CARDIAC DISEASE DIAGNOSIS USING CANFIS MODEL

BY

S. AROGYA SWARNA
0820108002

OF

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641006

A PROJECT REPORT

Submitted to the

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

In partial fulfillment of the requirements
For the award of the degree

OF

MASTER OF ENGINEERING

IN

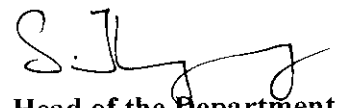
COMPUTER SCIENCE AND ENGINEERING

MAY 2010

BONAFIDE CERTIFICATE

Certified that this project report entitled "**CARDIAC DISEASE DIAGNOSIS USING CANFIS MODEL**" is the bonafide work of **Ms. S. AROGYA SWARNA**, who carried out the research under my supervision. Certified further, that to best of my knowledge the work reported herein is not from any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature of the Guide
Mrs. D. Chandrakala M.E.,
Assistant Professor,
Department of Computer
Science and Engineering.



Head of the Department
Dr.S.Thangasamy, Ph.D.,
Dean and Professor,
Department of computer
Science and Engineering

The candidate with **University Register No. 0820108002** was examined by us in the project viva-voce examination held on 17/5/10

INTERNAL EXAMINER

EXTERNAL EXAMINER

QET Dr. Mahalingam College of Engineering & Technology
TECHNOLOGY
POLLACHI-642003

CERTIFICATE

certify that Dr. / Mr. / Ms. ...S..A.ROGYA..SWARNA..... *of*

AGURU..COLLEGE..OF..TECHNOLOGY. *has participated and presented a paper titled* CARDIAC

E..DIAGNOSIS...CLASSIFICATION...BY..CANFIS....MODEL.....

ational Conference on **INFORMATION,NETWORKING AND COMMUNICATION**

IOLOGIES (NCINCT-10), *organized by Departments of ECE and IT during 16th -17th*

010.

Convener

Principal

ABSTRACT

The CANFIS (Co-Active Neuro-Fuzzy Inference System) model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. Fuzzy inference systems are also valuable as they combine the explanatory nature of rules (membership functions) with the power of "black box" neural networks. The diagnosis of diseases is a vital and intricate job in medicine. The recognition of heart disease from diverse features or signs is a multi-layered problem that is not free from false assumptions and is frequently accompanied by impulsive effects. Thus the attempt to exploit knowledge and experience of several specialists and clinical screening data of patients composed in databases to assist the diagnosis procedure is regarded as a valuable option. A proficient methodology for the classification of type the heart disease from heart disease has been proposed.

Initially, the data warehouse is pre-processed in order to make it suitable for the mining process. Once the preprocessing gets over, the membership values for all the attributes in the dataset are calculated with the aid of the fuzzy C means clustering algorithm, which will extract the data appropriate to heart attack from the warehouse. Consequently the weightage applicable to heart disease are calculated from the membership values. In addition, the rulesets vital to heart disease classification are selected on basis of the weightage with the aid of MAFIA algorithm. The neural network is trained with the fuzzy input-the membership values for the effective classification of heart attack types.

The dataset used was Cleveland database from UCI Machine Learning Repository. PSO is utilized to optimize the results. The results thus obtained have illustrated that the designed system is capable of classifying the type of heart attack more accurately. For the experiments, a computer having Intel Pentium 4 processor and 512MB RAM and JAVA 1.6 was used

ஆய்வுச்சுருக்கம்

தகவல் தேடல் தொழில்நுட்பத்தின் முன்னேற்றமானது இன்றைய வர்த்தகத்தில் முக்கிய பங்கு வகிக்கின்றது.

நோய்கண்டறிதல் என்பது மருத்துவத்துரையில் மிக நுண்ணியமான, முக்கியமான வேலை. பல வகைப்பட்ட குணங்களையும்,தன்மைகளையும் கொண்ட இருதய நோயின் வகைகளை கண்டறிதல் பல நிலைகள் கொண்ட பிரச்சினை. பல சிறப்பு வல்லுனர்களின் அறிவுத்திறனையும், அனுபவத்தையும், நோயாளிகளை சோதித்து அறிந்த தகவல்களையும் சேகரித்து அதிலிருந்து இருதய நோயினை வகைப்படுத்த கான்ஃபிஸ் என்ற மாதிரியை முன்வைக்கிறது இந்த அறிக்கை.

கான்ஃபிஸ் மாதிரி இருவெவ்வேறு உத்திகளை இணைத்து, பலதரப்பட்ட சிக்கலான செயல்பாடுகளை தெளிவாக்குகிறது. குழப்பமான ஏரணம், மற்றும் கணிணி நரம்பியல் கட்டமைப்பு என்ற இறண்டையும் கூட்டமைத்து, வேகமான, துல்லியமான முடிவுகளை கொடுக்கிறது. இருதய நோயை வகைப்படுத்த தகவல் தளத்தில் இருந்து கிடைத்த தகவல்களை பதப்படுத்தி, பின்பு தெளிவில்லாத உள்ளீட்டுகளின் உறப்புறிமை மதிப்பை கணித்து அதை கணிணி நரம்பு கட்டமைபில் இட்டு, கணிணி கட்டமைப்பை பயிற்சி செய்யப்படுகிறது. இவ்வாறு உறுவாக்கிய கான்ஃபிஸ் மாதிரி இருதய நோயின் வகையை துல்லியமாக கணிக்கிறது.

ACKNOWLEDGEMENT

I express my profound gratitude to our Chairman **Padmabhusan Arutselvar Dr. N. Mahalingam B.Sc, F.I.E** for giving this opportunity to pursue this course.

I thank, **Dr. S.Ramachandran, Ph.D.,** Principal, and **Prof. R. Annamalai,** Vice Principal, Kumaraguru College of Technology, Coimbatore, for being a constant source of inspiration and providing me with the necessary facilities to work on this project.

I would like to express a special acknowledgement and my honest thanks to **Dr. S. Thangasamy, Ph.D.,** Professor and Dean of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

I convey my special thanks to **Ms.V.Vanitha, M.E.,** Assistant Professor and Project Coordinator, Department of Computer Science and Engineering for her valuable suggestions and guidance.

I express my deep sense of gratitude and gratefulness to my Project Guide, **Ms. D.Chandrakala, M.E.,** Assistant professor, Department of Computer science and Engineering, for her kind support, supervision, tremendous patience, active involvement and guidance.

I would like to convey my honest thanks to all **members of staff** of the Department for their unlimited enthusiasm, friendship and experience from which I have greatly benefited.

I express my profound gratitude to my **parents, husband, kid and friends** for their moral support.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS	
ABBREVIATIONS	**EXPANSION**
KDD	Knowledge Discovery in DataBases
MF	Membership Function
GBELL	Generalized bell function
FCM	Fuzzy C Means
CANFIS	Co-Active Neuro Fuzzy Inference System
HD I	Heart Disease type
ANFIS	Adaptive Neuro-Fuzzy Inference Systems
PSO	Particle Swarm Optimization
NN	Neural networks
ANN	Artificial Neural Networks
MAFIA	Maximal Frequent Itemset Algorithm

LIST OF FIGURES

LIST OF TABLES

1. PROBLEM DEFINITION

A major challenge facing healthcare organizations (hospitals, medical centers) is the provision of quality services at affordable costs. Quality service implies diagnosing patients correctly and administering treatments that are effective. Poor clinical decisions can lead to disastrous consequences which are therefore unacceptable. Clinical decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and reduces diagnosis process.

Most hospitals today employ some sort of hospital information systems to manage their healthcare or patient data . Unfortunately, these data are rarely used to support clinical decision making. The main objective of this research is to develop a prototype Intelligent Heart Disease Prediction System with CANFIS and genetic algorithm using historical heart disease databases to make intelligent clinical decisions which traditional decision support systems

The cost of management of HD is a significant economic burden and so prevention of heart disease is very important step in the management. Prevention of HD can be approached in many ways including health promotion campaigns, specific protection strategies, life style modification programs, early detection and good control of risk factors and constant vigilance of emerging risk factors.

2. INTRODUCTION

2.1 Introduction to Data Mining [13, 9]:

Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information – information that can be used to increase revenue, cut costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

2.1.1 Data, Information, and Knowledge:

Data:

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- Operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- Non-operational data, such as industry sales, forecast data, and macro economic data
- Meta data – data about the data itself, such as logical database design or data dictionary definitions

Information:

The patterns, associations, or relationships among all this data can provide information. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

Knowledge:

Information can be converted into knowledge about historical patterns and future trends. For example, summary information on retail supermarket sales can be analyzed in light of promotional efforts to provide knowledge of consumer buying behavior. Thus, a manufacturer or retailer could determine which items are most susceptible to promotional efforts.

2.1.2 Data Warehouses:

Dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases into data warehouses. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining.

2.1.3 What can data mining do?

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining

demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

2.1.4 How does data mining work?

Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries.

Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table

2.2 Data Mining Techniques [14] :

The ultimate goal of data mining is prediction. Predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages: (1) Initial exploration (2) Model building with validation/verification (3) Deployment

2.2.1 Exploration:

This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and – in case of data sets with large numbers of variables ("fields") – performing some preliminary feature selection operations to bring the number of variables to a manageable range . Then, depending on the nature of the analytic problem, this first stage of the process of data mining may

involve anywhere between a simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods, in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

2.2.2 Model Building and Validation:

This stage involves considering various models and choosing the best one based on their predictive performance. There are a variety of techniques developed to achieve that goal – many of which are based on so-called "competitive evaluation of models". This means that applying different models to the same data set and then comparing their performance to choose the best. These techniques – which are often considered the core of predictive data mining – include: Bagging (voting, averaging), Boosting (to generate multiple models or classifiers (for prediction or classification), and to derive weights to combine the predictions from those models into a single prediction or predicted classification), Stacking and Meta-Learning (to combine the predictions from multiple models. It is particularly useful when the types of models included in the project are very different. In this context, this procedure is also referred to as Stacking (Stacked Generalization)).

2.2.3 Deployment:

The final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

The concept of Data Mining is becoming increasingly popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Data Mining is still based on the conceptual principles of statistics including the traditional Exploratory Data Analysis (EDA) and modeling and it shares with them both some components of its general approaches and specific techniques.

However, an important general difference in the focus and purpose between Data Mining and the traditional Exploratory Data Analysis (EDA) is that Data Mining is more oriented towards applications than the basic nature of the underlying phenomena. In other words, Data Mining is relatively less concerned with identifying the specific relations between the involved variables. For example, uncovering the nature of the underlying functions or the specific types of interactive, multivariate dependencies between variables are not the main goal of Data Mining. Instead, the focus is on producing a solution that can generate useful predictions. Therefore, Data Mining accepts among others, a "black box" approach to data exploration or knowledge discovery and uses not only the traditional Exploratory Data Analysis (EDA) techniques, but also uses techniques such as Neural Networks which can generate valid predictions but are not capable of identifying the specific nature of the interrelations between the variables on which the predictions are based.

2.3 Introduction to neural networks

Artificial Neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks are constructed for solving artificial intelligence problems.

A *neural network* (NN). in the case of artificial neurons called *artificial neural network* (ANN) or *simulated neural network* (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

However, the paradigm of neural networks - i.e., *implicit*, and not *explicit* learning is stressed - seems more to correspond to some kind of *natural intelligence* than to the traditional *Artificial Intelligence*, which would stress, instead, rule-based learning.

Applications of artificial neural networks

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

Real life applications

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modelling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind signal separation and compression.

Application areas of ANNs include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition, etc.), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

2.3.1 learning paradigms

Supervised learning

In supervised learning, we are given a set of example pairs and the aim is to find a function in the allowed class of functions that matches the examples. In other words, we wish to *infer* the mapping implied by the data; the cost function is related to the mismatch

between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the mean-squared error which tries to minimize the average squared error between the network's output, f(x), and the target value y over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called Multi-Layer Perceptrons, one obtains the common and well-known backpropagation algorithm for training neural networks.

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher," in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

Unsupervised learning

In unsupervised learning we are given some data x and the cost function to be minimized, that can be any function of the data x and the network's output, f.

The cost function is dependent on the task (what we are trying to model) and our *a priori* assumptions (the implicit properties of our model, its parameters and the observed variables).

As a trivial example, consider the model $f(x) = a$, where a is a constant and the cost $C = E[(x - f(x))^2]$. Minimizing this cost will give us a value of a that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between x and y, whereas in statistical modelling, it could be related to the posterior probability of the model given the data. (Note that in both of those examples those quantities would be maximized rather than minimized).

Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

Reinforcement learning

In reinforcement learning, data x are usually not given, but generated by an agent's interactions with the environment. At each point in time t, the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t, according to some (usually unknown) dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost; i.e., the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated.

More formally, the environment is modeled as a Markov decision process (MDP) with states and actions with the following probability distributions: the instantaneous cost distribution $P(c_t \mid s_t)$, the observation distribution $P(x_t \mid s_t)$ and the transition $P(s_{t+1} \mid s_t, a_t)$, while a policy is defined as conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the policy that minimizes the cost; i.e., the MC for which the cost is minimal.

ANNs are frequently used in reinforcement learning as part of the overall algorithm.

Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

See also: dynamic programming, stochastic control

Learning algorithms

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms

available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

Most of the algorithms used in training artificial neural networks employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

Evolutionary methods, simulated annealing, expectation-maximization and non-parametric methods are some commonly used methods for training neural networks. See also machine learning.

Temporal perceptual learning relies on finding temporal relationships in sensory signal streams. In an environment, statistically salient temporal correlations can be found by monitoring the arrival times of sensory signals. This is done by the perceptual network.

2.4. Fuzzy Logic Concepts:

2.4.1. Fuzzy Logic:

Fuzzy logic starts with and builds on a set of user supplied human language rules. The fuzzy systems convert these rules to their mathematical equivalents. This simplifies the job of the system designer and the computer, and results in much more accurate representation of the way systems behave in the real world.

Additional benefits of fuzzy logic include its simplicity and its flexibility. Fuzzy logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity.

A fuzzy system can create to match any set of input data. The Fuzzy Logic Toolbox makes this particularly easy by supplying adaptive techniques such as adaptive neuro-fuzzy inference systems (ANFIS) and fuzzy subtractive clustering. Fuzzy logic models, called fuzzy inference systems, consist of a number of confidential "if then" rules.

In fuzzy logic, unlike standard conditional logic, the truth of any statement is a matter of degree. The inference rule is the form of p -- > q (p implies q). For example, the rule if (weather is cold) then (heat is on). both variables, cold and on, has ranges of values. Fuzzy inference systems rely on membership functions to explain to the computer how to calculate the correct value between 0 and 1. The degree to which any fuzzy statement is true is denoted by a value between 0 and 1. Not only do the rule-based approach and flexible membership function scheme make fuzzy systems straightforward to create, but they also simplify the design of systems and ensure that it can easily update and maintain the system over time.

2.4.2. Fuzzy Set [15]:

Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically. For example, Fig. 2.1 illustrates bivalent sets to characterize the temperature of a room.

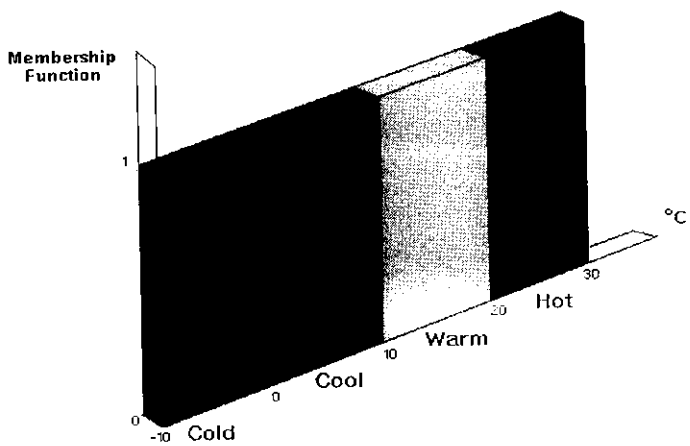


Fig. 2.1 Bivalent sets to characterize temperature of a room

The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' hence the expert knowledge we need to define our system is

mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Fahrenheit of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur. This natural phenomenon can be described more accurately by Fuzzy Set Theory. Fig. 2.2 shows how fuzzy sets quantifying the same information can describe this natural drift.

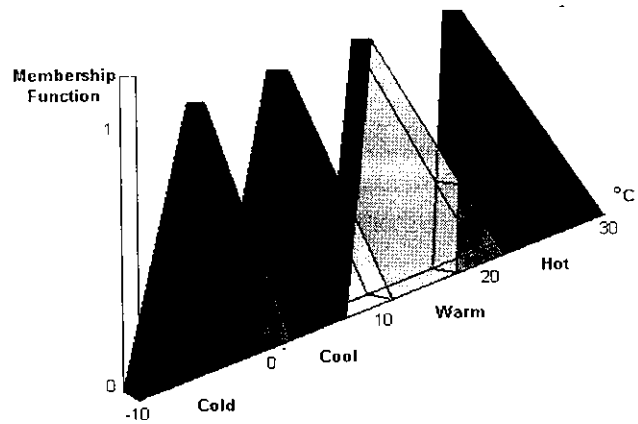


Fig. 2.2 Fuzzy sets to characterize temperature of a room

The whole concept can be illustrated with this example. Let's talk about people and "youthness". In this case the set S (the universe of discourse) is the set of people. A fuzzy subset YOUNG is also defined, which answers the question "to what degree is person x young?" To each person in the universe of discourse, we have to assign a degree of membership in the fuzzy subset YOUNG. The easiest way to do this is with a membership function based on the person's age.

young(x)= { 1, if age(x) <= 20,
(30-age(x))/10, if 20 < age(x) <= 30,
0, if age(x) > 30 }

A graph of this looks like the one in Fig. 2.3:

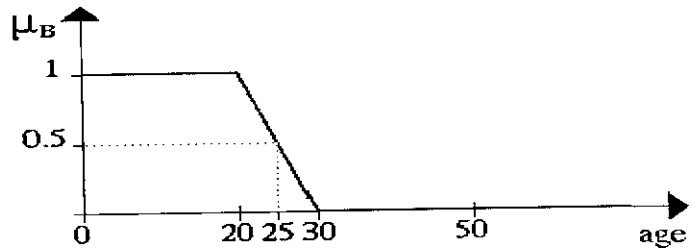


Fig. 2.3 Graph to Characterize Person's Age.

2.4.3 Fuzzy Set Operations[16] :

Universe Of Discourse :

The Universe of Discourse is the range of all values for an input to a fuzzy system.

Fuzzy Set :

A fuzzy Set is any set that allows its member to have different grades of membership in the interval.

Membership Function :

The membership function μA(x) quantifies the grade of membership of the elements x to the fundamental set X. An element mapping to the value 0 means that the member is not included in the given set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members as shown in figure 2.4

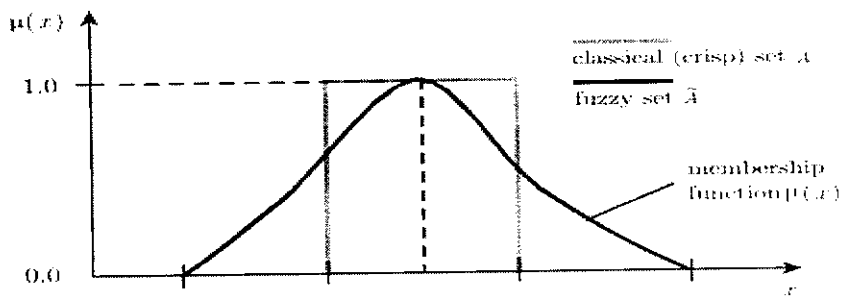


Fig. 2.4 Membership Function

Types of membership functions [18]:

1. Numerical definition (discrete membership functions)

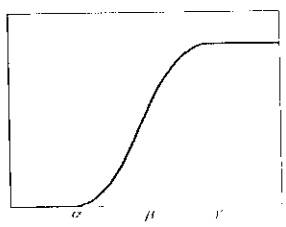
$$A = \sum_{x_i \in X} \mu_A(x_i) / x_i$$

2. Function definition (continuous membership functions)

Including of S function, Z Function, Pi function, Triangular shape, Trapezoid shape, Bell shape.

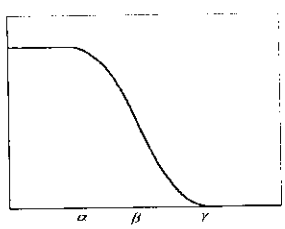
$$A = \int_X \mu_A(x) / x$$

(1) **S Function**: monotonical increasing membership function



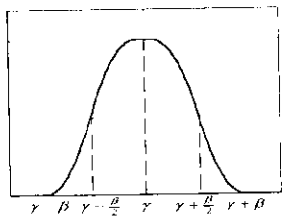
$$S(x;\alpha,\beta,\gamma) = \begin{cases} 0 & for\ x \le \alpha \\ 2(\frac{x-\alpha}{\gamma-\alpha})^2 & for\ \alpha \le x \le \beta \\ 1-2(\frac{x-\alpha}{\gamma-\alpha})^2 & for\ \beta \le x \le \gamma \\ 1 & for\ \gamma \le x \end{cases}$$

(2) **Z Function**: monotonical decreasing membership function



$$Z(x;\alpha,\beta,\gamma) = \begin{cases} 1 & for\ x \le \alpha \\ 1-2(\frac{x-\alpha}{\gamma-\alpha})^2 & for\ \alpha \le x \le \beta \\ 2(\frac{x-\alpha}{\gamma-\alpha})^2 & for\ \beta \le x \le \gamma \\ 0 & for\ \gamma \le x \end{cases}$$

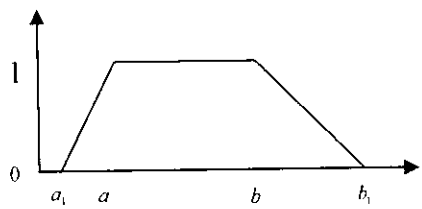
(3) **Π Function**: combine S function and Z function, monotonically increasing and



$$\Gamma(x;\beta,\gamma)=\begin{cases} S(x;\gamma-\beta,\gamma-\frac{\beta}{2},\gamma) & for\ x\le\gamma \\ 1-S(x;\gamma,\gamma+\frac{\beta}{2},\gamma+\beta) & for\ x\ge\gamma \end{cases}$$

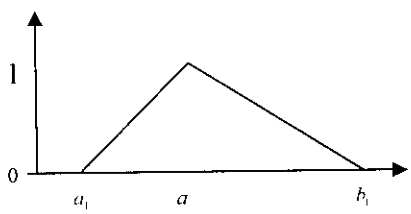
Piecewise continuous membership function

(4)Trapezoidal Membership Function



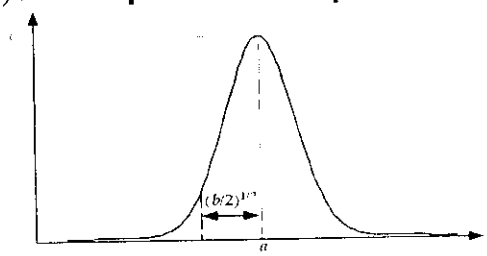
$$\mu_A(x)=\begin{cases} 0 & for\ x\le a_1 \\ \frac{x-a_1}{a-a_1} & for\ a_1\le x\le a \\ 1 & for\ a\le x\le b \\ \frac{b_1-x}{b_1-b} & for\ b\le x\le b_1 \\ 0 & for\ b_1\le x \end{cases}$$

(5) Triangular Membership Function



$$\mu_A(x)=\begin{cases} 0 & for\ x\le a_1 \\ \frac{x-a_1}{a-a_1} & for\ a_1\le x\le a \\ \frac{b_1-x}{b_1-a} & for\ a\le x\le b_1 \\ 0 & for\ b_1\le x \end{cases}$$

(6) Bell-shaped membership function



3. LITERATURE SURVEY

3.1 Efficient Supanova Kernel For Heart Disease Diagnosis [3]

This paper presented a new heuristic for computing efficiently sparse kernel in SUPANOVA.. On this data, 83.7% predictions were correct, exceeding the results obtained using the standard Support Vector Machine and equivalent kernels. It was assumed that there are N training data points in the form of vectors $xi=[xi1,xi\ 2,....,xi\ n]$, $i=1,2,...,N$. Each vector represents values of n features and has the corresponding output value, y1 they denoted the matrix containing these vectors (or training data points) as X and the vector of the corresponding output values as y.

The SUPANOVA method represents the solution as a sum of kernels that decompose functions of the order n into a sum of terms that are *unitary, 2- ary,....n-ary* order functions of the original arguments. Each function higher than first order uses a product of spline functions to represent its arguments. Hence, kernel function is replaced with a sum of kernels that measure similarity of argument vectors on a subset of features.

Limitation : Elements with small magnitude are dropped to approximate the error.

3.2 Classification & prediction using Naïve bayes, Decision trees and Neural networks [7]

Decision Tree algorithms include CART (Classification and Regression Tree). ID3 (Iterative Dichotomized 3) and C4.5. These algorithms differ in selection of splits, when to stop a node from splitting, and assignment of class to a non-split node. CART uses Gini index to measure the impurity of a partition or set of training tuples . It can handle high dimensional categorical data. Decision Trees can also handle continuous data (as in regression) but they must be converted to categorical data.

Naive Bayes or Bayes' Rule is the basis for many machine-learning and data mining methods . The rule (algorithm) is used to create models with predictive capabilities. It provides new ways of exploring and understanding data. It learns from the "evidence" by calculating the correlation between the target (i.e., dependent) and other (i.e., independent) variables.

Neural Networks consists of three layers: input, hidden and output units (variables). Connection between input units and hidden and output units are based on relevance of the assigned value (weight) of that particular input unit. The higher the weight the more important it is. Neural Network algorithms use Linear and Sigmoid transfer functions. Neural Networks are suitable for training large amounts of data with few inputs. It is used when other techniques are unsatisfactory.

Limitations :

(i)The current version of combination is based on the 15 attributes listed in . This list may need to be expanded to provide a more comprehensive diagnosis system.

(ii)Another limitation is that it only uses categorical data. For some diagnosis, the use of continuous data may be necessary.

(iii) Another limitation is that it only uses three data mining techniques The size of the dataset used in this research is still quite small.

3.3 Medical Data Classification Methods Based on Decision Tree and System Reconstruction Analysis [5]

This paper studies medical data classification methods, comparing decision tree and system reconstruction analysis as applied to heart disease medical data mining. The paper uses data from 1,723 coronary heart disease patients' cases. Each case contains about 71 attributes. The data come from a hospital clinic's observations and allow us to get a good classification of patients' status and behavior, from which we can determine the relationships among the factors. A data mining method to analyze the medical data is used. A system-reconstruction method is used to do data preprocessing and use decision-tree algorithms to do the classification. A comparison made on the classification correction rate on weighted and not weighted data, which is preprocessed by the system-reconstruction method. In this paper, first we introduce the system-reconstruction method and show how the coronary heart disease data are to be processed, and we discuss the theory and algorithms of decision trees, including, ID3, C4.5, CART, CHAID, and Exhausted-CHAID. We also apply these methods to medical data mining problems by

designing some experiments to compare the correction rate, tree depth, and leaf number of weighted and not weighted data gotten by decision tree.

Limitation : From the data – it concludes that data weighted by the system-reconstruction method can get a higher correction rate but will have little effect on the leaf number and tree depth of the decision tree.

3.4 Predicting survival causes after out of hospital cardiac arrest using data mining method [6].

The main objective of their work was to represent the relationship between variables to determine which variables were the most important for patient survival, with data mining methods. Other goals were firstly to build a decision tree to evaluate which was main factor and its impact on the patient activity in the service or physician activity. Secondly, planned to produce a real patient profile to determine they the best practice for these patients. One interest of the Bayesian network is the possibility of monitoring the propagation of the impact of an event in the network on other variables or events.

Analysis performed in three steps:

- Step 1: Learning step
- Step 2: Analysis of associations
- Step 3: Inference and prediction

For the learning step, they use the Taboo Order method to build the network . The graph of the variables showed that the probability of being alive after heart failure is directly associated with five variables: age sex, the initial cardiac rhythm, the origin of heart failure, and the type of specialized resuscitation employed. By monitoring the main node (Alive/death), it can infer conclusions about the patient profile. The patient profile observed after cardiac arrest. Forty nine point one six percent of the patients died, 6.04 % left the hospital and the 44.80 % represent non cardiac arrest or cardiac arrest where resuscitation was not possible. For each variable directly related with this node, the rhythm was equal to 1 asystole in 62.32 % of the cases, equal to 2 ventricular tachy arrhythmias FV/TV in 31.03 % and equal to 3 pulseless electrical activity in 6.65 % of

the cases. Nineteen point six four percent of the patients were younger than 46 years and 87.37 % of the patients were not hospitalised in intensive care. The hidden Markov layer (Fig. 4) demonstrated that one node appears to be highly related to the main node.

Limitation : The main limit of these tools is the necessity to have enough data to find regularity in the relationships.

3.5 A multi-class heartbeat classifier employing hybrid fuzzy - neural network [4]

Electrocardiogram (ECG) diagnosis is used to study the condition of the heart and thus its present state in addition to being inexpensive and non-invasive technique. This paper proposed a novel strategy for automatic heartbeat classification to palliate the mentioned problems. Ten types of heartbeats considered for automatic classification are Atrial Premature Contraction (APC), Fusion(F), Left Bundle Branch Block type I and type II (LBBBB I& LBBBB II), Normal(N), Paced(p), Right Bundle Branch Block type I and type II (RBBBB I & RBBBB II) , Premature Ventricular Contraction type I. and type II (PVC I & PVC II). Fuzzy cmeans clustering (FCM) is employed for feature extraction of the individual ECG cycles and these extracted features are then used for training multilayer perceptron. A detailed study undertaken to find the optimum number of clusters and optimal MLP configuration with the metric of overall percentage classification accuracy. The best FCM-MLP topology exhibited overall classification accuracy of 98.25°This network was tested for performance in presence of additive white Gaussian noise and was found to be very robust. For comparison, a well-known method of Principal Component Analysis (PCA) was also experimented with. FCM-MLP performs better than PCA-MLP in classifying the correct heartbeats. The proposed scheme exhibited an average accuracy of more than 98%.

3.6 Combination data mining methods with new medical data to predicting outcome of Coronary Heart Diseas [8] .

The prediction of survival of Coronary Heart Disease (CHD) has been a challenging research problem for medical society. The goal of this paper is to develop

data mining algorithms for predicting survival of CHD patients based on 1000 cases .We carry out a clinical observation and a 6-month follow up to include 1000 CHD cases. The survival information of each case is obtained via follow up. Based on the data. we employed three popular data mining algorithms to develop the prediction models using the 502 cases. We also used 10-fold cross-validation methods to measure the unbiased estimate of the three prediction models for performance comparison purposes. The results indicated that the SVM is the best predictor with 92.1 % accuracy on the holdout sample artificial neural networks came out to be the second with91.0% accuracy and the decision tress models came out to be the worst of the three with 89.6% accuracy. The comparative study of multiple prediction models for survival of CHD patients along with a 10-fold crossvalidation provided us with an insight into the relative prediction ability of different data.

4. OBJECTIVE

The objectives of this system are

- To preprocess the dataset to make the mining process easier.
- To determine membership values for the attributes to train the network.
- The critical rules are determined which are used to train the neural network
- To prove that the system produces consistent result .
- The system is designed to analyze the patient data and predict type of disease that may attack that patient. The Heart disease data set is used for the system

5. SYSTEM METHODOLOGY

Knowledge Discovery in Databases (KDD) means the application of non-trivial procedures for identifying effective, coherent, potentially useful, and previously unknown patterns in large databases. The KDD process generally consists of the following three phases [11].

(1) Pre-processing: This consists of all the actions taken before the actual data analysis process starts. It may be performed on the data for the following reasons: solving data problems that may prevent us from performing any type of analysis on the data, understanding the nature of the data, performing a more meaningful data analysis, and extracting more meaningful knowledge from a given set of data.
(2) Data-mining: This involves applying specific algorithms for extracting patterns or rules from data sets in a particular representation.
(3) Post-processing: This translates discovered patterns into forms acceptable for human beings. It may also make possible visualization of extracted patterns.

The CANFIS model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. Fuzzy inference systems are also valuable, as they combine the explanatory nature of rules (MFs) with the power of neural networks. These kinds of networks solve problems more efficiently than neural networks when the underlying function to model is highly variable or locally extreme .
The fundamental component of CANFIS is a fuzzy axon, which applies membership functions to the inputs. The output of a fuzzy axon is computed using the following formula: $f_j(x,w) = \min \forall_i (MF(x_i, w_{ij}))$.
where i = input index, j = output index, xi = input i, wij=weights (MF parameters) corresponding to the j th MF of input i and MF=membership function of the particular subclass of the fuzzy axon. This system can be viewed as a special three-layer feed forward neural network. The first layer represents input variables, the middle (hidden)

CANFIS Architecture

A CANFIS structure with *n* inputs and one output. For model initialize, a common rule set with *n* inputs and *m* IF-THEN rules . Layers in CANFIS structure can be adaptive or fixed and their functions are:

Layer 1 (Premise Parameters): Every node in this layer is a complex-valued membership function (μij) with a node function: $O1,ij = |\mu Aij(zi)| \llcorner \mu Aij(zi)$ for $(1 \le i \le n,\ 1 \le j \le m)$. Each node in layer 1 is the membership grade of a fuzzy set (Aij) and specifies the degree to which the given input belongs to one of the fuzzy sets.

Layer 2 (Firing Strength): Every node in this layer is product of all the incoming signals. This layer receives input in the form of the product of all the output pairs from the first layer: $O2,j = wj = \mu Ai1(z1)\mu Ai2(z2), \ldots, \mu Ai1n(zn)$ for $(1 \le i \le m)$

Layer 3 (Normalized Firing Strength): Every node in this layer calculates rational firing strength:

$$O_{3,j} = \overline{w_j} = \frac{w_j}{\sum_{j=1}^{m} w_j} \quad \text{for } (1 \le j \le m).$$

Layer 4 (Consequence Parameters): Every node in this layer is multiplication of Normalized Firing Strength from the third layer and output of neural network:

$$O_{4,j} = \overline{w_j}\, u_j = \overline{w_j}\,(P_{j1}Z_1 - P_{j2}Z_2 - \cdots \quad P_{jn}Z_{2n} - q_j)$$

for $(1 \le j \le m)$

Layer 5 (Overall Output): The node here computes the output of CANFIS network:

$$O_{5,1} = \Sigma\, \overline{w_j}\, u_j$$

The bell fuzzy axon used in this study is a type of fuzzy axon that uses a bell-shaped curve as its membership function. Each MF takes three parameters stored in the weight vector of the bell fuzzy axon

$$MF(x,w)=\frac{1}{1+\left|\frac{x-w_2}{w_0}\right|^{2w_1}}$$

where x =input and w=weight of the bell fuzzy axon. Fuzzy axons are valuable because their MF can be modified through back propagation during network training to expedite the convergence. The second major component of CANFIS is a modular network that applies functional rules to the inputs. The number of modular networks matches the number of network outputs and processing elements in each network corresponding to the number of MFs. Two fuzzy structures are mainly used: the Tsukamoto model and the Sugeno (TSK) model. Finally, a combiner is used to apply the MF outputs to the modular network outputs. The combined outputs are then channeled through a final output layer, and the error is back propagated to both the MF and the modular network

The system is divided into four major modules

- Data preprocessing
- Training the network
- Optimization
- Performance evaluation

5.1.1 Description of the database

The data base consists of 303 cases. It represent seventy two different types of information about the patient. The selected fourteen attributes was collected from UCI machine learning database. The attributes and its details are shown in Table 5.1

Table 5.1 Attributes and its description

No.	Attributes	Details
1	Sex	(value 1:male ; value 0 : female)
2	Chest pain type	(value 1:typical type1 angina ; value 2: typical angina ; value 3: non angina pain ; value 4: asymptomatic)
3	Fasting blood	sugar (value 1: > 120 mg/dl; value 0: < 120 mg/dl)
4	Restecg	(value 0: normal; value 1:havng st wave normality ;value 2: showing probable or defnite left ventrcular hypertrophy)
5	Exang	exercise nduced angina (value 1:yes: value 0: no)
6	Slope	The slope of the peak exercise ST segment (value 1: unsloping ; value 2: flat; vlaue 3: downsloping)
7	CA	Number of major vessels coloured by fluroscopy(value 0-3)
8	Thal	(value 3: normal; value 6: fxed defect ; value 7: reversible defect)
9	Trest blood pressur	(in mm hg)
10	Serum cholestral	(mm/hg)
11	Thalach	maximum heart rate achieved
12	Old peak	ST depression induced by exercise relative to rest
13	Age	In years

The overall system structure of the project is shown in Fig 5.1. Each one the steps are explained in detailed n the following sections.

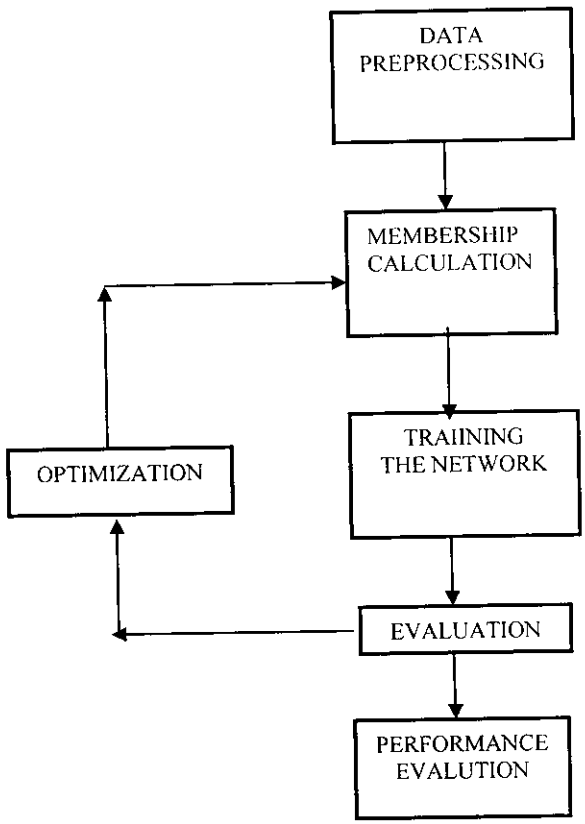


Fig 5.1 System Architecture

5.2 Data preprocessing

Real world data is incomplete, inconsistent and noisy. For quality mining result quality data is needed. So data preprocessing is an important role in data mining. There are different stages in preprocessing which are described in Figure 5.2

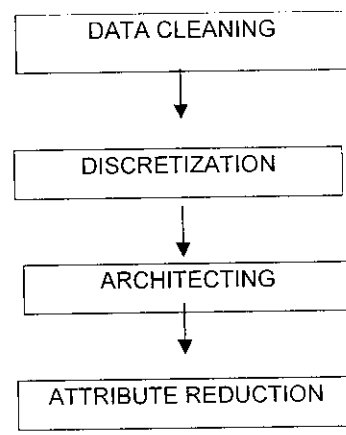


Figure 5.2 Preprocessing steps

5.2.1 Data cleaning

Among the data of 303 cases, the order numbers with 167,193,288,303 has missing values for attribute CA. Mean \mode and Inference based fill is used to treat CA column. Both the methods gave the same results for filling the column.

5.2.2 Discretization

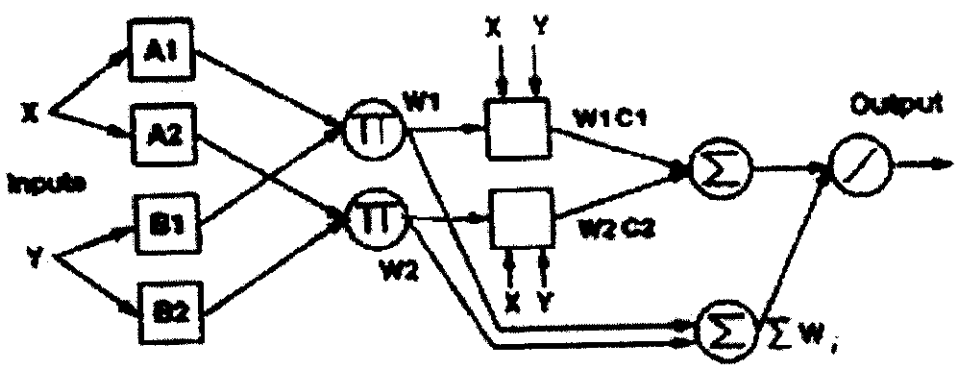
Certain continuous attribute values are not suitable for mining hence these attributes are discretized. Entrophy based discretization, Histogram discretization, Equal frequency binning algorithm were used .Equal frequency binning algorithm provided good result without overlapping intervals when compared to other methods.

5.2.3 Attribute reduction

Cleveland database contains thirteen attributes, not all attributes are essential. Therefore some redundant attributes are reduced using Rosetta software Johnson's algorithm to attain 9 attributes . Selected Attributes are AGE, CP, TRESTBPS, FBS, RESTECG, THALACH, OLDPEAK, THAL, CA .Four attributes SEX, CHOL,EXANG, SLOPE were deleted.

5.3 Training the network

The hidden layer of the neural network consist of five different stages . The overall view of the system with two input and one output is shown in figure 5.3 .



5.3.1 Stages in Training the network

There are different stages in training the network with various methodologies. These steps are shown in figure 5.3

Step 1 :Membership Calculation

Membership values are calculated using gbell, Gaussian Function and FCM methods. Only gbell function resulted in calculating the membership values. The sample result .

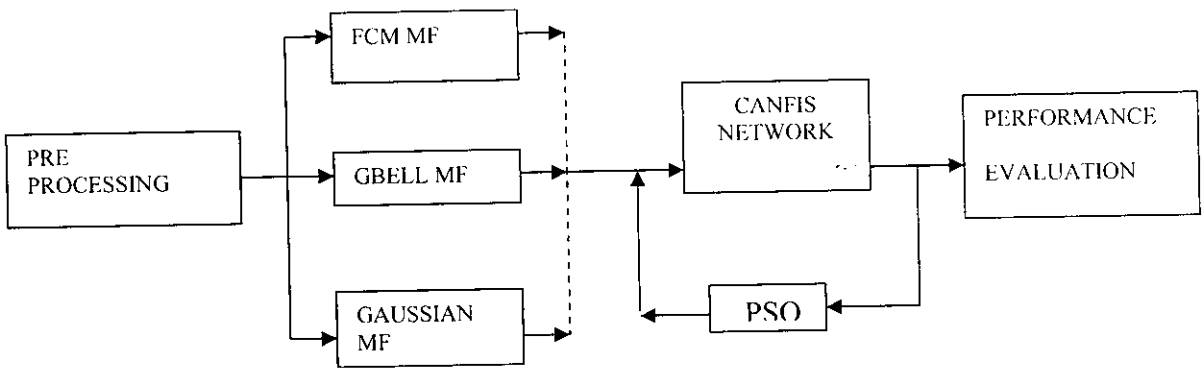


Fig 5.4 Membership functions used in network

The formula for gbell function is mentioned below where x = input and a,b,c are the weight of the gbell fuzzy axon. Fuzzy axons are valuable because their MF can be modified through back propagation during network training.

$$f(x;a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$

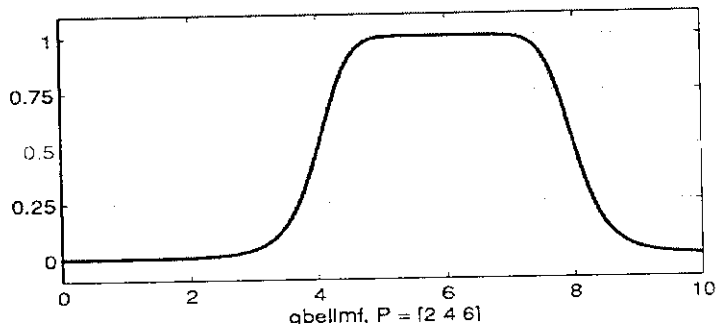


Fig 5.5 **gbell mf curve**

GAUSSIAN MF

The formula for Gaussian function is mentioned below where x = input and c and α centre and width are they are the parameters of the Gaussian functions.

$$\mu_{Ai}(x) = \exp(-\frac{(c_i - x)^2}{2\sigma_i^2})$$

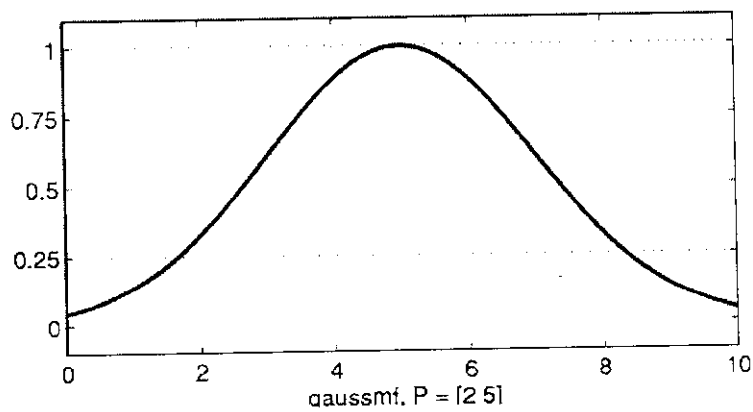


Figure 5.6 Gaussian MF curve

FUZZY C-MEANS

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. The algorithm is given below, where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, algorithm

1. *Initialize* $U=[u_{ij}]$ *matrix,* $U^{(0)}$
2. *At k-step: calculate the centers vectors* $C^{(k)}=[c_j]$ *with* $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

3. *Update* $U^{(k)}$, $U^{(k+1)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

4. *If* $|| U^{(k+1)} - U^{(k)}|| < \varepsilon$ *then STOP; otherwise return to step 2.*

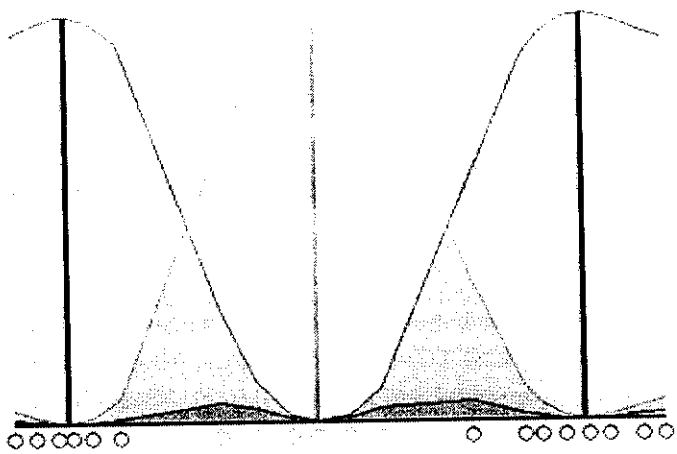


Figure 5.7 FCM graph

Step 2: Weight calculation

This layer receives input in the form of the product of all the output pairs from the first layer:

$$W(j) = f(x;a,b,c)(z1)\ f(x;a,b,c)(z2)\ldots\ldots\ldots\ldots\ f(x;a,b,c)(zn)$$

for ($1 \leq j \leq$ no of attrbutes).

Step 3 :Consequent layer

Rules are formulated accordingly using MAFIA. The rules determines the type of classification. Input given to the algorithm is the dataset. The transactions are the records represented in rows. The algorithm is shown below.

MAFIAlgorithm

Input : transactions D, mn_sup =3

Output : support of items, maxitems

Step A: For the sake of efficient implementation, the items are assumed that the set of items I is an ordered set and if X = {x1,...,xn} is an itemset, then x1 < ... < xn.

Step B: Count the support of each item I in a given set of transactions

Step C : if support(itemset I) > = min_sup and support(itemset I+1) > = min_sup
then go to step3

Step D : find all permutations of the items I and I+1

Step E: find the intersection of the transactions of each items in each permutation

Step F: Count the supports for I and (I+1) itemsets in a single pass
If the itemsets are infrequent, go to step C.

Step G: Return supports and items

Step 4: output calculation and Fitness evaluation

Output of the fuzzy network is calculated using the consequent part z1,z2 calculations and weight obtained from the membership calculation w1 w2.. the formula is given below

$$\text{Output} = \frac{w1\ z1 + w2\ z2}{w1 + w2}$$

Fitness evaluation made using minimum error function $E = \frac{1}{2} (y_t - y)^2$

Step 6 : Optimization

Particle swarm optimization (PSO) is a method for performing numerical optimization without explicit knowledge of the gradient of the problem to be optimized. PSO optimizes a problem by maintaining a population of candidate solutions called particles and moving these particles around in the search-space according to simple formulae. The movements of the particles are guided by the best found positions in the search-space, which are continually updated as better positions are found by the particles.

5.3.2 Performance evaluation

Performance of the CANFIS model with various membership values are found using sensitivity, specificity,recall accuracy formulas from confusion matrix and dataset are partitioned in various percentage for training and testing the CANFIS. The formula are mentioned below.

Sensitivity = (TP / TP + FN) 100

Specificity = (TN / TN + FP) 100

Accuracy = ((TP+TN) / (TP+FN+TN+FP))100

Recall = (TP / (TP + FP)) 100

6. SIMULATION RESULTS

6.1 Experimental Results:

This chapter is devoted to explain the simulation results because results of the project promise the effective operation of the model that has been designed. The classification of the patterns is a multi-stage system with data preprocessing module, parameter extracting and membership value calculation module, Rule generation module, weight calculation module and classification module which are clearly analyzed in this chapter.

The pattern classification system is tested on Intel Core 2 Duo (3846 MHz clock speed), 3 GB RAM desktop PC. For testing the validity of the proposed model the Cleaveland database is selected from the UCI Machine Learning Repository [14]. The dataset is preprocessed for elimination of unknown values. The preprocessing modules consisting of various steps like cleaning, discretization, attribute reduction and architecting. The determination of the membership function through Gaussian function, Bell Function and C-Means fuzzy logic methods. The proposed method is a hybrid version of CANFIS model and the Particle Swarm Optimization algorithm.

Data Preprocessing:

Real world data is incomplete, inconsistent and noisy. Data preprocessing techniques can improve the quality of the data thereby helping to improve the accuracy and efficiency of the subsequent mining process. For getting quality results we need to preprocess the dataset first. A summary of the datasets taken from the UCI Machine learning repository is shown in Table 6.1. The datasets are selected in such a way that the problems chosen are with atleast 5 classes and only 4 missing values in 'ca' attribute. The database consisting of values for 74 attributes of a patient from that only the most important 14 attributes are selected.

Table 6.1 Database summary

Datasets	Number of instances	Number of attributes	Number of attributes chosen for classification	Number of classes	Missing values	Area
Cleveland	303	74	14	5	4	Heart Disease

The preprocessing is done to convert the available data into the form that is required for the classification process. The datasets are preprocessed by available methods namely data cleaning, data discretization, data architecting and data reduction. From the Database selected to construct the decision making table, of which 5 cases of data are shown in Table 6.2.

Table 6.2 Cleveland database of 5 cases of Heart Disease data

Sl No	age	sex	cp	trestbps	chol	fbs	recg	thalach	exang	oldpeak	slope	ca	thal	dia	dtype
1	63	Male	Angina	145	233	True	Hyp	150	Fal	2.3	Down	0	Fix	Buff	H
2	67	Male	Asympt	160	286	Fal	Hyp	108	True	1.5	Flat	3	Norm	Sick	S2
3	67	Male	Asympt	120	229	Fal	Hyp	129	True	2.6	Flat	2	Rev	Sick	S1
4	37	Male	Notang	130	250	Fal	Norm	187	Fal	3.5	Down	0	Norm	Buff	H
5	41	Fem	Abnang	130	204	Fal	Hyp	172	Fal	1.4	Up	0	Norm	Buff	H

(i) Data Cleaning:

Among the data of 303 instances, the instance numbers 167,193,288 and 303 have null values for attribute 'ca'. The following are the various methods to fill null values.

a.Ignore the tuple:

This is usually done when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

b.Fill in the missing value manually:

This approach is time consuming and may not be feasible given a large dataset with many missing value.

c. Use global constant to fill in the missing value:

Replace all missing attribute values by the same constant such as a label like " unknown: or "-∞".

d.Use attribute mean to fill in the missing value:

The mean value of the particular attribute having missing value can be calculated. Use this value to replace the missing value for that attribute.

e.Use the most probable value to fill in the missing value :

This is determined with Bayesian formula.

Among these various methods available for filling the null values the last two methods are suitable for medical field and these two methods yield same result as shown in Table 6.3.

Table 6.3 Filling missing values for CA

TRAINING SAMPLES	Number of CA values missing	Mean fill	Inference based (Bayesian formula)
153	2	0	.004
203	3	0	.002
303	4	0	.002

ii) Data Discretization:

The discretization of continuous attributes to integer attributes, to get the partition point of these attributes. The continuous attribute values are discretized using Entrophy based discretization, Histogram discretization, Equal Frequency Binning algorithm. Equal Frequency binning algorithm provided good result without overlapping intervals when compared to other methods. The discretized attributes and their intervals are shown in Table 4.

a. Equal Frequency Binning Algorithm:

The attribute values can be discretized by distributing the values into bins, and replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians. respectively. These techniques can be applied recursively to the resulting partitions in order to generate concept hierarchies.

Table 6.4

Table 6.4: Discretization using Equal Frequency Binning algorithm

Attributes	Ranges of Discretization			Number of Ranges
	First range	Second range	Third range	
Age	(-∞, 52)	(52, 60)	(60, +inf)	3
trestbps	(-∞, 123)	(123, 139)	(139, +inf)	3
chol	(-∞, 223)	(223, 265)	(265, +inf)	3
Thalch	(-∞ ,143)	(143, 162)	(162, +inf)	3
oldpeak	(-∞, 0.1)	(0.1, 1.4)	(1.4, +inf)	3

b. Entropy based discretization:

An information based measure called entropy can be used to recursively partition the values of a numeric attribute A, resulting in a hierarchical discretization. Such a discretization forms a numerical concept hierarchy for the attribute. This method can reduce data size. This is shown in Table 6.5.

Table 6.5 Entropy based discretizatioin results

Attributes	First Ranges	Second Ranges	Third Ranges	Fourth Ranges	Number of ranges
age	(-∞, 50)	(52, 61)	(60,+ ∞)	-	2
trestbps	(-∞, 120)	(123, 139)	(139, 140)	(138,140)	4
chol	(-∞, 223)	(223, 261)	(260, 265)	(267,+∞]	4
thalch	(-∞140)	(143, 150)	(150, +∞)		3
oldpeak	(-∞, 0.01)	(0.021, .04)	(0.14, +∞)		3

C .Histogram based discretization

Histograms present a graphical representation of data, providing useful information about the distribution of a random variable. A histogram is visualized as a bar graphthat shows frequency data. The basic algorithm to construct an histogram consists of sorting the values of the random variable and place them into bins. Then we count the number of data points in each bin. The height of the bar drawn on the top of each bin is proportional to the number of observed values in that bin. The results of the histogram method s shown in Table 6.6

Table 6.6 Histogram discretization results

Attributes	First Ranges	Second Ranges	Third Ranges	Fourth Ranges	Number of ranges
age	(+∞, 55)	(60, -∞)			2
trestbps	(+∞, 125)	(124, 140)	(139, -∞)		3
chol	(+∞, 230)	(229, 225)	(260, -∞)		3
thalch	(+∞142)	(143, 152)	(160, 165)	(166,-∞)	34
oldpeak	(+∞, 0.2)	(0.2, 1.2)	(1.3, -∞)		3

Equal Frequency Binning algorithm provides good result without overlapping intervals when compared to other methods.

iii) Data Architecting:

Various values of all attributes can be mapped as symbols. As the attribute value after the discretization, such as age with interval (-∞, 52) mapped to 0, (52, 60) mapped to 1 and (60, +inf) mapped to 2. same way the remaining continuous value parameters are changed to discrete values so that it will be easy for mining results.

iv) Attribute Reduction:

Patient data from Cleveland database contains 14 attribute, compared to decision making table, not all of the attributes are essential, and therefore we can remove some redundant attributes by attribute reduction . Rosetta software's Johnson's algorithm is used to reduce attributes of data of 303 cases, and to attain 9 reduced attributes as age, cp, trestbps, fbs, restecg, thalach,oldpeak,ca,thal. After that, the four attributes (sex, chol, exang,alope) were deleted.

The final preprocessed results are shown in Table 6.7 with sample records.

Table 6.7 Preprocessed dataset

SL NO	age	sex	cp	trestbps	chol	fbs	recg	thalach	exang	oldpeak	slope	ca	thal	dtype
1	2	1	1	2	1	1	2	1	0	3	3	0	6	0
2	2	1	4	2	2	0	2	0	1	3	2	3	3	2
3	2	1	4	0	1	0	2	0	1	3	2	2	7	1
4	0	1	3	1	1	0	0	2	0	3	3	0	3	0
5	0	0	2	1	0	0	2	2	0	3	1	0	3	0

Membership value calculation

In order to handle the fuzzy data, it is necessary to convert the actual data into fuzzy data based on certain membership functions. Membership values are calculated using gbell, Gaussian Function and FCM methods. Only gbell function resulted in proper the membership values.

a. Gbell MF

The results for gbell MF are shown in Table 6.8 where x = input and a,b,c are the width, slope, centre respectively of the gbell fuzzy axon . Table 6.9 shows values of MF for various x values.

Table 6.8 Gbell MF values for Age when x = 0

Width(a)	Slope (b)	Cetre(c)	Membership function
0.003	0.01	0.4	0.726
0.004	0.12	0.6	0.3068
0.1	0.2	1.0	0.3981
0.2	0.4	1.6	0.1592
0.3	0.6	2.0	0.0930

Table 6.9 Gbell MF for Age with various x values

X	MF	MF + PSO
0	0.398	0.42
1	0.5283	0.124
2	0.8247	0.8111

b.Gaussian MF

Gaussian membership function is defined by 2 parameters c and σ control the center and width of the membership function. A plot of the Gaussian is shown in figure 5.6 Gaussian MF values for Age attribute is shown in Table 6.10 where σ is the width and C is the centre of the curve. Table 6.11 shows MF values for other few attributes.

Table 6.10 Gaussian MF values for Age attribute where x=0

Width (σ)	Centre (c)	Membership function
0	1.20	0.548
1	1.50	0.314
2	1.84	0.0008
3	20.0	0.0003

Table 6.11 Gaussian MF values for other attributes

Attribute name	Width (σ)	Centre(c)	Membership function
age	2	1.84	0.0008

c. Fuzzy C means clustering

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, Fuzzy partitioning is carried out through an iterative manner , with the update of membership u_{ij} and the cluster centers c_j . For FCM MF values for only age attribute is shown in Table 6.12, where ε is stopping criteria which ranges from zero to one. Table 6.13 shows MF for other few attributes.

Table 6.12 FCM MF Values for Age attribute

Attribute name	Stopping criteria (ε)	No of iterations	Membership function
age	0.5	17	0.81
cp	0.4	18	0.53
trestbps	0.3	25	0.23
fbs	0.1	39	0.51

Table 6.13 FCM MF values for other attributes

Stopping criteria (ε)	No of iterations	Membership function
0.9	10	0.57
0.7	11	0.59
0.5	17	0.81
0.4	17	0.44

Rule Generation

Rules are formulated accordingly using MAFIA. The rules determines the type of classification. The IF (antecedent) part is fuzzy in nature, whereas the THEN (consequent) part is a crisp function of an antecedent variable The study presented here could be written as:

Rule-1:if THAL is LOW and TRESTBPS is LOW and OLDPEAK is HIGH and THALACH is HIGH

Then Z1 = THAL * 0 .2 + TRESTBPS * 0.3 + OLDPEAK * 0.6 + THALACH * 0.7

Support = 45 ; Confidence = 0.818

Rule-2: if CP is LOW and AGE is LOW and FBPS is HIGH and OLDPEAK is HIGH

Then Z1 = CP * 0 .3 + AGE * 0.3 + FBPS * 0.7 + OLDPEAK * 0.9

Support = 28 ; Confidence = 0.778

Rule-3: if RESTECG is LOW and CP is LOW and TRESTBPS is HIGH and HTYPE is HIGH

Then Z1 = RESTECG * 0 .2 + CP * 0.2 + TRESTBPS * 0.8 + DTYPE * 0.9

Support = 29 ; Confidence = 0.828

Rule -4 : if CP is LOW and AGE is LOW and FBS is HIGH

Then Z1 = CP * 0 .4 + AGE * 0.2 + FBS * 0.8

Support = 10 ; Confidence = 0.769

Rule -5 : if TRESTBPS is LOW and FPS is LOW and THAL is LOW and AGE is LOW

Then Z1 = TRESTBPS * 0 .1 + FBPS * 0.2 + THAL * 0.1 + AGE * 0.2

Support = 158 ; Confidence = 0.936

Performance evaluation

Performance of the CANFIS model with various membership values are found using sensitivity, specificity,recall accuracy formulas from confusion matrix and dataset are partitioned in various percentage for training and testing the CANFIS. The actual classification are shown in table 6.14.The results of CANFIS for different training and testing sets shown in Table 6.15. the performance of CANFIS for Different MF values are also shown in Table 6.16.

Table 6.14 Classification of heart disease

Data set	Type H	Type S1	Type S2	Type S3	Type S4
303	164	55	36	35	13

Table 6.15 Accuracy and Timing for different training dataset

TRAINING SAMPLES	TESTING SAMPLES	CANFIS Classification Without PSO			CANFIS Classification With PSO		
		Accuracy %	Training Time in sec	Testing Time in sec	Accuracy %	Training Time in sec	Testing Time in sec
30 %	70%	83	0.122	0.31	85	0.12	0.532
50%	50%	86	0.225	0.29	88	0.24	0.32
60%	40	88	0.229	0.22	90	0.28	0.25
80%	20%	92	0.357	0.14	94	0.32	0.12

Performance of the CANFIS model with various membership values are found using sensitivity, specificity,recall accuracy formulas from confusion matrix and dataset are partitioned in various percentage for training and testing the CANFIS.

The formula are mentioned below.

Sensitivity = (TP / TP + FN) 100

Specificity = (TN / TN + FP) 100

Accuracy = ((TP+TN) / (TP+FN+TN+FP))100

Recall = (TP / (TP + FP)) 100

Table 6.16 Performance Evaluation Number of training samples considered 80%

	Gbell		**Gaussian**		**FCM**	
	CANFIS	**CANFIS + PSO**	**CANFIS**	**CANFIS + PSO**	**CANFIS**	**CANFIS + PSO**
Positive	289	291	284	285	290	292
Negative	13	12	19	18	13	11
True positive classified	280	286	281	279	285	286
True negative classified	9	5	3	6	5	6
False positive classified	6	5	8	10	8	5
False negative classified	5	7	11	8	5	6
Accuracy (%)	95	96	93	94	96	97
Sensitivity (%)	98	97	96	97	98	97
Specificity (%)	60	50	27	37	38	54
Recall (%)	97	98	97	96	97	98

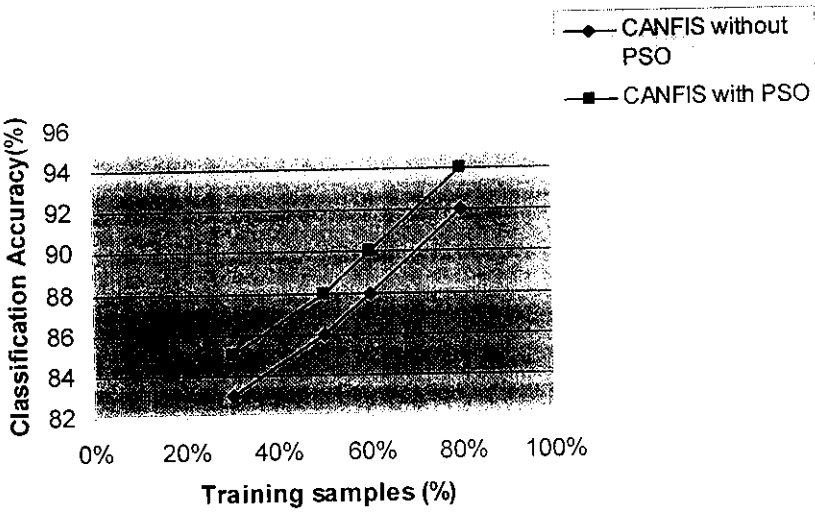


Figure : 6.1 comparison between CANFIS Vs CANFIS with PSO

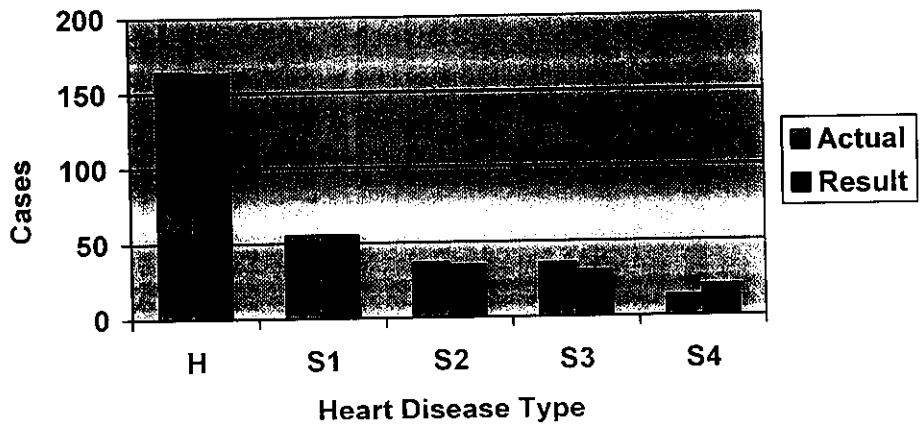


Figure 6.2 : Comparison of Actual cases Vs. Obtained cases through CANFIS with PSO

6.2 Conclusion:

In this project work, classification of various heart disease by combining nueral networks and fuzzy sets was proposed. To get accurate classification results, this method uses Genetic algorithm. Experiments conducted on the heart disease dataset illustrated that the proposed approach produces meaningful results and has reasonable efficiency. The results of the proposed model is consistent and hence encouraging.

6.3 Future Scope:

This model can be further trained with more data set to get cent percent result and can be further developed for predicting unknown patients record , further user interface can be developed for easy access.

APPENDIX – I

SCREEEN SHOTS

FCM showing centroid value and membership values for attributes

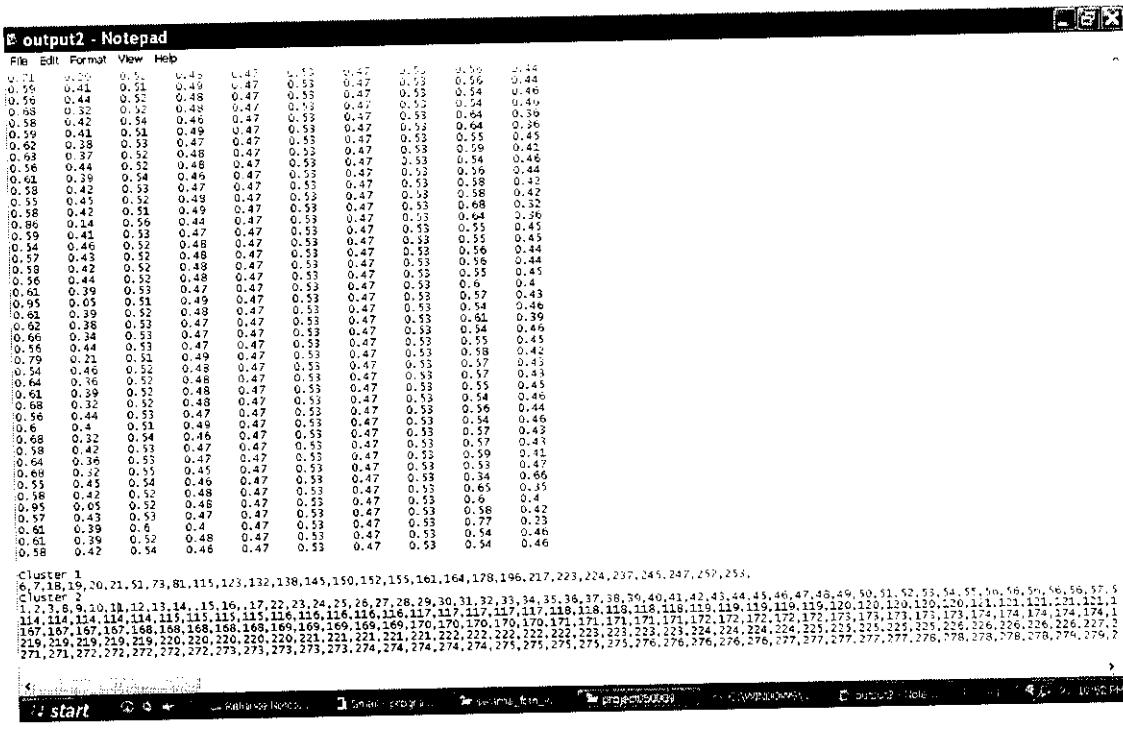
output - Notepad

File Edit Format View Help

```
centroid of cluster 1 = 106.75664
centroid of cluster 2 = 105.41201
centroid of cluster 1 = 108.425545
centroid of cluster 2 = 103.09779
After C-Means mf output
*****************************************
```

0.57	0.43	0.52	0.48	0.47	0.53	0.47	0.53	0.56	0.44
0.55	0.45	0.51	0.49	0.47	0.53	0.47	0.53	0.99	0.01
0.68	0.32	0.52	0.48	0.47	0.53	0.47	0.53	0.61	0.39
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.53	0.47
0.61	0.39	0.53	0.47	0.47	0.53	0.47	0.53	0.54	0.46
0.68	0.32	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.58	0.42	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.68	0.32	0.51	0.49	0.47	0.53	0.47	0.53	0.55	0.45
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.56	0.44
0.58	0.42	0.53	0.47	0.47	0.53	0.47	0.53	0.55	0.45
0.58	0.42	0.53	0.47	0.47	0.53	0.47	0.53	0.56	0.44
0.58	0.42	0.51	0.49	0.47	0.53	0.47	0.53	0.56	0.44
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.57	0.43
0.68	0.32	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.54	0.46	0.53	0.47	0.47	0.53	0.47	0.53	0.55	0.45
0.56	0.44	0.54	0.46	0.47	0.53	0.47	0.53	0.54	0.46
0.95	0.05	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.58	0.42	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.58	0.42
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.95	0.05	0.53	0.47	0.47	0.53	0.47	0.53	0.57	0.43
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.68	0.32	0.51	0.49	0.47	0.53	0.47	0.53	0.55	0.45
0.6	0.4	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.61	0.39	0.53	0.47	0.47	0.53	0.47	0.53	0.6	0.4
0.68	0.32	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.68	0.32	0.51	0.49	0.47	0.53	0.47	0.53	0.54	0.46
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.79	0.21
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.95	0.05	0.54	0.46	0.47	0.53	0.47	0.53	0.79	0.21
0.58	0.42	0.52	0.48	0.47	0.53	0.47	0.53	0.56	0.44
0.72	0.28	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.58	0.42	0.51	0.49	0.47	0.53	0.47	0.53	0.55	0.45
0.59	0.41	0.52	0.48	0.47	0.53	0.47	0.53	0.55	0.45
0.61	0.39	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.58	0.42	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.68	0.32	0.54	0.46	0.47	0.53	0.47	0.53	0.68	0.32
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.86	0.14
0.6	0.4	0.51	0.49	0.47	0.53	0.47	0.53	0.6	0.4
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.58	0.42
0.56	0.44	0.52	0.48	0.47	0.53	0.47	0.53	0.79	0.21
0.58	0.42	0.53	0.47	0.47	0.53	0.47	0.53	0.54	0.46
0.55	0.45	0.51	0.49	0.47	0.53	0.47	0.53	0.55	0.45
0.56	0.44	0.53	0.47	0.47	0.53	0.47	0.53	0.55	0.45
0.61	0.39	0.51	0.49	0.47	0.53	0.47	0.53	0.54	0.46
0.86	0.14	0.52	0.48	0.47	0.53	0.47	0.53	0.54	0.46
0.95	0.05	0.54	0.46	0.47	0.53	0.47	0.53	0.65	0.35

FCM showing records belonging to clusters



APPENDIX - II

SAMPLE CODING

FCM code

```
import java.awt.Point;

import java.util.Random;
import java.io.*;


/**
 * This class implements a basic Fuzzy C-Means clustering algorithm
 */
public class FCMC
{
 private int width,height,numBands;

 private int maxIterations,numClusters;

  private float fuzziness; // "m"
  private float[][][] membership;
  private int iteration;
  private double j = Float.MAX_VALUE;
  private double epsilon;
  private boolean hasFinished = false;
  private long position;
  private float[][] clusterCenters;
  private int[] inputData;
  private float[] aPixel;
  private short[][] outputData;
```

```
String datalist="";
public float[][] cout;
int[] data:

public FCMC(int numClusters,int maxIterations,float fuzziness,double epsilon)
   {

    // Get the  dimensions.
    width = 5; // no of attributes
    height =10; // no of patients
    numBands = 1; //

    // Get some clustering parameters.
    this.numClusters = numClusters;
    this.maxIterations = maxIterations;
    this.fuzziness = fuzziness;
    this.epsilon = epsilon;
    iteration = 5;

    // We need arrays to store the clusters' centers, validity tags and membership values.
    clusterCenters = new float[numClusters][numBands];
    membership = new float[width][height][numClusters];

    inputData = new int[width*height*numBands];
    aPixel = new float[numBands];

    //get the input from the document


 File tfile=new File("attributes.txt");
 loaddata(tfile);
```

```
String[] data1=datalist.split(",");
data=new int[data1.length];

for(int i=0;i<data1.length;i++)
data[i]=Integer.parseInt(data1[i]);

displayattributes();

   for(int i=0;i<data.length;i++)
   inputData [i]=data[i];

// Gets memory for the output data (cluster indexes).
   outputData = new short[width][height];
cout=new float[width][height];
   // Initialize the membership functions randomly.
   Random generator = new Random(); // easier to debug if a seed is used
   // For each data point (in the membership function table)
   for(int h=0;h<height;h++)
     for(int w=0;w<width;w++)
       {
       // For each cluster's membership assign a random value.
       float sum = 0f;
       for(int c=0;c<numClusters;c++)
         {
         membership[w][h][c] = 0.01f+generator.nextFloat();
         sum += membership[w][h][c];
         }
       // Normalize so the sum of MFs for a particular data point will be equal to 1.
       for(int c=0;c<numClusters;c++) membership[w][h][c] /= sum;
       }
```

```
// Initialize the global position value.
position = 0;

System.out.println("\n");
 System.out.print("Normalised output\n");
System.out.print("*****************************************\n");

 for(int h=0;h<height;h++)
 {
  for(int w=0;w<width;w++)
   {
    for(int k=0;k<numClusters;k++)
   {
   //System.out.print(cout[w][h]);
    System.out.print((double) (int)((membership[w][h][k]+0.005)*100.0)/100.0);
   System.out.print("\t");
   }

  }
System.out.print("\n");
  }

  }

 public void run()
  {
  double lastJ;
  lastJ = calculateObjectiveFunction();
  for(iteration=0;iteration<maxIterations;iteration++)
   {
```

```
  calculateClusterCentersFromMFs();
  calculateMFsFromClusterCenters();
  j = calculateObjectiveFunction();
  if (Math.abs(lastJ-j) < epsilon) break;
  lastJ = j;
  }
 hasFinished = true;
 position = getSize();
 }

private void calculateClusterCentersFromMFs()
 {
 float top,bottom;
 // For each band and cluster
 for(int b=0;b<numBands;b++)
  for(int c=0;c<numClusters;c++)
   {
   top = bottom = 0;
   for(int h=0;h<height;h++)
    for(int w=0;w<width;w++)
     {
     int index = (h*width+w)*numBands;
     top += Math.pow(membership[w][h][c],fuzziness)*inputData[index+b];
     bottom += Math.pow(membership[w][h][c],fuzziness);
     }
   clusterCenters[c][b] = top/bottom;
System.out.println("\n centroid of cluster "+(c+1)+" = "+clusterCenters[c][b]);
   position += width*height;
   }
  }
```

```
private void calculateMFsFromClusterCenters()
 {
 float sumTerms;
 for(int c=0;c<numClusters;c++)
  for(int h=0;h<height;h++)
   for(int w=0;w<width;w++)
    {
    int index = (h*width+w)*numBands;
    for(int b=0;b<numBands;b++)
     aPixel[b] = inputData[index+b];
    float top = calcDistance(aPixel,clusterCenters[c]);
    sumTerms = 0f;
    for(int ck=0;ck<numClusters;ck++)
     {
     float thisDistance = calcDistance(aPixel,clusterCenters[ck]);
     sumTerms += Math.pow(top/thisDistance,(2f/(fuzziness-1f)));
     }
    membership[w][h][c] =
    (float)(1f/sumTerms);
    position += (numBands+numClusters);
    }
 }


private double calculateObjectiveFunction()
 {
 double j = 0;
 for(int h=0;h<height;h++)
  for(int w=0;w<width;w++)
```

```
  for(int c=0;c<numClusters;c++)
    {
    int index = (h*width+w)*numBands:
    for(int b=0;b<numBands;b++)
     aPixel[b] = inputData[index+b];
    float distancePixelToCluster = calcDistance(aPixel,clusterCenters[c]);
    j += distancePixelToCluster*Math.pow(membership[w][h][c],fuzziness);
    position += (2*numBands);
    }
 return j;
 }


private float calcDistance(float[] a1,float[] a2)
 {
 float distance = 0f;
 for(int e=0;e<a1.length;e++) distance += (a1[e]-a2[e])*(a1[e]-a2[e]);
 return (float)Math.sqrt(distance);
 }


public long getSize()
 {
 return (long)maxIterations* // The maximum number of iterations times
  (
    (numClusters*width*height*(2*numBands))+ // Step 0 of method run()
    (width*height*numBands*numClusters)+ // Step 1 of method run()
    (numClusters*width*height*(numBands+numClusters))+ // Step 2 of run()
    (numClusters*width*height*(2*numBands))  // Step 3 of method run()
  );
 }
```

```
public long getPosition()
  {
  return position;
  }


 public boolean isFinished()
  {
  return (position == getSize());
  }


public void getRankedImage(int rank)
  {
  int[] pixelArray = new int[numBands];

  for(int h=0;h<height;h++)
  for(int w=0;w<width;w++)
     {
     int aCluster = getRankedIndex(membership[w][h],rank);
     for(int band=0;band<numBands;band++) pixelArray[band] =
(int)clusterCenters[aCluster][band];
     cout[w][h]=pixelArray[0];
     }
  }


 public void getRankedMFImage(int rank)
```

```
{
for(int h=0;h<height;h++)
 for(int w=0:w<width:w++)
  {
  float aCluster = (getRankedMF(membership[w][h],rank));
   cout[w][h]=aCluster;
  }

 // written by me.....

  System.out.print("After C-Means output\n");
System.out.print("*****************************************\n");
System.out.print("\n");

  for(int h=0;h<height;h++)
  {
   for(int w=0;w<width;w++)
    {
     for(int k=0;k<numClusters;k++)
    {
     //System.out.print(cout[w][h]);
      System.out.print((double) (int)((membership[w][h][k]+0.005)*100.0)/100.0);
     System.out.print("\t");
     }
   }
System.out.print("\n");
  }


   int cluster1[]=new int[100];
   int cluster2[]=new int[100];
```

```
int x=0;
int y=0;
 for(int h=0;h<height;h++)
{
 for(int w=0;w<width;w++)
  {
 if(membership[w][h][0] < epsilon)
 {
       cluster1[x]=h+1;
       x++;
 }
 else
 {
       cluster2[y]=h+1;
       y++;
 }
 }
}

  System.out.println("Cluster 1");
  for(int i=0;i<x;i++)
     System.out.print(cluster1[i]+",");

   System.out.print("\n");
  System.out.println("Cluster 2");
  for(int i=0;i<y;i++)
     System.out.print(cluster2[i]+",");

  System.out.println("\n\n*********");
```

```
}

private int getRankedIndex(float[] data.int rank)
 {
 // Create temporary arrays for the indexes and the data.
 int[] indexes = new int[data.length];
 float[] tempData = new float[data.length];
 // Fill those arrays.
 for(int i=0;i<indexes.length;i++)
  {
  indexes[i] = i;
  tempData[i] = data[i];
  }
 // Sort both arrays together, using data as the sorting key.
 for(int i=0;i<indexes.length-1;i++)
  for(int j=i;j<indexes.length;j++)
   {
   if (tempData[i] < tempData[j])
    {
    int tempI= indexes[i];
    indexes[i] = indexes[j];
    indexes[j] = tempI;
    float tempD = tempData[i];
    tempData[i] = tempData[j];
    tempData[j] = tempD;
    }
   }
  // Return the cluster index for the rank we want.
  return indexes[rank];
  }
```

```
private float getRankedMF(float[] data,int rank)
 {
 // Create temporary arrays for the indexes and the data.
 int[] indexes = new int[data.length];
 float[] tempData = new float[data.length];
 // Fill those arrays.
 for(int i=0;i<indexes.length;i++)
  {
  indexes[i] = i;
  tempData[i] = data[i];
  }
 // Sort both arrays together, using data as the sorting key.
 for(int i=0;i<indexes.length-1;i++)
  for(int j=i;j<indexes.length;j++)
   {
   if (tempData[i] < tempData[j])
    {
    int tempI= indexes[i];
    indexes[i] = indexes[j];
    indexes[j] = tempI;
    float tempD = tempData[i];
    tempData[i] = tempData[j];
    tempData[j] = tempD;
    }
   }
  // Return the cluster index for the rank we want.
  return tempData[rank];
  }

 public double getPartitionCoefficient()
  {
```

```
double pc = 0;
// For all data values and clusters
for(int h=0;h<height;h++)
 for(int w=0;w<width;w++)
  for(int c=0;c<numClusters;c++)
   pc += membership[w][h][c]*membership[w][h][c];
pc = pc/(height*width);
return pc;
}

public double getPartitionEntropy()
 {
 double pe = 0;
 // For all data values and clusters
 for(int h=0;h<height;h++)
  for(int w=0;w<width;w++)
   for(int c=0;c<numClusters;c++)
    pe += membership[w][h][c]*Math.log(membership[w][h][c]);
 pe = -pe/(height*width);
 return pe;
 }

 public double getCompactnessAndSeparation()
  {
  double cs = 0;
  // For all data values and clusters
  for(int h=0;h<height;h++)
   for(int w=0;w<width;w++)
    {
    // Get the current pixel data.
    int index = (h*width+w)*numBands;
```

```
    for(int b=0;b<numBands;b++)
     aPixel[b] = inputData[index+b];
    for(int c=0;c<numClusters;c++)
     {
     // Calculate the distance between a pixel and a cluster center.
     float distancePixelToCluster = calcSquaredDistance(aPixel,clusterCenters[c]);
     cs += membership[w][h][c]*membership[w][h][c]*
         distancePixelToCluster*distancePixelToCluster;
     }
    }
cs /= (height*width);
// Calculate minimum distance between ALL clusters
float minDist = Float.MAX_VALUE;
for(int c1=0;c1<numClusters-1;c1++)
 for(int c2=c1+1;c2<numClusters;c2++)
  {
  float distance = calcSquaredDistance(clusterCenters[c1],clusterCenters[c2]);
  minDist = Math.min(minDist,distance);
  }
cs = cs/(minDist*minDist);
return cs;
}

private float calcSquaredDistance(float[] a1,float[] a2)
 {
 float distance = 0f;
 for(int e=0;e<a1.length;e++) distance += (a1[e]-a2[e])*(a1[e]-a2[e]);
 return (float)distance;
 }

private void loaddata(File tfile)
```

```
int ff=0;
try
{
if(tfile.exists()==true)
{
FileInputStream fin=new FileInputStream(tfile.getPath());
while(true)
{
ff++;
String tstr=IO_Utils.readLine(fin);
if(tstr.length()==0) break;
if(ff>1)
datalist=datalist+","+tstr;
else
datalist=tstr;
}
fin.close();
}
catch(Exception err)
{
System.out.println("Error: "+err);
System.exit(-1);
}
}

private void displayattributes()
{
for(int i=0;i<data.length;i++)
{
if(i%width==0)
System.out.println("");
```

REFERENCES

[1] Latha Parthiban and R.Subramanian, "Intelligent Heart Disease Prediction System using CANFIS and Genetic Algorithm", International Journal of Biological, Biomedical and MedicalSciences, Vol. 3, No. 3, 2008

[2] Obenshain, M.K: "Application of Data Mining Techniques to Healthcare Data", Infection Control and Hospital Epidemiology, 25(8), 690–695, 2004. Sellappan, P., Chua, S.L.: "Model-based Healthcare

[3] Boleslaw Szymanski, Long Han, Mark Embrechts, Alexander Ross, Karsten Sternickel, Lijuan Zhu, "Using Efficient Supanova Kernel For Heart Disease Diagnosis", proc. NNIE 06, intelligent engineering systems through artificial neural networks, vol. 16, pp:305-310, 2006.

[4] Rajesh Ghongade and Dr. Ashok Ghatol, "A multi-class heartbeat classifier employing hybrid fuzzy -neural network" , International Conference on Intelligent and Advanced Systems 2007

[5] Tzung-I Tang, Gang Zheng, Yalou Huang, Guangfu Shu, Pengtao Wang, "A Comparative Study of Medical Data Classification Methods Based on Decision Tree and System Reconstruction Analysis", IEMS, Vol. 4, No. 1, pp. 102-108, June 2005.

[6] Franck Le Duff, Cristian Munteanb, Marc Cuggiaa, Philippe Mabob, "Predicting Survival Causes After Out of Hospital Cardiac Arrest using Data Mining Method", Studies in health technology and informatics, Vol. 107, No. Pt 2, pp. 1256-9, 2004.

[7] Sellappan Palaniappan, Rafiah Awang, "Intelligent Heart Disease Prediction System Using Data Mining Techniques", IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.8, August 2008

[8] Yanwei Xing, Jie Wang , Zhihong Zhao, Yonghong Gao, "Combination data mining methods with new medical data to predicting outcome of Coronary Heart Disease ", 2007 International Conference on Convergence Information Technology

[9] http://www.statsoft.com/textbook/stdatmin.html

[10]http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/cleveland.data

[11] A.M.J. Md. Zubair Rahman and P. Balasubramanie, "An Efficient Algorithm for Mining Maximal Frequent Item Sets ", Journal of Computer Science 4 (8): 638-645, 2008, ISSN 1549-3636 , © 2008 Science Publications

[12] S.Shankar and T.Purusothaman, "Utility Sentient Frequent Itemset Mining and Association Rule Mining: A Literature Survey and Comparative Study ", International Journal of Soft Computing Applications ISSN: 1453-2277 Issue 4 (2009), pp.81-95 © EuroJournals Publishing, Inc. 2009

[13] http://www.mathworks.com

[14]Arun and K.Pujari, "Data mining Techniques", University Press, First Edition, 2001.

[15] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/_report.fuzzysets.html

[16] http://www.centerforpbbefr.rutgers.edu/Jan11-2008%20papers/4-2.doc