

# **1. INTRODUCTION**

Disease diagnosis is a very complex and tedious task as it requires lots of experience and knowledge. One of the traditional ways for diagnosis is doctor's examination or a number of blood tests. The main task is to provide disease diagnosis at early stages with higher accuracy. But all the time, patients cannot afford the cost and time. Today computer science has changed our world and computers become a big part of our life. By the help of computer it is easy for us to analyze and diagnose the medical problems and diseases. Various attempts are made by researchers to make such system which can help medical problems. These systems are now widely used in clinics and hospitals. They are considered to be very useful for patient as well as for medical experts in making the decisions. The information required for diagnosis is typically gathered from history of the physical examination of the person and person's symptoms and signs. Diagnosis is often challenging, because many signs and symptoms are non-specific.

A medical diagnosis can be regarded as an attempt in classifying an individual's condition into separate and distinct categories that allow medical decisions about treatment and forecast of the future course of a disease to be made. Subsequently, a diagnostic opinion is often described in terms of a disease or other condition, but in the case of a wrong diagnosis, the individual's actual disease or condition is not the same as the individual's diagnosis.

## **1.1 Soft Computing**

Soft computing is a novel paradigm to deal with the existing real world problems which are hard to be solved logically. Such problems may possess good theoretical solutions but because of high resource requirement and high computational time complexity, their practical solutions are difficult to obtain. For such kind of complex problems soft computing acts as a helping hand. Soft computing refers to the science of reasoning thinking and deduction that recognises and uses the real world phenomenon of groupings, memberships, and classification of various quantities under study.

Soft computing is made up of some predominate techniques like fuzzy logic, neural networks, Bayesian statistics and evolutionary computing. Based on the nature of the problem this techniques are used .They can be used independently or can be combined with other techniques. There is no need of strict mathematical definitions and distinction of system components in case of soft computing which makes it better suitable to complex problems. Non linear problems which do not have mathematical models can be solved using soft computing techniques which is the main cause of adopting soft computing techniques in solving major problems. Intelligent systems like autonomous vehicles are easy by soft computing because it introduced a new concept of human knowledge such as recognition, understanding, learning, and cognition into the fields of computing. Both natural and artificial ideas can be used for crafting the soft computing techniques.

## **1.2 Data Mining**

Data Mining implies mining or extracting knowledge from large amounts. Data mining is the process of analysing data from different perspectives and subsequent enumeration of the major points into useful information. Data mining is the process of exploration and analysis by automatic or semi-automatic means of large quantities of data to discover meaningful patterns and rules. Data mining can be regarded as a result of the natural evolution of information technology. It is also termed as data dredging, information harvesting, business intelligence, etc.

The major part of data mining is related with the analysis of data and the use of software techniques for finding patterns and regularities in sets of data. Data Mining is an integral part of knowledge discovery in databases which is the overall process of converting raw data into useful information. The tasks performed by the data mining are: Classification, Clustering, Regression, Summarization, Anomaly detection, Association rules learning. Data mining and machine learning depend on classification which is the most important and essential task. Many experiments are performed on medical datasets using multiple classifiers and feature selection techniques.

### **1.2.1 How does data mining work**

Data mining refers to the extracting or “mining” knowledge from large amounts of data .Data mining is simply viewed as an important step in the knowledge discovery.

Knowledge discovery consists of an iterative sequence of the following steps.

1. Data Pre-processing.
2. Data mining (an essential process where intelligent methods are applied in order to extract data patterns).
3. Pattern Evaluation (to identify the truly interesting patterns representing knowledge based on some interesting measures).
4. Knowledge Representation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user).

### **1.3 Data Pre-Processing:**

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real world data is often incomplete, inconsistent, and lacking in certain behaviours or trends, and is likely to contain many errors. Data Pre-processing is a proven method of resolving such issues. Data pre-processing prepares raw data for further processing.

Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income – 100), impossible data combinations (e.g., Sex– Female, Pregnant- Yes), missing values, etc.

Analysing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data are first and foremost before running analysis.

If there is much irrelevant and redundant information present or noisy and unreliable data, then training phase is more difficult.

Data pre-processing includes cleaning, normalization, and transformation of data.

### **1.3.1 Why pre-processing**

Real world data are generally

- Incomplete – lacking attribute values, lacking certain attributes of interest, or containing only aggregate data.
- Noisy – containing errors and outliers.
- Inconsistent - containing discrepancies in codes or names.

### **1.3.2 Data pre-processing methods**

Raw data is highly susceptible to noise, multiple values, and inconsistency. The quality of data affects the data mining results. In order to help improvement the quality of data and consequently, of the mining results raw data is pre-processed so as to improve the efficiency and the ease of mining process. It is one of the critical steps in data mining which deals with the preparation of initial dataset. Data pre-processing methods are divided into following categories.

- Data Cleaning
- Data Transformation
- Data Discretization

#### **1.3.2.1 Data Cleaning**

Data cleaning is a process used to determine inaccurate, incomplete or unreasonable data and then improve the quality through correcting of detected errors and omissions. Generally data cleaning reduces errors and improves the data quality.

#### ***Handling missing values***

- Predict the missing value by using a learning algorithm: Consider the attribute with the missing value as class variable and run a learning algorithm to predict the missing value.
- Ignore the missing value during analysis( Generally done when class label is missing )

- Use the attribute mean to fill missing values belonging to the same class.
- Use the attribute mean to fill the missing value.

#### **1.3.2.2 Data Discretization**

Discretization refers to the process of converting or partitioning continuous attributes, features or variables to discretized or nominal attributes or features or variables or intervals. It is a part of data reduction, replacing numerical attributes with nominal ones. The labels in turn, can be recursively organised into higher level concepts, resulting in a concept hierarchy for the numeric attribute. More than one concept hierarchy can be defined on the same attribute to accommodate the needs of various users. There are methods for discretizing the data, which includes:

*Clusters:* It divides the data into groups by sampling the training data and a discrete number to each group.

*Equal Areas:* The algorithm divides the data into groups that contain an equal number of values.

### **1.4 Feature Selection**

#### **1.4.1 Why Feature Selection**

Feature selection deletes spurious features from the original dataset without degrading generalization performance. In real world problems, Feature selection is essentially due to the existence of following factors:

- Abundance of noise
- Spurious information
- Irrelevant and redundant features in the original feature set.

#### **1.4.2 What is Feature Selection**

Feature selection is generally used in machine learning, especially when the learning task envelopes high-dimensional datasets. The intention of feature selection is to reduce the complexity and augment the quality of a dataset by selecting prominent features. Optimization methods are used in the process of feature selection to compute the most significant set features

from the data set while maintaining adequate accuracy rate represented by the original set of features.

Accordingly, Feature selection has become an area of active research spreading throughout many fields including pattern recognition, data mining, image mining and text categorization. The feature selection methods are typically presented in three classes based on how they combine the selection algorithm and the model building:

- **Filter approach:** Filter methods tend to select redundant variables because they do not consider the relationships between variables.
- **Wrapper approach:** Wrapper methods evaluate subsets of variables which allows, unlike filter approaches, to detect the possible interactions between variables.
- **Hybrid approach:** These try to combine the advantages of both previous methods.

One of the sub classes of wrapper approach is “random search” which includes meta-heuristic methods such as

- Genetic algorithm
- Simulated annealing
- Ant colony optimization
- Social cognitive optimization
- Variable neighbourhood search.

The Feature subset selection problem directs to the task of identifying and opting useful subset of features to be used to represent patterns from a larger set of often mutually redundant, possibly irrelevant features with different associated measurement costs and or risks.

In our Feature Selection, it comprises of

- **Fuzzification** followed by
- **Genetic Algorithm.**

Fuzzification is needed to deal with Uncertainty.

Genetic Algorithm is used for selecting Optimum Feature Subsets.

### **1.4.3 Advantages of Feature Selection**

- It reduces the dimensionality of the feature space, to limit storage requirements and increase algorithm speed.
- It eliminates the redundant, irrelevant and noisy data.
- It enhances the data quality and increasing the accuracy of the result.

### **1.4.4 FUZZIFICATION**

#### **1.4.4.1 Why Fuzzification**

Human decisions may not always be either yes or no but also includes all intermediate possibilities between the digital values yes or no. So, real world data may not be binary valued data (0 or 1) all the time. Fuzzy logic helps the computer to deal with this type of data and resulting in an output to the human equivalent intermediate possibilities. Fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. It handles:

- Uncertainty and Imprecise nature.
- To describe grades of Truth.
- It is useful in Complex Non-linear applications.

#### **1.4.4.2 Fuzzification**

Fuzzy means of or relating to a form of set theory and logic in which predicates may have degrees of applicability rather than simple being true or false. Fuzzy logic is logic of fuzzy sets. A fuzzy set has potentially an infinite

range of truth values between zero and one. A fuzzy set A in a universe of discourse X is defined as the following set pairs.

$$\mu_A(x) \in [0, 1] \quad (A = \mu_A(x)|x \in X) \quad \text{-----} \quad (1)$$

Where  $\mu_A$ , is a mapping called the membership function of fuzzy set A and is called the degree of belongingness or membership value or degree of membership of in the fuzzy set A.

Fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. The membership function is used to associate a grade to each linguistic term. Generally, the fuzzification process involves three basic concepts: Association of linguistic variables, and membership functions.

### ***Linguistic Variables***

Whereas an algebraic variable takes numbers as values, a linguistic variable takes words or sentences as values [84]. The name of such a linguistic variable is its label. The set of values that it can take is called its term set. Each value in the term set is a linguistic value or term defined over the universe. In summary: A linguistic variable takes a linguistic value, which is a fuzzy set defined on the universe. As for example Let x be a linguistic variable labeled 'Age'. Its term set T could be defined as

T (age) = {young, very young, not very young, old, more or less old}. Each term is defined on the universe, for example the integers from 0 to 100 years.

Assume a discrete universe U = (0, 20, 40, 60, 80) of ages. We can assign u = [0 20 40 60 80] and

$$\text{'young'} = [1 \ .6 \ .1 \ 0 \ 0]$$

The discrete membership function for the set 'very young' is,

$$\text{'very young'} = [1 \ 0.36 \ 0.01 \ 0 \ 0]$$

The set 'very very young' is, by repeated application,

$$\text{very very young} = [1 \ 0.13 \ 0 \ 0 \ 0]$$

The derived sets inherit the universe of the primary set.



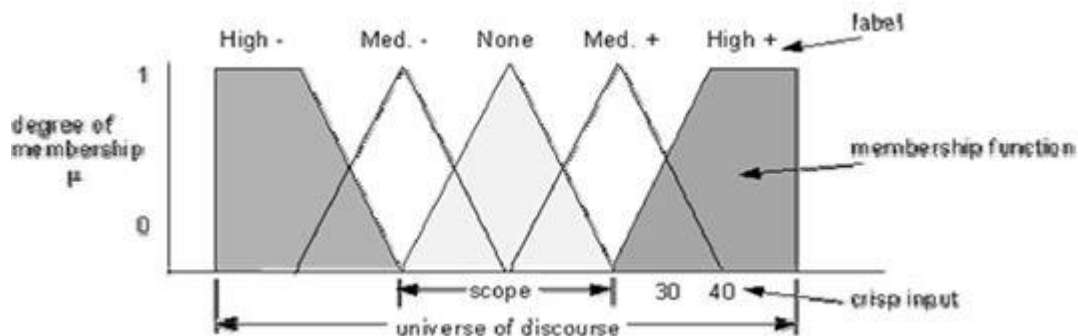
### **Membership Function:**

Membership functions allow you to quantify linguistic term and represent a fuzzy set graphically. A membership function for a fuzzy set  $A$  on the universe of discourse  $X$  is defined as. Here, each element of  $X$  is mapped to a value between 0 and 1. It is called membership value or degree of membership. It quantifies the degree of membership of the element in  $X$  to the fuzzy set  $A$ .

- X axis represents the universe of discourse.
- Y axis represents the degrees of membership in the  $[0, 1]$  interval.

There can be multiple membership functions applicable to fuzzify a numerical value. Simple membership functions are used as use of complex functions does not add more precision in the output.

All membership functions for **High-**, **Medium -**, **None**, **Medium +**, and **High+** are shown as below



**Fig 1.1 Membership Function Structure**

Each linguistic pattern has membership degree in the range  $[0, 1]$ . The type of the membership function is used depending on the base set patterns. If the base set contains many values, or if this set is continuous, then a parametric representation, which can be adapted by changing the parameters, is appropriate. Mostly this type of membership functions are triangular or trapezoidal functions that are defined by three and four parameters respectively.

There are different forms of membership functions such as :

- Triangular membership function

- Trapezoidal membership function
- Piecewise linear membership function
- Gaussian membership function
- Singleton membership function

We are using Triangular Membership function which indicates 3 possible ranges. Where three ranges which includes upper limit, mode and lower limit of an attribute.

### **1.4.5 Genetic Algorithm**

Genetic Algorithms (GAs) was invented by John Holland and developed this idea in his book “Adaptation in natural and artificial systems” in the year 1975. Holland proposed GA as a heuristic method based on “Survival of the fittest”. GA was discovered as a useful tool for search and optimization problems.

A genetic algorithm is a problem solving method that uses genetics as its model of problem solving. It's a search technique to find approximate solutions to optimization and search problems. GA handles a population of possible solutions. Each solution is represented through a chromosome. Coding all the possible solutions into a chromosome is the part. Reproduction operators are applied directly on the chromosomes, and are used to perform mutations and recombination over solutions of the problem.

#### **1.4.5.1 Biological Background**

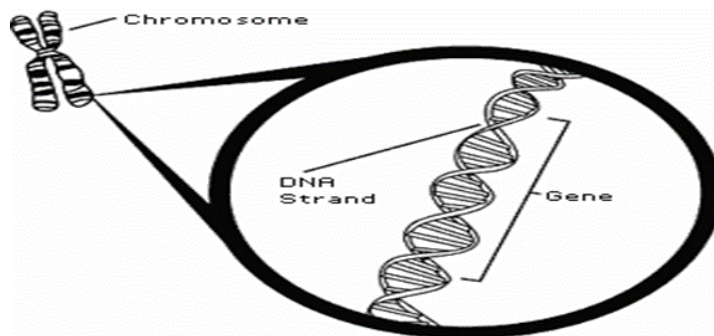
The word “genetics” is derived from the Greek word “genesis” meaning “to grow” or “to become”. The science of genetics helps us to differentiate between heredity and variations and seeks to account for the resemblances and differences due to the concepts of Genetic Algorithms and directly derived from natural heredity, their source and development. The concepts of Genetic Algorithms are directly derived from natural evolution.

## **The Cell**

Every animal/human cell is a complex of many “small” factories that work together. The center of all this is the cell nucleus. The genetic information is contained in the cell nucleus.

### **Chromosomes**

All the genetic information gets stored in the chromosomes. Each chromosome is build of Dioxy Ribo Nucleic Acid (DNA). In humans, a chromosome exists in the form of pairs (23 pairs found). The chromosomes are divided into several parts called genes. Genes code the properties of species i.e., the characteristics of an individual. The possibilities of the genes for one property are called allele and a gene can take different alleles. This gene pool can determine all the different possible variations for the future generations. The size of the gene pool helps in determining the diversity of the individuals in the population. The set of all the genes of a specific species is called genome. Each and every gene has an unique position on the genome called locus.



**Fig 1.2 Model of Chromosome**

### **Genetics**

For a particular individual, the entire combination of genes is called genotype. The phenotype describes the physical aspect of decoding a genotype to produce the phenotype. One interesting point of evolution is that selection is always done on the phenotype whereas the reproduction recombines genotype. Thus morphogenesis plays a key role between selection and reproduction. In higher life forms, chromosomes contain two sets of genes. This is known as diploids. In case of conflicts between two values of the same

pair of genes, the dominant one will determine the phenotype whereas the other one, called recessive, will still be present and can be passed on to the offspring.

## **Reproduction**

Reproduction of species via genetic information is carried out by,

- Mitosis
- Meiosis

In Mitosis the same genetic information is copied to new offspring. There is no exchange of information. This is a normal way of growing of multi cell structures, like organs. When meiotic division takes place 2 gametes appears in the process. When reproduction occurs, these two gametes conjugate to a zygote which becomes the new individual.

## **Natural Selection**

The origin of species is based on “Preservation of favourable variations and rejection of unfavourable variations”. The variation refers to the differences shown by the individual of a species and also by offspring’s of the same parents. There are more individuals born than can survive, so there is a continuous struggle for life. Individuals with an advantage have a greater chance for survive i.e., the survival of the fittest. For example, Giraffe with long necks can have food from tall trees as well from grounds, on the other hand goat, deer with small neck have food only from grounds. As a result, natural selection plays a major role in this survival process.

Natural evolution	Genetic algorithm
Chromosome	String
Gene	Feature or character
Allele	Feature value
Locus	String position
Genotype	Structure or coded string
Phenotype	Parameter set, a decoded structure

**Table.1.1: Comparison of Natural evolution and GA**

#### 1.4.5.2 Basic Operators of Genetic Algorithm

- **Reproduction :**

The first step consists in selecting individuals for reproduction. This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction than poor ones. It is based on Darwin's evolution theory of "Survival of the fittest". Therefore, this operator is also known as 'Selection Operator'.

- **Crossover :**

During crossover we create new individuals by combining aspects of our selected individuals. By combining certain traits from two or more individuals we will create an even 'fitter' offspring which will inherit the best traits from each of its parents.

- **Mutation :**

We need to add a little bit randomness into our populations' genetics otherwise every combination of solutions we can create would be in our initial population. Mutation typically works by making very small changes at random to an individual genome.

#### 1.4.5.3 BASIC GENETIC ALGORITHM

**STEP 1:** Genetic random population of  $n$  chromosomes (suitable solutions) for the problem

**STEP 2:** Evaluate the fitness  $f(x)$  of each chromosome  $x$  in the population

**STEP 3:** Create a new population by repeating following steps until the STEP3 is complete

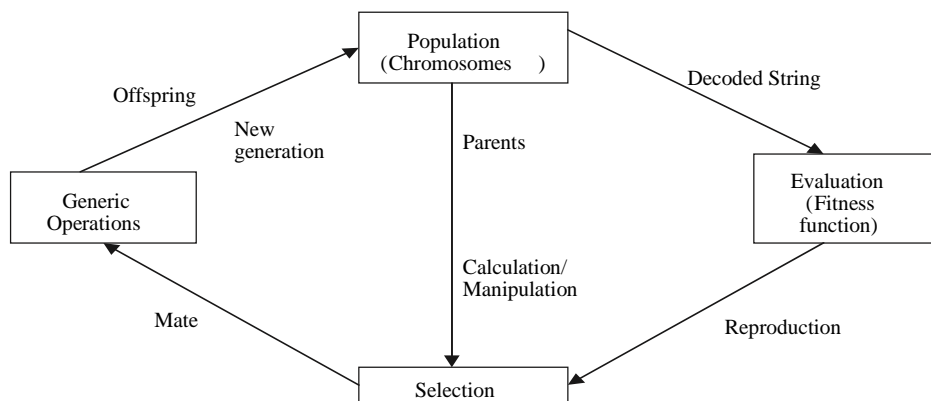
- **Selection:** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
- **Crossover:** With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

- **Mutation:** With a mutation probability, mutate new offspring at each locus (position in chromosome)
- **Accepting:** Place new offspring in the new population.

**STEP 4 :** Use new generated population for a further action of the algorithm.

**STEP 5:** Go to *Step 2*, and repeat the process until the termination criterion is satisfied.

The Genetic algorithm process is discussed through the GA cycle



**Fig 1.3 Genetic algorithm cycle**

#### 1.4.5.6 Comparison of Genetic Algorithm with Other Optimization Techniques

1. GAs operate with coded versions of the problem parameters rather than parameters themselves i.e., GA works with the coding of solution set and not with the solution itself.

2. Almost all conventional optimization techniques search from a single point but GAs always operate on a whole population of points(strings) i.e., GA uses population of solutions rather than a single solution searching. This plays a major role to the robustness of genetic algorithms. It improves the chance of reaching the global optimum and also helps in avoiding local stationary point.

3. GA uses fitness function for evaluation rather than derivatives. As a result, they can be applied to any kind of continuous or discrete optimization

problem. The key point to be performed here is to identify and specify a meaningful decoding function.

4. GAs use probabilistic transition operates while conventional methods for continuous optimization apply deterministic transition operates i.e., GAs does not use deterministic rules.

#### **1.4.5.7 Advantages of Genetic Algorithm**

The advantages of genetic algorithm includes,

1. Solution space is wider
2. Easy to discover global optimum
3. Easily modified for different problems.
4. Handles noisy functions well.
5. Handles large, poorly understood search spaces easily
6. Very robust to difficulties in the evaluation of the objective function.
7. They are resistant to becoming trapped in local optima.

#### **1.4.5.8 Applications of Genetic Algorithm**

Genetic algorithms have been used for difficult problems (such as NP-hard problems), for machine learning and also for evolving simple programs. They have been also used for some art, for evolving pictures and music. A few applications of GA are as follows:

- Nonlinear dynamical systems–predicting, data analysis
- Robot trajectory planning
- Evolving LISP programs (genetic programming)
- Finding shape of protein molecules

- Control–gas pipeline, pole balancing, missile evasion, pursuit
- Design–semiconductor layout, aircraft design, keyboard configuration, communication networks
- Machine Learning–Designing neural networks, both architecture and weights, improving classification algorithms, classifier systems
- Combinatorial Optimization–set covering, travelling salesman (TSP), Sequence scheduling, routing, bin packing, graph colouring and partitioning.

## 1.5 Classification

Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and it can be used for classifying newly available data. As in this project of medical diagnosis we need to implement classification phase. Classification is one of the Data Mining techniques that is mainly used to analyze a given data set and takes each instance of it and assigns this instance to a particular class such that classification error will be least. There are different classification techniques

- Id3
- C4.5
- K-nearest neighbours.
- Naïve Bayes Algorithm.
- Support vector machine
- Artificial neural network.
- Decision tree

There are several advantages and disadvantages in every classification technique.

- **Artificial neural networks:** An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological



nervous systems, such as the brain, process information. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

- **Decision trees:** A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.
- **Nearest neighbour method:** A technique that classifies each record in a dataset based on a combination of the classes of the  $k$  record(s) most similar to it in a historical dataset (where  $k \geq 1$ ). Sometimes called the  $k$ -nearest neighbour technique.

E.g.: (I) The advantage of using decision tree as a classification technique is it is easy to implement and takes less time. But when complex problems occurred there is no use of using decision as it produce local optimum results. The main problem with decision it will inconsistent solutions to the non-polynomial problems, there the machine learning techniques plays a crucial role.

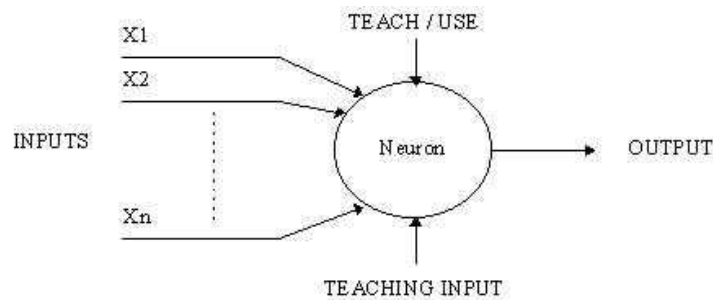
Here in this project, we are choosing artificial neural networks as a classification technique. The main disadvantage with artificial neural networks is it takes more time. For that for faster accessing we are using feature selection.

### 1.5.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

**Simple neuron:**

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.



**Fig. 1.4 Simple neuron**

There are several firing rules, this rules determine how one calculates whether a neuron should fire for any input pattern.

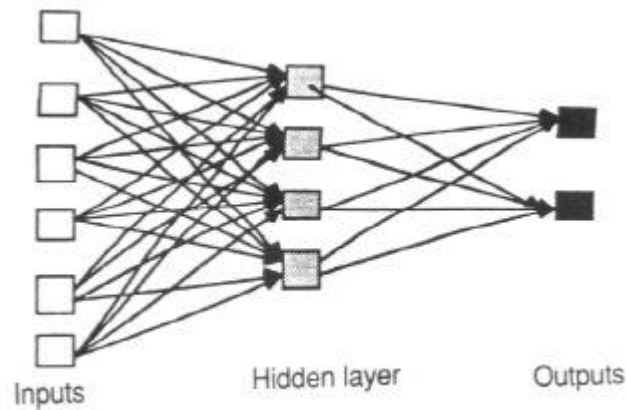
E.g.: Hamming distance technique.

There are two types of neural networks

1. Feed-forward networks.
2. Feedback networks.

**Feed-forward networks:**

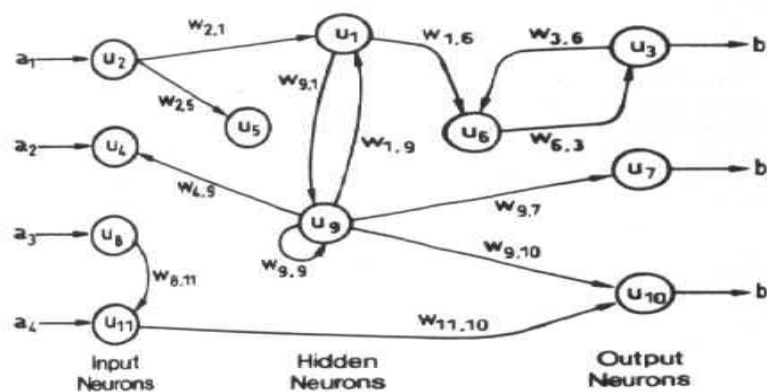
Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs.



**Fig.1.5 Feed-forward Networks**

### ***Feedback networks:***

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic (their 'state' is changing continuously) until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.



**Fig. 1.6 Feedback Networks**

### **1.5.1.1 Architecture OF NEURAL NETWORK**

There are three layers present in neural network.

1. Input layer.
2. Hidden layer.

### 3. Output layer.

#### **Input layer:**

The activity of the input units represents the raw information that is fed into the network.

#### **Hidden layer:**

The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

#### **Output layer:**

The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

#### **Activation function:**

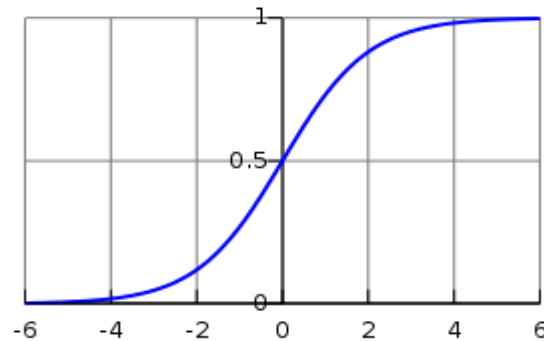
In computational networks, the **activation function** of a node defines the output of that node given an input or set of inputs. It is also called transfer function. There are different activation functions.

- Eg:
  - Binary step.
  - Logistic function.
  - tanH function.
  - ArcTan function.
  - Softsign function.
  - SoftPlus function.
  - Bent function.
  - Sinusiod function.
  - Gaussian function.
  - Step function or threshold function.

In this project, we are using sigmoid function because for this medical diagnosis ,this is binary classification. So in sigmoid the result is in the range of 0 to 1.

**Sigmoid function:**

$$F(x) = 1/(1+e^{-x}) \quad \text{-----} \quad (2)$$



**Fig 1.7 Sigmoid Function**

Training the system and error calculation:

There are total of four methods with which we can train the neural network.

1. Nelder-Mead method.
2. Simulated annealing method.
3. Genetic algorithm.
4. Back propagation.

Among all back propagation is the mostly used one and in this project we are implementing it.

**1.5.1.2 BACKPROPAGATION ALGORITHM:**

The feed forward back-propagation network undergoes supervised training, with a finite number of pattern pairs consisting of an input pattern and a desired or target output pattern. An input pattern is presented at the input layer. The neurons here pass the pattern activations to the next layer neurons, which are in a hidden layer. The outputs of the hidden layer neurons are obtained by using a bias, and also a threshold function with the activations determined by the weights and the inputs. These hidden layer outputs become inputs to the output neurons, which process the inputs using an optional bias and a threshold function. The final output of the network is determined by the activations from the output layer. The computed pattern

and the input pattern are compared, a function of this error for each component of the pattern is determined, and adjustment to weights of connections between the hidden layer and the output layer is computed.

### **Problems solved by back propagation:**

The only problem with back propagation is we may get local minima as an output. This can be solved by gradient descent.

### **ALGORITHM:**

The back-propagation algorithm consists of four steps:

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$EA_j = \frac{\partial E}{\partial y_j} = y_j - d_j \quad \text{----- (3)}$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = EA_j y_j (1 - y_j) \quad \text{----- (4)}$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$EW_{ij} = \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_j} \times \frac{x_j}{w_{ij}} = EI_j y_i \quad \text{----- (5)}$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to

multilayer networks. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j W_{ji} \quad \text{----- (6)}$$

## 1.6 Dataset

### 1.6.1 Diabetes

Diabetes is a group of diseases characterized by high blood sugar. When a person has diabetes, the body either does not make enough insulin or is unable to use its own insulin well. If blood sugar builds up in the body and its levels are not controlled, it can lead to serious health complications, such as heart disease, stroke, kidney disease, blindness, amputations of the legs and feet, and early death. CDC programs and other scientific activities support improvements in health outcomes for people with Type 1 diabetes, Type 2 diabetes, Gestational diabetes, and Pre-diabetes.

Type 1 diabetes is an autoimmune condition where the body attacks and destroys insulin-producing cells, meaning no insulin is produced. This causes glucose to quickly rise in the blood. About 10 per cent of people with diabetes have Type 1. In Type 2 diabetes, the body doesn't make enough insulin, or the insulin it makes doesn't work properly, meaning glucose builds up in the blood. About 90 per cent of people with diabetes have Type 2. In Gestational diabetes, a form of high blood sugar affecting pregnant women. In Pre-diabetes, a condition in which blood sugar is high, but not high enough to be type 2 diabetes.

Generally doctors consider the following test results for diagnosing the Diabetes. Pregnancies , PG Concentration , Diastolic BP, Tri Fold Thick , Serum Ins , BMI , DP Function , Age , Diagnosis .

## **Diabetes Dataset**

Each test result of the diabetes is considered as an attribute in the dataset. The dataset has been taken from the from UCI machine learning repository. This dataset contains 768 number of instances. It consists of 9 attributes including the class attribute. The dataset contains the attributes whose values are discrete and continuous. The attributes such as Diagnosis possess discrete value and the remaining attributes possess continuous values. Data set contains Missing values which are represented by '?' in the dataset.

### **Attribute Explanation**

*Pregnancies:* Number of pregnancies

*PG Concentration:* Plasma glucose at 2 hours in an oral glucose tolerance test

*Diastolic BP (Diastolic Blood Pressure):* It measures the pressure in your blood vessels when your heart rest between beats

*Tri Fold Thick (Triceps Skin Fold Thickness):* A value used to estimate body fat, measured on the right arm halfway between the olecranon process of the elbow and the acromial process of the sculpa. Normal thickness in males is 12mm, in females is 23mm.

*Serum INS (Serum Insulin):* Serum Insulin is a hormone that is produced and stored in the beta cells of the pancreas.

*BMI (Body Mass Index):* Body Mass Index is used to determine a person's weight in regard to their height.

*DP Function (Diabetes Pedigree Function):* It provided some data on diabetes mellitus history in relatives and genetic relationship of those relatives to the patient.

*Diagnosis:* Diagnosis serves for professionals to categorize a collection of certain common symptoms in order to be able to 'treat'.



Attribute Name		Possible Values
Pregnancies	:	continuous
PG Concentration	:	continuous
Diastolic BP	:	continuous
Tri Fold Thick	:	continuous
Serum INS	:	continuous
BMI	:	continuous
DP Function	:	continuous
Age	:	continuous
Diagnosis	:	sick, healthy

No. of Instances : 768

No of Attributes : 9

### 1.6.2 THYROID

The thyroid gland, or simply the thyroid is an endocrine gland in the neck consisting of two lobes connected by an isthmus. The most common Thyroid problems involve abnormal production of thyroid hormones. Thyroid disorders can range from a small, harmless goitre (enlarged gland) that needs no treatment to life-threatening cancer. The most common disease cause being are Hypothyroidism and Graves' disease.

Generally doctors consider the following 29 test results for diagnosing the Thyroid: AGE, SEX, ON THYROXINE, QUERY ON THYROINE, ON ANTITHYROID MEDICATION, SICK, PREGNANT, THYROID SURGERY, I131 TREATMENT, QUERY HYPOTHYROID, QUERY HYPERTHYROID, LITHIUM, GOITRE, TUMOR, HYPOPITUITARY, PSYCH, TSH MEASURED, TSH, T3 MEASURED, T3, TT4 MEASURED, TT4, T4U MEASURED, T4U, FTI MEASURED, FTI, TBG MEASURED, TBG, REFERRAL SOURCE.

## **Thyroid Dataset**

Each test result of the Thyroid is considered as an attribute in the dataset. The dataset has been taken from the UCI machine learning repository. This dataset consists of 29 attributes. The dataset contains the attributes whose values are discrete and continuous. Data set contains Missing values which are represented by '?' in the dataset.

## **Attribute Explanation**

*Referral Source:* It consists of a string of letters indicating diagnosed conditions. A diagnosis "-" indicates no condition requiring comment.

Hyperthyroid conditions (A, B, C, D), Hypothyroid conditions (E, F, G, H), Binding protein (I, J) , General health (K) , Replacement therapy (L, M, N), Antithyroid treatment(O,P,Q), Discordant results (R,S,T) .

*Lithium:* It is a common treatment for bipolar disorder .It is known to cause enlarged thyroid and autoimmune thyroiditis, an inflammatory condition of the thyroid gland.

*Goitre:* It describes a condition where the thyroid gland located in the neck becomes enlarged .Iodine deficiency is the cause of goitre.

*Thyroxine:* It is the main hormone secreted into the bloodstream by the thyroid gland .It play vital roles in regulating the body's metabolic rate, heart and digestive functions, muscle control, brain development and maintenance of bones.

*Antithyroid Medication:* It is to prevent the thyroid from producing excess amounts of hormone.

*Tsh Measured:* A means of measuring thyroid function is to measure how much iodine is taken up by the thyroid gland.

*T3 Measured:* Thyroid creates hormones and controls how your body uses energy and your body's sensitivity to other hormones. The hormone is called triiodothyronine.

*T4U Measured:* It is a hormone that plays a role in several body functions including growth and metabolism.

*TT4 Measured:* A total T4 test measures the T4 that has bonded to protein and the free T4.

*FTI Measured:* It hasn't bonded to protein in your blood.

*TBG:* Thyroid Binding Globulin blood test measures the level of a protein that moves thyroid hormone throughout your body.

*Hypopituitary :* It is a condition in which small gland at the base of the brain does not produce one or more of its hormones or else not enough of them.

*I131 Treatment:* Your thyroid gland absorbs nearly all of the iodine in your body. When radioactive iodine also known as I131 treatment.

*Thyroid Surgery:* During this procedure, part or all of the thyroid gland is removed.

*Hypothyroid:* A Condition in which the thyroid gland doesn't produce enough thyroid hormone. Nearly more than 10 million cases recorded per year in India.

*Hyperthyroid:* A Condition that occurs due to excessive production of thyroid hormone by the thyroid gland.

#### **Attribute Name**

-----

#### **Possible Values**

-----

Age	:	continuous.
Sex	:	M, F.
On thyroxine	:	f, t.

Query on thyroxine	:	f, t.
On antithyroid medication	:	f, t.
Sick	:	f, t.
Pregnant	:	f, t.
Thyroid surgery	:	f, t.
I131 treatment	:	f, t.
Query hypothyroid	:	f, t.
Query hyperthyroid	:	f, t.
Lithium	:	f, t.
Goiter	:	f, t.
Tumor	:	f, t.
Hypopituitary	:	f, t.
Psych	:	f, t.
TSH measured	:	f, t.
TSH	:	continuous.
T3 measured	:	f, t.
T3	:	continuous.
TT4 measured	:	f, t.
TT4	:	continuous.
T4U measured	:	f, t.
T4U	:	continuous.
FTI measured	:	f, t.
FTI	:	continuous.
TBG measured	:	f, t.
TBG	:	continuous.
Referral source	:	WEST, STMW, SVHC, SVI, SVHD, other

No of Attributes : 29

## **2. LITERATURE SURVEY**

In [1], proposed that data mining is a multidisciplinary effort to extract the knowledge from data. The proliferation of large data sets within many domains poses unprecedented challenges to data mining. Not only are data sets getting larger, but new types of data become prevalent, such as streams on the Web, microarrays in genomics and proteomics, and networks in social computing and system biology. Researchers are realizing that in order to achieve successful data mining, feature selection is an indispensable component.

In [2], Feature selection is a process that selects a subset of original features. The optimality of a feature subset is measured by an evaluation criterion. As the dimensionality of a domain expands, the number of features  $N$  increases. Finding an optimal feature subset is usually intractable and many problems related to feature selection have been shown to be NP hard.

In [3] Proposed a hybrid Neuro-Genetic system for stock trading. They examine the system with 36 companies in stock exchange from 1992 to 2004. They used a recurrent neural network as a prediction model. They also optimize the weights of the neural network by Genetic Algorithm. The hybrid system showed prominent improvement on the average over the buy and hold approach.

In [4] Proposed data mining and fuzzy system based techniques for incurable Type-2 diabetes disease. They used Genetic algorithms optimization of chromosome to restrict in new population to get chromosomal accuracy from old rate of old population diabetes.

In [5] Proposed diagnosis system for predicting the risk of heart disease (like Cardiovascular). They use Genetic Algorithms and multi-layered neural network. In this work, Genetic algorithms used for determining the weights of NN, Because genetic algorithms finds tolerably good set of weights in fewer iterations.

In [6] Proposed an improved hybrid genetic algorithms and multilayer perception in their study to get better accuracy of the intelligent diseases diagnostic system. They also proposed a novel crossover technique that is SMCC (Segmented multi-chromosome crossover) that allows offspring to inherit information from multiple parent chromosomes with maintaining the information contained in gene segments.

By [7] Using Genetic algorithm and Decision Tree data mining technique in their model to reduce the number of tests which were needed to be taken by a heart patient.

Mario A Garcia et al [8] explained that by using ESDIABETES, patients can enter their symptoms any time and get diagnosis about their blood sugar levels. Medical trainees can use this software to study about diabetes. Thus, this tool proves advantageous for both testing and training purposes.

PetricǎVizureanu et al [9] Presented the task of extracting the hidden patterns underlying the data has become more and more difficult. The situation is further complicated by the complexity of biological data (numerous different types, sources, quality, etc.) and the creation of tools that can exploit the accumulated human expertise in this field is now crucial.

M.Kalpana et al [10] propose a fuzzy expert system framework which constructs large scale knowledge based system effectively for diabetes. The knowledge is constructed by using the Fuzzification to convert crisp values into fuzzy values. By applying the fuzzy verdict mechanism, diagnosis of diabetes becomes simple for medical practitioners.

Sa'di et.al. [11] used data mining techniques such as Naïve Bayes, J48 and Radial Basis Function Artificial Neural Networks for diagnosing diabetes type 2. They took advantage of a data set with 768 data samples, 230 of them selected for test phase. Naive Bayes algorithm with 76.95% accuracy outperformed J48 and RBF with 76.52% and 74.34% accuracies, respectively.

Al-Rofiye, et al. [12] use multilayer perceptron (MLP) artificial neural networks for identifying diabetes type 2. .In order for artificial neural network to learn non-linear, it should designed as multi-layer.

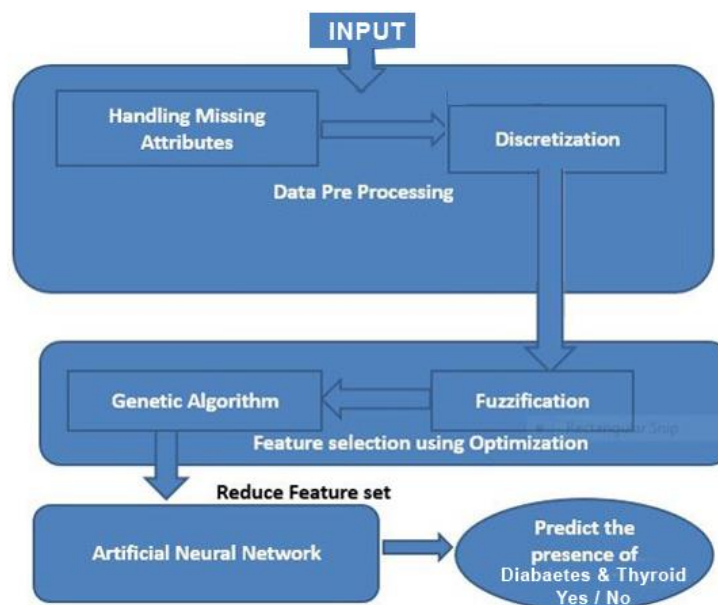
### 3. SYSTEM ANALYSIS & REQUIREMENTS

#### 3.1 Existing system

Generally, doctors consider all the symptoms in order to diagnose the presence of disease even though some of the symptoms are not necessary to diagnose the disease. Due to these unnecessary symptoms, it usually takes more time to decide the presence of disease.

#### 3.2 Proposed System

The redundant or unnecessary symptoms makes the diagnosing system to take more time to decide the presence of disease. By applying the Feature selection techniques, such as Genetic Algorithm, the unnecessary symptoms can be eliminated by considering necessary symptoms for diagnosing the disease. After applying the techniques to the dataset, it will produce the optimum feature set .By using Artificial Neural Network, we will find the decision about the disease for the optimum feature set.



**Fig.3.1 Proposed System Architecture**



### 3.3 MODULES

#### 3.3.1 Pre-processing:

Data pre-processing is a data mining technique that implicates transformation of raw data into an understandable format.

- **Handling missing values:** Using attribute mean the missing values are replaced by the mean of all attribute values.
- **Discretization:** It refers to the conversion of continuous attributes discretized or nominal attributes. Equal areas is used to discretize the continuous attributes in the dataset.

#### 3.3.2 Feature selection

##### 3.3.2.1 Fuzzification

Fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. It handles:

- Uncertainty and Imprecise nature.
- To describe grades of Truth.
- It is useful in Complex Non-linear applications.

We are using Triangular Membership function which indicates 3 possible ranges. The triangular distribution is a continuous probability distribution with lower limit a, upper limit c and mode b , where  $a < c$  and  $a \leq b \leq c$ .

Here we are considering a, b, c values as minimum, standard deviation and maximum values of an attribute.

$$\mu(x, a, b, c) = \begin{cases} 0 & x < a, x > c \\ \frac{x - a}{b - a} & a \leq x \leq b \\ \frac{c - x}{c - b} & b < x \leq c \end{cases} \dots\dots\dots(7)$$

### 3.3.2.2 Genetic Algorithm

Dataset contains 23 features and these subset of features are represented as Chromosome. Each Chromosome represents the solution, which contain series of 0's and 1's where 0 indicates the absence of an attribute and 1 indicates the presence of attribute

1	1	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Genetic Algorithm consists of 3 operators are

1. Selection
2. Cross Over
3. Mutation

#### Selection:

This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction than poor ones.

Chromosome 1 = 

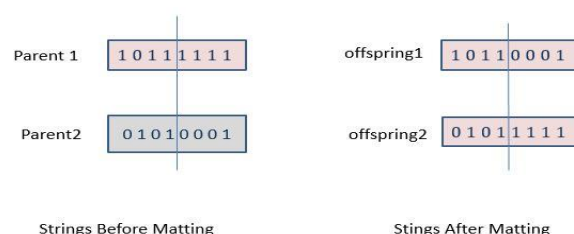
1	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

Chromosome 2 = 

0	1	0	1	0	0	0	1
---	---	---	---	---	---	---	---

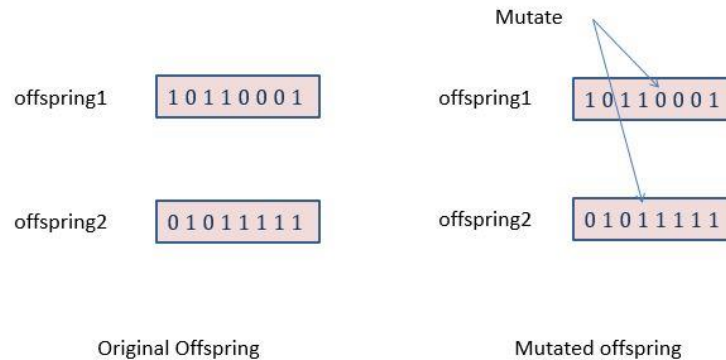
#### Cross Over:

After selection phase, population is enriched with better individuals. It makes clones of good strings but does not create new ones. Cross over operator is applied to the mating pool with a hope that it would create better strings.



## Mutation

After cross over, the strings are subjected to mutation. Mutation of a bit involves flipping it, changing 0 to 1 and vice-versa

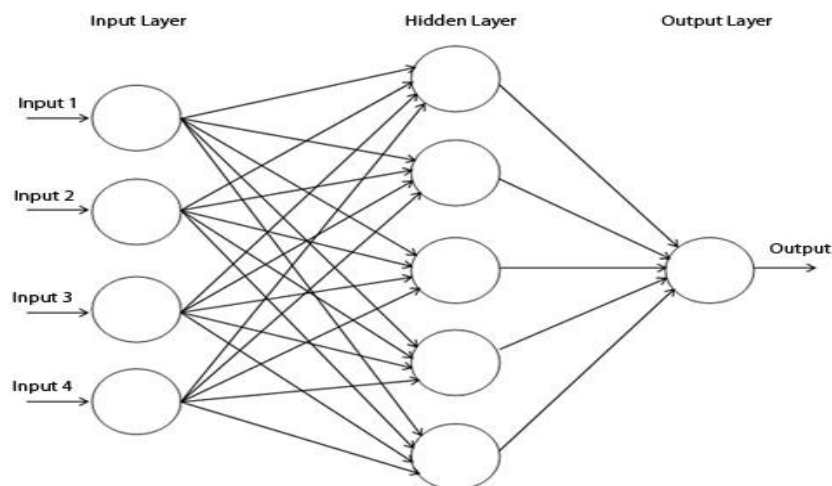


### 3.3.2.3.1 Artificial Neural Networks

From the fuzzified and feature selection in this project we have decreased the number of attributes to 5 and 12 for diabetes and thyroid respectively.

**Step 1:** Construct an network:

We have created neural network with single hidden layer with 2 nodes.



**Step 2:** Assign the random weights.

3,6,4,2,1,5,1,2.

Let us consider an sample network.

Patients	Symptom1	Symptom2	Symptom3
Patient1	0	6	4
Patient2	2	7	9
Patient3	0	0	5

**Table 3.1: Sample Patient Database**

**Step 3:** Multiply the weights with the respected symptoms .

$$\text{Node1} = 0*3+6*6+4*4$$

$$\text{Node 1} = 52$$

$$\text{Node 2} = 2*0+6*1+5*4$$

$$\text{Node 2} = 26$$

**Step 4:** Apply sigmoid function.

$$\text{Node 1} = 0.9999$$

$$\text{Node 2} = 0.9998$$

**Step 5:** Again check the value for output layer.

$$\text{Output} = 0.9986.$$

**Step 6:** Take the cost function and apply mean square error.

$$mse = \frac{1}{Q} \sum_{k=1}^Q e(k)^2 = \frac{1}{Q} \sum_{k=1}^Q (t(k) - a(k))^2 \quad \text{.....(8)}$$

**Step 7:** Then apply backpropagation and repeat the same process from step 2 to step 6 until output reaches the threshold value or limited number of iterations.

### **3.4.REQUIREMENTS :**

#### **3.4.1 Hardware Requirement**

Processor	-	Intel CORE <sup>TM</sup> i3-3230M 2.6GHz to 3.2GHz
RAM	-	2GB
HDD	-	500GB

#### **3.4.2 Software Requirement**

Java	-	jre6 and above
Operating system	-	Windows XP and above
Programming tool	-	MATLAB 2012b and above
Tool	-	Weka 3.6 and above

## **4 METHODOLOGY**

### **4.1. Problem Statement**

Medical datasets are often classified by a large number of disease measurements and a relatively small number of patient records. All these measurements (features) are not important or irrelevant/noisy. This extreme number of features carries the problem of memory usage, time complexity in order to represent the dataset. Feature Selection (FS) is a solution that involves finding a subset of prominent features to improve predictive accuracy and to remove the redundant features.

The objective of this project is to investigate the fuzzy logic based Genetic algorithm to generate reduced number of features with improvement in diagnostic performance. The performance of the model can be increased by selecting optimum features using optimization techniques to better diagnoses of disease. The goal of this project is to select Optimum features using the optimization techniques such as:

- Genetic Algorithm
- Artificial Neural Network techniques

The above techniques are used for better diagnoses of disease. Each test result of the disease is considered as an attribute in the dataset. It includes of following modules.

- Pre – processing.
- Feature Selection.
- Classification.

### **4.2 Module 1: Pre-processing**

In this phase, the dataset undergoes several tasks of pre-processing which includes:

- Handling Missing values
- Discretization

### 4.2.1 Handling Missing Values

The missing values are handled by computing the attribute mean of all the other attribute values in the same column of the dataset. Here '?' represents the missing value of an attribute.

Example:

- Before cleaning missing data in the dataset.

Column of a dataset: 2, 40, 1, 2.08, ?, 4.0

- Calculate attribute mean

$$\text{Mean} = 2+40+1+2.08+4.0 / 5 = 9.816$$

- Now, the missing value (?) is replaced by the mean of the data

Data after Cleaning: 2, 40, 1, 2.08, 9.816, 4.0

### 4.2.2 Discretization

The attribute consist of continuous values and are now converted into discrete values by applying discretization to the attributes in the dataset.

Example:

- Data before applying discretization

0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

- Discretizing the data using equal areas method

0.00 – 0.33 = Low

0.33 – 0.66 = Medium

0.66 – 0.99 = High

- Data after discretization

Low, Low, Medium, Medium, high, high, high

## 4.3 Module 2: Feature Selection

The data that is pre-processed is now applicable for applying optimization techniques such as Fuzzification, genetic algorithm and ANN thereby feature reduction is done.

### 4.3.1 Fuzzification

We are using Triangular Membership function which indicates 3 possible ranges.

$$\mu(x, a, b, c) = \begin{cases} 0 & x < a, x > c \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b < x \leq c \end{cases} \dots\dots\dots(9)$$

Where a, b, c are minimum, Standard deviation and maximum values of an attribute.

### Working Example

P-ID	Age	Gen	BG	W	S	F	BP
111	18	M	5	54	108	101	90
112	32	M	1	65	123	104	80
113	43	M	3	70	116	97	60
114	54	F	2	84	94	90	100
115	65	F	7	66	82	107	110
116	36	M	3	66	237	107	120
117	47	F	1	72	109	92	140
118	68	M	6	81	58	100	105
119	79	F	2	69	71	100	100
120	40	M	7	40	85	99	86

**Table 4.1: Fuzzy Input Table**

Let a,b,c are membership variables as shown in the Fig 1. The boundary and core value of membership assign into following attributes BP = {a=60,b=100,c=200}

Fever( $F^{\circ}C$ ) = {a=90,b=97,c=107}

Sugar={a=30,b=100,c=320}

Age= {a=18,b=54,c=90}

Weight= {a=40, b=54,c=92}



Blood Group= {a=0,b=4,c=7}

Consider a = 20, b = 40, c = 60 for age .

Calculating fuzzy value for  $\mu = 32$  using triangular membership function.

$$\mu(32,20,40,60) = \frac{32-20}{40-20} \text{ ( since } 20 \leq 32 \leq 40 \text{ )} = 0.6.$$

Patient id	Age	Gender	BloodGrp	Weight	Sugar	Fever	Bp
111	0.0	1	0.6666	1.0	0.0	1.0	1.0
112	0.4516	1	0.0	0.7105	0.9380	0.5	0.6666
113	0.8064	1	1.0	0.5789	0.9714	0.0	0.0
114	0.0	0	0.6666	0.2105	0.9142	0.0	0.0
115	0.7575	0	0.0	0.6842	0.7428	0.0	1.0
116	0.5806	1	1.0	0.6842	0.3952	0.0	0.8888
117	0.9354	0	0.3333	0.5263	0.0	0.3333	0.6666
118	0.6666	1	0.3333	0.2894	0.4	0.0	0.0
119	0.3333	0	0.6666	0.6052	0.5857	0.0	0.0
120	0.7096	1	0.6666	0.0	0.7857	0.0	0.8666

**Table 4.2: Fuzzy Output Result**

#### 4.3.2 Genetic Algorithm

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

**STEP 1:** Represent the problem variable domain as a chromosome of a fixed length, choose the size of a chromosome population  $N$ , the crossover probability  $p_c$  and the mutation probability  $p_m$ .

**STEP 2:** Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

$$\text{Fitness}(X) = \text{Accuracy} + (\text{No of absent attributes} * \text{Balancing factor1}) + (\text{no of Present attributes} * \text{Balancing factor2}) \dots\dots\dots(10)$$

$\text{Accuracy}(X)$  = Accuracy is calculated by using one of the classification techniques such as C4.5.

**STEP 3:** Randomly generate an initial population of chromosomes of size  $N$   
 $x_1, x_2, \dots, x_n$

**STEP 4 :** Calculate the fitness of each individual chromosome  
 $F(x_1), F(x_2), \dots, F(x_n)$

**STEP5:** Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness.

**STEP 6:** Create a pair of offspring chromosomes by applying the genetic operators - **crossover** and **mutation**.

**STEP 7:** Place the created offspring chromosomes in the new population.

**STEP 8:** Repeat *Step 5* until the size of the new chromosome population becomes equal to the size of the initial population,  $N$ .

**Step 9:** Replace the initial (parent) chromosome population with the new (offspring) population.

**Step 10:** Go to *Step 4*, and repeat the process until the termination criterion is satisfied.

### Step 1: Initialization

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the number of Instances in the dataset. The population is generated randomly, covering the entire range of possible solutions '0' and '1' where '0' is the absent of attribute and '1' is the present of attribute in the dataset.

1	0	1	1	0	1	0	1	0
---	---	---	---	---	---	---	---	---

### Step 2: Defining Fitness Function

The fitness function simply defined is a function which takes a candidate solution to the problem as input and produces as output how “fit” our how “good” the solution is with respect to the problem in consideration. It measures the quality of the represented solution. It is always problem dependent. The fitness of a solution is measured, how best it gives the result.

$$\text{Fitness}(X) = \text{Accuracy} + (\text{No of absent attributes} * \text{Balancing factor1}) + (\text{no of Present attributes} * \text{Balancingfactor2}) \dots\dots\dots(11)$$

Accuracy is calculated by using one of the classification techniques such as C4.5.

### C4.5 Algorithm

Pruned or unpruned C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by J48 can be used for classification. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set  $S=S_1, S_2$ .

Each sample  $S_i$  consists of  $p$ -dimensional vector, where the  $X_i$  represents attributes or features of the sample, as well as the class in which  $S_i$  falls. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is

chosen to make the decision. The C4.5 algorithm then resources on the smaller sub lists. This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain .In the case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously -unseen class encountered. Again C4.5 creates a decision node higher up the tree using the expected value.

This algorithm selects the attribute having the more the information gain as splitting attribute. Information gain tells how important the attribute in the feature set is. For calculating the information gain, entropy is used, which can be calculated using the following formulae.

$$\text{Entropy (E)} = - \sum_{i=1}^n ((P_i * \log_2 P_i))$$

Where  $P_i$  is the probability of class  $i$ .

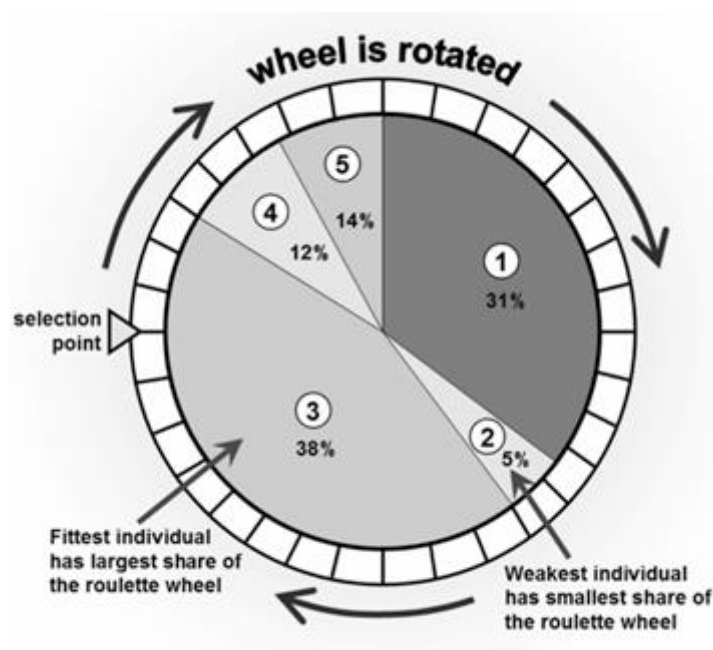
$$\text{Information gain} = \text{Entropy (parent)} - [\text{average entropy (children)}]$$

### **Step 3 : Selection**

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions.

## Roulette Wheel Selection

The basic part of the selection process is to stochastically select from one generation to create the basis of the next generation. The requirement is that the fittest individuals have a greater chance of survival than weaker ones. This replicates nature in that fitter individuals will tend to have a better probability of survival and will go forward to form the **mating pool** for the next generation. Weaker individuals are not without a chance. In nature such individuals may have genetic coding that may prove useful to future generations.



**Fig. 4.1 Roulette Wheel Selection**

Chromosome with bigger fitness will be selected more times.

This can be simulated by following algorithm.

**Step 1:** Calculate sum of all chromosome fitness in population - sum **S**.

**Step 2:** Generate random number from interval  $(0, S)$  - **r**.

**Step 3:** Go through the population and sum fitness from **0**-sum **s**. When the sum **s** is greater than **r**, stop and return the chromosome where you are.

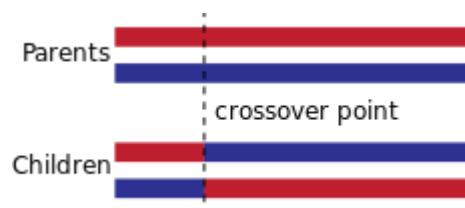
Of course, step **1** is performed only once for each population.

#### **Step 4: Cross Over**

Cross over is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. Cross over is a process of taking more than one parent solutions and producing a child solution from them. First, the crossover operator randomly chooses a crossover point where two parent chromosomes “break”, and then exchanges the chromosome parts after that point. As a result, two new offspring are created. If a pair of chromosomes does not cross over, then the chromosome cloning takes place, and the offspring are created as exact copies of each parent.

##### ***Single-point crossover***

A single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children:



**Fig. 4.2 Single-point Crossover**

#### **Step 5: Mutation**

Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

### **Bit String Mutation**

The mutation of bit strings ensue through bit flips at random positions.

Example:

```
1 0 1 0 0 1 0
      ↓
1 0 1 0 1 1 0
```

The probability of a mutation of a bit is  $1/L$ , where  $L$  is the length of the binary vector. Thus, a mutation rate of  $1/L$  per mutation and individual selected for mutation is reached.

#### **4.3.2.1 Working Example for Genetic Algorithm**

Genetic Algorithm technique is applied to a sample dataset to show its working.

- Number of Chromosomes : 5
- Number of Cycles : 5
- Balancing Factor1 : 0.1
- Balancing Factor2 : 0.4
- Accuracy is calculate by applying J48(C4.5) algorithm using Weka Library functions
- $\text{Fitness}(X) = \text{Accuracy} + (\text{No of absent attributes} * \text{Balancing factor1}) + (\text{no of Present attributes} * \text{Balancingfactor2}) \dots\dots\dots(12)$
- $\text{Probability} = \text{fitness of each chromosome} / \text{sum of all chromosomes} \dots\dots\dots(13)$
- The Probability of selecting chromosome in Crossover is  $P_c = 1.1$  and
- The Probability of Mutation is  $P_m = 0.014$

\*\*\*\*\*Initialization\*\*\*\*\*

Chromosome Label	Chromosome(Bit String)
X1	1 1 0 0
X2	0 1 0 0
X3	0 0 0 1
X4	0 1 1 1
X5	1 0 0 1

**Table 4.3: Initialization of Chromosome**

\*\*\*\*\*Cycle 1 \*\*\*\*\*

labe l	old generatio n	missing attribute s	accurac y	Fitness	Probabilit y	new generatio n
X1	1 1 0 0	2	81.8	82.8	0.2198	1 1 0 1
X2	0 1 0 0	3	81.8	82.5	0.2190	0 1 0 1
X3	0 0 0 1	3	69.6	70.3	0.1866	0 0 0 1
X4	0 1 1 1	1	69.4	70.7	0.1877	0 1 1 0
X5	1 0 0 1	2	69.3	70.3	0.1866	1 0 0 0

**Table 4.4: Genetic Algorithm 1<sup>st</sup> cycle result table**

Best Chromosome : [ 1 1 0 1 ]

Best Fitness : 82.8

\*\*\*\*\*Cycle 2 \*\*\*\*\*

Label	old generation	missing attributes	accuracy	fitness	Probability	new generation
X1	1 1 0 1	1	81.9	83.2	0.2145	1 1 0 0
X2	0 1 0 1	2	80.3	81.3	0.2096	0 1 0 0
X3	0 0 0 1	3	69.4	70.1	0.1807	0 0 1 0
X4	0 1 1 0	2	81.8	82.8	0.2135	0 1 0 1
X5	1 0 0 0	3	69.7	70.4	0.1815	1 0 0 0

**Table 4.5: Genetic Algorithm 2<sup>nd</sup> cycle result table**



Best Chromosome: [1 1 0 0]

Best Fitness: 83.2

\*\*\*\*\*Cycle 3 \*\*\*\*\*

Label	old generation	missing attributes	accuracy	fitness	Probability	new generation
X1	1 1 0 0	2	81.8	82.8	0.2141	0 1 1 0
X2	0 1 0 0	3	80.4	81.1	0.2097	0 1 0 0
X3	0 0 1 0	3	69.6	70.3	0.1818	0 0 1 0
X4	0 1 0 1	2	81.2	82.2	0.2126	0 1 1 0
X5	1 0 0 0	3	69.5	70.2	0.1815	1 0 0 0

**Table 4.6: Genetic Algorithm 3<sup>rd</sup> cycle result table**

Best Chromosome: [ 0 1 0 0]

Best Fitness: 82.8

\*\*\*\*\*Cycle 4 \*\*\*\*\*

Label	old generation	missing attributes	accuracy	Fitness	probability	new generation
X1	0 1 1 0	2	81.8	82.8	0.2125	1 1 0 1
X2	0 1 0 0	3	81.6	82.3	0.2120	0 1 1 0
X3	0 0 1 0	3	69.6	70.3	0.1814	0 0 1 0
X4	0 1 1 0	2	81.5	82.5	0.2125	0 1 1 1
X5	1 0 0 0	3	69.5	70.2	0.1808	1 0 0 1

**Table 4.7: Genetic Algorithm 4<sup>th</sup> cycle result table**

Best Chromosome: [ 1 1 0 1]

Best Fitness: 82.8

\*\*\*\*\* Cycle 5 \*\*\*\*\*

Label	old generation	missing attribute	Accuracy	fitness	Probability	new generation
X1	1 1 0 1	1	81.8	83.1	0.2082	0 1 0 1
X2	0 1 1 0	2	81.7	82.7	0.2072	0 0 1 1
X3	0 0 1 0	3	80.8	81.5	0.2042	0 0 0 1
X4	0 1 1 1	1	79.8	81.1	0.2032	0 1 1 0
X5	1 0 0 1	2	69.6	70.6	0.1769	1 0 0 1

**Table 4.8 : Genetic Algorithm 5<sup>th</sup> cycle result table**

Best chromosome: [0 1 0 1]

Best fitness: 83.1

#### **4.4 Module 3: Implementation Of Artificial Neural Networks With BackPropagation**

In neural networks implementation, we are implementing with variable input perceptrons, and one hidden layer containing variable number of perceptrons and one output perceptron.

##### **Initialization of Network Weights:**

Although the ideal initial values for weights (i.e., those that will maximize the effectiveness and speed with which a neural network learns) cannot yet be determined theoretically, it is general practice to assign randomly-generated positive and negative quantities as the initial weight values. Such a random distribution can help minimize the chances of the network becoming stuck in local minima. Typically, values are selected from a range  $[-a, +a]$  where  $0.1 < a < 2$ . The reason for using random initial weights is to break symmetry, while the reason for using small initial weights is to avoid immediate saturation of the activation function.

We are initialising the weights with the random values because in the process of backpropagation we have to decrease the value that is weight dependent if we initialise with zeros or same weights the change in weights will not effect the output and it will decrease the accuracy. Initialising the weights with same value will lead to saturation of weights.

In this project we are using sigmoid function as an activation function because the outcome after applying sigmoid will be in the range of [0,1]. As in medical diagnosis we need only binary classification and we want either it is 'yes' or 'no'. Suppose if we had used linear function, it is very difficult to find the bounds for this function and we cannot predict the ranges of output and we have to put ranges for the input also.

Eg: let us consider  $5x^2$  as an activation function the value for that function will increase with increase in value and we cannot bound or predict the output is in some range for any value of input.

The cost function here we used in this project is mean squared error. The cost function is used as an measure where to stop training and an measure to decrease the error.

$$mse = \frac{1}{Q} \sum_{k=1}^Q e(k)^2 = \frac{1}{Q} \sum_{k=1}^Q (t(k) - a(k))^2 \dots\dots\dots(14)$$

# ALGORITHM:

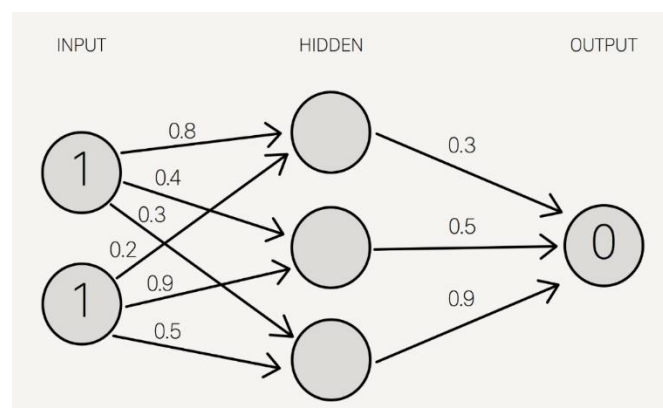
- **Step 1:** Take an input from already selected features from defuzzified data.
- **Step 2:** Apply the decision patterns in the hidden layer.
- **Step 3:** Obtain the output and find the difference between the cost function.

- **Step 4:** If the difference is greater than threshold value the again change the weights and bias value and iterate the process till we get difference less than threshold value.
- **Step 5:** If less than threshold value the obtain the output and give the result.
- **Step 6:** After that we will use backpropagation to increase the efficiency of the system.

#### 4.4.1 Working Example for FeedForward and then BackPropagation:

Let us consider 2 input neurons, 3 hidden neurons and 1 output neuron.

**Step 1:** Initializing weights with random variables as in fig 4.3.



**Fig 4.3 Initialising the weights for input**

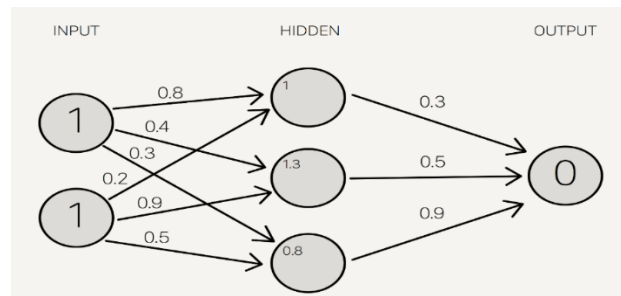
**Step 2:** Assigning the values for the hidden layers by using the values of input layers.

$$1*0.8+1*0.2 = 1$$

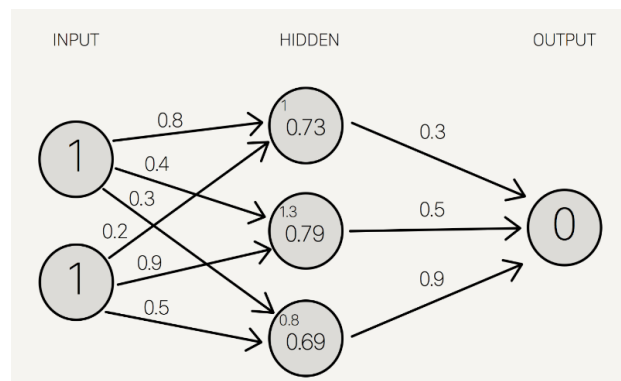
$$1*0.4+1*0.9 = 1.3$$

$$1*0.3+1*0.5 = 0.8$$

**Step 3:** Apply the activation function and assign the values to the hidden layers.

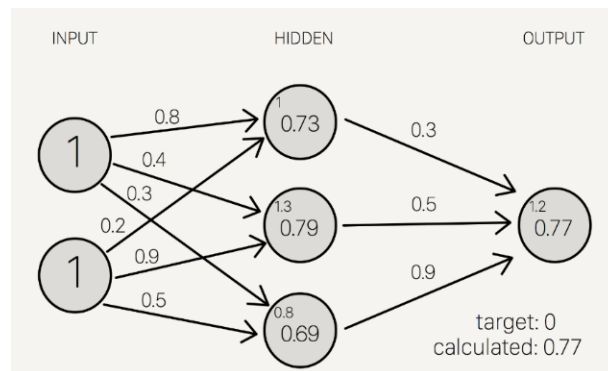


**Fig 4.4 Apply the sigmoid function**



**Fig 4.5 Obtain the output**

**Step 4:** Apply same for the output layer also.



**Fig 4.6 Values after applying the activation function in output layer**

### BACKPROPAGATION:

To improve and accuracy and decrease the error to get the output we want we are using backpropagation. For that we are using mean square error i.e.,

$$mse = \frac{1}{Q} \sum_{k=1}^Q e(k)^2 = \frac{1}{Q} \sum_{k=1}^Q (t(k) - a(k))^2 \dots\dots\dots(15)$$

Output sum margin of error = target-calculated .....**(16)**

Target = 0

Calculated = 0.77

Target-calculated = -0.77

Delta output sum = s'(output sum)\*(output sum margin of error).....**(17)**

Delta Output sum = -0.13439890643886018

Hidden result 1 = 0.73105857863

Hidden result 2 = 0.78583498304

Hidden result 3 = 0.68997448112

Delta weights = delta output sum \* hidden layer results. ....**(18)**

Delta weights = -0.1344 \* [0.73105 0.78583 0.69997]

Old w7 = 0.3

Old w8 = 0.5

Old w9 = 0.9

New w7 = 0.202

New w8 = 0.394

New w9 = 0.806

Delta hidden sum = delta output sum \* hidden-to-outer weights \* s'(hidden sum).....**(19)**

Delta hidden sum = [-0.0403, -0.0672, -0.1209] \* [0.1966, 0.1683, 0.2139]

Delta hidden sum = [-0.0079, -0.0113, -0.0259]

The final total values input 1 = 1

Input 2 = 1

Delta weights = delta hidden sum \* input .....(20)

Delta weights = [-0.0079, -0.0113, 0.0259] \* [1, 1]

Old w1 = 0.8	New w1 = 0.7921
Old w2 = 0.4	New w2 = 0.3887
Old w3 = 0.3	New w3 = 0.2741
Old w4 = 0.2	New w4 = 0.1921
Old w5 = 0.9	New w5 = 0.8887
Old w6 = 0.5	New w6 = 0.4741

**Table 4.9: Comparision of weights in Neural Networks**

## 5. IMPLEMENTATION

### 5.1 Sample Code

#### 5.1.1 Genetic Algorithm

```
options = gaoptimset;  
  
options = gaoptimset(options,'PopulationSize', PopulationSize_Data);  
options = gaoptimset(options,'Generations', Generations_Data);  
options = gaoptimset(options,'CreationFcn', @popf);  
options = gaoptimset(options,'StallGenLimit', StallGenLimit_Data);  
options = gaoptimset(options,'SelectionFcn', @selectionroulette);  
options = gaoptimset(options,'CrossoverFcn', @crossoversinglepoint);  
options = gaoptimset(options,'MutationFcn', { @mutationuniform [] });  
options = gaoptimset(options,'Display', 'iter');  
options = gaoptimset(options,'PlotFcns', { @gaplotbestf });  
  
[x,fval,exitflag,output,population,score]=ga(@ObjFun,nvars,[],[],[],[],[],[],[],[],options);
```

#### 5.1.2 ARTIFICIAL NEURAL NETWORKS:

```
function theta=ann(x,l)  
    p=size(x,2);  
    y=x(:,p);  
    out=zeros(l+1,1);  
    out=double(out);  
    temp=zeros(l,p);  
    m=size(x,1); %number of rows  
    iterations=1000;  
    alpha=0.1;  
    %theta=zeros(56,1);  
    theta=randi([-10,10],[(p)*1,1]);  
    theta=double(theta);  
    for i=1:m  
        net=zeros(l+1,1);  
        net=double(net);  
        for j=1:iterations  
            %Forward Propagation Start%  
            for k = 1:l  
                for c=1:(p-1)
```



```

        net(k,1) = theta((k-1)*(p-2)+c)*x(i,c);
    end
end
n=7;
for k=1:l
    out(k,1)=sigmoid(net(k,1));
end
for s=1:l
    net(l+1,1)=net(l+1,1)+theta((k)*(p-1)+s)*out(s,1);
end
out(l+1,1)=sigmoid(net(l+1,1));

%Forward Propagation End%
*****BACKPROPAGATION PROGRAM*****

```

### 5.1.3 GRAPHICAL USER INTERFACE:

```

function diabetes(handles)
a = str2double(get(handles.edit1,'string'));
b = str2double(get(handles.edit2,'string'));
c = str2double(get(handles.edit3,'string'));
d = str2double(get(handles.edit4,'string'));
e = str2double(get(handles.edit5,'string'));
theta=xlsread('dia_theta.xls');
net=zeros(8,1);
out=zeros(8,1);
for k = 1:7
    net(k,1) = theta((k-1)*5+1,1)*a+theta((k-1)*5+2,1)*b+theta((k-1)*5+3,1)*c+theta((k-1)*5+4,1)*d+theta((k-1)*5+5,1)*e;
end
for k= 1:7
    out(k,1)=sigmoid(net(k,1));
end
net(8,1)=theta(36,1)*out(1,1)+theta(37,1)*out(2,1)+theta(38,1)*out(3,1)+theta(39,1)*out(4,1)+theta(40,1)*out(5,1)+theta(41,1)*out(6,1)+theta(42,1)*out(7,1);
out(8,1)=sigmoid(net(8,1));
disp(out(8,1));
if out(8,1)>=0.5
    disp('yes');
else
    disp('no');
end
end
end

*****GRAPHICAL INTERFACE*****

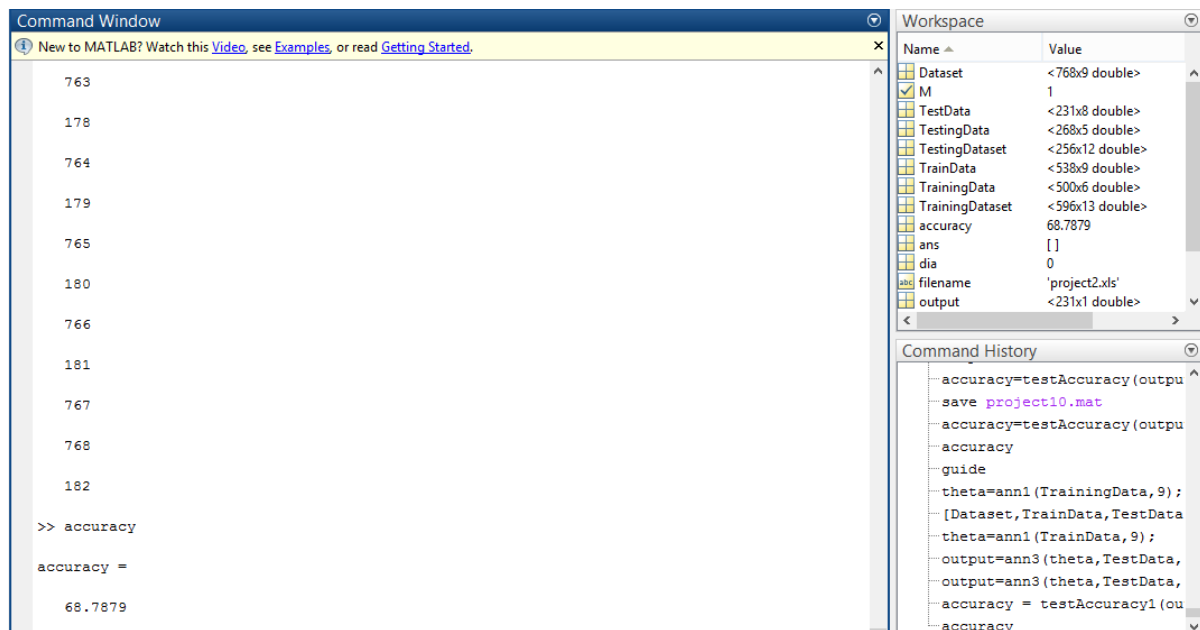
```

## 6. RESULTS

### 6.1 RESULT ANALYSIS

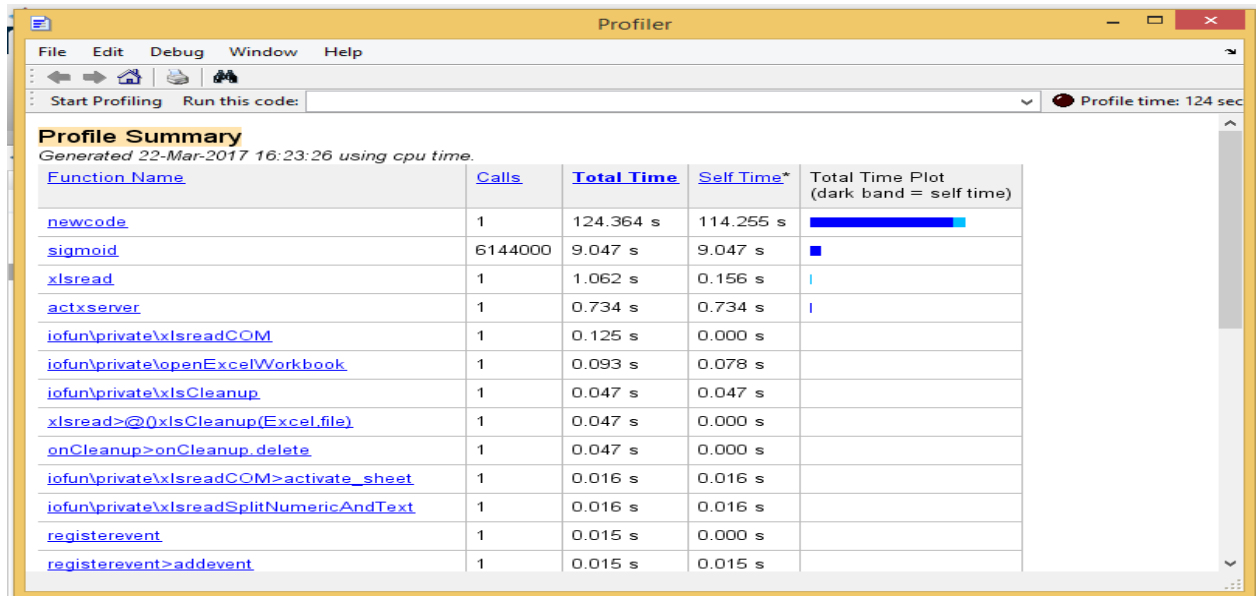
In this project, the classification part is the most important. First consider the raw data for the diabetes which contains the 768 instances and 8 attributes or features.

Using the 8 features of Original diabetes dataset, we can predict the presence of diabetes with 68.78 % accuracy with neural networks.

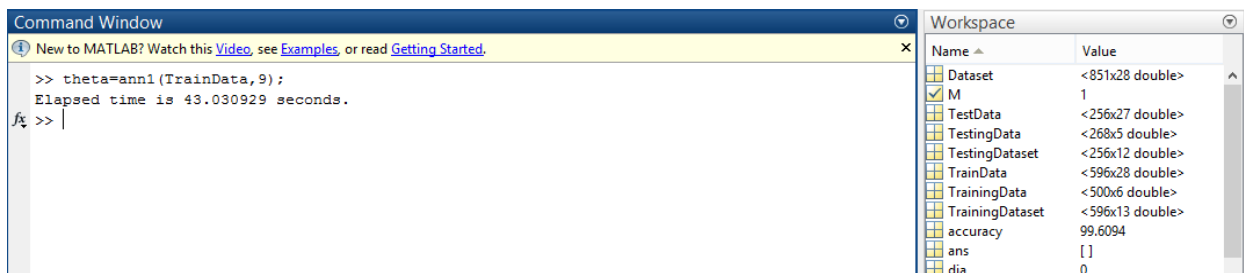


### 6.1 Accuracy of original dataset

Time for building the model using the original feature set (8 features) obtained by Neural Networks is 114.255 seconds.



## 6.2 Time complexity of original dataset of diabetes



## 6.3 Time complexity of original dataset for thyroid

In this project, size and learning time are considered as important factors for analysing the optimization techniques. These important factors can be increased by reducing overall irrelevant features which is actually done through Feature Selection.

### 6.1.1.1 For diabetes:

#### 6.1.1.1 Fuzzification :

Fuzzification removes uncertainty and datasets are taken as input and provide fuzzified values as an input to Genetic Algorithm.

Diabetes dataset before Fuzzification :

6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1

**Fig 6.4 Dataset before fuzzification**

Diabetes dataset after Fuzzification :

0.807018	0.305339	0.48712	0.770641	0	0.565727	0.85844	0.44772	yes
0.296773	0.682523	0.545574	0.842888	0	0.683939	0.990582	0.722128	no
0.660288	0.095793	0.565059	0	0	0.739667	0.836896	0.707686	yes
0.296773	0.658575	0.545574	0.915136	0.815661	0.658608	0.351322	0.866554	no
0	0.371197	0.798876	0.770641	0.927806	0.405297	0.063198	0.693243	yes
0.880384	0.496925	0.467635	0	0	0.700826	0.485535	0.736571	no
0.890319	0.724432	0.701452	0.806764	0.763597	0.609634	0.671065	0.794341	yes
0.513557	0.502912	0	0	0	0.537018	0.221057	0.751013	no
0.593546	0.011974	0.506604	0.650228	0.414639	0.618078	0.315795	0.404392	yes
0.660288	0.443041	0.253302	0	0	0	0.607906	0.389949	yes
0.953749	0.532847	0.292272	0	0	0.498178	0.446061	0.736571	no
0.513557	0.185598	0.467635	0	0	0.491423	0.90153	0.678801	yes
0.513557	0.359223	0.40918	0	0	0.675495	0.468719	0.346622	no
0.296773	0.05987	0.604028	0.915136	0	0.624833	0.96808	0.317736	yes
0.880384	0.197572	0.48712	0.963301	0.918227	0.697449	0.877591	0.433277	yes
0.733653	0.592717	0	0	0	0.626522	0.926905	0.707686	yes
0	0.48495	0.370211	0.626146	0.842963	0.359701	0.894827	0.722128	yes
0.733653	0.550808	0.467635	0	0	0.633277	0.69475	0.722128	yes
0.296773	0.574756	0.8963	0.734517	0.720211	0.401919	0.414481	0.693243	no
0.296773	0.502912	0.506604	0.830847	0.833015	0.54884	0.90536	0.707686	yes

**Fig 6.5 Dataset after fuzzification**

#### 6.1.1.2.Genetic algorithm :

Genetic Algorithm eliminates irrelevant features thereby increasing overall time complexity and size reduction.

Optimum features to diabetes dataset obtained by Genetic Algorithm :

- Pregnancies.
- PG Concentration.
- Diastolic BP

- Serum Ins
- Age

0.6603	0	0.7397	0.8369	0.7077
0.2968	0.9151	0.6586	0.3513	0.8666
0	0.7706	0.4053	0.0632	0.6932
0.8804	0	0.7008	0.4855	0.7366
0.8903	0.8068	0.6096	0.6711	0.7943
0.5136	0	0.5370	0.2211	0.7510
0.5935	0.6502	0.6181	0.3158	0.4044
0.6603	0	0	0.6079	0.3899
0.9537	0	0.4982	0.4461	0.7366
0.5136	0	0.4914	0.9015	0.6788
0.5136	0	0.6755	0.4687	0.3466
0.2968	0.9151	0.6248	0.9681	0.3177
0.8804	0.9633	0.6974	0.8776	0.4333
0.7337	0	0.6265	0.9269	0.7077
0	0.6261	0.3597	0.8948	0.7221
0.7337	0	0.6333	0.6947	0.7221
0.2968	0.7345	0.4019	0.4145	0.6932
0.2968	0.8308	0.5488	0.9054	0.7077
0.8903	0.6984	0.4695	0.8216	0.7799
0.6603	0	0.5353	0.9729	0.4477
0.7337	0	0.4610	0.9427	0.5777
0.5869	0.7706	0.6434	0.7303	0.7510

**Fig 6.6 Dataset after feature selection and fuzzification**

#### **6.1.1.3.Artificial neural networks:**

Using the 5 optimum features obtained from Genetic Algorithm we can predict the presence of diabetes with 67.9104 % accuracy with neural networks.

```
>> theta=NeuralNetwork_Train(TrainingDataset);
Elapsed time is 14.644760 seconds.
>> theta

theta =

    7.0000
    9.0000
   -8.0000
    9.0000
    3.0000
   -8.1884
   -5.0422
    0.9256
    9.9980
    9.8360
   -6.9891
   10.0000
   10.0009
    0.0001
    6.0115
  -11.6130
   -2.4981
    7.1757
    5.9428
    8.2170
```

**Fig 6.7 Sample weights for constructed neural network**

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>> accuracy

accuracy =

    78.7879

fx >> |
```

**Fig 6.8 Accuracy for the feature selected data**

Here by using feature selection we are decrease the time complexity and space complexity.

#### **6.1.1.1.1. Size reduction:**

Size of the Original diabetes dataset by considering all 9 features is 24KB. Diabetes dataset size after applying the feature selection is 15KB as the features are reduced to five. Here the size is reduced by 62%.

Iteration	No. of features selected	Selected Attributes
1	5	1,2,3,5,8
2	6	1,2,3,4,5,8
3	6	1,2,3,5,6,8

**Table 6.1 Iterations performed for diabetes dataset**

#### 6.1.1.1.2 Time Reduction:

Time for building the model using the reduced feature set (5 features) obtained by Neural Networks is 14.6447seconds.

Time for building the model using the original feature set (8 features) obtained by Neural Networks is 114.255 seconds. Here the time is reduced to 12.8%

Accuracy obtained for building the original diabetes feature set (8 features) obtained by Artificial Neural Networks is 68.7479 %.

Accuracy obtained for building the optimum diabetes feature set (5 features) obtained by Artificial Neural Networks is 67.901%.

#### 6.1.2.For thyroid:

##### 6.1.2.1Fuzzification :

Thyroid Dataset before Fuzzification :

41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.3	1	2.5	1	125	1	1.14	1	109	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4.1	1	2	1	102	1	0.96	1	108	1	1
46	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.98	1	1.9	1	109	1	0.91	1	120	1	1
70	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.16	1	1.9	1	175	1	0.96	1	108	1	1
70	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.72	1	1.2	1	61	1	0.87	1	70	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.03	1	1.9	1	183	1	1.3	1	141	1	1
59	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.4	1	1.9	1	72	1	0.92	1	78	1	1
80	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2.2	1	0.6	1	80	1	0.7	1	115	1	1
66	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.6	1	2.2	1	123	1	0.93	1	132	1	1
68	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2.4	1	1.6	1	83	1	0.89	1	93	1	1
84	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.1	1	2.2	1	115	1	0.95	1	121	1	1
67	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.03	1	1.9	1	152	1	0.99	1	153	1	1
71	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.03	1	3.8	1	171	1	1.13	1	151	1	1
59	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2.8	1	1.7	1	97	1	0.91	1	107	1	1
28	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3.3	1	1.8	1	109	1	0.91	1	119	1	1
65	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.2	1	1.9	1	99	1	1.14	1	87	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.2	1	1.8	1	70	1	0.86	1	81	1	1
63	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.5	1	1.2	1	117	1	0.96	1	121	1	1
80	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	1	1.6	1	99	1	0.95	1	104	1	1
28	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2.1	1	2.6	1	121	1	0.94	1	130	1	1

**Fig 6.9 Dataset of thyroid before fuzzification**

## Thyroid Dataset after Fuzzification :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab
0.7119	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0518	1	0.8357	1	0.7877	1	0.4528	1	0.8065	1 yes
0.9536	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.1638	1	0.8873	1	0.8471	1	0.5559	1	0.8093	1 yes
0.6447	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.039	1	0.8976	1	0.829	1	0.5846	1	0.7755	1 yes
0.3224	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0062	1	0.8976	1	0.6585	1	0.5559	1	0.8093	1 yes
0.3224	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0286	1	0.9639	1	0.9529	1	0.6075	1	0.9165	1 yes
0.9165	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001	1	0.8976	1	0.6379	1	0.3611	1	0.7163	1 yes
0.4701	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0558	1	0.8976	1	0.9245	1	0.5789	1	0.8939	1 yes
0.188	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0878	1	0.6411	1	0.9039	1	0.7049	1	0.7896	1 yes
0.3761	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0238	1	0.8667	1	0.7928	1	0.5731	0	0.7416	1 yes
0.3492	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0958	1	0.9286	1	0.8961	1	0.596	0	0.8516	1 yes
0.1343	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0438	1	0.8667	1	0.8135	1	0.5617	0	0.7726	1 yes
0.3627	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001	1	0.8976	1	0.7179	1	0.5387	0	0.6824	1 yes
0.3089	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.001	1	0.7016	1	0.6689	1	0.4585	0	0.6881	1 yes
0.4701	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.1118	1	0.9183	0	0.86	1	0.5846	0	0.8121	1 yes
0.8865	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.1318	1	0.908	0	0.829	1	0.5846	0	0.7783	1 yes
0.3895	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4739	1	0.8976	0	0.8548	1	0.4528	0	0.8685	1 yes
0.6984	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0478	1	0.908	0	0.9237	1	0.6132	0	0.8854	1 yes
0.4164	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.0598	1	0.9639	0	0.8083	1	0.5559	1	0.7726	1 yes
0.188	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.2398	1	0.9286	0	0.8548	1	0.5617	1	0.8206	1 yes

**Fig 6.10 Dataset for thyroid after fuzzification**

### 6.1.2.2. Genetic feature selection for thyroid dataset :

Optimum features to thyroid dataset obtained by Genetic Algorithm :

- Age
- On thyroxine
- Query on thyroxine
- On antithyroid medication
- Sick
- Thyroid Surgery
- I131 treatment
- Query hypothyroid
- Query hyperthyroid
- Psych
- TSH
- T3
- TT4 measured



0.4970	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.4029	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.5910	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.4432	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.2149	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.3358	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.7656	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.7925	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.3492	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.5238	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.6984	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.7656	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.9268	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.3492	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.1209	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.3492	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.7790	0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.6581	0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.3089	0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0
0.1477	0	1.0000	1.0000	1.0000	1.0000	0	1.0000	1.0000
0.7925	0	1.0000	1.0000	1.0000	1.0000	0	1.0000	1.0000

**Fig 6.11 Dataset after feature selection**

### **6.1.2.3.Artificial neural network for thyroid dataset:**

Using the 13 optimum features obtained from Genetic Algorithm we can predict the presence of thyroid with 89.6 % accuracy with neural networks.

```

253
850
254
851
255
>> accuracy
accuracy =
93.0094
fx >>

```

**Fig 6.12 Accuracy for thyroid dataset after feature selection**

```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

249
846
250
847
251
848
252
849
253
850
254
851
255

>> theta=ann1(TrainData,9);
Elapsed time is 25.785659 seconds.
fx >>

```

**Fig 6.13 Time complexity for feature selected dataset**

#### **6.1.2.3.1 Size reduction:**

Size of the Original thyroid dataset by considering all 27 features is 57KB. Thyroid dataset size after applying the feature selection is 38KB as the features are reduced to thirteen. Here the size is reduced by 66.7%.

Iteration	No. of features selected	Selected Attributes
1	15	1,4,5,6,10,14,15,16,18,19,20,22,23,26,27
2	17	4,5,6,7,8,9,14,16,17,18,19,20,21,22,23,25,27
3	15	1,3,4,5,6,13,15,16,18,20,21,22,24,25,26
4	13	1,3,4,5,6,8,9,10,11,16,18,20,21

**Table 6.2 Iterations performed for thyroid dataset**

### **6.1.2.3.2 Time Reduction:**

Time for building the model using the reduced feature set (5 features) obtained by Neural Networks is 14.6447seconds.

Time for building the model using the original feature set (8 features) obtained by Neural Networks is 43.9 seconds. Here the time is reduced to 12.8%.

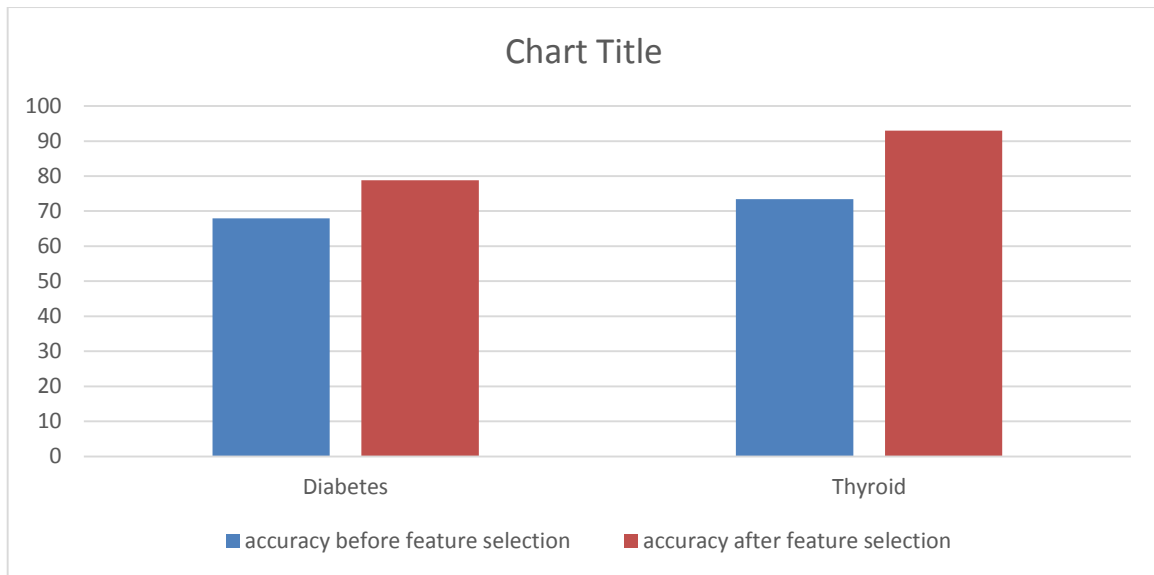
### **6.1.3 Summarisation of proposed system:**

Disease	Input perceptrons	Hidden perceptrons	Output perceptrons	Accuracy	Time	Feature selection
Diabetes	9	7	1	67.9104	14.644	No
Diabetes	5	7	1	68.6542	114.225	Yes
Diabetes	5	9	1	78.7879	15.391	Yes

**Table 6.3: Comparisons of the different Classification approaches of Diabetes dataset**

Disease	Input perceptrons	Hidden perceptrons	Output perceptrons	Accuracy	Time	Feature selection
Thyroid	27	8	1	73.4523	43.0309	No
Thyroid	13	8	1	89.6094	25.7856	Yes
Thyroid	13	14	1	93.01	44.7596	Yes

**Table 6.4: Comparisons of the different Classification approaches of Thyroid dataset**



**Fig 6.14 Graphical analysis for diabetes and thyroid before and after feature selection**

## ***6.2 Screenshots For Graphical User Interface For Medical Diagnosis:***

Using the optimum features obtained by genetic algorithm and fuzzification techniques, a model is built for diagnosing the thyroid and diabetes using the optimum features.

Graphical User interface is built for this model using the matlab GUI. The user interface will accepts the user input for the reduced feature set and it identify presence of diabetes and thyroid using the learning model inside the user interface.

User interface for diagnosing the diabetes and hepatitis using the reduced feature set obtained by the genetic algorithm is shown below.

**Diabetes**

Pregnancies

Glucose

Blood Pressure

Insulin

Age

**Thyroid**

Age

Thyroxine

Query on Thyroxine

Anti Thyroid

Sick

Thyroid

T131

Hypothyro

Hyperthyr

Psych

TSH

T3

TT4

**Fig 6.15 Screenshot of graphical user interface**

#### **6.2.1 User interface for diagnosing the diabetes :**

The user has to enter his test results in the user interface for knowing presence of hepatitis. If the user presses the submit button without entering the details, he will get the following message.

**Diabetes**

Pregnancies

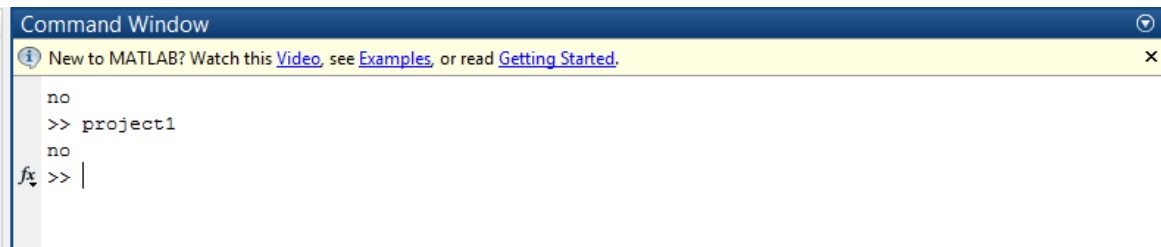
Glucose

Blood Pressure

Insulin

Age

**Fig 6.16 GUI For Entering Values in Diabetes**



**Fig 6.17 Output for Diabetes**

### User interface for thyroid:

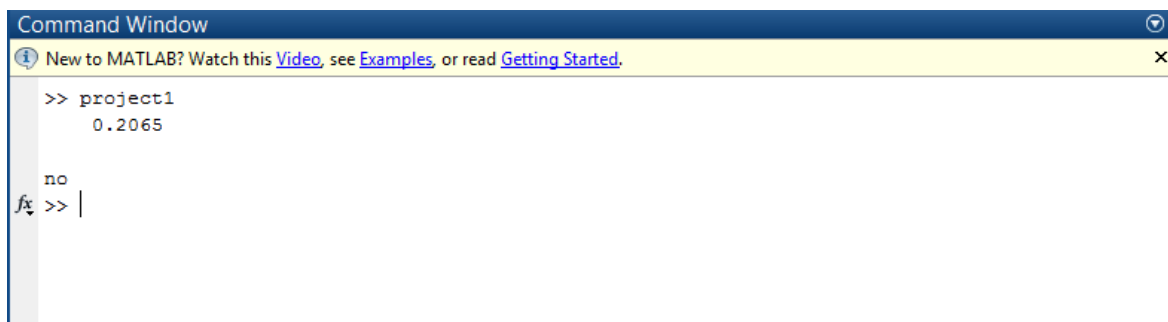
The user has to enter his test results in the user interface for knowing presence of hepatitis. If the user presses the submit button without entering the details, he will get the following message.

Panel

**Thyroid**

Age	<input type="text" value="0.71187"/>	Hypothyroi	<input type="text" value="1"/>
Thyroxine	<input type="text" value="1"/>	Hyperthyr	<input type="text" value="1"/>
Query on Thyroxine	<input type="text" value="1"/>	Psych	<input type="text" value="0.051808"/>
Anti Thyroid	<input type="text" value="1"/>	TSH	<input type="text" value="0.83573"/>
Sick	<input type="text" value="1"/>	T3	<input type="text" value="1"/>
Thyroid	<input type="text" value="1"/>	TT4	<input type="text" value="1"/>
T131	<input type="text" value="1"/>		

**Fig: 6.18 GUI for Diagnosing Thyroid**



**Fig: 6.18 Output presenting the diagnosis of Thyroid**

## **7.CONCLUSION**

This work presented a fuzzification method followed by feature selection method based on genetic algorithm. The results shown that a reduced number of features can achieve classification accuracy superior to that of using the full set of features. A comparison is also done between different datasets and between the outputs of data before and after feature selection. The results shows that after feature selection there is an great change in accuracy and time complexity also decreased. Here we considered diabetes and thyroid datasets containing of 9 attributes and 768 instances and 27 attributes and 851 instances respectively. And then applied genetic algorithm which resulted in a feature subset of 5 attributes and 13 attributes for diabetes and thyroid dataset respectively.

## **8. FUTURE WORK**

This work can be further enhanced for future work, to investigate alternative mechanisms to explore more classification techniques and improve the working of the neural networks and more machine learning techniques to solve problems of soft computing i.e., solving the NP-Hard problem. It can also further enhanced by finding the better fitness functions to obtain more accurate features.



## 9. REFERENCES

- [1]Aghdani,M.,GhasemAghaee,N.,Basiri,E.,(2009).Text feature selection using ant colony optimization. Expert System Application.
- [2] A.M. Natarajan (2011), artificial bee colonyalgorithm integrated with fuzzy c-mean operator for data clustering.
- [3]Andreas, W. (2003). The Shortest Common Super sequence Problem. DOI=<http://www.update.uu.se/~shikaree/Westling>
- [4]A. Singh, A.S. Baghel (2008), New meta heuristic approaches for the leaf-constrained minimum spanning tree problem, Asia-Pacific Journal of Operational Research.
- [5]BanharnsakunA,Achalakul,T.,Sirinaovakul,B.,(2011).Thebest-so farselection Artificial Bee Colony Algorithm applications for Soft Computing.
- [6] Celal Oz Turk In this study, an optimization method based on the swarm intelligence algorithm is proposed for the purpose of clustering.
- [7] D Karaboga, B Akay, (2012) A modified artificial bee colony algorithm for realparameter optimization. Inf. Sci. 192,120-142.
- [8] D.Karaboga, (2005). An idea based on honey bee swarm for numerical optimization, Technical report TR06, Erciyes University,Engineering Faculty, Computer Engineering Department.
- [9]Mario A Garcia, AmitJ.Gandhi, Tinu Singh, Leo Duarte, RuiShen, MaruthiDantu, Steve Ponder, and Hilda Ramirez ,*An Expert SystemIn diabetes*, JCSC pp.825-3478, March 2001.
- [10]H.Yalçın, and ,S. Taşdemir"*Fuzzy Expert System Approach for Determination of LinolenicAcid Content of Eggs Obtained from Hens by Dietary Flaxseed*" *Food Sci Tech Int.* 26 September 2006".
- [11]Albert P. C. Chan, Daniel W. M. Chan, M.ASCE, and John F. Y. Yeung " Overview of the Application of “*Fuzzy Techniques*” in *Construction Management Research* "Journal of construction Engineering and Management,Nov 2009
- [12] PetricǎVizureanu ”,*Expertsystems*,” published by intech,2010.
- [13]M.Kalpana, Dr. A.V Senthil Kumar, *Fuzzy Expert System for Diabetes using FuzzyVerdict Mechanism*, Int.J.
- [14]Advanced Networking and ApplicationsVolume: 03, Issue: 02, Pages:1128-1134 (2011)

- [15] S. Sa'di, A. Maleki, R. Hashemi, Z. Panbechi, K. Chalabi, "Comparison of Data Mining Algorithms in the Diagnosis of Type Ii Diabetes", *International Journal on Computational Science & Applications (IJCSA)*, Vol.5, No.5,(2015), pp. 1-12.
- [16]A. Al-Rofiyee, M. Al-Nowiser, N. Al-Mufadi, M. A. AL-Hagery, "using prediction methods in data mining for diabetes diagnosis", *posters*, (2014).
- [17]Yung-Keun Kwon and Byungo-Ro Moon, "A Hybrid Neuro-Genetic Approach for Stock Forecasting", *IEEE Transactions on Neural Networks*,Vol.18, No.3, pp.851-864, 2007.
- [18]Sapna, S., A. Tamilarasi, and M. Pravin Kumar. "Backpropagation learning algorithm based on Levenberg Marquardt Algorithm." *Comp Sci Inform Technol (CS and IT)* vol. 2 (2012): 393-398.
- [19]Amma, NG Bhuvaneswari. "Cardiovascular disease prediction system using genetic algorithm and neural network." *International Conference on Computing, Communication and Applications*. IEEE, 2012
- [20]Ahmad, Fadzil, et al. "Intelligent medical disease diagnosis using improved hybrid genetic algorithm-multilayer perceptron network." *Journal of medical systems*, 37.2, 1-8, 2013 .
- [21]Anbarasi et. al, Iyengar, "Enhanced Prediction of Heart Disease with Feature Subset Selection using Genetic Algorithm," *International Journal of Engineering Science and Technology*, 2(10), 5370-5376, 2010