRAMMING DATA CONSTRUCTION METHOD (GPDCM)

The lack of reference data is very often cause a poor performance of learning. Therefore, How to develop a theoretically and practically valid tool for analyzing small sample data remains a critical and challenging issue for researches and it consider being one of the still open problems. Data construction methods can be roughly divided into two groups, according to construction strategy: **hypothesis-driven methods** and **data-driven methods**.

Hypothesis-driven methods construct new attributes out of previously-generated hypotheses (discovered rules or another kind of knowledge representation). In general they start by constructing a hypothesis, for instance a decision tree, and then examine that hypothesis to construct new attributes [25] . ***By contrast***, data-driven methods do not suffer from the problem of depending on the quality of previous hypotheses. They construct new attributes by directly detecting relationships in the data.

The process of attribute construction can also be roughly divided into two approaches, namely the **interleaving approach** and the **preprocessing approach**.

In the preprocessing approach the process of attribute construction is independent of the inductive learning algorithm that will be used to extract knowledge from the data. In other words, the quality of a candidate new attribute is evaluated by directly accessing the data, without running any inductive learning algorithm. In this approach the attribute construction method performs a preprocessing of the data, and the new constructed attributes can be given to different kinds of inductive learning methods. ***By contrast***, in the interleaving approach the process of attribute construction is intertwined with the inductive learning algorithm. The quality of a candidate new attribute is evaluated by running the inductive learning algorithm used to extract knowledge from the data, so that in principle the constructed attributes' usefulness tends to be limited to that inductive learning algorithm. An example of data construction method following the interleaving approach can be found in [33].

In this work we follow the data-driven strategy and the preprocessing approach, mainly for two reasons**. First**, using this combination of strategy/approach the constructed data have a more generic usefulness, since they can help to improve the predictive accuracy of any kind of inductive learning algorithm. **Second**, data construction method following the preprocessing approach tends to be more efficient than its interleaving counterpart, since the latter requires many executions of an inductive learning algorithm.

The work proposes a GPDCM to create more visual samples the original small dataset; the idea is depend on the represent each individual as decision tree. After that, we use tournament selection and apply the genetic operations (reproduction, crossover and mutation). There are three types of crossover operations each one of them give different result, therefore the work focuses on the following types:-

A. The Node Crossover

The following, steps can be represented the implementation of the node crossover method.

Step1: Select two parents from population.

Step2: Select random one crossover node from the first parent and search randomly in the second parent for an exchangeable.

Step3: Swap the crossover node.

Step4: The child is a copy of the modified its first parent.

B. The Branch Crossover

The following, steps can be represented the implementation branch crossover method.

Step1: Select two parents from population.

Step2: Select random one crossover node from the first parent and search randomly in the second parent for an exchangeable.

Step3: Cutoff the branch with the crossover nodes.

Step4: Calculate the size of the expected child (remind size of first parent + size of branch cutoff from the second parent).

Step5: If the size of child is accepted created the child by appending the prang cutoff from second parent t o the ramming of first parent otherwise try again starting from (STEP 2).

C. The Mixed Crossover

The following, steps can be represented the implementation mixed crossover method.

Step1: Select two parents from population.

Step2: Select random one crossover node a terminal node in the second parent.

Step3: Generate the child by replacing the branch with the crossover node in first parent with the terminal node select from second parent.

The fitness function used in this work is information gain ratio [Fre02] , which is a well-known attribute-quality measure in the data mining and machine learning literature. It should be noted that the use of this measure constitutes a data-driven strategy. As we mentioned above, an important advantage of this kind of strategy is that it is relatively fast, since it avoids the need for running a data mining algorithm when evaluating an attribute (individual). In particular, the information gain ratio for a given attribute can be computed in a single scan of the training set

As a result, if the population consist of 20 individuals (No of records in database or No of parents) and each crossover between two parents yields one child this mean generation 10 children when apply node crossover, 10 children when apply branch crossover, 10 children when apply mixed crossover at first iteration of GPDCM.

The size of new population =the size of old population + No of children for all the three crossover approaches. The size of new population=20 +30=50 individuals. While the

size of population in second iteration =50+75=125 individuals. See figure 3.

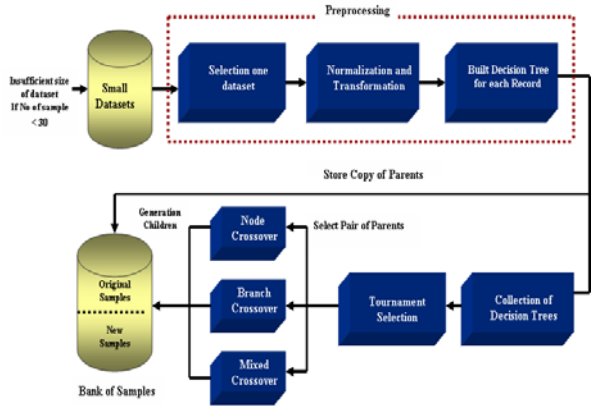

Figure3. Block Diagram of the GPDCM

Each individual generation by any crossover method is tested if it is valid or not by given the value, where(-1) for each operation and (+1) for each operand and find the summation of these values if the result is +1 the expression is valid otherwise is not valid. The generation process is continues, until it depends on the value of the statistical measures such as mean, standard deviations and Roc Curve analysis.

VI. Dimension Reduction using Principal Components Analysis

The use of too many predictor variables to model a relationship with a response variable can unnecessarily complicate the interpretation of the analysis and violates the principle of parsimony: that one should consider keeping the number of predictors to a size that could easily be interpreted. Also, retaining too many variables may lead to overfitting, in which the generality of the findings is hindered because the new data do not behave the same as the training data for all the variables.

Further, analysis solely at the variable level might miss the fundamental underlying relationships among predictors. For example, several predictors might fall naturally into a single group (a *factor* or a *component*) that addresses a single aspect of the data. [2]

Dimension reduction methods have the goal of using the correlation structure among the predictor variables to accomplish the following:

A. To reduce the number of predictor components
B. To help ensure that these components are independent
C. To provide a framework for interpretability of the results.

Definition 1: Principal components analysis (PCA) seeks to explain the correlation structure of a set of predictor variables using a smaller set of linear combinations of these variables. These linear combinations are called *components*.

Suppose that the original variables $X1, X2, \ldots, Xm$ form a coordinate system in m-dimensional space. The principal components represent a new coordinate system, found by rotating the original system along the directions of maximum variability. When preparing to perform data reduction, the analyst should **first** standardize the data so that the mean for each variable is zero and the standard deviation is 1. Let each variable Xi represent an $n \times 1$ vector, where n is the number of records. Then represent the standardized variable as the $n \times 1$ vector Zi , where $Zi = (Xi - \mu i) / \sigma ii$, μi is the mean of Xi , and σii is the standard deviation of Xi . In matrix notation,this standardization is expressed as $\mathbf{Z} = (\mathbf{V}1/2) - 1(\mathbf{X} - \boldsymbol{\mu})$, where the "–1" exponent refers to the matrix inverse, and $\mathbf{V}1/2$ is a diagonal matrix (nonzero entries only on the diagonal), the $m \times m$ *standard deviation matrix*:

$$\mathbf{V}^{1/2} = \begin{bmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{pp} \end{bmatrix}$$

Let $\sum$ *refer* to the symmetric *covariance matrix*:

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1m}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \cdots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1m}^2 & \sigma_{2m}^2 & \cdots & \sigma_{mm}^2 \end{bmatrix}$$

Where σ_{ij}^2 , $i \neq j$ refers to the *covariance* between Xi and Xj :

$$\sigma_{ij}^2 = \frac{\sum_{k=1}^{n} (x_{ki} - \mu_i)(x_{kj} - \mu_j)}{n}$$

The covariance is a measure of the degree to which two variables vary together. Positive covariance indicates that when one variable increases, the other tends to increase. Negative covariance indicates that when one variable increases, the other tends to decrease.

Note that the covariance measure is not scaled, so that changing the units of measure would change the value of the covariance.

The *correlation coefficient* $ri\ j$ avoids this difficulty by scaling the covariance by each of the standard deviations:

$$r_{ij} = \frac{\sigma_{ij}^2}{\sigma_{ii}\sigma_{jj}}$$

Then the *correlation matrix* is denoted as ρ:

$$\rho = \begin{bmatrix} \frac{\sigma_{11}^2}{\sigma_{11}\sigma_{11}} & \frac{\sigma_{12}^2}{\sigma_{11}\sigma_{22}} & \cdots & \frac{\sigma_{1m}^2}{\sigma_{11}\sigma_{mm}} \\ \frac{\sigma_{12}^2}{\sigma_{11}\sigma_{22}} & \frac{\sigma_{22}^2}{\sigma_{22}\sigma_{22}} & \cdots & \frac{\sigma_{2m}^2}{\sigma_{22}\sigma_{mm}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sigma_{1m}^2}{\sigma_{11}\sigma_{mm}} & \frac{\sigma_{2m}^2}{\sigma_{22}\sigma_{mm}} & \cdots & \frac{\sigma_{mm}^2}{\sigma_{mm}\sigma_{mm}} \end{bmatrix}$$

HOW CAN YOU COMPUTE PCA?

In order to compute the PCA we can follow the following steps [18] :

Step1: Compute the standardized data matrix $\mathbf{Z}$ =[$Z1, Z2, \ldots, Zm$] base on $Zi = (Xi - \mu i) / \sigma ii$ *from the original dataset.*

Step2: Compute Eigenvalues. Let $\mathbf{B}$ be an $m \times m$ matrix, and let $\mathbf{I}$ be the $m \times m$ identity matrix (diagonal matrix with 1's on the diagonal). Then the scalars (numbers of dimension 1×1) $\lambda 1, \lambda 1, \ldots, \lambda m$ are said to be the *Eigenvalues* of $\mathbf{B}$ if they satisfy $|\mathbf{B} - \lambda \mathbf{I}| = 0$.

Step3: Compute Eigenvectors. Let $\mathbf{B}$ be an $m \times m$ matrix, and let λ be an Eigenvalues of $\mathbf{B}$. Then nonzero $m \times 1$ vector $\mathbf{e}$ is said to be an *eigenvector* of $\mathbf{B}$ if $\mathbf{Be} = \lambda \mathbf{e}$.

Step4: The *i*th *principal component* of the standardized data matrix $\mathbf{Z} = [Z1, Z2, \ldots, Zm]$ is given by $Yi = \mathbf{e}^{\mathrm{T}} i\mathbf{Z}$, where $\mathbf{e}i$ refers to the *i*th *eigenvector* (discussed below) and $\mathbf{e}^{\mathrm{T}} i$ refers to the transpose of $\mathbf{e}i$. The principal components are linear combinations $Y1, Y2, \ldots, Yk$ of the standardized variables in $\mathbf{Z}$ such that (1) the variances of the Yi are as large as possible, and (2) the Yi are uncorrelated. The first principal component is the linear combination $Y1 = \mathbf{e}^{\mathrm{T}} 1Z = e11Z1 + e12Z2 + \cdots + e1m\, Zm$, which has greater variability than any other possible linear combination of the Z variables. Thus:

- The first principal component is the linear combination $Y1 = \mathbf{e}^{\mathrm{T}} 1Z$, which maximizesVar($Y1$) = $\mathbf{e}^{\mathrm{T}} 1\boldsymbol{\rho}\, \mathbf{e}1$.
- The second principal component is the linear combination $Y2 = \mathbf{e}^{\mathrm{T}} 2Z$, which *is independent of* $Y1$ and maximizes Var($Y2$) = $\mathbf{e}^{\mathrm{T}} 2\boldsymbol{\rho}\, \mathbf{e}2$.
- The *i*th principal component is the linear combination $Yi = \mathbf{e}\_i\mathbf{X}$, which *is independent of all the other principal components* Yj , $j < i$, and maximizes Var(Yi) = $\mathbf{e}^{\mathrm{T}} i\boldsymbol{\rho}\, \mathbf{e}i$.

The criteria used for deciding how many components to extract are the following:

- Eigenvalue criterion
- Proportion of variance explained criterion
- Minimum communality criterion
- Scree plot criterion

The eigenvalue criterion states that each component should explain at least one variable's worth of the variability, and therefore the eigenvalue criterion states that only components with eigenvalues greater than 1 should be retained. For the proportion of variance explained criterion, the analyst simply selects the components one by one until the desired proportion of variability explained is attained. The minimum communality criterion states that enough components should be extracted so that the communalities for each of these variables exceed a certain threshold (e.g., 50%). The scree plot criterion is this: The maximum number of components that should be extracted is *just prior to* where the plot begins to straighten out into a horizontal line.

VII. FP-GROWTH ALGORITHM

After, we get the best features for database base on PCA (i.e., reduce number of features). We using FP-Growth algorithm to discovering frequency item sets and generation all association rules of that database.

FP-Growth adopts a divide and conquer strategy by (1) compressing the database representing frequent items into a structure called FP-tree (frequent pattern tree) that retains all the essential information and (2) dividing the compressed database into a set of conditional databases, each associated with one frequent itemset and mining each one separately. It scans the database only twice. In the first scan, all the frequent items and their support counts (frequencies) are derived and they are sorted in the order of descending support count in each transaction. In the second scan, items in each transaction are merged into an FP-tree and items (nodes) that appear in common in different transactions are counted. Each node is associated with an item and its count.

Nodes with the same label are linked by a pointer called a node-link. Since items are sorted in the descending order of frequency, nodes closer to the root of the FP tree are shared by more transactions, thus resulting in a very compact representation that stores all the necessary information. Pattern growth algorithm works on FP-tree by choosing an item in the order of increasing frequency and extracting frequent itemsets that contain the chosen item by recursively calling itself on the conditional FP-tree, that is, FP-tree conditioned to the chosen item. FP-growth is an order of magnitude faster than the original Apriori algorithm.

VIII. CLASSIFICATION BASE ON ASSOCIATION RULES

After that, adboosting data mining algorithm used to find best class for the association rules ,result from FP-Growth (i.e., find the class base on volt principle) as follow:

Step1: initialize equal weights for all N rules in association rule database (ARD). w = {wj = 1/N | j = 1, 2,. . . , N}

Step2: set number of iteration equal K

Step3: for i=1 to K do

Step4: Great training set Di by sampling (with replacement) from (ARD) according to w.

STEP 5: Train a base classifier Ci on Di.

Step 6: Apply Ci to all rules in the original training set, D.

Step 7: Calculate the weighted error

$$\epsilon_i = \frac{1}{N}\left[\sum_j w_j\, \delta\left(C_i(x_j) \neq y_j\right)\right]$$

Step 8: If $\epsilon_i > 0.5$ then

Step 9: Reset the weights for all N rules: w = {wj = 1/N | j = 1, 2, . . . ,N}: Go back to Step 4.

STEP 10: end if

STEP 11: $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$

Step 12: Update the weight of each rule as follow:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(\mathbf{x_i}) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(\mathbf{x_i}) \neq y_i \end{cases},$$

13: end for

Step 14:

$$C^*(\mathbf{x}) = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y))$$

IX. Experiment Results

To test performance of the suggest methodology, we apply it on the two small dataset (Heart, Iris dataset) and two huge dataset (Diabetes, DNA). Table I shows the results of each small dataset generation by GPDCM

TABLE I. Result of each Small Dataset

Name of DB	# Sample	# Features	# Class	Max # Generation.	Total #New Samples	#New True Samples
Heart	24	14	2	3	375	278
Iris	29	5	3	2	180	121

#NTS=T#NS –N#NS

Where

T#NS: total Number of New Samples,

N#NS: Neglected Number of New Samples

(Ex: #NTS of Heart=375-97 =278)

A. Result of Diabetes Dataset

We use Diabetes dataset as example to test the method, where these dataset consist of 768 samples and 8 features belong to two classes. Table II shows the Correlation Matrix of these dataset while, Table III explains egien value for each feature (Principal Factor Analysis).

TABLEII. Correlation Matrix of Diabetes Dataset

	TIMSPREG	PLASMAG	DBLDPRS	TRICEPS	H2SERUM	BODYMASS	PEDIGREE	AGE
TIMSPREG	1.0000	0.1295	0.1413	-0.0817	-0.0735	0.0177	-0.0335	0.5443
PLASMAG	0.1295	1.0000	0.1526	0.0573	0.3314	0.2211	0.1373	0.2635
DBLDPRS	0.1413	0.1526	1.0000	0.2074	0.0889	0.2818	0.0413	0.2395
TRICEPS	-0.0817	0.0573	0.2074	1.0000	0.4368	0.3926	0.1839	-0.1140
H2SERUM	-0.0735	0.3314	0.0889	0.4368	1.0000	0.1979	0.1851	-0.0422
BODYMASS	0.0177	0.2211	0.2818	0.3926	0.1979	1.0000	0.1406	0.0362
PEDIGREE	-0.0335	0.1373	0.0413	0.1839	0.1851	0.1406	1.0000	0.0336
AGE	0.5443	0.2635	0.2395	-0.1140	-0.0422	0.0362	0.0336	1.0000

TABLE III. Principal Factor Analysis

Factor	Eigen value	Variance	Cumulative
1	1.44848	46.081	46.081
2	1.30536	41.528	87.609
3	0.27192	8.651	96.259
4	0.09984	3.176	99.435
5	0.01775	0.565	100.000
6	-0.02709		
7	-0.10687		
8	-0.25555		

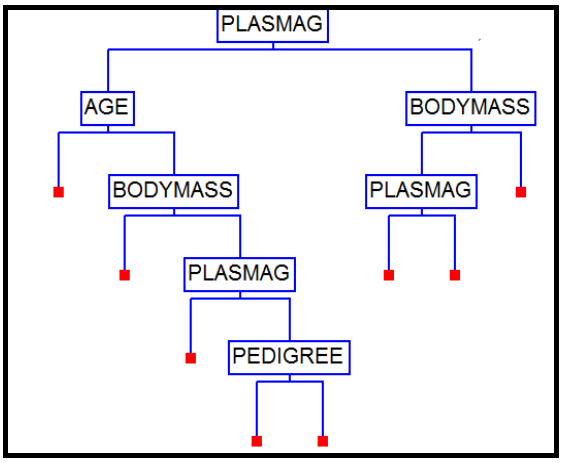

Figure4. Main Tree Split Variables *of Diabetes* Datset

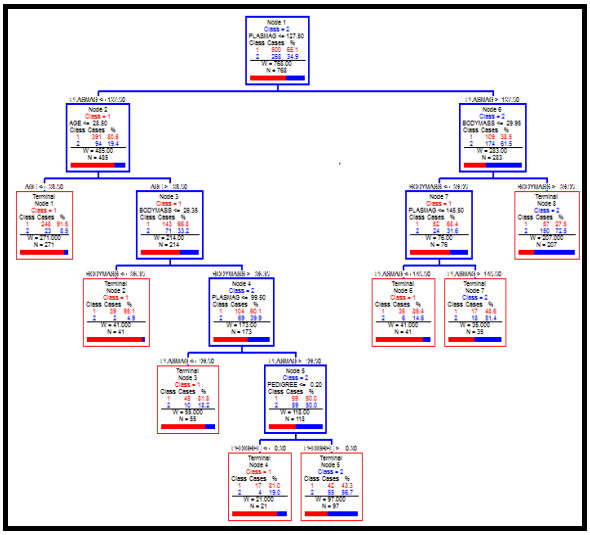

Figure5.Main Tree of Classification base on Association Rules

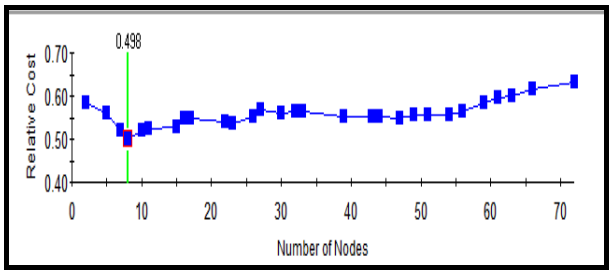

Figure6. Relation between Number of Nodes and Relative Cost

Knowledge Base Generation by methodology:

Rule1: *IF PLASMAG<=172.5 AND AGE<=28.50 THEN* ***CLASS 1***

Rule2: *IF PLASMAG<=172.5 AND AGE>28.50 AND BODYMASS<=26.35 THEN* ***CLASS 1***

Rule3: *IF PLASMAG<=172.5 AND AGE>28.50 AND BODYMASS>26.35 AND BODYMASS<=99.50 THEN* ***CLASS 1***

Rule4: *IF PLASMAG<=172.5 AND AGE>28.50 AND BODYMASS>26.35 AND BODYMASS<=99.50AND PEDIGREE <=0.20 THEN* ***CLASS 1***

Rule5: *IF PLASMAG<=172.5 AND AGE>28.50 AND BODYMASS>26.35 AND BODYMASS<=99.50AND PEDIGREE > 0.20 THEN* ***CLASS 2***

Rule6: *IF PLASMAG>172.5 AND BODYMASS<=29.95AND PLASMAG <= 145.5 THEN* ***CLASS 1***

Rule7: *IF PLASMAG>172.5 AND BODYMASS<=29.95AND PLASMAG >145.5 THEN* ***CLASS 2***

Rule8: *IF PLASMAG>172.5 AND BODYMASS>29. 95THEN* ***CLASS 2***

B. Result of DNA Dataset

We use DNA dataset as another example to test the method, where these dataset consist of 2000 samples and 181 features belong to three classes. Table IV shows the Correlation Matrix of these dataset while, Table V explains Principal Factor Analysis for each feature.

A105	A104	A100	A98	A97	A95	A94	A93	A91	A89	A88	A85	A83	A82	A75	A74	A73	A72	A0N	A63	
0.0704	-0.0682	0.0596	-0.0348	-0.0053	-0.0363	-0.0324	0.0388	-0.0347	0.0111	0.0148	-0.0734	-0.0892	0.0598	0.0249	-0.0064	0.0374	0.0918	-0.0148	1.0000	A63
-0.0398	0.0684	-0.0724	0.0307	-0.0188	0.0566	-0.0047	-0.0140	0.0419	-0.0113	-0.0859	0.1027	0.0867	-0.0684	-0.2334	0.0742	0.0151	-0.3618	1.0000	-0.0148	A71
0.0794	-0.0428	0.0669	-0.0442	0.0223	-0.0606	-0.0143	0.0733	-0.1015	0.0430	0.0025	-0.0907	-0.0586	0.0409	0.0350	-0.0280	0.1085	1.0000	-0.3618	0.0918	A72
0.0495	-0.0240	0.0640	-0.0364	0.0168	-0.0256	0.0305	0.0000	0.0055	0.0460	0.0608	-0.1079	-0.1415	0.0921	-0.2615	-0.3319	1.0000	0.1085	0.0151	0.0374	A73
-0.0709	0.0767	-0.0672	0.0794	-0.0479	0.0356	-0.0390	-0.0246	-0.0148	-0.0395	-0.0308	0.1174	0.2040	-0.1223	-0.3548	1.0000	-0.3319	-0.0280	0.0742	-0.0064	A74
0.0692	-0.0512	0.0224	-0.0116	-0.0060	-0.0158	-0.0058	0.0376	-0.0606	0.0432	0.0365	-0.1259	-0.1224	0.0544	1.0000	-0.3548	-0.2615	0.0350	-0.2334	0.0249	A75
0.1127	-0.0828	0.1022	-0.0378	0.1035	-0.0459	-0.0204	0.0400	-0.0534	-0.0174	0.0755	-0.0990	-0.4512	1.0000	0.0544	-0.1223	0.0921	0.0409	-0.0684	0.0598	A82
-0.0586	0.0518	-0.0603	0.0322	-0.0447	0.0176	-0.0250	0.0347	0.0642	-0.1323	-0.1507	0.3052	1.0000	-0.4512	-0.1224	0.2040	-0.1415	-0.0586	0.0867	-0.0892	A83
-0.0209	0.0096	-0.0209	-0.0085	0.0041	-0.0535	-0.0110	0.1582	0.0503	-0.2657	-0.1619	1.0000	0.3052	-0.0990	-0.1259	0.1174	-0.1079	-0.0907	0.1027	-0.0734	A85
-0.0639	0.0228	-0.0019	-0.0011	0.0418	0.0446	0.0841	-0.1240	0.0261	-0.1682	1.0000	-0.1619	-0.1507	0.0755	0.0365	-0.0308	0.0608	0.0025	-0.0859	0.0148	A88
-0.0835	0.0674	-0.0845	0.0827	-0.0814	0.0880	-0.0142	-0.3020	0.1129	1.0000	-0.1682	-0.2657	-0.1323	-0.0174	0.0432	-0.0395	0.0460	0.0430	-0.0113	0.0111	A89
-0.1333	0.0353	-0.0789	0.0673	-0.0535	0.0358	0.0889	-0.4718	1.0000	0.1129	0.0261	0.0503	0.0642	-0.0534	-0.0606	-0.0148	0.0055	-0.1015	0.0419	-0.0347	A91
0.2837	-0.0980	0.2234	-0.1459	0.1543	-0.1653	-0.0827	1.0000	-0.4718	-0.3020	-0.1240	0.1582	0.0347	0.0400	0.0376	-0.0246	0.0000	0.0733	-0.0140	0.0388	A93
-0.1646	0.0157	-0.0415	0.0067	0.0190	-0.2326	1.0000	-0.0827	0.0889	-0.0142	0.0841	-0.0110	-0.0250	-0.0204	-0.0058	-0.0390	0.0305	-0.0143	-0.0047	-0.0324	A94
-0.1087	0.1212	-0.1241	0.1072	-0.0451	1.0000	-0.2326	-0.1653	0.0358	0.0880	0.0446	-0.0535	0.0176	-0.0459	-0.0158	0.0356	-0.0256	-0.0606	0.0566	-0.0363	A95
0.1312	-0.0638	0.0778	-0.3395	1.0000	-0.0451	0.0190	0.1543	-0.0535	-0.0814	0.0418	0.0041	-0.0447	0.1035	-0.0060	-0.0479	0.0168	0.0223	-0.0188	-0.0053	A97
-0.1518	0.0599	-0.0744	1.0000	-0.3395	0.1072	0.0067	-0.1459	0.0673	0.0827	-0.0011	-0.0085	0.0322	-0.0378	-0.0116	0.0794	-0.0364	-0.0442	0.0307	-0.0348	A98
0.2854	-0.1374	1.0000	-0.0744	0.0778	-0.1241	-0.0415	0.2234	-0.0789	-0.0845	-0.0019	-0.0209	-0.0603	0.1022	0.0224	-0.0672	0.0640	0.0669	-0.0724	0.0596	A100
-0.4065	1.0000	-0.1374	0.0599	-0.0638	0.1212	0.0157	-0.0980	0.0353	0.0674	0.0228	0.0096	0.0518	-0.0828	-0.0512	0.0767	-0.0240	-0.0428	0.0684	-0.0682	A104
1.0000	-0.4065	0.2854	-0.1518	0.1312	-0.1087	-0.1646	0.2837	-0.1333	-0.0835	-0.0639	-0.0209	-0.0586	0.1127	0.0692	-0.0709	0.0495	0.0794	-0.0398	0.0704	A105

TABLE V. PRINCIPAL FACTOR ANALYSIS OF DNA DATASET

Factor	Eigen value	Variance	Cumulative
1	1.69511	27.138	27.138
2	1.37372	21.993	49.130
3	1.01262	16.211	65.342
4	0.45369	7.263	72.605
5	0.44395	7.107	79.712
6	0.38872	6.223	85.936
7	0.3307	5.294	91.230
8	0.25676	4.111	95.340
9	0.18103	2.898	98.239
10	0.10788	1.727	99.966
11	0.00214	0.034	100.000
12	-0.00108	.	.
13	-0.17904	.	.
14	-0.21039	.	.
15	-0.22791	.	.
16	-0.25475	.	.
17	-0.31549	.	.
18	-0.31921	.	.
19	-0.31984	.	.
20	-0.36296	.	.

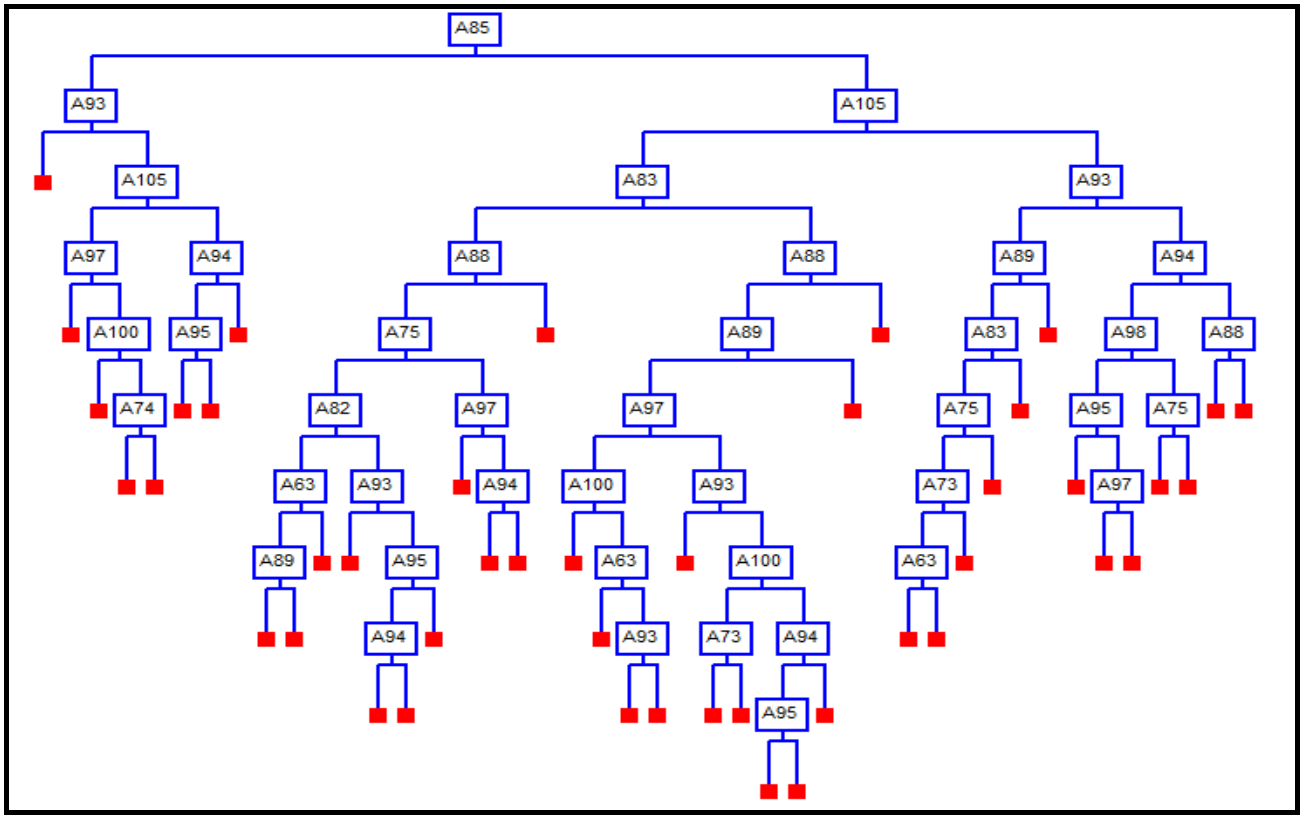

Figure7.Main Tree Split Variables *of DNA* Dataset

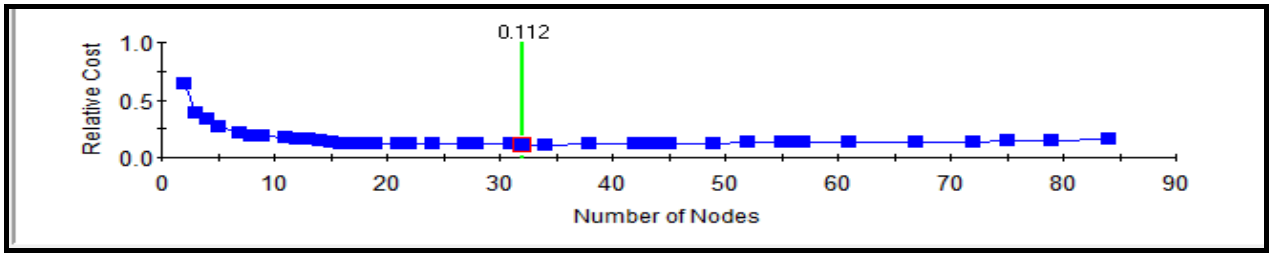

Figure8. Relation between Number of Nodes and Relative Cost for DNA Dataset

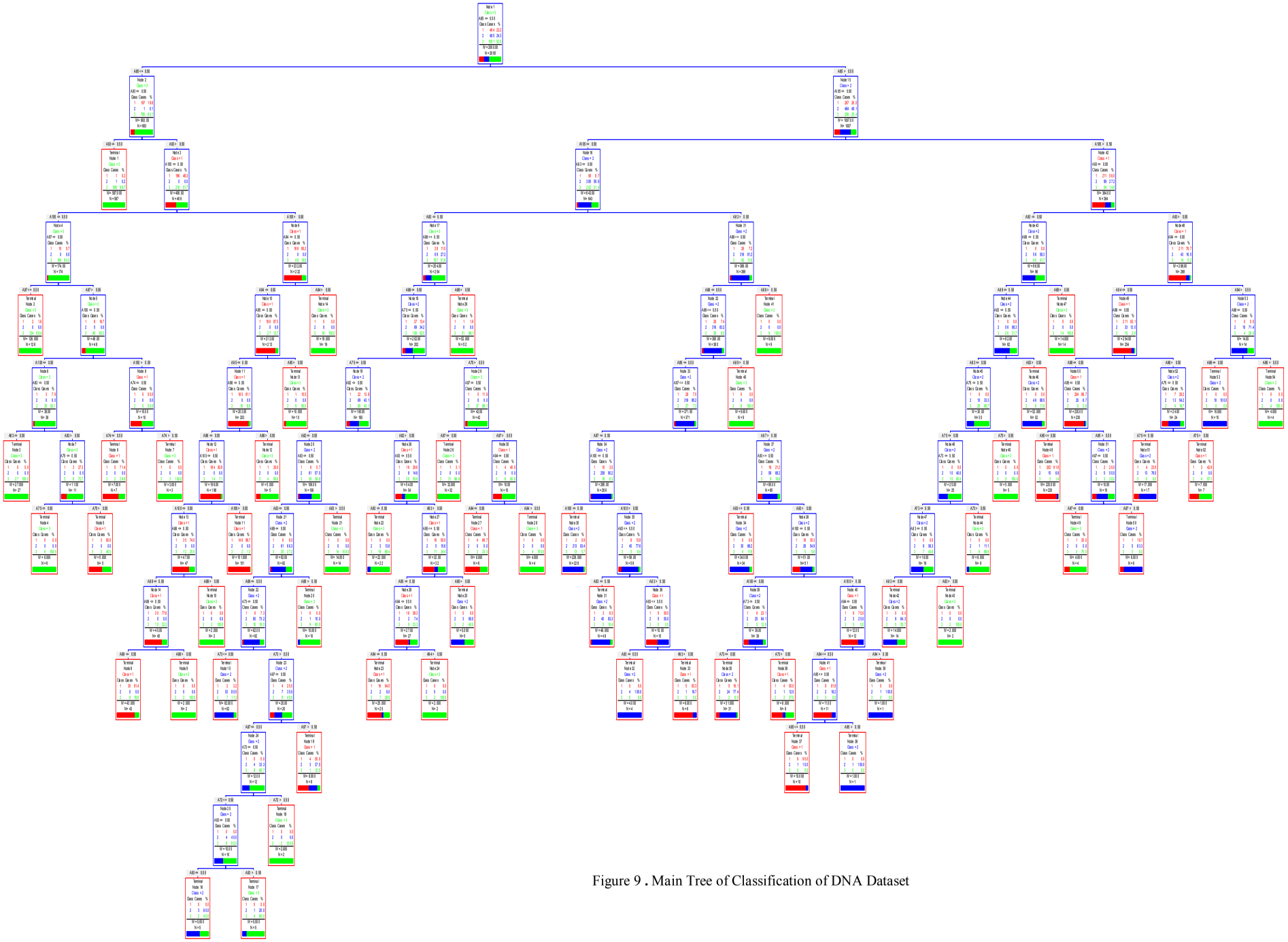

Figure 9 . Main Tree of Classification of DNA Dataset

Knowledge Base Generation by Methodology:

Rule1: *IF A85<=0.50 AND A93<=0.50 THEN* ***CLASS 1***

Rule2: : *IF A85<=0.50 AND A93>0.50 AND A105 <=0.50 AND A97<=0.50 THEN* ***CLASS 3***

Rule3: *IF A85<=0.50 AND A93>0.50 AND A105 <=0.50 AND A97>0.50 AND A100<=0.50 THEN* ***CLASS 3***

Rule4: : *IF A85<=0.50 AND A93>0.50 AND A105 <=0.50 AND A97>0.50 AND A100>0.50 AND A74<=0.50 THEN* ***CLASS 1***

Rule5: *IF A85<=0.50 AND A93>0.50 AND A105 <=0.50 AND A97>0.50 AND A100>0.50 AND A74>0.50 THEN* ***CLASS 3***

Rule6: *IF A85<=0.50 AND A93>0.50 AND A105 >0.50 AND A94<=0.50 AND A95<=0.50 THEN* ***CLASS 1***

Rule7: *IF A85<=0.50 AND A93>0.50 AND A105 >0.50 AND A94<=0.50 AND A95>0.50 THEN* ***CLASS 3***

Rule8: *IF A85<=0.50 AND A93>0.50 AND A105 >0.50 AND A94>0.50 THEN* ***CLASS 3***

Rule9: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82<=0.50 AND A63<=0.50 AND A89<=0.50 THEN* ***CLASS 2***

Rule10 *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82<=0.50 AND A63<=0.50 AND A89>0.50 THEN* ***CLASS 3***

Rule11: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82<=0.50 AND A63>0.50 THEN* ***CLASS 3***

Rule12: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82>0.50 AND A93<=0.50 THEN* ***CLASS 3***

Rule13: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82>0.50 AND A93>0.50 AND A95<=0.50 AND A94<=0.50 THEN* ***CLASS 1***

Rule14: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82>0.50 AND A93>0.50 AND A95<=0.50 AND A94>0.50 THEN* ***CLASS 3***

Rule15: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75<=0.50 AND A82>0.50 AND A93>0.50 AND A95>0.50 THEN* ***CLASS 2***

Rule16: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75>0.50 AND A97<=0.50 THEN* ***CLASS 3***
Rule17: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75>0.50 AND A97>0.50 AND A94<=0.50 THEN* ***CLASS 1***

Rule18: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88<=0.50 AND A75>0.50 AND A97>0.50 AND A94>0.50 THEN* ***CLASS 3***

Rule19: *IF A85>0.50 AND A105<=0.50 AND A83 <=0.50 AND A88>0.50 THEN* ***CLASS 3***

Rule20: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97<=0.50 AND A100<=0.50 THEN* ***CLASS 2***
Rule21: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97<=0.50 AND A100>0.50 AND A63<=0.50 THEN* ***CLASS 2***

Rule22: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97<=0.50 AND A100>0.50 AND A63>0.50 AND A93<=0.50 THEN* ***CLASS 2***
Rule23: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97<=0.50 AND A100>0.50AND A63>0.50 AND A93>0.50 THEN* ***CLASS 1***

Rule24: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93<=0.50 THEN* ***CLASS 2***

Rule25: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93>0.50 AND A100<=0.50 AND A73<=0.50 THEN* ***CLASS 2***

Rule26: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93>0.50 AND A100<=0.50 AND A73>0.50 THEN* ***CLASS 1***

Rule27: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93>0.50 AND A100>0.50 AND A94<=0.50 AND A95 <=0.50 THEN* ***CLASS 1***

Rule28: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93>0.50 AND A100>0.50 AND A94<=0.50 AND A95 >0.50 THEN* ***CLASS 2***

Rule29: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89<=0.50 AND A97>0.50 AND A93>0.50 AND A100>0.50 AND A94>0.50 THEN* ***CLASS 2***

Rule30: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88<=0.50 AND A89>0.50 THEN* ***CLASS 3***

Rule31: *IF A85>0.50 AND A105<=0.50 AND A83 >0.50 AND A88>0.50 THEN* ***CLASS 3***

Rule32: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89>0.50 THEN* ***CLASS 3***

Rule33: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89<=0.50 AND A83>0.50 THEN* ***CLASS 2***

Rule34: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89<=0.50 AND A83<=0.50 AND A75>0.50 THEN* ***CLASS 3***

Rule35: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89<=0.50 AND A83<=0.50 AND A75<0.50 AND A73>0.50 THEN* ***CLASS 3***

Rule36: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89<=0.50 AND A83<=0.50 AND A75<0.50 AND A73<=0.50 AND A63>0.50 THEN* ***CLASS 3***

Rule37: *IF A85>0.50 AND A105>0.50 AND A93<=0.50 AND A89<=0.50 AND A83<=0.50 AND A75<0.50 AND A73<=0.50 AND A63<0.50 THEN* ***CLASS 2***

Rule38: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94<=0.50 AND A98<=0.50 AND A95<=0.50 THEN* ***CLASS 1***

Rule39: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94<=0.50 AND A98<=0.50 AND A95>0.50 AND A97<=0.50 THEN* ***CLASS 3***

Rule40: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94<=0.50 AND A98<=0.50 AND A95>0.50 AND A97>0.50 THEN* ***CLASS 2***

Rule41: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94<=0.50 AND A98>0.50 AND A75<=0.50 THEN* ***CLASS 2***

Rule42: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94<=0.50 AND A98>0.50 AND A75>0.50 THEN* ***CLASS 1***

Rule43: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94>0.50 AND A88<=0.50 THEN* ***CLASS 2***

Rule44: *IF A85>0.50 AND A105>0.50 AND A93>0.50 AND A94>0.50 AND A88>0.50 THEN* ***CLASS 3***

X. CONCLUSION

This paper suggests solution of the three still open problems:

First, how you can find the actual value of the missing values? By develop of Random forest data mining algorithm.

Second, solve the problem of taken intelligent analysis of small size of dataset (i.e, insuffizent size of dataset), by propose new method for generation data called GPDCM.

Third, how you can get accurate and comprehensible rule base? These done by three stages: first stage, using PCA to reduction dimensional of database. Second stage, FP-Growth data mining algorithm using to generating association rules from the components of PCA of database. Third stage,, classification that association rules by TreeNet algorithm.

As result, proposed methodology provides fast, Accurate and comprehensible classification rules. Also, it can use to compression dataset in two dimensions (number of features"by PCA", number of records" by FP-Growth").

By experiments, found "the values of correlation matrix among the predicators (components of PCA) are 95% equivalent the rules result from FP-growth", therefore, we can use correlation matrix as alternative tool in generation association rules.

REFERENCES

[1] Breiman, Leo., "Random Forests". Machine Learning 45 (1): 5–32. 2001: doi:10.1023/A:1010933404324.

[2] DANIEL T. LAROSE,"Data Mining Methods and Models" Department of Mathematical Sciences Central Connecticut State University 2006.

[3] Erica C. And Falk H., " Using "Blackbox" Algorithms Such as TreeNet and Random Forests for Data-Mining and for Finding Meaningful Patterns, Relationships, and Outliers in Complex Ecological Data", Information science reference, Hershey • New York, 2009

[4] Freitas.A. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, 2002.

[5] Georgios K, Eleftherios K and Vassili L " Ant Seeker: An algorithm for enhanced Web Search", IFIP International Federation for information processing, Vol 204 , Artifical Intellegent Application and Innovations ,pp. 649-656, 2006.

[6] Hu, Y-J. A Genetic Programming Approach to Constructive Induction . In Proceeding of 3rd Annual Genetic Programming Conference, pp. 146–151, 1998.

[7] Hussein K., "Knowledge Discovery in database by using data mining" Ph.D. Thesis, University of Technology, 2002.

[8] Isao Okina "Extracting uncertain knowledge in database using Binary Causal Network Model", Ph.D thesis Linköpings university , Sweden, 2002.

[9] Jeril " Adboosting an exclusive implementation of jerome Friedman's MART methodology"., Salford Systems, version 2.0., 2008.

[10] Jiaxiong Pi, Yong Shi and Zhengxin Chen, From similarity retrieval to cluster analysis: The case of R*-trees, IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 2007

[11] Konar A., "Artificial Inelegant and Soft Computing: Behavioral and Cognitive of the Human Brain," CRC Press, Florida, 2000.

[12] Mannila H., "Data mining: Machnine learning, statistics and databases ", in Proc. Of 8th Int. Con. On science and statistical database management, p 2-9, Stockholm, Sweden, 1997.

[13] McGarry K., Tait J., Wermter S., MacIntyre J. "Rule Extraction from Radial Basis Function Networks," Proceedings of the International Conference on Artificial Neural Networks. p. 613-618, Edinburgh, UK, September 1999.

[14] McGarry K. "The Analysis of Rules Discovered by the Data Mining Process," 4th International Conference on Recent Advances in Soft Computing, Nottingham, UK, December 2000.

[15] Mitra P., Mitra S., and Sankar K., "Staging of Cervical Cancer with Soft Computing," IEEE Transactions On Biomedical Engineering, Vol.47, No. 7, July 2000.

[16] S. Mitra, P. Mitra, and S. K. Pal, Data Mining In Soft Computing Framework: A Survey, IEEE Transactions On Neural Networks, Vol.13, No. 1, January 2002.

[17] Mitra S., Mitra P., and Pal S., Data Mining In Soft Computing Framework: A Survey, IEEE Transactions On Neural Networks, Vol.13, No. 1, January 2002.

[18] Mitra S., "Data Mining: Multimedia, Soft Computing, and Bioinformatics," John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.

[19] Mutthew G. and Larry B., "Feature construction and selection using genetic programming and a genetic algorithm", LNCS, P 229-237, 2003.

[20] Malone J., McGarry K. and Bowerman C., "Using an Adaptive Fuzzy Logic System to Optimise Knowledge Discovery in Proteomics," International Conference on Recent Advances in Soft Computing, pp. 80-85, November 2004.

[21] Malone J., McGarry K, Bowerman C., Wermter S. "Rule Extraction from Kohonen Neural Networks. Automated Trend Analysis of Proteomics Data Using Intelligent Data Mining Architecture,"Neural Computing Applications Journal, 2005.

[22] Mahdi A.," Extracting Rules from Databases Using Soft Computing", M.Sc Thesis, University of Babylon, 2005.

[23] Nomura T. and Miyoshi T., "An Adaptive fuzzy rue extraction using Hybrid Model of Fuzzy Self-Organizing Map and The Genetic Algorithm With numerical Chromosomes," Kyoto 619-02, Japan,1998.

[24] Nian Yan : Classification Using Neural Network Ensemble with Feature Selection., Ph.D thesis Linköpings university , Sweden, 2004.

[25] Pagallo, G. & Haussler, D. Boolean Feature Discovery in Empirical Learning. In Machine Learning 5, pp. 71–99.1990.

[26] Pal S., Mitra S., and Mitra P., "Rough fuzzy MLP: Modular evolution, rule generation and evaluation," IEEE Trans. Knowledge Data Eng., 2001.

[27] Samaher H. and Mahdi A.," Knowledge Discovery in Data Mining using Fuzzy C-Means Model and Genetic Programming" , IEEE, 4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, March 25-29, 2007.

[28] Samaher hussein " Designing a Software for Knowledge Discovery in Database Using Data Mining and Soft Computing Techniques .IEEE, 2nd International Conference: E-Medical Systems October 29-31, 2008

[29] Usama F., Gregory P. and Padhraic S., "Knowledge discovery and data mining: Towards a unifinying framework ". pages 82-88, Portland, Oregon, USA, August 1996.

[30] Ulf Johansson, "Obtaining Accurate and Comprehensible Data Mining Models – An Evolutionary Approach", Ph.D thesis Linköpings universitet SE-581 83 Linköping, Sweden, 2007.

[31] Wen Xiong, Cong Wang, A Hybrid Improved Ant Colony Optimization and Random Forests Feature Selection Method for56 v/' Microarray Data. IEEE Computer Society , Fifth International Joint Conference on INC, IMS and IDC, 2009.

[32] Xindong WU and Vipin K. "The top Ten Algorithm in Data Mining" , Chaman and Hall/CRC, Data Mining and Knowledge Discovery Series, Taylor & Francis Group, LLC,2009.

[33] Zheng, Z. Constructing X-of-N attributes for decision tree learning. Machine Learning 40 1-43.2000