SWARM OPTIMIZED FUZZY REASONING MODEL (SOFRM) FOR DIABETES DIAGNOSIS

By

ATIQ UR REHMAN

2010-NUST-MS PhD-ComE-01

Submitted to the Department of Computer Engineering

In partial fulfillment of the requirements for the degree of

Master of Science in Computer Engineering

Thesis Supervisor

Dr.Aasia Khanum

College of Electrical & Mechanical Engineering

National University of Sciences and Technology 2013

DECLARATION

We hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification of this of any other university or other institute of learning. If any act of plagiarism found, we are fully responsible for every disciplinary action taken against us depending upon the seriousness of the proven offence, even the cancellation of our degree.

COPYRIGHT STATEMENT

This Thesis is Dedicated to My Parents and Teachers

ACKNOWLEDGMENTS

First of all, I am really very thankful to my advisor Dr. Aasia Khanam for the guidance, support and motivation she has provided me during this thesis project. It was her knowledge and guideline which enabled me to carry out this research. She also supervised writing of thesis in a professional and helpful manner.

I would also like to thank other members of my thesis committee Dr. Arslan Shaukat, Dr. Saad Rehman and Dr. Shahzad Khalid for their comments and observations about my work.

Finally, I am really grateful to my family for the support they provided me especially during this phase of my studies.

Atiq ur Rehman

College of E&ME, NUST

April 2013

ABSTRACT

Early diagnosis of Diabetes is important as it reduces the chances of related complications to arise. Swarm Intelligence is being widely used for medical diagnostic purposes. Many classifiers are being optimized by Swarm Intelligence techniques to reduce the cumbersome procedure of defining complex rules and frameworks. Cuckoo Search is a recently developed algorithm which uses the concept of Swarm Intelligence. Cuckoo Search mimics the brooding behavior of some Cuckoo species. Cuckoo Search is enhanced by Levy Flights which follows Levy distribution.

Fuzzy Reasoning Model represents some expert knowledge via simple linguistic rules. This makes the system more understandable. Employing Fuzzy Reasoning Model to represent a system for diabetes diagnosis will help an expert who has to consider a large number of factors before making a decision. In this way human error will be minimized.

Objective of this thesis is to employ Fuzzy Reasoning Model for Diabetes Diagnosis. Cuckoo Search has been used to optimize fuzzy model for better results. Pima Indians Diabetes Data set has been used to evaluate the accuracy of proposed classifier.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1. Background

Diabetes mellitus is a group of metabolic diseases indicated by blood sugar level higher than the normal. Diabetes is developed by shortage of insulin, decreased ability to use insulin or both. Insulin is a hormone produced by pancreas which regulates blood sugar level [1].

There are three types of diabetes:

I. In type 1 Diabetes mellitus (T1DM) enough insulin is not produced because of the loss of beta cells in pancreas. These beta cells are responsible for insulin production. T1DM needs insulin injection to keep blood sugar within normal range.

II. In type 2 Diabetes mellitus insulin production is normal but body fails to utilize it for glucose conversion into glycogen.

III. In the third type called as gestational diabetes a pregnant woman shows symptoms of diabetes with no previous record of diabetes.Type1 and Type2.

Diabetes mellitus can strike children and adults alike. If people are unaware of symptoms or ignore them deliberately blood sugar level may reach a dangerous level. This situation leads to hospitalization and may result in early death as well [2].

1.2. Condition of Diabetes World-wide

An estimated 347 million people of this planet are bearing the effects of diabetes. Low income and middle income people are prime targets of this disease as 80% of deaths from diabetes occur in developing and under developed countries. According to WHO diabetes will be the 7th biggest killer disease by 2030 [34].

In order to demonstrate the worsening diabetes situation on earth fig1.1 is really very helpful.

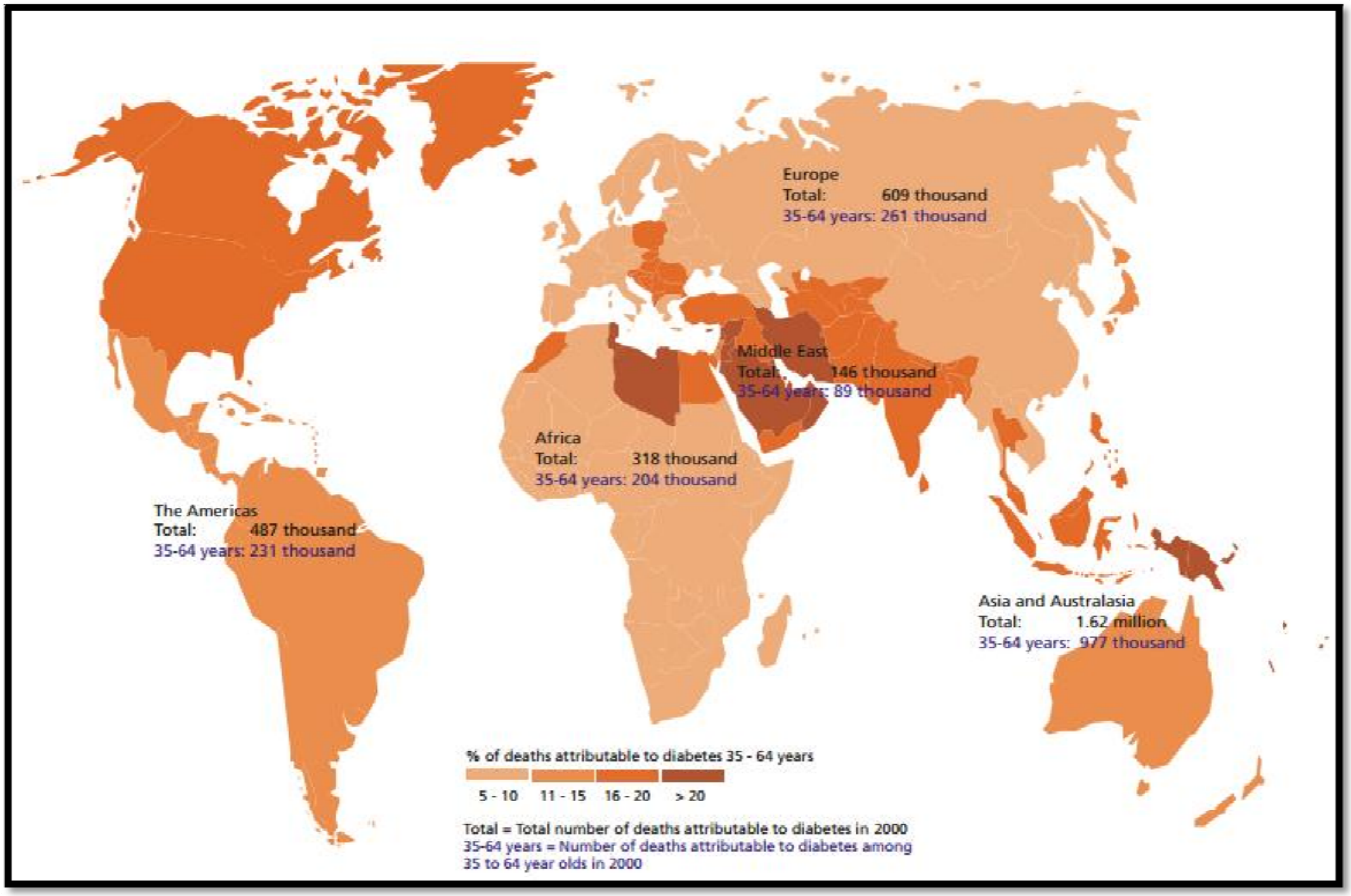
Europe
Total: 609 thousand
35-64 years: 261 thousand
Middle East
Total: 146 thousand
35-64 years: 89 thousand
Africa
Total: 318 thousand
35-64 years: 204 thousand
The Americas
Total: 487 thousand
35-64 years: 231 thousand
Asia and Australasia
Total: 1.62 million
35-64 years: 977 thousand
% of deaths attributable to diabetes 35 - 64 years
5 - 10
11 - 15
16 - 20
> 20
Total = Total number of deaths attributable to diabetes in 2000
35-64 years = Number of deaths attributable to diabetes among 35 to 64 year olds in 2000

Figure 1.1 Deaths attributable to diabetes in year 2000 [34]

According fig 1.1 Middle Eastern countries and Sudan had highest percentage deaths due to diabetes in the year 2000. South Asia, Central Asia and USA stood second while USA, Latin America and Australia had 3rd position. Rest of the world (China, Russia, Europe and most of Africa) had lowest percentages of death because of diabetes. But lower percentages do not mean lower numbers for populous countries like India and China. Currently world's top 10 diabetic countries (w.r.t. number of people affected) include India, China, USA, Indonesia, Japan, Pakistan, Russia, Brazil, Italy and Bangladesh. So our Homeland and two of our close neighbors are in this list which means an alarming situation in this region of the world [34].

Unfortunately I was unable to find any open source diabetes dataset for Pakistan or any other South Asian country. Only available open source dataset is based upon medical tests conducted on a section of American population [28]. This dataset has been explained in section 4.1.

In United States about 25.8 million people (8.3 %) have diabetes. Out of these 25.8 million about 7 million have undiagnosed diabetes. If this trend continues by 2050 about 1/3 of American population will be diabetic [1].

1.3. Importance of Early Diabetes Diagnosis

In most cases when diabetes is diagnosed it is found that patient had already developed chronic complications [3]. Hypoglycemia, diabetic ketoacidosis and hyperosmolar nonketotic state are common complications related to diabetes. Damage to arteries result in long term complications. Diabetes increases the risk of cardiovascular disease [17]. Diabetes is also responsible for microvascular complications (damage to small blood vessels). Diabetic retinopathy affects blood vessels in retina. Such a patient has an increased chance of reduced vision and potential blindness. Diabetic nephropathy damages kidneys. Patient loses protein in the urine, ultimately leading him to chronic kidney disease. Impact of diabetes on nervous system (neuropathy) causes Paresthesia (numbness) and pain in the feet. Neuropathy together with vascular disease causes foot problems which might need amputation in severe cases.

In a research conducted at Funagata Machi, Yamagata Prefecture (Japan) ratio of newly diagnosed diabetes patients with retinopathy and microalbuminuria was found to be at 7% and 31% respectively [4]. Harris et al [5] found that of the newly diagnosed diabetes patients 7% already had diabetes and were diabetic for about 4-7 years.

Thus not only diabetes is increasing but complications related to it are also getting worse. Early diagnosis of diabetes will help a patient in adopting a proper life style to tackle this diabetes related problems. This life style includes appropriate changes in diet, regular exercise and proper medication. Special attention should be given to health problems as they make effects of diabetes worse. These include high blood pressure, high cholesterol level, obesity, smoking and lack of exercise. Therefore early diagnosis of diabetes is needed not only to manage diabetes but also to also handle the complications related to it.

1.4. Introduction to Swarm Intelligence

Term “Swarm Intelligence” was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [6]. Swarm intelligence is the behavior of a community where though each individual cannot itself behave intelligently, it interacts with its nearest neighbors via minimal communication and ultimately the entire swarm/community behaves intelligently to achieve a certain goal. An example of this behavior is an ant colony. An ant initially wanders unintelligently to find a food resource. Upon finding the food source, ant returns to its colony lying down Pheromone trails. If another ant finds such a path it also takes the same route to the food source. On return it will also lay down Pheromone trail, enhancing the attractiveness of this path for other ants.

But as the time passes on Pheromone evaporates. Evaporation is more for longer paths resulting in smaller paths becoming more attractive for ants. Thus an ant colony (at the community level) behaves intelligently in search of food despite the fact that each ant at the individual level was unable to do so (in an intelligent way).

Swarm intelligence was declared a formal research field by Kazadi in 2000 [7]. According to Kazadi Swarm Engineering is a two step process: generation of swarm condition and modification of swarm behavior to satisfy the swarm condition [8].

Traditionally top-down approach is used for finding solutions of a problem but Swarm Intelligence is a bottom-up approach. In bottom up approach agent ignorance is useful as it uses local information to ultimately achieve global goal (optima). Thus chaos turns into a complex order. While in top-down approach agent ignorance is harmful and leads order into a chaos [8].

1.5. Advantages Of Swarm Intelligence

- **Computational efficiency:** availability of multiple processors (particles) in a swarm provides better computational power [8].
- **Reliability:** even if a single agent fails there is no single point of failure for the entire system. It is because of decentralized control, shared sensor data, redundancy because of large number of agents and simplicity of the individual agents
- **Scalability:** agents can be added or removed to the system without making changes in the program structure.

- **Self-organizing:** a swarm solves a problem without the need of a centralized command.
- **Economical:** such systems have a simple design so they need less hardware and are more economical than the traditional systems.

1.6. Example Algorithms of Swarm Intelligence

A brief overview of some important example algorithms of swarm intelligence is given below:

1.6.1. Particle Swarm Optimization

In Particle Swarm Optimization all the particles are initialized randomly in an n-dimensional space. All these particles have initial velocities and communication channels between these particles. After an iteration, particles change their positions according to the computed velocities evaluated using problem-specific fitness criterion. Ultimately all of the particles are accelerated to the particles which have better fitness values [9].

1.6.2. Intelligent Water Drops

A natural river has a lot many options of paths to flow from source to destination. The optimal or near optimal path from all of these available options is found using interactions of water drops with themselves and with the riverbeds. Intelligent Water Drops (IWD) algorithm is inspired by the above mentioned behavior of the natural water drops [10]. In IWD water drops use co-operation among them to establish a path which has minimum soil on its links.

Applications of IWD include finding the solution of n-queen puzzle and travelling salesman problem (TSP).

1.6.3. Firefly Algorithm

Firefly Algorithm (FA) was proposed by Xin-She Yang [11]. This algorithm uses fireflies' flashing behavior. Light intensity of a firefly decides it's attractiveness to other fireflies. This attractiveness serves the criterion for division of a group of fireflies into sub-groups. Each subgroup is associated to its own local swarm. This feature of FA makes it useful for optimizing multimodal problems.

1.6.4. Cuckoo Search

In Cuckoo Search (CS) algorithm brooding behavior of some Cuckoo species is used for problem optimization. Some Cuckoo species lay their eggs in other species' nests. Such parasitic species use various techniques to avoid the discovery of alien eggs by the host species. Over various generations, parasitic Cuckoo species try to discover set of host nests which provide them with maximum reproduction rate [14]. This strategy is inspiration of Cuckoo Search (CS) algorithm. CS tries to minimize/maximize a fitness function by discovering most suitable solutions (nests) over a certain number of iterations (generations).

Recent studies have shown that CS performs better than PSO and ant bee algorithm for numerical optimization problems [15].

1.6.5. Krill Herd Algorithm

Krill Herd algorithm is one of the latest inclusions in the field of swarm intelligence. It has been proposed by Gandomi and Alavi [16]. This algorithm simulates the herding behavior of Krill. In this algorithm minimum distance of each Krill individual from food source and from highest density area of Krill herd is used as objective function. An important enhancement of Krill Herd algorithm is that time interval is the only parameter needed to be fine tuned. It is because of the fact that Krill Herd algorithm carefully simulates Krill Behavior and coefficient values are found using real world empirical studies.

1.7. Fuzzy Logic

1.7.1. Background

In 1965 Zadeh [12] introduced fuzzy logic to approximate the behavior of a system which is too complex to be described in the form of a precise mathematical model. In contrast to logical calculus which takes only two values (0 and 1) Fuzzy Logic is a many-valued logic. As an example we can say that if traditional logic deals with truth and false, fuzzy logic deals with values which range from complete truth to partial truth, partial false and ultimately to complete false.

Fuzzy logic is being widely used in different fields of research like machine learning, medical diagnostic, control theory etc.

1.7.2. Example of Fuzzy Logic

Let's suppose we have to find the risk value associated with starting a new business. We have two inputs: country's economic conditions and company's investment amount. We have to develop a fuzzy model for evaluation of risk associated with starting this business when economic conditions and investment amount will be represented in percentage (0-100%) [13].

1.7.2.1. Defining Fuzzy Sets And Their Member-Ship Functions:

Fuzzy set for economic conditions are: inadequate, satisfactory and good. Fuzzy set for investment amount are: small and large and finally fuzzy sets for output business risk are: Low, Normal and High. Membership function for economic condition μ_{eco}, investment μ_{inv} and business risk $\mu_{\text{risk}}$are shownin the diagrams below:

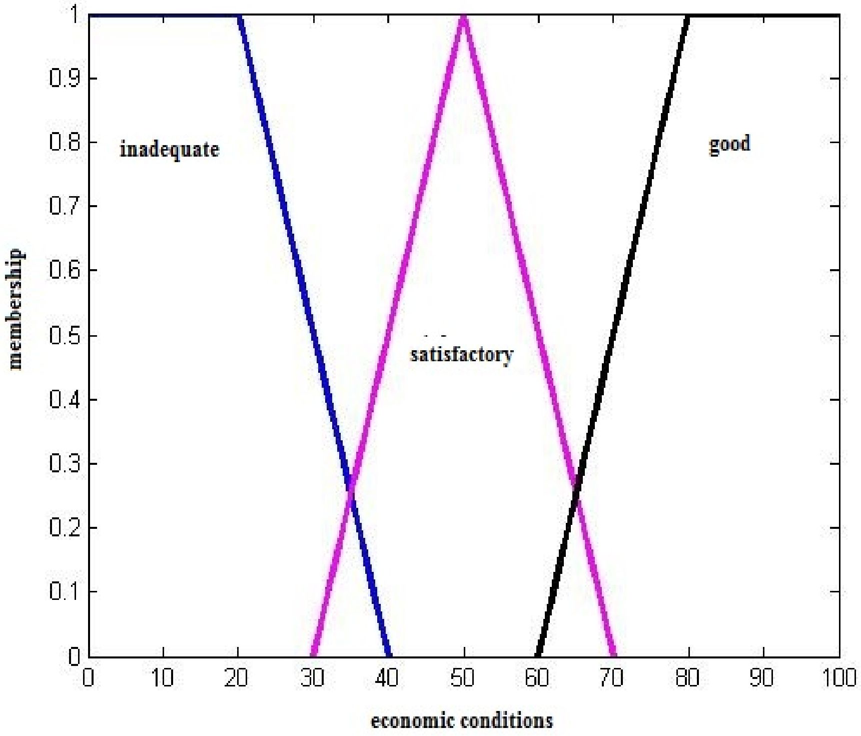

Figure 2.2 Membership function for input attribute 'economic condition' [13]

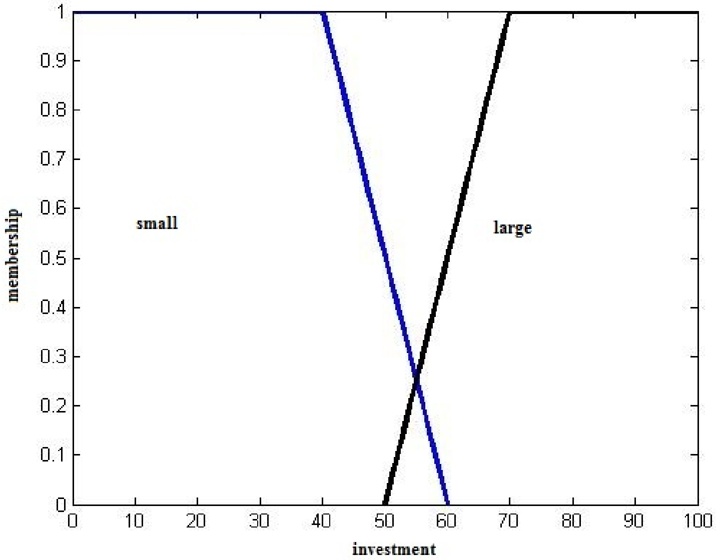

Figure 1.3 Membership function for input attribute 'investment amount' [13]

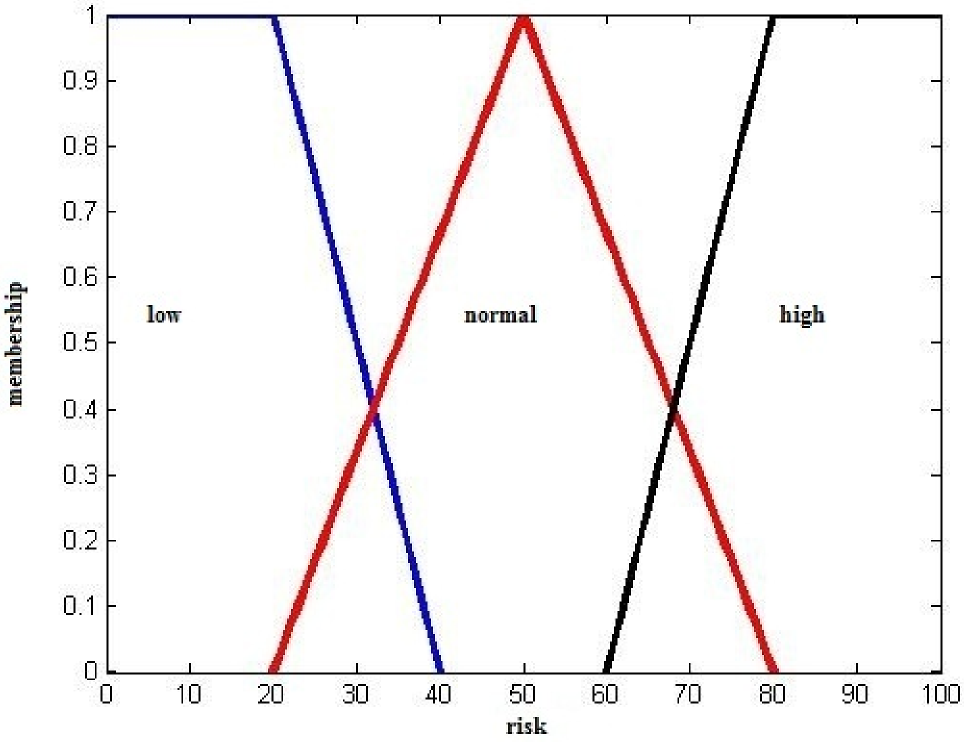

Figure 1.4 Membership function for output 'risk associated to starting business' [13]

1.7.2.2. Defining Fuzzy Logic Rules

Following rules are defined for the given example problem:

1 **If** (economic conditions = good) **Or** (investment = small) **Then**(risk =low)

2 **If** (economic conditions = satisfactory) **And** (investment = large) **Then** (risk = normal)

3 **If** (economic conditions = inadequate) **Then** (risk = high)

1.7.2.3. Evaluating Rules

Each rule will be computed for given input values. Let's suppose that economic condition = 40 % and investment = 55 % then,

I. For the first rule:

$$\mu_{risk=low} = \mu_{eco=good}(.4) + \mu_{inv=small}(.55) - \mu_{eco=good}(.4) * \mu_{inv=small}(.55) \quad (1.1)$$

$$\mu_{risk=low} = 0 + .3 - 0 * .3 = .3 \quad (1.2)$$

II. For the second rule:

$$\mu_{risk=normal} = \mu_{eco=satisfactory}(.4) * \mu_{inv=large}(.55) \quad (1.3)$$

$$\mu_{risk=normal} = 0.3 * .3 = .09 \quad (1.4)$$

III. For the third rule:

$$\mu_{risk=high} = \mu_{eco=inadequate}(.4) \quad (1.5)$$

$$\mu_{risk=high} = 0 \quad (1.6)$$

1.7.2.4. Defuzzification

Defuzzification i.e. to infer results from the memberships of different fuzzy sets of risk can be done by many different methods like centroid method, smallest value of maximum, largest value of maximum, mean of maximum and bisector method etc. We will use centroid method (center of gravity of area under curve):

$$\text{Center of gravity} = \frac{\sum_{\mathrm{i=a}}^{\mathrm{b}} \mu_{\mathrm{A}}(I)i}{\sum_{\mathrm{i=a}}^{\mathrm{b}} \mu_{\mathrm{A}}(I)} \quad (1.7)$$

In our example:

$$\text{Risk of starting business} = \frac{(0+10+20+30)*.3 + (40+50+60+70)*.09 + (80+90+100)*0}{.3*4+.09*4+0*3} \quad (1.8)$$

$$\text{Risk of starting business} = 24.23\% \quad (1.9)$$

Risk factor of starting a new business in the given input conditions is found to be at 49.73%. This final result is a non-fuzzy value found by using centroid method of defuzzification.

1.8. Fuzzy logic optimized by Swarm intelligence

It can be observed from the above example of fuzzy logic that a large number and variations of rules, outputs, fuzzy sets and membership functions of fuzzy sets are possible. It is not easy for an expert of a field to keep in mind all these variations and complexities. So Swarm Intelligence can be used to optimize fuzzy logic proposed for a given problem. In this way an expert’s job regarding decision making is made easier.

1.9. Thesis Organization

In the first chapter concepts of Swarm intelligence and Fuzzy logic have been discussed. 2nd chapter will focus on to the previous work done in the Medical Diagnostic Machine Learning using Swarm and Fuzzy techniques. Afterwards, in the 3rd chapter proposed classifier will be elaborated. Then, 4th chapter will focus the implementation of this classifier plus its experimental results. Finally last chapter will be consisted of conclusions and future aspects for related research.

CHAPTER 2

Literature Review

In the past few years use of Swarm Intelligence to optimize medical diagnostic systems is increasing. Some of these are discussed as under:

2.1. Fuzzy Logic Optimized By Ant Colony Optimization

Fathi et al [18] used Ant Colony Optimization to optimize Fuzzy Logic for rule selection of diabetes diagnosis. They named the system as Fuzzy Ant Colony Optimization for Diagnosis of Diabetes Disease (FADD). FADD proposed a new framework for learning fuzzy rules. This classifier showed better classification rate and more comprehensibility of fuzzy rules as compared to other contemporary methods. This implementation of Ant Colony Optimization paid more attention to cooperation than to the aspect of competition among ants. Strong fuzzy rules were evolved in FADD. Separate training was done for both the classes (diabetic and non-diabetic).

2.1.1. Algorithm of FADD

For each class *k*, main function calls a function named as FADD. FADD develops rules for the class and returns them to the main function. The main function discards the samples for class *k*. This process is repeated for all the classes in the dataset. Ultimately, a final set of rules is established which act as a classifier for diabetes. Algorithm for FADD function is described below:

```
Begin
    Set of learned rules is initialized as an empty set.
    While (stopping condition not satisfied)
        iteration = 0
        All of the cells in the pheromone table initialized to an equal value
        Create rule R with all terms = Don’t care
        Repeat
            iteration = iteration + 1
            Modify terms of created rule R according to max_change parameter
            Calculate classification rate of the rule R and update pheromone table
        Until (iteration>Number of Ant)
        If classification rate of Rule ‘R’ is above a certain threshold add this rule to the set of learned rules
    End While
    Return Set of learned rules
End
```

Figure 2.1 Pseudo code for Ant Colony Optimization

2.1.2. Fuzzy Sets

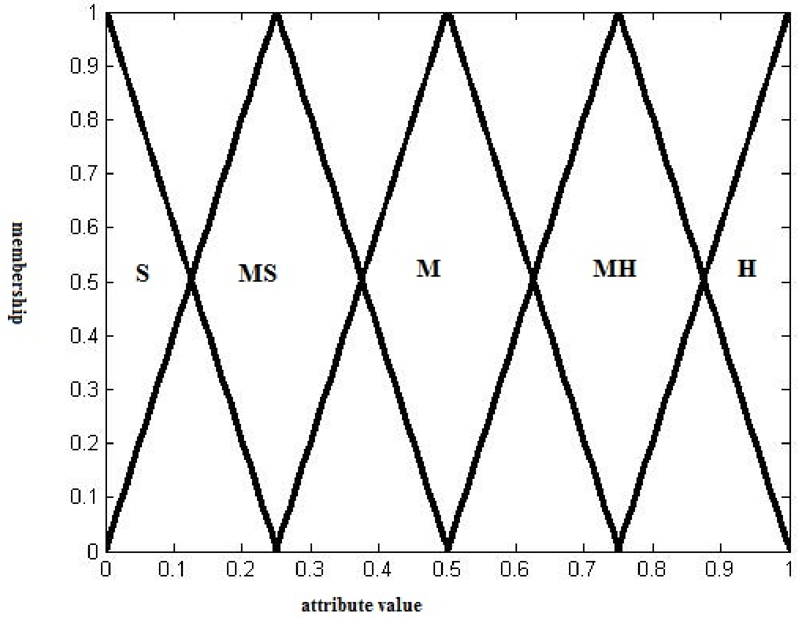

Figure 2.2 Membership function for any input attribute
S = small, MS = medium small, M = medium, MH = medium high, H = high [18]

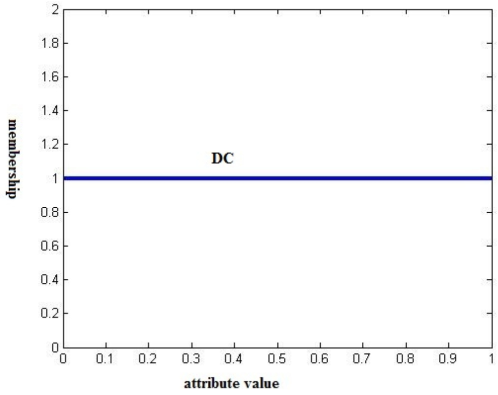

Figure 2.3 Membership function for any input attribute [18]

Fuzzy sets for are obtained by partitioning each attribute's value homogenously into triangular membership functions as shown in fig. 2.2 and 2.3. These fuzzy sets have not been tailored for simplicity and high performance. However, these membership functions of these sets can be tailored for use in other fuzzy logic based classifiers.

2.1.3. Pheromone Table Initialization

Before rule learning is started pheromone table is initialized for all the paths at an equal value by using following equation:

$$T_{i,j}(\text{iteration} = 0) = \frac{1}{\sum_{i=1}^{x} z_i} \tag{2.1}$$

'x' denotes number of attributes in the dataset while z_i represents the number of possible values in the domain of attribute 'x'.

2.1.4. Constructing Rules

In the first iteration of FADD all the terms in newly constructed rules are equal to don't care (DC). In the next iterations terms of generated rule are modified according to max_chage parameter. In the implementation used by Fathi et al [18] max_chage = 2. Chance of the term $T_{i,j}$to be modified in an iteration is given by:

$$P_{i,j} = \frac{T_{i,j}(\text{iteration}).\eta_{i,j}}{\sum_{i}^{x}\sum_{j}^{b_i} T_{i,j}(\text{iteration}).\eta_{i,j}} \quad ,\forall\, i\, \in I \tag{2.2}$$

Where,

$\eta_{i,j}$ = problem dependant heuristic, for DC it is equal to .5 else .1.

$T_{i,j}$ = value of pheromone on the path from I to j.

I = set of attributes which have not been used yet.

2.1.5. Computing Classification Rate

After modification of terms of a rule its quality for the given class is computed [18].

$$quality\ of\ rule\ Rj = \frac{TP}{TP+FN} . \frac{TN}{TN+FP} \quad (2.3)$$

In eq. 2.3, **TP** denotes number of instances in the training set of given class which are correctly classified by rule R_j, **FP** denotes number of instances training set of given class which are incorrectly classified by rule $R_{j,}$ **FN** denotes number of instances outside the training set of given class which are correctly classified by rule R_j, **TN** denotes number of instances outside the training set of given class which are incorrectly classified by rule R_j**.**

2.1.6. Pheromone Table Update

If quality calculated in eq. 2.3 has increased for a rule 'R', then amount of pheromone on the corresponding path$\mathrm{T}_{i,j}$ in the pheromone table is increased.

$$\mathrm{T}_{i,j}(\text{iteration} + 1) = \mathrm{T}_{i,j}(\text{iteration}) . \Delta \mathrm{Q}. \mathrm{C} \quad (2.4)$$

Where,

$\Delta\mathrm{Q}$ = change in quality before and after rule modification

C = weightage of $\Delta\mathrm{Q}$ in pheromone updation

Furthermore, Ant Colony Optimization algorithm needs pheromone of the paths not helpful in achieving goal to be decreased. This is done by dividing the amount of pheromone across each path by summation of pheromones along all the paths in the pheromone table.

2.2. Fuzzy Reasoning Model optimized by Hybrid Particle Swarm Optimization

Ling et al [19] used Fuzzy Reasoning Model optimized by Hybrid Particle Swarm Intelligence with Wavelet Mutation (HPSOWM) for hypoglycaemic detection. Their synthetic dataset had two parameters heart rate and corrected QT interval of ECG signal. They used an enhanced form of Particle Swarm Optimization (PSO) called as Hybrid Particle Swarm Optimization with Wavelet Mutation (HPSOWM). HPSOWM overcomes the drawback of trapping in local optima in PSO. Varying number of member-ship function used in Fuzzy Reasoning Model they enhanced sensitivity and specificity of their classifier.

2.2.1. Algorithm of HPSOWM

HPSOWM models the flocking behavior of birds. This algorithm constitutes a number of particles. Each of these particles tries to find the global optimum in the search space. Global optimum point is position of a particle which yields best results according the given problem.

```
Begin

t = 0

Initialize swarm S (t)

Evaluate initial fitness function f(S(t))

Initialize maximum velocity (Vmax) and minimum velocity (Vmin) of particles

Repeat (until stopping condition not satisfied)

t = t + 1

Compute velocity matrix v(t) of swarm

Update position matrix s (t) of swarm based upon computed velocity matrix

If (velocity of a particle >Vmax)
```

```
Set, velocity >= Vmax
Else
set,  velocity = Vmin
Apply wavelet mutation to the position of swarm particles
Again update position matrix s (t) of particles
Check fitness function f(S (t)) for the new (updated) swarm
End
```

Figure 2.4 Pseudo code for HPSOWM

In the pseudo code of HPSOWM S(t) represents swarm at the iteration number t. S (t) consists of particles which denoted by $s^p(t)$. Each particle $s^p(t)$ consists of multiple dimensions which are represented by $s_j^p(t)$. Here p = 1,2,… ,∂ and j = 1,2,… ,k. ∂ is total number of particles in the swarm and ‘k’ is total number of dimensions of each particle. Vmax and Vmin are used to select the resolution of area to be searched. Vmax is usually between .1 to .2 and Vmin is equal to – Vmax.

2.2.2. Mathematical Equations for Computation of Velocity and Position Vector of Particles

In order to compute velocity $v_j^p(t)$ corresponding to each particle $s_j^p(t)$ following equation is used:

$$v_j^p(t) = l\left(w.v_j^p(t-1) + \emptyset_1.r_1.\left(s_j^{\sim} - s_j^p(t-1)\right) + \emptyset_2.r_2.\left(s_j^{\wedge} - s_j^p(t-1)\right)\right) \quad (2.5)$$

Where, $s^{\sim}$ =[$s_1^{\sim}$,$s_2^{\sim}$,… ,$s_k^{\sim}$] and $s^{\wedge}$ = [$s_1^{\wedge}$,$s_2^{\wedge}$,… ,$s_k^{\wedge}$]. Best previous position of an individual particle is denoted as s~ while the position of best particle of all the particles in the swarm is represented by s^. Similarly, r1andr2 generate two random numbers between 0 and 1. $\emptyset_1$ and $\emptyset_2$ are acceleration constants. Parameter l is used to ensure that swarm optimization does not converge prematurely. Value of l is found using following equation.

$$l = \frac{2}{2-\emptyset-\sqrt{\emptyset^2-4\emptyset}} \tag{2.6}$$

Here, $\emptyset$ is summation of $\emptyset_1$ and $\emptyset_2$ and its value should be greater than 4.

Value of 'w' is set dynamically during the program execution by using following formula:

$$w = (w\_max) - (\frac{w\_\max - w\_min}{T} * t) \tag{2.7}$$

Here,w_max = upper limit of inertia weight, w_min = lower limit of inertia weight, t = number of current iteration and T = total number of iterations. In the implementation of HPSOWM for hypoglycaemia value of w_max is set at 1.2 and w_min is set at 0.2.

In the next step, velocity computed in eq. 2.5 is used to update position matrix of all the particles:

$$s_j^p(t-1) + v_j^p(t) \tag{2.8}$$

2.2.3. Applying Wavelet Mutation

In order to fine tune the particles of swarm wavelet mutation is used. Wavelet mutation avoids the chance of particles to be trapped in local optima. Chance that an individual can be mutated lies in the range 0 to 1. Let's suppose j-th dimension and p-th particles$s_j^p(t)$has been randomly selected for wavelet mutation. Value of $s_j^p(t)$ lies in the range [ρ_{min},ρ_{max}]. $\bar{s}_j^p(t)$ is the particle's dimension after wavelet mutation and is represented as:

$$\bar{s}_j^p(t) = \begin{cases} s_j^p(t) + \gamma * \left(\rho^{j}_{max} - s_j^p(t)\right), \gamma > 0 \\ s_j^p(t) + \gamma * \left(s_j^p(t) - \rho^{j}_{max}\right), \gamma \leq 0 \end{cases} \tag{2.9}$$

In eq. 2.9 value of γ is depends upon Morlet wavelet function:

$$\gamma = \omega_{a,0}(\emptyset) = \frac{\omega\left(\emptyset/_{a}\right)}{\sqrt{a}} = \frac{1}{\sqrt{a}} e^{-\frac{(\frac{\emptyset}{a})^2}{2}} \cos(5(\frac{\emptyset}{a})) \tag{2.10}$$

In order to enhance the performance of searching the global optima, value of 'a' in eq. 2.10 is increased to reduce the amplitude of ω.

$$mutated\ value\ of\ s_j^p(t) = \begin{cases} \rho_{max} \ , \gamma \text{ approaches } 1 \\ \rho_{min} \ , \gamma \text{ approaches } \ -1 \end{cases} \tag{2.11}$$

It is implied from eq. 2.11 that larger |γ| means larger search space and vice versa. 99 % energy of Morlet wavelength is contained in the interval [-2.5, 2.5] so value of ∅ should be generated randomly in the interval [-2.5, 2.5] × a. Furthermore, when value of t / T becomes large value of 'a' must also be large. It is because at large values of't' effect of mutation will hinder needed local search. Thus in order to make value of 'a' fluctuate monotonically with the parameter t / T following function is used:

$$a = e^{-\ln(g)*(1-t/T)^{\Psi}+\ln(g)} \tag{2.12}$$

Here, 'g' is the maximum value that 'a' can attain and ψ is the shape parameter of this monotonic function.

2.3. Cluster Analysis (Clustering)

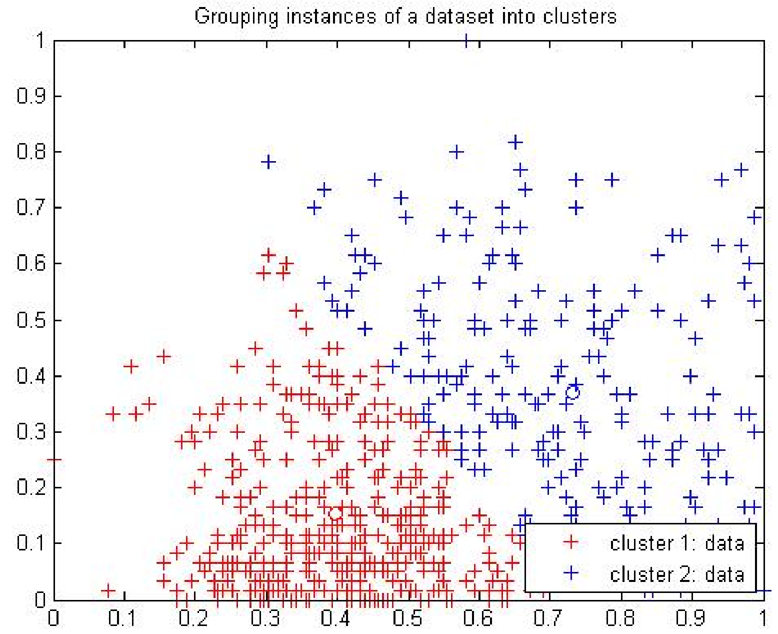

Figure 2.5 Example of clustering (Plotted using Matlab)

Clustering is the process of organizing objects into groups in such a way that objects in the same group (cluster) are more similar to each other as compared to the objects assigned to other groups [20]. Clustering is backbone of data mining. In the fields of image analysis, pattern recognition, machine learning and bioinformatics etc. clustering is being widely used for statistical analysis of data.

2.3.1. Clustering Types

Superficially, there are two types of clustering:

1) **Hard clustering:** each instance of dataset can belong to only one cluster.

2) **Soft clustering:** each instance of dataset belongs to all the clusters with certain of relationship. This type of clustering is also called as fuzzy clustering.

When we look into finer details of clustering there four types:

1) **Strict partitioning:** each instance is affiliated to only one cluster.

2) **Strict partitioning with outliers:** each instance is affiliated to only one cluster but there are certain instances which don't belong to any of the clusters and are called as 'outliers'.

3) **Hierarchal clustering:** each instance is a member of a cluster who is itself member child cluster of a parent cluster**.**

4) **Overlapped clustering:** it is a type of hard clustering where an instance of a dataset can belong to more than one cluster**.**

2.3.2. Models of Clustering

We need to understand clustering models in order to understand the difference in techniques and applications of different clustering algorithms.

1) **Connectivity models**: these models are based on connectivity distances.
2) **Centroid models**: these models use mean vector as a criterion for grouping objects.
3) **Distribution models**: statistical distributions like multivariate normal distributions are used to build these types of models.
4) **Density models**: clusters are defined as connected dense regions inside the data space of the given data set.
5) **Subspace models**: these types of models use simultaneous clustering of rows and columns of a dataset represented as a matrix. Subspace models generate bi-clusters i.e. a

cluster consists of a subset of rows which exhibit same sort of behavior as found in a subset of columns or vice-versa.

6) **Group models**: these models do not provide detailed justification of why objects were grouped together in a cluster. These simply provide groups as their output.

7) **Graph-based models**: this model uses clique (subgrpahs) i.e. subsets of nodes of a graph in which every two nodes are connected by an edge. These subsets of the graph act as clusters [21].

2.3.3. Usage of Clustering to Remove Noise in Pima Indians Diabetes Dataset

As presence of noise and missing values in the dataset deteriorates the accuracy of any classifier pre-processing dataset is an important step of classification. Balakrishnan et al [22] discusses missing values in Pima Indians Diabetes dataset. They also used k-means clustering for removing class outliers from the dataset. Removing anomalies will obviously improve results of classification.

2.3.4. K-means Clustering

K-means clustering is a simple technique of unsupervised learning. It is used to classify instances of a given data set into ‘k’ number of clusters. Each cluster has its own center called as centroid and each instance is assigned to a cluster whose centroid is nearest to the instance. To partition data into ‘k’ clusters, ‘k’ numbers of centroids are needed to be initialized in an intelligent way. Initial positions of centroids strongly affect the final results. It is a good practice to keep the centroid initial positions as far away from each other as possible.

2.3.5. Steps of Clustering

```
Begin
    Initialize 'k' number of clusters and assign all data points randomly to the cluster so that all clusters have almost same number of data points [23].
Repeat (until centroids are stable)
    Calculate the Euclidean distance of each data point to all of the centroids.
    Assign each data point to the cluster whose centroid's distance is minimum from the data point.
    Check if the centroids are stable i.e. during an iteration no data point changes its cluster as a result of calculation of its distance to all of the centroids.
End
```

Figure 2.6 Example of clustering

Euclidean distance is the ordinary distance between two points found using Pythagoras Theorem. Let's suppose there are two points 'n-dimensional' points **x** = (x_1,x_2,… ,x_n) and **y** = (y_1,y_2,… ,y_n), then Euclidean distance between them will be given by:

$$d\,(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{2.13}$$

2.3.6. Properties of K-means Clustering

In each cluster there must be at least one instance of the dataset.

1) Clusters formed by using k-means technique must not be overlapping.
2) Cluster should not hierarchal.

3) An instance must be placed in a cluster from which it has minimum distance.

2.3.7. Example of K-means Clustering

Technique of k-means clustering can be explained by following example. Let’s suppose there are 15 points in the 2-dimensional space which have to partition into 3 clusters. Process of clustering can be explained by following figures [24]:

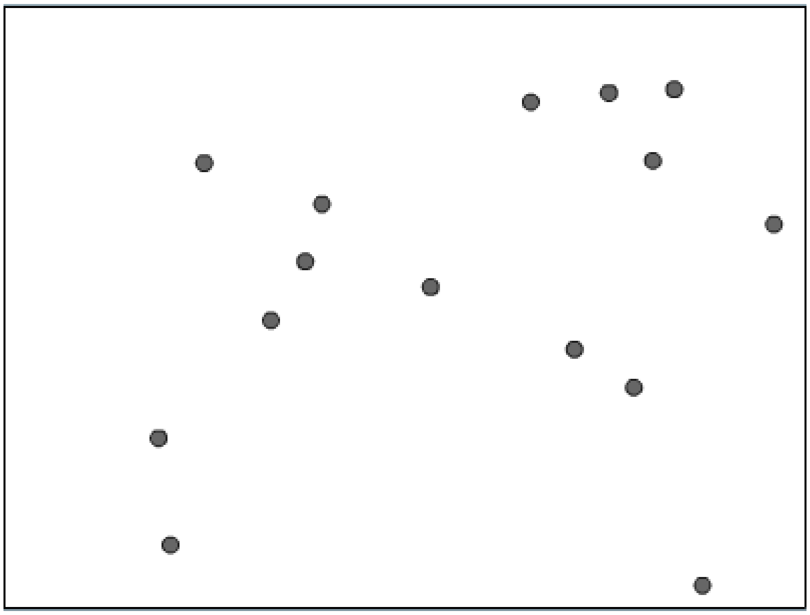

Figure 2.7(a) dataset points before clustering started

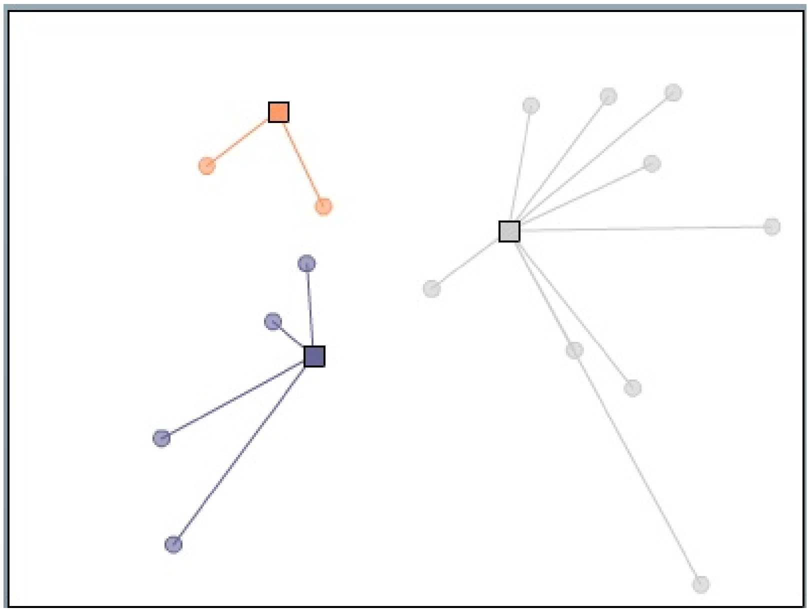

Figure 2.7(b) dataset points and corresponding centroids after 1st iteration of clustering

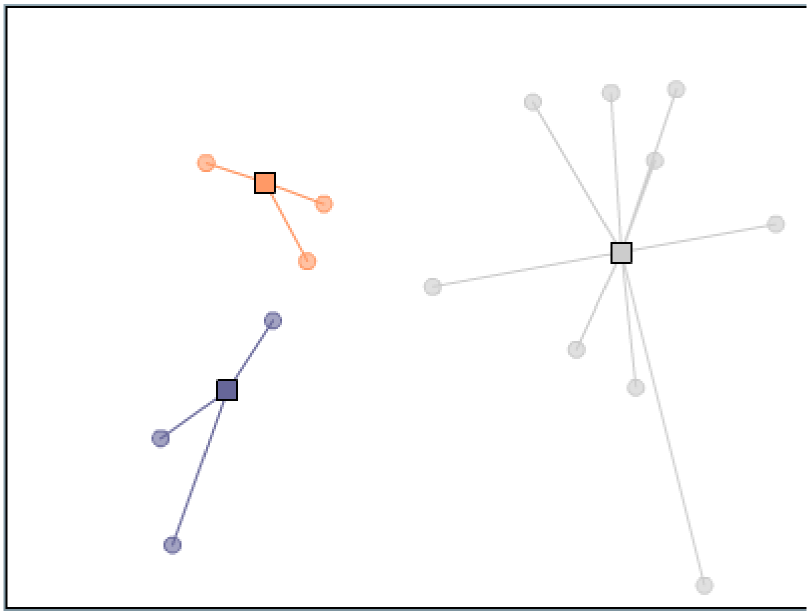

Figure 2.7(c) dataset points and corresponding centroids after 2nd iteration of clustering

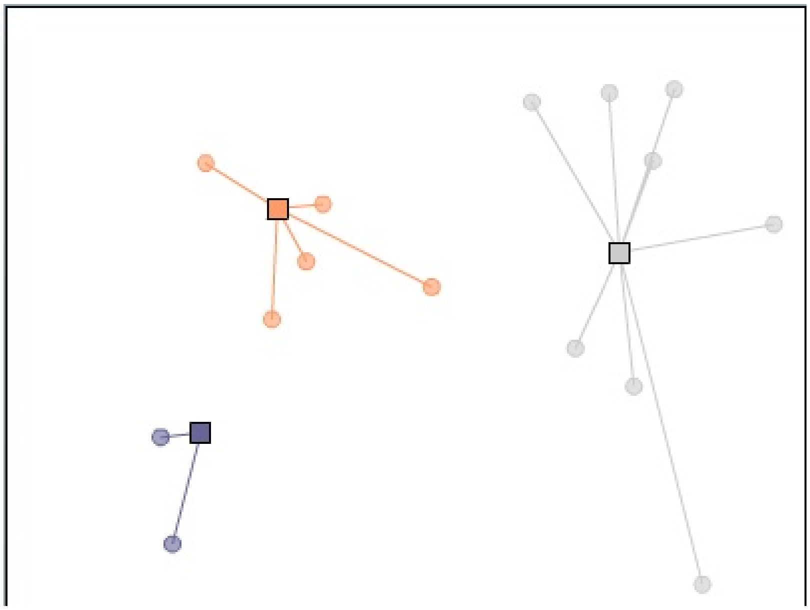

Figure 2.7(d) dataset points and corresponding centroids after 3rd iteration of clustering

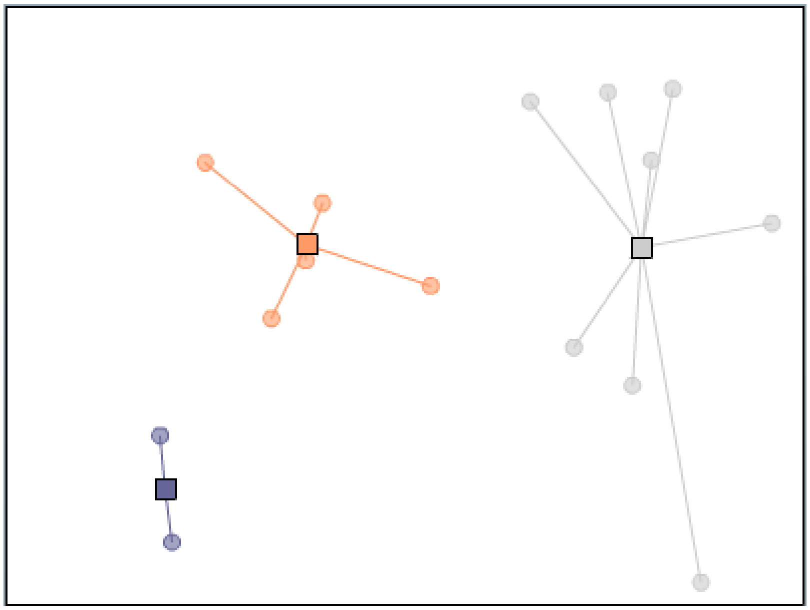

Figure 2.7(e) dataset points and corresponding centroids after 4th iteration of clustering

It can be seen in Figure 2.7(e) that during 4th iteration, no point changes centroid assigned to it in the 3rd iteration. It means that cluster centroids have stabilized and clustering process has been completed.

2.4. Problem statement

Based upon the introduction, background and related work presented about the diagnosis of diabetes I can present my problem statement as: "To apply Fuzzy Theory and Swarm Intelligence for creating an Optimized Fuzzy Rule Based System for early diagnosis of diabetes with high accuracy and acceptable comprehensibility". Here *high accuracy* means a good classification rate of the disease and *acceptable comprehensibility* stands for a classifier model fulfilling optimal time and computational requirements.

2.5. Summary

In this chapter, it has been discussed that how swarm intelligence techniques have been used in medical diagnostic systems. In the beginning of this chapter use of Ant Colony Optimization (ACO) algorithm for developing fuzzy rules regarding diabetes diagnosis has been discussed in detail. Then it has been elaborated that how a variant of Particle Swarm Optimization has been employed for detecting expected hypoglycemia episodes. In the last part of this chapter an overview of clustering has been given. K-means clustering technique has been discussed in detail. K-means clustering is a simple technique to detect 'class outliers' in a dataset. Detecting outliers (noise) can be used to enhance classification rate of a classifier.

Chapter 3

Proposed System

In this thesis I have proposed a classifier named as ‘Swarm Optimized Fuzzy Reasoning Model (SOFRM) for Diabetes Diagnosis’. In order to understand the working of this classifier we have to understand the working of ‘Fuzzy Reasoning Model’ and the concept of ‘Cuckoo Search’ algorithm (swarm intelligence).

3.1.1. Fuzzy Reasoning Model

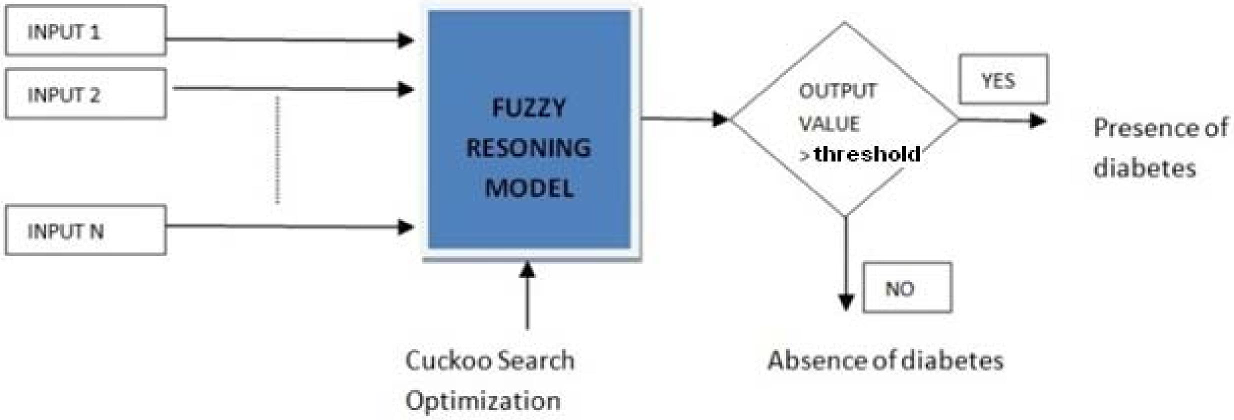

Figure 3.1 Fuzzy Reasoning Model for diabetes diagnosis, optimized by Cuckoo Search [19]

Fuzzy Reasoning Model (FRM) used for early detection of diabetes is shown in Fig. 3.1. This FRM consists of ‘N’ inputs INPUT 1, INPUT 2 ,…, INPUT N. Output tells presence or absence of diabetes. FRM consists of 3 phases: Fuzzification, Reasoning by if-then rules, and defuzzification

1) Fuzzification

First step is to determine the degree of membership of each input to the corresponding fuzzy sets through membership functions. In our case inputs are selected (informative) attributes of Pima Indians Diabetes dataset. Feature selection has discussed later in this thesis.

Degree of membership for any input attribute 'I' when $m_I^K \neq \max\{m_I\}$ or min $\{m_I\}$ is given as:

$$\mu_{N_I^K}(I(t)) = e^{\frac{-\left(I(t)-m_I^K\right)^2}{2\sigma_I^k}} \tag{3.1}$$

Here, $m_I = [m_I^1, m_I^2, \ldots.., m_I^k, \ldots.., m_I^{mf}]$,K = 1, 2,……, $m_{f,}$ m_f = number of membership functions, t = 1, 2,……, $n_{d,,}$ n_d= number of input-output data pairs, m_I^K= mean value of member function,σ_I^K= standard deviation of member function.

For, m_I^K= $\max\{m_I\}$

$$\mu_{N_I^K} I(t) = \begin{cases} 1,\ I(t) => max\{m_I\} \\ e^{\frac{-\left(I(t)-m_I^K\right)^2}{2\sigma_I^k}},I(t) < \max\{m_I\} \end{cases} \tag{3.2}$$

For, $m_I^K = \min\{m_I\}$

$$\mu_{N_I^K} I(t) = \begin{cases} 1,\ I(t) =< min\{m_I\} \\ e^{\frac{-\left(I(t)-m_I^K\right)^2}{2\sigma_I^k}},I(t) > min\{ m_I \} \end{cases} \tag{3.3}$$

Degree of membership for all inputs, $I_1, I_{2,\ldots,}$ I_n is given by equations 3.1 to 3.3. Where, n = number of input attributes of FRM. Fig. 3.2 shows the relationship between degrees of membership of any of the input attributes.

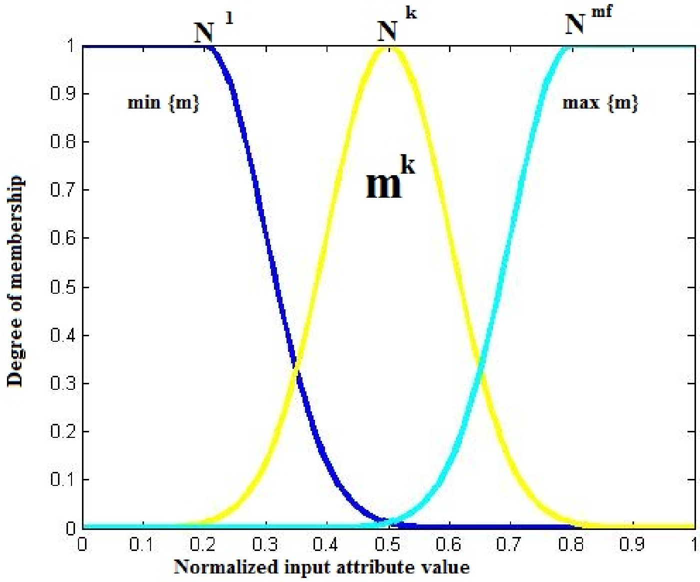

Figure 3.2 Generalized graph of degree of member-ship for any normalized input 'I'

2) Reasoning by if-then rules

As, I_1, I_2,…, I_n are fuzzy inputs and y(t) represents the fuzzy output then, fuzzy rules are defined by:

$$\textbf{Rule } \boldsymbol{\gamma} : \textbf{IF } I_1(t) = N_{I_1^k}(I_1(t)) \textbf{\textit{AND}} I_2(t) = N_{I_2^k}(I_2(t)) \textbf{\textit{AND….AND}} \ I_n(t) = N_{I_n^k}\big(I_n(t)\big) \textbf{THEN}$$

$$y(t) = w_\gamma \tag{3.4}$$

Here, $\gamma = 1, 2, \ldots, n_r$, n_r is number of rules given by:

$$n_r = (m^f)^n \tag{3.5}$$

‘n’ is number of inputs of FRM and w_γ is fuzzy singleton to be determined.

3) Defuzzification

In defuzzification output of fuzzy rules is interpreted if it represents diabetes or not. Presence or absence of diabetes is represented by equations 3.6 to 3.8 [19]:

$$\mathrm{h(t)} = \begin{cases} 0, y(t) < \text{threshold} \\ 1, y(t) \geq \text{threshold} \end{cases} \tag{3.6}$$

Threshold in equation 3.6 is defined by Centroid method of defuzzification. h(t) is output class i.e diabetic or non-diabetic.

$$\mathrm{y(t)} = \sum_{\gamma=1}^{n_r} m_\gamma(t) w_\gamma \tag{3.7}$$

$$m_\gamma(\mathrm{t}) = \frac{\left[\!\left[(I_1(t))\mu_{N_{I_1}^\gamma} \times (I_2(t))\mu_{N_{I_2}^\gamma} \times \ldots . \times (I_n(t))\mu_{N_{I_n}^\gamma} \right]\!\right]}{\sum_{\gamma=1}^{n_r} \left[\!\left[(I_1(t))\mu_{N_{I_1}^\gamma} \times (I_2(t))\mu_{N_{I_2}^\gamma} \times \ldots . \times (I_n(t))\mu_{N_{I_n}^\gamma} \right]\!\right]} \tag{3.8}$$

Cuckoo Search [14] has been employed to optimize w_γ, m_I and σ_I where I=1,…,n and γ=1,…,n_r.

3.2. Cuckoo search

3.2.1. Introduction

Cuckoo search was proposed by Xin-She Yang and Suash Deb 2010 [14]. They stressed that Cuckoo Search is superior to PSO and Genetic Algorithms. Firstly because, though Cuckoo Search is population based it uses selection similar to that used in Harmony Search. Secondly, exploring random solutions in Cuckoo Search is more efficient since its step length is heavy tailed and thus step-size can be very large. Cuckoo Search is more generic than genetic algorithm and PSO as it needs lesser parameters to be tuned.

3.2.2. Inspiration

Cuckoo search algorithm is based upon parasitic reproductive strategy of certain Cuckoo species. They lay their eggs in some other species' (host) nest. There are two types of parasitic behavior of such Cuckoo species [25]:

1) **Obligate Brood Parasitism**

   Cuckoo species which show this type of parasitic behavior depend totally on the nests of others species for their reproduction.

2) **Non-obligate Brood Parasitism**

   Cuckoo species which show this type of parasitic behavior can use their own nests for breeding but also use host species' nests to increase their reproduction rate.

3.2.3. Inserting Eggs Into Host Nests

Depending upon hosts' defense mechanism parasitic Cuckoo species have evolved various strategies to insert their eggs safely into host nests. Thickness of egg-shell is two-layered to prevent damage to the Cuckoo egg at the time of its dropping into the host nest. To keep host nest un-aware of intrusion various secretive and quick in action strategies have been developed by female Cuckoos. In some cases male Cuckoos help females Cuckoos by luring host adults away from their nests and in the absence of adult hosts Cuckoo females lay their eggs into the host nests.

3.2.4. Protecting Parasitic Cuckoo Eggs

To increase the reproduction ability of their eggs present in the alien nests Cuckoo species use various techniques. Cuckoo eggs hatch earlier than the hosts' eggs and the resulting chicks propel away host nests to increase their own chance of survival. This behavior cannot be taught by parent Cuckoo to its offspring so, this methodology is passed on via genes.

Female Cuckoos lay eggs similar in size and color to the host nests. It minimizes the chance of parasitic Cuckoo eggs to be thrown away by the host. Many hosts remove alien eggs from their nests. They detect such eggs from their sizes and colors etc. Female Cuckoos prefer to lay their eggs into the nest of a host whose eggs are similar to its own eggs.

Cuckoo Search uses the above mentioned behavior of Cuckoo species to search for the best nest (solution) in order to optimize the reproduction (classifier) over many generations (iterations).

3.2.5. Pseudo code of Cuckoo Search

```
Begin
Generate initial population of 'n' number of host nests (solutions) $x_{i,}$ where ($i = 1, 2,\dots, n$) with random values.
Use fitness function, $f(x)$ to determine best nest amongst 'n' randomly generated nests.
While (minimum fitness achieved or number of iterations above a certain value)
    Generate new nests using Levy Flight but never discard the best nest
    Find current best nest amongst the new 'n' nests
    Discard a fraction $p_a$ ε [0,1] of the worst nests and replace them by new solutions using biased/selective random walks
```

```
Now again, find current best nest amongst the new ‘n’ nests

If (fitness (current best nest) < fitness (best nest))

best nest = current best nest

End If

End While

stop training and use best nest for testing

End
```

Figure 3.3 Pseudo code of Cuckoo Search Algorithm

In this pseudo-code each nest represents a single solution. P_a represents the probability with which worst nest will be discarded in search of better nests, p_a ε [0, 1].

3.3. Cuckoo Search and Levy Flights

Levy flight is performed to generate new solutions. Flight behaviors of many animals and insects show characteristics of Levy Flight [14]. A Lévy flight is a random walk in which the step-lengths have a probability distribution that is heavy-tailed [26]. Fig. 3.4 shows the path traced and turning points of Levy Flight.

Jumps in Levy Flight are distributed according to the power law of following equation [27]:

$$\mathrm{Prob}(Y > u) = u^{-h} \tag{3.9}$$

Far right sides (the tails) of three hyperbolic distributions for h=.5, h=1 and h=1.5 are shown in Fig. 3.4. Additionally normal distribution (black curve) is also shown for comparison [27]. With probability one, Levy distribution has infinite variance and an infinite mean.

Generating new solution $x_i(t+1)$ using Levy Flight is done by following equation:

$$x_i(t+1) = x_i(t) + \alpha \oplus \text{Levy}(\lambda) \tag{3.10}$$

Here, α is step length and is used to scale effect of Levy Flight. Its value is greater than 0 (usually equal to 1). Operator $\oplus$ denotes point to point multiplication.

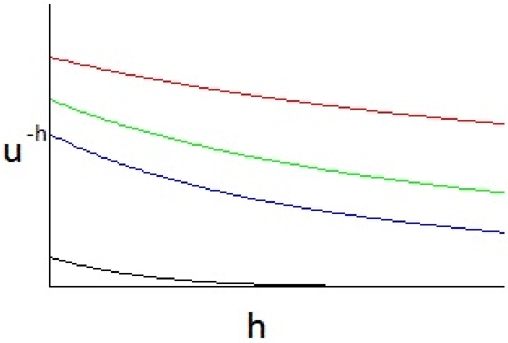

Figure 3.4 Tails of hyperbolic distributions
for h=.5(red), h=1(green), h=1.5(blue) and normal distribution (black) [27]

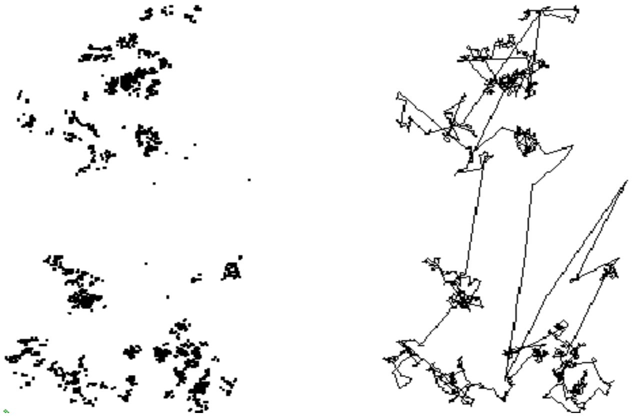

Figure 3.3 Levy Flight
On the right there is path traced out by a Levy flight; on the left, the turning points [27]

Random step length is generated using following equation [14].

$$\text{Levy} \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \tag{3.11}$$

Equation 3.11 is same as equation 3.9 but written in the context of Cuckoo Search Algorithm.

3.4. Summary

In this chapter a new method for detection of diabetes has been proposed called as 'Swarm Optimized Fuzzy Reasoning Model (SOFRM) for Diabetes Diagnosis'. First a fuzzy reasoning model (FRM) has been explained then it has been discussed that Cuckoo Search algorithm can be used to optimize certain parameters of FRM. Algorithm of Cuckoo search been explored in detail. Furthermore, Levy flight has been explained to show how Cuckoo performs flights with heavy-tailed distribution in search of better and superior solutions.

Chapter 4

Implementation

4.1. Description of Dataset

In this study Pima Indians Diabetes dataset has been used to find the usefulness of our work. This dataset is owned by National Institute of Diabetes and Digestive and Kidney Diseases [28]. A population of Pima Indians living near Phoenix, Arizona was used to collect the samples for this dataset. All of the persons whose medical tests were performed were females with age above 21. Reason for selecting this population in this survey is Pima Indians' higher rate of diabetes occurrence as compared to other Americans [31].

Pima Indians Diabetes data set has 8 features [28]:

1. Number of times pregnant

2. Plasma glucose concentration

3. Diastolic blood pressure

4. Triceps skin fold thickness

5. Two hour serum insulin

6. Body mass index

7. Diabetes pedigree function

8. Age.

All of these attributes have numeric values. There are 768 samples for these 8 attributes in the dataset. Total number of classes is 2, denoted by 0 for non-diabetic and 1 for patients with diabetes. Many instances in this dataset show 'zero' values for one or more attributes. But many a times these zero values are improbable such as for blood pressure attribute. In this situation we have to consider these 'zeros' as missing values. Decision regarding missing has been discussed in the section 4.2.1.

Out of these 768 samples, 268 samples represent diabetes patients while remaining 500 samples were found to be of non-diabetic persons.

4.2. Pre-processing

Before implementing Swarm Optimized Fuzzy Reasoning Model (SOFRM) for Diabetes Diagnosis, Pima Indians Diabetes data is analyzed for missing values (as missing values do not help in development of a useful classifier). Then informative features of Pima Indians diabetes data set are extracted to get a simpler and more effective implementation of FRM. Afterwards, noise is removed from data set using k-means clustering.

4.2.1. Removing missing values from data set

Out of the eight attributes of the dataset serum-insulin and triceps skin fold have very high occurrence of missing values (374 and 227, respectively). Furthermore instances for other attributes which have missing values are also removed. Thus 625 instances and six features are left [22].

Before moving on to 'extracting informative features' each column of (excluding the column for classes) Reduced Pima Indians Diabetes is normalized using the following formula:

$$\text{x\_norm} = \frac{\text{x−x\_min}}{\text{x\_max−x\_min}} \qquad (4.1)$$

In eq. 4.1 x_min is the minimum value of an attribute and x_max is the maximum one. Original value is denoted as x whereas normalized value is represented by x_norm.

4.2.2. Extracting informative features

We need to select most informative features of Pima Indians Diabetes set because lesser number of informative features will provide simpler FRM implementation (see eq.3.5 and table 4.1) and lesser clustering error as will be discussed later in section 4.2.3.

No of features in FRM	No of rules*
2	16
3	64
4	256
5	1024
6	4096

* Given by equation 3.5

Table 4.1 No. of rules of FRM for different no of features when no. of membership functions = 4

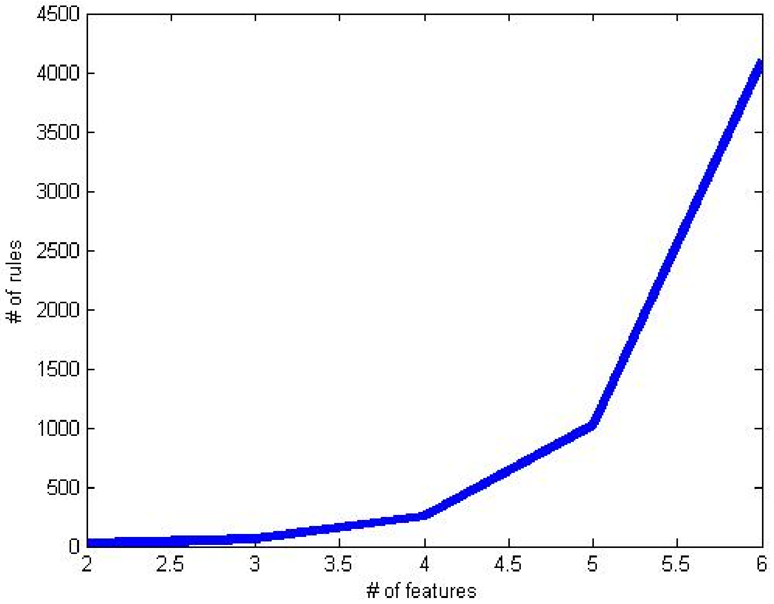

Figure 4.1 Exponential increase in number of rules of FRM with increase in number of features

Weiss/Indurkhya 'independent features' significance testing method [29] has been used to select most informative features to be used in FRM as implemented by Will Dwinnell [30]. Significance of an attribute in discriminating between two classes 'A' and 'B' is found using the following formula:

$$significance = \frac{abs(mean(classA)-mean(classB))}{\sqrt{\frac{var(classA)}{nA}+\frac{var(classB)}{nB}}} \tag{4.2}$$

Where <u>means(classA)</u> represents means of all the samples belonging to class 'A' and <u>nA</u> represents sum of the samples of classA. Results of independent features significance testing have been shown in table 4.2.

FEATURE	SIGNIFICANCE
F2: Glucose tolerance test	13.1853
F6: Body mass index	8.1062
F8:Age	7.1470
F1:No. of times pregnant	6.4516
F3: Diastolic blood pressure	4.7087
F7: Diabetes pedigree function	4.1669

Table 4.2 Significance values of different attributes for distinguishing two output classes of Pima Indians Diabetes Dataset [29]

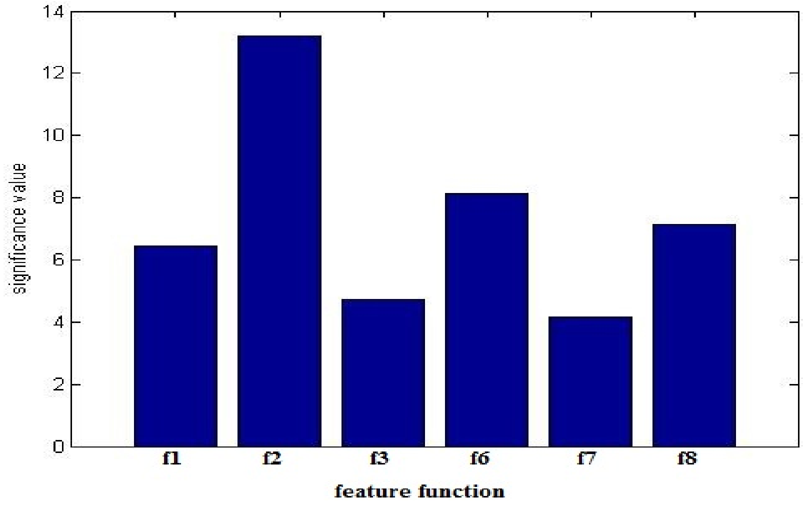

Figure 4.2 Significance value for 6 attributes of Pima Indians Diabetes dataset

4.2.3. Removal of Noise

K-means clustering [22] algorithm has been used to detect outliers. During clustering we initially use all the six attributes seen in table 4.2. Then we remove the least significant attribute of table 4.2 and cluster the dataset again. This process is continued until we perform clustering using the only two top most informative attributes of table 4.2. Results of this clustering approach are shown in table 4.3.

No Of Attributes Used	Clustering Error (%)
6	26.08
5	27.5
4	28.9
3	25.6
2	24.32

Table 4.3 Clustering error using different number of attributes of Pima Indians Diabetes dataset

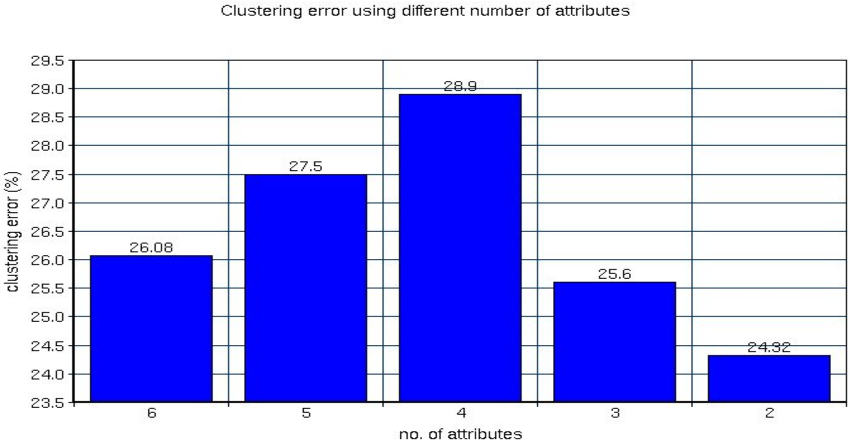

Figure 4.3 Graph of clustering error observed using different no. of attributes

24.32 % (152) of 625 instances of reduced Pima Indians Diabetes Dataset are incorrectly clustered. These outliers are thrown out as noise and remaining 473 will be used by FRM to determine the symptoms of diabetes.

4.3. Implementation of the Proposed Classifier

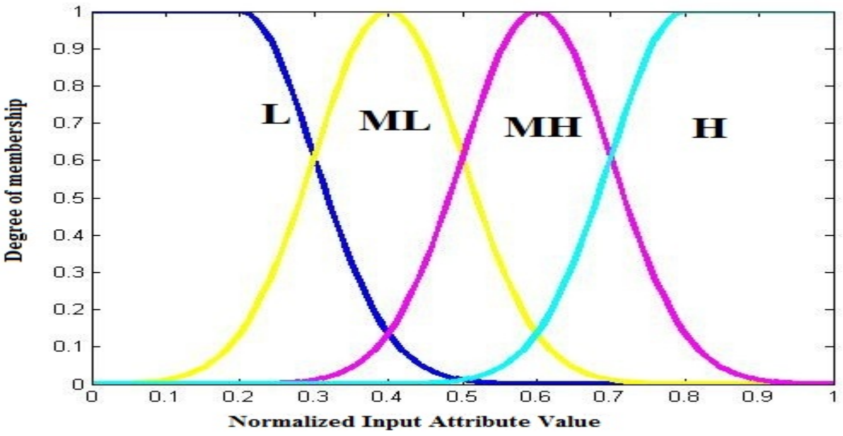

Figure 4.4 Graph of degree of member-ships for normalized input 'I' of our implementation of FRM

In our implementation of FRM (Swarm Optimized Fuzzy Reasoning Model, SOFRM) we will use the two top most informative attributes. No of membership functions is 4. The four membership functions are **L=low, ML=medium low, MH=medium high and H = high**.

Cuckoo Search has been used to optimize the output of Fuzzy rules as well as the means and variances of membership functions. In the original implementation of Cuckoo Search [14] each solution has only one type of data. In our study, each solution has three types of data: the fuzzy rule outputs, means of membership functions and corresponding variances.

As we have selected two most informative attributes and four membership functions to be used in FRM the equations (3.4) and (3.8) will be simplified as:

Rule γ: **IF** $I_1(t) = N_{I_1^k}(I_1(t))$ **AND** $I_2(t) = N_{I_2^k}(I_2(t))$ **THEN** $y(t) = w_\gamma$ (4.3) and as $m_f = 4$, k=1,…,4, $n_{r\,=}\ (4)^2 = 16$

$$m_{\gamma}(\mathrm{t}) = \frac{\left[\!\left[\left(\mathrm{I}_1(\mathrm{t})\right)\mu_{\mathrm{N}^{\gamma}_{\mathrm{I}_1}} \times \left(\mathrm{I}_2(\mathrm{t})\right)\mu_{\mathrm{N}^{\gamma}_{\mathrm{I}_2}}\right]\!\right]}{\sum_{\gamma=1}^{16}\left[\!\left[\left(\mathrm{I}_1(\mathrm{t})\right)\mu_{\mathrm{N}^{\gamma}_{\mathrm{I}_1}} \times \left(\mathrm{I}_2(\mathrm{t})\right)\mu_{\mathrm{N}^{\gamma}_{\mathrm{I}_2}}\right]\!\right]} \tag{4.4}$$

Fitness function is defined by the classification rate of Pima Indians Diabetes Dataset.

$$classification\,rate = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.5}$$

Where,

TP: TRUE POSITIVE i.e sample assigned as diabetic by the classifier which was actually diabetic

FN: FALSE NEGATIVE i.e sample assigned as non-diabetic by the classifier which was actually diabetic

TN: TRUE NEGATIVE i.e sample assigned as non-diabetic by classifier which weas actually non-diabetic

FP: FALSE POSITIVE i.e sample assigned as diabetic by classifier which was actually non-diabetic

$$fitness\,function = \frac{1}{classificationrate} \tag{4.6}$$

This fitness function will be minimized by Cuckoo Search to optimize the FRM.

4.4. Summary

In the beginning of this chapter a description of the dataset is given. Then implementation steps of the proposed method (SOFRM) are explained. First preprocessing is used for data cleaning. In this phase, missing (disguising) values are removed from the dataset and informative attributes are selected for a simpler implementation of the classifier. Then k-means clustering is employed to remove the noise (outliers) from the dataset. Finally, the proposed classifier is discussed which will use the pre-processed dataset to diagnose the diabetes.

Chapter 5

Experimental Setup &Results

5.1. Experimental Setup

Proposed method was simulated using:

- MATLAB 7 (R2010b)
- Intel(R) Core i3 processors
- processor speed = 5 GHz ,and
- RAM = 2 GB

Dataset was tested using tenfold cross validation. Entire dataset was divided into 10 equal partitions at random. During an iteration of cross validation one partition was used as testing set and the remaining 9 partitions were assigned to be training set. There were a total of 10 iterations and testing partition was different each time.

5.2. Evaluation Measures

In order to find the usefulness of our classifier, classification rate given by equation 4.6 in section 4.3 is used. When SOFRM (proposed classifier) is compared to FADD [18] number of rules is more but rule length is less. This comparison of SOFRM and FADD is being focused because both of these techniques are using classifiers based upon fuzzy logic.

Classifier	No. of Rules	Length of Rules
FADD	8	2.571
SFORM	16	2

Table 5.1 SFORM v/s FADD In Terms of No. of Rules and Length of Rules

In FADD initially number of rules is zero. After a certain number of iterations of the swarm 8 useful rules were developed. Mean length of these rules is 2.571. In contrast to this number of rules in SFORM is fixed at 16. But these rules evolve over many 'generations' of a swarm to reach an acceptable classification rate. Length of each of these rules is '2' which is lesser than 'Mean Rule Length' of FADD.

In addition to above comparisons an enhancement of SFORM over FADD is data cleansing used in preprocessing phase (See section 4.2). This data cleansing removes disguising values from the dataset.

5.3. Comparison of Results

Comparison of SFORM's classification rate with respect to some of the classifiers developed in the recent years is shown in the following table:

Author	Approach Used	No. Of Attributes Used	Year of Publication	Publication Type	Classification Rate (%)	Used Data Cleansing	Used Feature Selection?
Balakrishnan et al. [22]	SVM	3	2011	Journal	98.09/98.92	YES	YES
Fathi et al. [18]	FADD(fuzzy logic)	8	2010	Conference	79.16	NO	NO
Davar et al. [35]	MI-MCS-FWSVM	4	2012	Journal	93.58	NOT Discussed	YES
Hosseinpour et al [36]	Bagging ensemble classifiers	8	2012	Journal	77.47	NO	NO
Kayaer et al [37]	Regression Neural Networks	8	2003	Conference	80.21	NO	NO
Madhavi et al [38]	Neural Network and Fuzzy k-Nearest Neighbor Algorithm	4	2012	Journal	72.59	YES	YES
Aslam et al [39]	GENETIC PROGRAMMING	8	2010	Conference	78.5	NO	NO
Proposed Method	Swarm optimized Fuzz	2			98.17*	YES	YES

Table 5.2 SFORM v/s Other Classifiers

*after data cleansing, without data cleansing it is 76.14 %.

5.4. Sensitivity, Specificity & ROC Curve

Sensitivity of SFORM is evaluated using following formula:

$$\text{sensitivity} = \frac{\text{No. of samples correctly diagnosed as diabetic (TP)}}{\begin{array}{c}\text{No. of samples correctly diagnosed as diabetic (TP)+} \\ \text{No. of samples incorrectly classified as healthy} \\ \text{(FN)}\end{array}} \qquad (5.1)$$

After implementation equation (5.1) gave the value of 95.17 %. Similarly specificity of SFORM is given as:

$$\text{specificity} = \frac{\text{No. of samples correctly classified as healthy(TN)}}{\begin{array}{c}\text{No. of samples correctly classified as healthy(TN)+} \\ \text{No. of samples incorrectly classified as diabetic} \\ \text{(FP)}\end{array}} \tag{5.2}$$

Eq. (5.2) produced a value of 99.14%.

Like specificity and sensitivity another very important aspect to judge any binary classifier is ROC (Receiver operating characteristic) [31].

Receiver operating characteristic (ROC) is the plot of TPR (True positive rate) v/s FPR (False Positive Rate) for a two class classifier. TPR is same as sensitivity while FPR = 1 – specificity.

Ideal position of ROC plot is at (0,1) i.e. 100% sensitivity and 100% specificity. If ROC of a classifier is above the diagonal line (shown in fig.5.1) it is considered to be a good classifier. Otherwise, if ROC of a classifier lies on the diagonal line result of the classifier is just a random guess like tossing of a coin and ROC below diagonal line means a poor classification approach.

ROC of our classifier is as shown in fig. 5.2. In this figure before the start of optimization process the ROC curve is at (0, 0.4) while as the optimization process continues (over iterations of Cuckoo Search Algorithm) ROC graphs improves and ultimately reaches (0.05,0.877) which is very close to the deal location of (0,1).

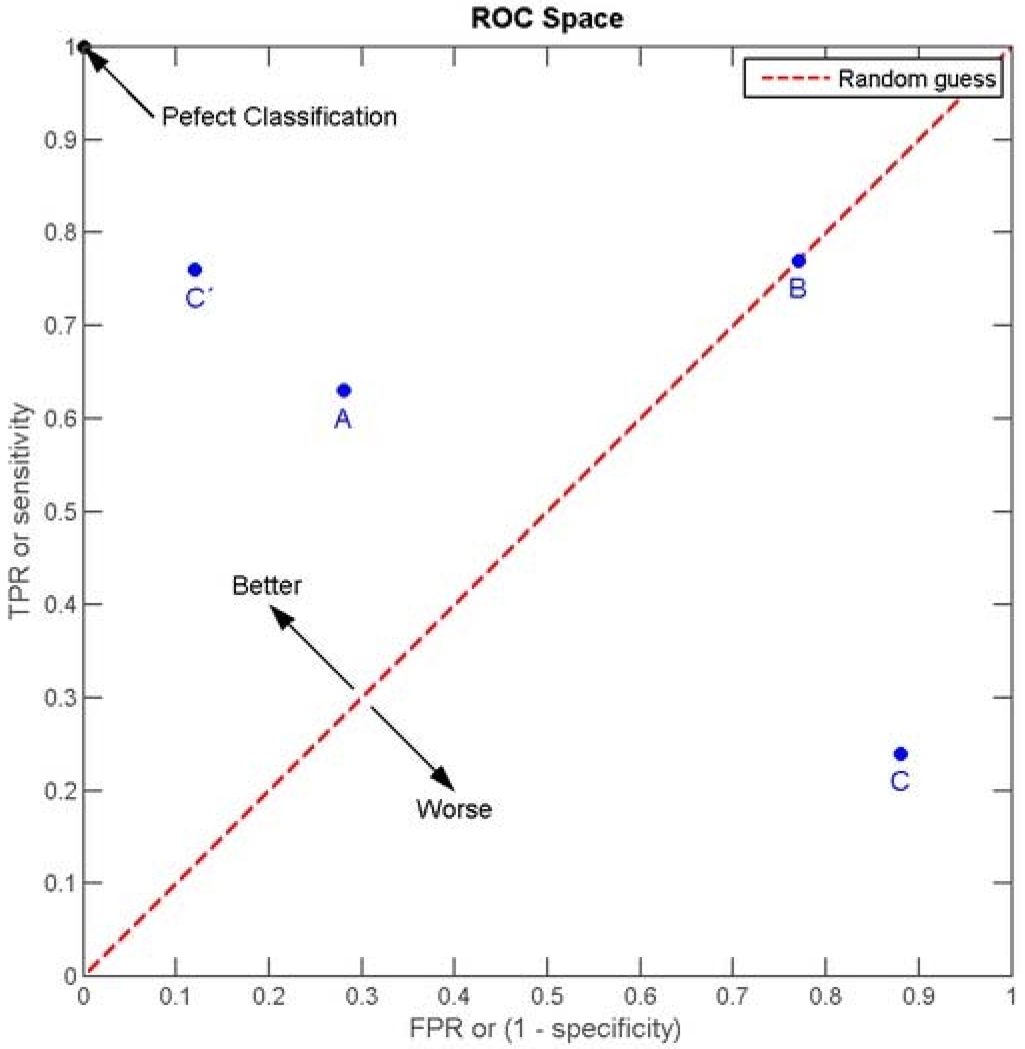

Figure 5.1 An example of ROC graph [32]

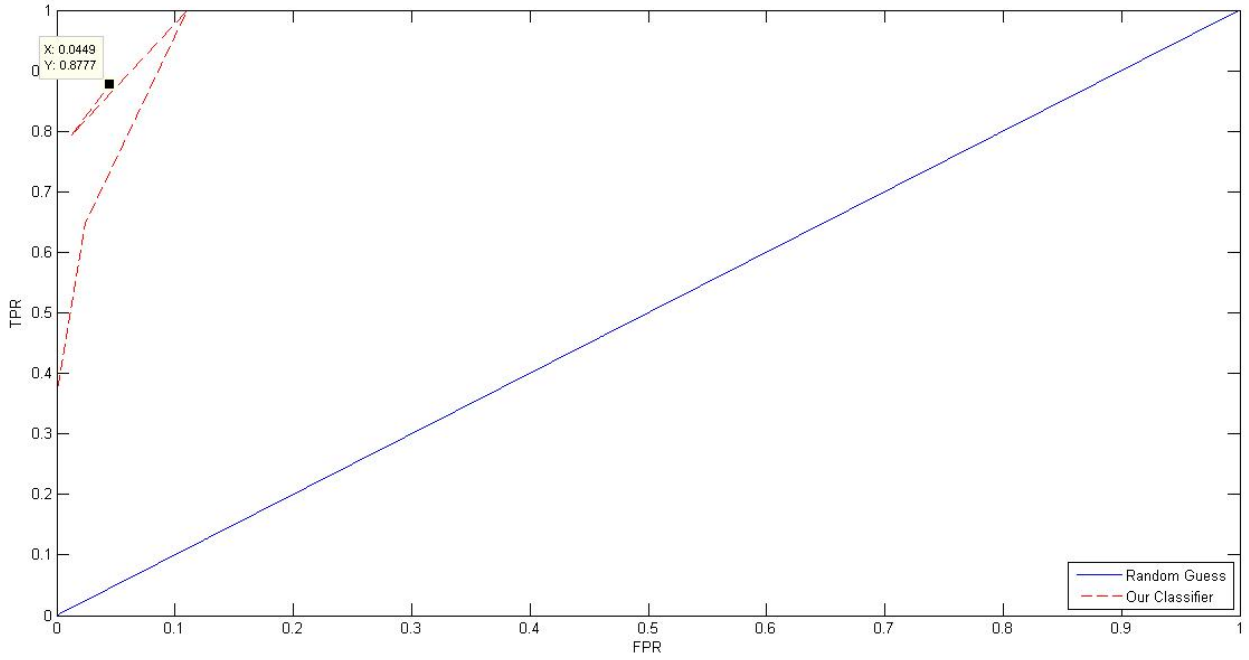

Figure 5.3 ROC plot of SFORM. Blue Diagonal line is a separation line between 'good' and bad binary classifiers

5.5. Summary

This chapter discusses experimental setup, classification results and comparisons of the proposed SFORM with other classifiers. We compared our classifier especially to FADD [18] as both the classifiers are fuzzy logic based. FADD has lesser number of rulers but rule length of SFORM is found to be better. Finally SFORM has enhanced classification accuracy as compared to different classifiers proposed in the recent past. At the end of this chapter we see sensitivity, specificity and ROC of SFORM are also showing satisfactory values.

Chapter 6

Conclusion and Future Work

In this thesis Fuzzy Reasoning Model (FRM) optimized by Swarm Intelligence (Cuckoo Search) has been used for diabetes diagnosis. I named this classifier as SOFRM. Use of Swarm intelligence techniques for optimizing medical classifiers has been discussed in detail. Fuzzy Logic as well as different mathematical models are utilizing Swarm Intelligence techniques to define outputs and tune different parameters in a simpler and less cumbersome manner than traditional techniques.

In the *relevant literature* chapter use of Ant Colony Optimization (ACO) for developing suitable fuzzy rules for diabetes diagnosis can been found [18]. This technique known as FADD has resulted in a classification rate of about 80%.Use of Fuzzy Reasoning Model optimized by a variant of PSO for hypoglycemia detection has shown a classification rate of about 76%. Dataset used in this method has only 2 attributes and thus only 25 rules if number of fuzzy set members is 5 for each of the two attributes.

Design of Fuzzy Reasoning Model (FRM) and Cuckoo Search were discussed in 3rd chapter (Proposed Classifier). After explaining both the concepts I moved on to the implementation phase (4th Chapter). In case of diabetes detection, data set available at the UCI website has 8 attributes and with 4 members of fuzzy set for each attributes it needs 4^8=65536 rules. It makes

FRM and its optimization process complex and time-consuming. Furthermore, anomalies present in the dataset are also decreasing classification rate of diabetes. In order to deal with both of these problems, firstly attributes were arranged in descending order of their influence on proper classification of data objects. In the second step k-means clustering was used to remove class outliers from the dataset. It was found that clustering error was maximum when all of the 8 attributes were used and error was minimum when the only two most informative attributes were used in clustering process.

As a result of the methodology used above we were left with 2 attributes and 16 rules (4 members of Fuzzy sets for each attribute). When this simplified and informative subset of Pima Indians Diabetes Data set was used in FRM optimized by Cuckoo Search classification rate rose up to 98%.

6.1. Future Work

This work on diabetes can be extended for other medical diagnostic purposes. The proposed classifier deals with two class scenario i.e. either positive or negative for the disease. But there will be cases for more than two classes as research continues. In my opinion Fuzzy rules would be very helpful in solving such cases.

In this thesis it was found that as the dataset gets bigger computational complexities worsen. Therefore future researches using Fuzzy Logic must have to cater for this in their implementations of Fuzzy classifiers.

Furthermore, from the aspect of general research in the field of machine learning/pattern recognitions: handling outliers is still a question needing too much research. Methods should be developed to find the process responsible for presence of outliers in an outlier. Otherwise, if outliers are to be thrown away there must be a strong mathematical justification for that.

References

[1] http://www.cdc.gov/chronicdisease/resources/publications/AAG/ddt.htm (last accessed July,2012)

[2] http://stephhicks68.hubpages.com/hub/Do-I-have-Diabetes--Early-Warning-Signs. (last accessed July,2012)

[3] Takeshi Kuzuya, “Early diagnosis,early treatment and the new diagnostic criteria of diabetes mellitus,” British Journal of Nutrition (2000), 84,Suppl.2,S177-S181

[4] Igarashi K, Abe T, Eguchi H & Tominaga M, “Chronic diabetic complications in patients with diabetes mellitus and impaired glucose tolerance found in a population-based Funagata Diabetes Study,” Journal of Japan Diabetes Society 41,159-163(in Japanese).

[5] Harris MI, Klein R, Welborn TA & KnuimanMW, “Onset of NIDDM occurs at least 4-7 years before clinical diagnosis,” DiabetesCare 151,815-819, 1992.

[6] Beni, G., Wang, J.,“Swarm Intelligence in Cellular Robotic Systems”, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)

[7] S.Kazadi, “Swarm Engineering”, Ph.D Thesis, California Institute of Technology, 2000

[8] Yan-fei Zhu, Xiong-min Tang, “Overview of Swarm Intelligence,” 2010 International Conference on Computer Application and System Modeling (ICCASM 2010)

[9] Parsopoulos, K. E.; Vrahatis, M. N. (2002). "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization". Natural Computing 1 (2-3): 235–306

[10] http://www.inderscience.com/offer.php?id=22775 (aceesed July,2012)

[11] Yang,X.S.:Fireflya lgorithms formultion modaloptimization. Stochastic Algorithms: Foundation and Applications,SAGA2009,Lecture Notes in ComputerSciences,5792,169--178,(2009).

[12] L.A.Zadeh, Fuzzysets, Inform. Control8(1965) pp. 338–353.

[13] Michael Negnevistky, Artificial Intelligence: A Guide to Intelligent Systems

[14] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search", Int. J. Mathematical Modelling and Numerical Optimisation," Vol. 1, No. 4, 330-343 (2010).

[15] P. Civicioglu and E. Besdok, " A conception comparison of the cuckoo search, particle swarm optimization, differential evolution and artificial bee colony algorithms,", Artificial Intelligence Review, DOI 10.1007/s10462-011-92760, 6 July (2011).

[16] Gandomi and Alavi, "Krill Herd Algorithm: A New Bio-Inspired Optimization Algorithm,", Communications in Nonlinear Science and Numerical Simulation. DOI10.1016/j.cnsns.2012.05.010

[17] http://www.thelancet.com/journals/lancet/article/PIIS0140-6736%2810%2960484-9/fulltext (accessed July, 2012)

[18] Mostafa Fathi Ganji and Mohammad Saniee Abadeh, "Using fuzzy Ant Colony Optimization for Diagnosis of Diabetes Disease," Proceedings of ICEE 2010, May 11-13, 2010.

[19] S. H. Ling, Nuryani, and H. T. Nguyen, "Evolved Fuzzy Reasoning Model for Hypoglycaemic Detection," 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, August 31 - September 4, 2010

[20] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html (accessed July' 2012)

[21] http://mathworld.wolfram.com/CliqueGraph.html (accessed July' 2012)

[22] S Balakrishnan, R Narayanaswamy, Ilango Paramasivam, "An Empirical Study on the Performance of Integrated Hybrid Prediction Model on the Medical Datasets," International Journal of Computer Applications (0975 – 8887) Volume 29– No.5, September 2011.

[23] http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/K-Means_Clustering_Overview.htm (accessed July' 2012)

[24] http://www.paused21.net/off/kmeans/bin/ (accessed July' 2012)

[25] Payne, Robert B, "The Cuckoos," Oxford University Press (2005). ISBN 0-19-850213-3.

[26] http://en.wikipedia.org/wiki/L%C3%A9vy_flight(last accessed July,2012)

[27] http://classes.yale.edu/fractals/randfrac/Levy/Levy.html (last accessed July,2012)

[28] Asuncion A and Newman D J, 2007. UCI Machine Learning repository. [http://www.ics.uci.edu/~mlearn/MLRepository.html].University of California, Irvine, CA

[29] Sholom M. Weiss,Nitin Indurkhya, Predictive Data Mining, 1st edition

[30] http://dwinnell.com/IndFeat.m (last accessed July,2012)

[31] http://diabetes.niddk.nih.gov/dm/pubs/pima/pathfind/pathfind.htm (last accessed Dec,2012)

[32] Swets, John A.; Signal detection theory and ROC analysis in psychology and diagnostics : collected papers, Lawrence Erlbaum Associates, Mahwah, NJ, 1996.

[33] http://en.wikipedia.org/wiki/Receiver_operating_characteristic (Accesssed Jan'13) .

[34] http://www.who.int/mediacentre/factsheets/fs312/en/

[35] Davar G., Hamid S., GholamReza B., Younes K," Automatic Detection of Diabetes Diagnosis using Feature Weighted Support Vector Machines based on Mutual Information and Modified Cuckoo Search", http://arxiv.org/ (Cornell University, New York, United States)

[36] Hosseinpour, Saeed, Karim, Mosleh," Diabetes Diagnosis by Using Computational Intelligence Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering, 2012

[37] K. Kayaer and T. Yildirim, "Medical diagnosis on Pima Indian diabetes using general regression neural networks," Proceedings of the international conference on artificial neural networks and neural information processing (ICANN/ICONIP), 2003, pp. 181-184.

[38] Madhavi, Ketki, Parag, Ajinkya, Eknath, "Design of Classifier for Detection of Diabetes using Neural Network and Fuzzy k-Nearest Neighbor Algorithm", International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5,2012.

[39] Aslam, Asoke, "Detection of Diabetes Using Genetic Programming", 18th European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark, August 23-27, 2010.