UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING AND INFORMATICS

A TYPE-2 DIABETES EARLY WARNING SYSTEM USING PARTICLE SWARM OPTIMIZED ARTIFICIAL NEURAL NETWORKS: A CASE OF TRANS NZOIA COUNTY OF KENYA

BY

DOUGLAS HUMPHREY MBIRIKA

REG NO: P58/76612/2012

SUPERVISOR: MR. SAMUEL RUHIU

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE OF THE UNIVERSITY OF NAIROBI

AUGUST 2014

DECLARATION

STUDENT

I declare that this project is my original work and has not been presented anywhere for academic or any other purpose.

DOUGLAS HUMPHREY MBIRIKA

P58/76612/2012

Sign: ……………………………………………………………………………………….

Date: …………………………………………………………………………..…………

SUPERVISOR

I declare that this work has been presented with my approval as the supervisor

MR. SAMUEL RUHIU

LECTURER, SCHOOL OF COMPUTING AND INFORMATICS

Sign: ……………………………………………………………………………………….

Date: ……………………………………………………………..………………………

DEDICATION

I give glory to the almighty God for the strength he has given me throughout this journey. Special thanks go to my family members especially my dear brother Rolland and my lovely daughter Jazleen.

ACKNOWLEDGEMENT

I first and foremost return glory to God the Almighty for breathing strength and determinations in me to enable me successfully undertake this research to conclusion.

Special acknowledgement goes to my supervisor Mr. Samuel Ruhiu, a Senior Lecturer in the School of Computing and Informatics, University of Nairobi, for his invaluable intellectual support throughout this research.

To my brother, Mr. Rolland Andayi, special thanks for the support you gave me linking me to various health experts who greatly assisted me throughout the study from the conceptualization of the idea to interpretation of the results. Mr. Kirwa and Evans of Kitale District Hospital thanks very much for your time and special contributions.

I also extend my special thanks to all health officers from Trans Nzoia County who aided me in different capacities to see to it that the research was a success.

May God Bless You All.

ABSTRACT

This research aimed at developing an early warning system for pre-diabetic and diabetics by analyzing simple and easily determinable signs and symptoms of diabetes among the people living in Trans Nzoia County of Kenya using Particle Swarm Optimized Artificial Neural Networks. With the skyrocketing prevalence of type 2 diabetes in Kenya the system can be used to encourage affected people to seek further medical attention to prevent the onset of diabetes or start managing it early enough to avoid the associated complications. The study sought to find out the best predictive variables of Type 2 Diabetes Mellitus, developed a system to diagnose diabetes from the variables using Artificial Neural Networks and tested the system on accuracy to find out the effectiveness of the system as an early warning system for the disease. Data was collected from diabetes clinics in hospitals within Trans Nzoia County of Kenya. The collected data was first preprocessed by R software to select the best generalizing attributes which were thereafter used to model an Artificial Neural Network. Particle Swarm Optimization algorithm was used to explore the global minima of the solution curve which was later exploited using the back propagation algorithm. The network attained a 70% and 66.23% accuracy on training and test data respectively. The network also attained a sensitivity of 70.23% and a specificity of 64.24%. This clearly shows that the system can be used as an early warning system for type 2 diabetes mellitus.

Keywords: *Artificial Neural Networks, Particle Swarm Optimization, Back Propagation Algorithm, Type 2 Diabetes*

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

MDG - Millennium Development Goals

NDS - National Diabetes Strategy

GDP- Gross Domestic Product

NCD - Non – Communicable Diseases

IDF – International Diabetes Federation

ANN – Artificial Neural Networks

PSO – Particle Swarm Optimization

MLP – Multilayer perceptron

PHE- Public Health of England

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 Background

Artificial Neural networks (ANN) are vital tools for classifying clinical data. They have demonstrated better results in terms of performance and accuracy when predicting diabetes diagnosis outcomes compared to other machine learning and statistical pattern recognition methods (Shankaracharya et al. 2010). They employ an information processing paradigm inspired by biological nervous system's components such as the brains whose main processing units are nerve cells interconnected to each other via axons and dendrites at synapses. In ANN models, the nodes form the processing units and are interconnected to each other by connection weights. The acquired knowledge is stored in the interconnections and nodes as weight and bias values respectively. Each neuron has a summation function which computes the product sum of the connection weights and the associated inputs. The neuron's activation function squashes the results to produce the output.

This study aims at developing an early warning system for type-2 undiagnosed diabetes and pre-diabetic cases for our local population using ANN with Trans Nzoia County as a case study. The World Health Organization (WHO) estimates that diabetes prevalence in Kenya stands at 3.3% currently and projects it to rise to 4.5% by the year 2025 (Tiffany 2013). According to Kenya Diabetes Management Information (DMI) (http://healthservices.uonbi.ac.ke/node/842) the figure stood at 1.8 million as at June 2013, however, going by IDF's findings, for every one diagnosed case there are two undiagnosed cases, the actual figure might be much higher (International Diabetes Federation [IDF] 2007). Clearly, if drastic measures are not taken to curb the escalating prevalence, diabetes is threatening Kenya's attainment of her millennium goals by attacking the most productive population and therefore lowering the GDP (Ministry of Health and Sanitation [MHS] 2010). One approach of managing the situation is to increase public awareness and encourage as many people as possible to test for diabetes so that timely medical attention can be sought to avoid diabetes related complications, prevent or delay its onset.

This study intends to take advantage of the current ubiquity of computers at workplaces, homes, social places and health centers and the high level of computer literacy among the Kenyan population to build an ease to use system on ANN that can detect pre-diabetics and undiagnosed diabetics from simple easily obtainable signs and symptoms.

Although some investors have already developed similar systems elsewhere, most of them are built using complex variables that cannot be easily understood by an average individual and are based on foreign populations' data. This narrows the usage of such systems to only experts in the medical sector. This study aims to use easily obtainable variables which require less or no calculation at all to increase the users of the system. The targeted variables include height, weight, waist circumference, age, sex, family history, hypertension, Body Mass Index, blood pressure and race/ethnicity.

1.2 Problem statement

Rapid urbanization, modernization, market globalization and economic transitions witnessed today have led to sedentary lifestyle and increased engagement in unhealthy diets due to availability of cheap chunk food, busy schedules and alcoholism. These compounded by scarcity of health facilities, lack of enough competent health personnel and low public awareness on diabetes has led to the escalating number of pre-diabetic and undiagnosed diabetics. Consequently stroke, cardiovascular diseases, blindness, lower limb amputations, kidney failure and other diabetes related complications are on the rise among the productive population lowering productivity and instituting unnecessary cost in the long run reducing Kenya's ability to reach her millennium goals. The most cost effective method to curb the escalating prevalence and the associated complications is to identify pre-diabetics and undiagnosed diabetics early enough and take proper measures to delay or prevent diabetes or control it before the onset of severe, expensive and irreversible complications. Unfortunately, there is lack of awareness among the public, inadequacy of competent health workers and facilities to meet the increased diabetes screening demands (Ministry of Health and Sanitation [MHS] 2010).

1.3 Purpose of the study

The purpose of this study is therefore to identify simple easily obtainable explanatory variables upon which a type-2 diabetes early warning system is built to publicize the courses of the disease and advise the affected people to undergo necessary medical attention on time.

1.4 Objectives

The objectives of this study are to:

a) Determine simple, easily obtainable variables that best distinguishes diabetics and pre-diabetics from healthy people among the residents of Trans Nzoia County

b) Develop an easy to use type-2 diabetes early warning system using a Particle Swarm Optimized trained MLP model using variables identified in objective (a) above as inputs

c) Determine the performance in terms of accuracy, sensitivity and specificity of the neural network model on local type-2 diabetes dataset with a view of establishing its viability as a decision support tool for local clinical diabetes diagnosis

1.5 Justification

The high prevalence rate of Diabetes Mellitus is lowering Kenya's productivity therefore reducing her ability to attain the millennium goals by affecting the productive age and causing expensive, social and health degrading complications such as blindness, renal failure, lower limb amputation and cardiovascular diseases.

Though the government through the Kenya National Diabetes Strategy (KNDS) 2010 – 2015 has outlined improvement on early diabetes detection as one of its key objectives to date very little impact has been felt on the ground. The test kits are still expensive making them unaffordable to most people. In addition, with the already overburdened public medical laboratory facilities, few screening is done in our public hospitals. The situation is further compounded by lack of enough trained personnel to deal with diabetic cases (Ministry of Health and Sanitation [MHS] 2010).

Although awareness campaigns are done through forums such as the world diabetes day, little has been achieved as demonstrated by the preliminary study in which although 96% of the respondents have heard of diabetes, 60% of them neither knew the causes nor the mechanisms of avoiding the disease. The great lack of awareness is also indicated by the Kenya National Diabetes Strategy 2010 – 2015. For increased screening and awareness we need an inexpensive, easily accessible and usable yet accurate and adaptive system for diabetes screening. The current ubiquity of computers makes this system easily accessible and with increased literacy among the Kenyan population this tool will really boost the people's awareness of the disease, increase screening which will in turn make more people aware of their diabetic status and seek proper medical redress where possible. The system's explanatory variables are easily comprehensible to an average Kenyan making it easy to operate as well as interpret the test results.

In addition, as more pre-diabetic and diabetic cases are disclosed, the government and other stakeholders will get a clearer picture of the disease situation in Kenya and therefore enable them to formulate policies and other measures to curb the disease's effects. With 60% of the cases undiagnosed (International Diabetes Federation [IDF] 2007), the government as well as the subjects are unable to plan and control the devastating complications posed by the disease. This tool is therefore handy in lowering undiagnosed cases and delaying or preventing the onset of the disease.

1.6 Assumptions, limitations and delimitations

This study is delimited to people living in Trans Nzoia County in Kenya. This county was chosen due to its cosmopolitan nature making the findings easily applicable to the rest of the country. Secondly the County has both the rural and urban aspects, with diabetes prevalence being related to demographic and lifestyles, the study is likely to capture the effects of the two. The study assumes that hospitals within the county have competent diabetes experts, sound procedures and testing equipment that yields accurate diabetes test results.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In the recent past, the ramifications of type 2 diabetes mellitus have been immensely felt world over including deaths, poverty and reduced productivity (International Diabetes Federation [IDF] 2007), as a result many studies have been carried out to mitigate these effects. Among the most pursued approach is the diagnosis of the disease early enough to prevent or delay its onset and for those already affected, reduce the effects of the associated complications. This chapter describes type-2 diabetes mellitus, chronologically examines the approaches exploited to enhance early diagnosis and later describes the proposed diagnostic model based on Particle Swarm Optimized ANN.

2.2 Type 2 diabetes mellitus

Diabetes Mellitus (DM) is a metabolic disorder characterized by chronic high blood sugar levels due to disturbances in proteins, carbohydrates and fats metabolism caused by either the body's inability to produce sufficient insulin or body cells' resistance to insulin action or both. If not controlled on time, diabetes mellitus leads to very debilitating chronic complication such as blindness, heart and blood vessels complications, kidney failure, nervous system complications, foot ulcers and lower limb amputation. In its acute state diabetes mellitus exhibits fatal complications such as diabetic ketoacidosis (DKA) resulting from high accumulation of ketone bodies in the blood and hyperosmolar hyperglycemic state (HHS) previously known as non-ketotic coma caused by too much dehydration as a result of high accumulation of glucose in the blood.

There are three main types of diabetes mellitus: -

2.2.1 Type-1 diabetes Mellitus

This type of diabetes mellitus is caused by total inability of the pancreas to secrete insulin. Insulin is a hormone that stimulates the body cells to take in blood glucose for energy generation. The inability of the pancreas to produce insulin occurs when the body's immune system auto attacks the beta cells of the pancreas, responsible for insulin production, and totally destroys them. The actual cause of this self-destruction is unknown and cannot be prevented once it has started making it difficult to prevent or even delay this type of diabetes. To manage it, patients are injected with insulin. Type-1 diabetes accounts for about 10% of DM cases and occurs mostly in children.

2.2.2 Type-2 diabetes Mellitus

Type-2 diabetes mellitus occurs when either the pancreas is unable to produce sufficient insulin for the body's use or sufficient insulin is produced but the body cells resists its action causing the elevated blood sugar levels. It is the most common type of diabetes accounting for about 85% of DM cases (Tiffany 2013). Patients with this type of diabetes are subjected to drugs and where there is insufficient insulin production they are injected with insulin to lower blood sugar levels.

2.2.3 Gestational diabetes Mellitus

Lastly, gestational diabetes mellitus occurs in expectant mothers but disappears immediately after delivery. Most women who suffer from gestational diabetes develop full blown type-2 diabetes mellitus later in their lives. Gestational diabetes accounts for about 5% of DM cases. Its causes are still unclear (Tiffany 2013; International Diabetes Federation [IDF] 2011). The fact that the causes of both type 1 and gestational diabetes are unclear at the moment makes it difficult to prevent and control them; however, they are less frequent accounting for between 10% to 15% of the DM cases compared to type-2 diabetes which accounts for between 85% to 90% of DM cases. This means that if we manage to control type-2 diabetes greater impacts will be felt. That is the main reason that has made this study to concentrate on type-2 diabetes mellitus.

2.3 Controlling type-2 diabetes mellitus

As indicated in section 2.2.3 above the overall prevalence of diabetes mellitus can be greatly lowered if type-2 diabetes is controlled. While it is ease to diagnose both type-1 and gestational diabetes, it is a bit difficult to diagnose type-2 diabetes mellitus due to the fact that in its early stages it is clinically silent and can remain asymptomatic for a period exceeding ten years (Shankaracharya et al. 2011). Most of the time medical practitioners make wrong diagnosis and eventually end up attending to the associated complications for example prescribing spectacles to a diabetic patient with visual problems therefore worsening the subject's condition instead of addressing the real problem.

Studies have shown that if diagnosed and attended to early enough type-2 diabetes can be delayed or prevented altogether. It has also been shown that if detected early and proper medical attention sought, the effects of the associated complications can be reduced (Shankaracharya et al. 2011*)*.

The problem is such an early detection is still elusive for our local population. First, incompetent medical personnel make wrong diagnosis, secondly, since the disease is asymptomatic many people don't see a reason as to why they should seek medical attention until the complications are so much pronounced and thirdly many people are unaware of the disease since most effort has been directed towards communicable diseases (Ministry of Health and Sanitation [MHS] 2010). There is therefore a need to have a simple system that can be available in people's homes, work places among other places where computers can be found to encourage self-testing and where the results indicate a higher risk of suffering from the disease proper medical attention sought.

2.4 Diabetes Risk Scores as tools for early type 2 diabetes diagnosis

Numerous solutions have been put forward to enable early type-2 diabetes diagnoses in order to prevent or delay diabetes or avoid the complications associated with it. To start with several diabetes risk scores have been proposed such as the Finnish Diabetes Risk Score (FDRISC), Atherosclerosis Risk in Community (ARIC), and the Epidemiological Study on Insulin Resistance Syndrome (ESIRS). Jou et al (2009) carried out a study on ten such risk factors and found that when the Risk scores are fed with simple and easily measurable clinical data they can be used to identify subjects at high risk of type-2 diabetes with sufficient accuracy and that their performance do not differ from each other by a significant margin in terms of accuracy, sensitivity and specificity. However, the involvement of human beings in determination of criteria and score subject them to human errors therefore lowering their reliability (Shankaracharya et al. 2011*)*. It is this limitation that made researchers start using machine language and statistical methods to develop automated systems that can predict ones risk of suffering from type-2 diabetes.

2.5 Machine learning tools for early type 2 diagnosis

The above limitation made researcher resort to machine learning approach to predict the risk of a subject suffering from type-2 diabetes. Several machine learning models have successfully been tried. Using the Third National Health and Nutrition Examination Survey dataset (NHANES III) Heikes et al (2009) contacted a study based on eighteen simple, easily measurable variables that neither requires laboratory tests nor professional expertise such as age, waist circumference, height and sex. The tool was built to encourage self-diabetes testing for the U.S population. The tool calculates the probability of either an individual being undiagnosed diabetic or pre-diabetic. For performance comparison purposes both logistic regression and classification trees analysis (CART) were used to process the results. Owing to the ease of use and presentation, CART was chosen to build a paper based tool for detection of diabetics or pre-diabetics and a specificity of 75% was obtained. Using v-fold cross examination validation, the study demonstrated that the tool is viable for detecting pre-diabetics and undiagnosed diabetics for U.S population. Despite its ease to use and associated accuracy, the tool might not be as user friendly as its electronic equivalent and can too result in erroneous outcome owing to its manual nature that elicits a lot of human decision making.

A significant amount of research work has been done on Pima Indian dataset (PID) using various strategies and a variety of both statistical and machine learning algorithms as demonstrated in a review work by Shankaracharya et al (2011) including logistic regression, support vector machine, Artificial neural networks, K-NN among others. Owing to high interrelation in the nature of clinical data and the ability of ANN to outperform the classical methods on such variables, neural nets showed better performance as compared to the rest in predicting type-2 diabetics.

Kazemnejad et al (2010) contacted a research to compare the performance of artificial neural network and binary logistic regression on discriminating between diabetic and pre-diabetic cases from non-diabetic cases. 7222 participants of between 30 to 88 years of age were used in the study. Though there was no significant performance difference between the two techniques, Artificial Neural Network had a better ROC value of 0.770 compared to 0.760 of the binary logistic regression.

Mansour et al (2013) contacted a study to analyze the efficiency and predictive power of Artificial Neural Network, Logistic Regression and Discriminant Analysis in determining the risk factors of type-2 diabetes. The data consisting of 100 diabetics and 100 pre-diabetics was collected from 17 rural health centers in Kermanshah city. Artificial Neural Networks attained the highest sensitivity of 95.2%.

2.6 PIMA Indian Diabetes Dataset

Despite of the numerous successes achieved in the above machine learning techniques in diabetes diagnosis, most of the studies focused on PID dataset. The PIMA dataset was obtained from the larger dataset held by National Institute of Diabetes and Digestive and Kidney disease and mainly contains women aged 21 years and above living near Phoenix Arizona, USA. It contains 268 positive cases and 500 negative cases (Kamer & Tuley 2011)

. Variables considered when classifying diabetics include: -

- Number of times pregnant
- Plasma glucose concentration at 2 hour in an oral glucose tolerance test
- Diastolic blood pressure (mmHg)
- Triceps skin fold thickness (mm)
- 2-h serum insulin (IU/ml)
- Body mass index (weight in kg/height in m)
- Diabetes pedigree function
- Age (years)

Although the dataset is well validated, it is based on variables which require expert involvement in both measurements and interpretation of the results. This restricts the number of people capable of using the resulting systems to only experts in diabetes or health care personnel. Secondly, the dataset is mainly composed of women aged 21 years and above leaving out men.

In addition studies have indicated that DM is also dependent on race/ethnicity (Abdulbari et al 2005; International Diabetes Federation [IDF] 2007; Hamman 1992). Cultural practices such as the type of food an ethnic community is used to eating and marriage coupled with genetic ties within members of the community can greatly influence individual's predisposition to diabetes *(*Abdulbari et al. 2005). With each country having its own unique set of ethnicity, clearly such systems cannot be applied to every nation and in this case our local Kenyan population. Therefore to get an effective model on a particular population, the model is supposed to be built on the local dataset of the population in question. It is this fact that informed the decision to build an ANN system based on data collected from the local hospitals.

2.7 Artificial neural networks

An Artificial neural network is a highly interconnected set of simple processing elements (also called nodes or units or neurons). This machine learning model attempts to simulate the structure and way of operations of biological nervous system components such as the brain, the spinal cord and the peripheral ganglia. Each neuron has three basic functions namely multiplication, summation and activation. Each input signal is multiplied by the corresponding interconnection weight on entry into the neuron, the summation function then sums up all the products and the bias to produce the weighted sum. The weighted sum is then fed into an activation function to finally produce the neuron's output. The figure below illustrates this process.

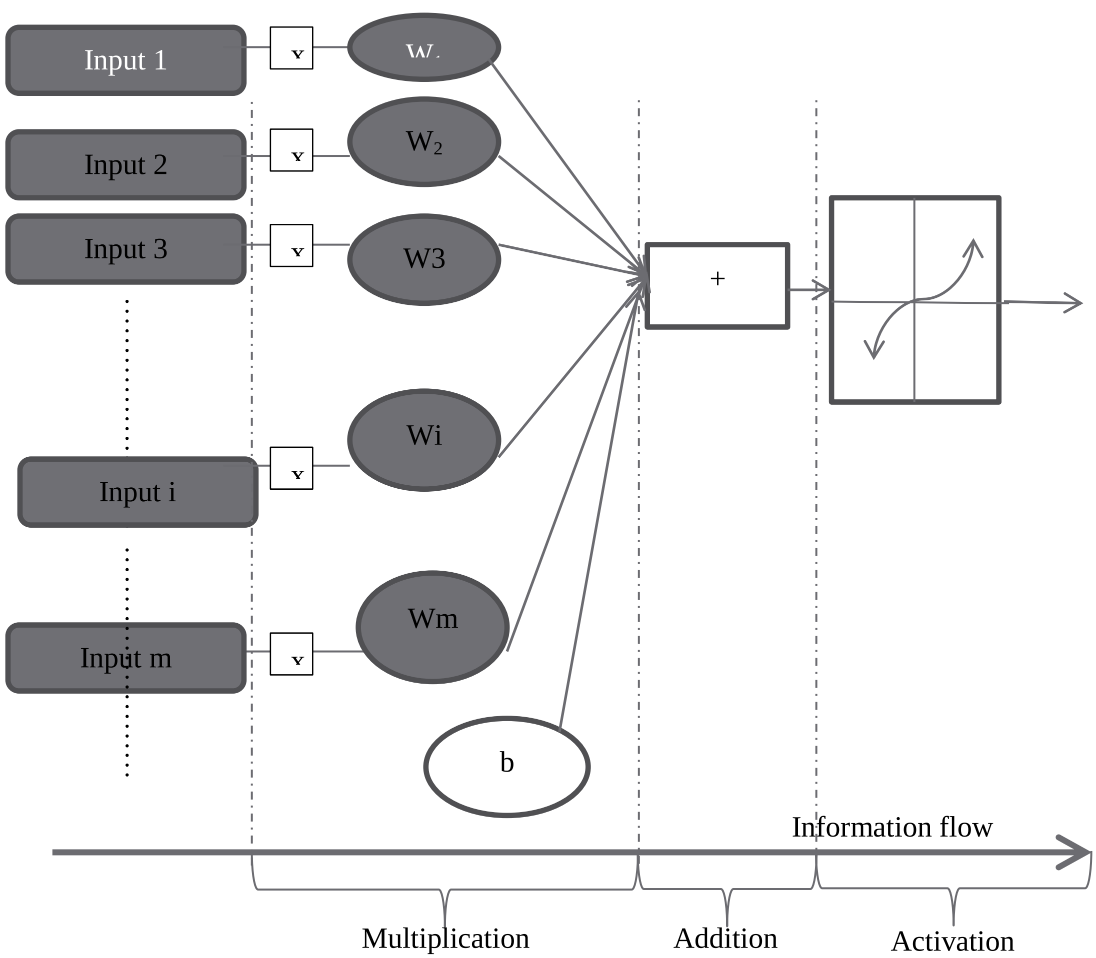


Fig. 2.1: Neuron basic operations

The real power of the ANN is manifested as the emergent characteristic of the whole network. They are praised for their high tolerance to noisy data. However, their main disadvantage is poor interpretability of the result since it is difficult for human beings to understand the rationale behind their classifications (Asha et al. 2011)

2.7.1 Artificial neural network topologies

The term topology in ANN refers to the way the nodes in the network are interconnected. This can also be referred to as a graph or the network's architecture. There are a number of ways to interconnect the network depending on the nature of the problem being solved. The most basic topologies are feed forward and recurrent topologies. The neurons in the ANN are grouped into layers. Members of one layer share the same characteristics. When signals flow only in one direction from the input layer through the hidden layers, if any, to the output layer the topology is referred to as a feed forward topology while when signals flow back from the higher layer to the lower layer or loop back on the same neuron the topology is referred to as a recurrent topology.

Artificial neural networks have at least two layers. The input layer is not usually counted therefore a network with two layers i.e. the input and the output layer is referred to as a single layer perceptron while a network with hidden layers (layers between the input and the output layer) is referred to as a multilayer perceptron. The figure below shows a multilayer perceptron with a single hidden layer. Single layer perceptions are only capable of classifying linearly separable data.

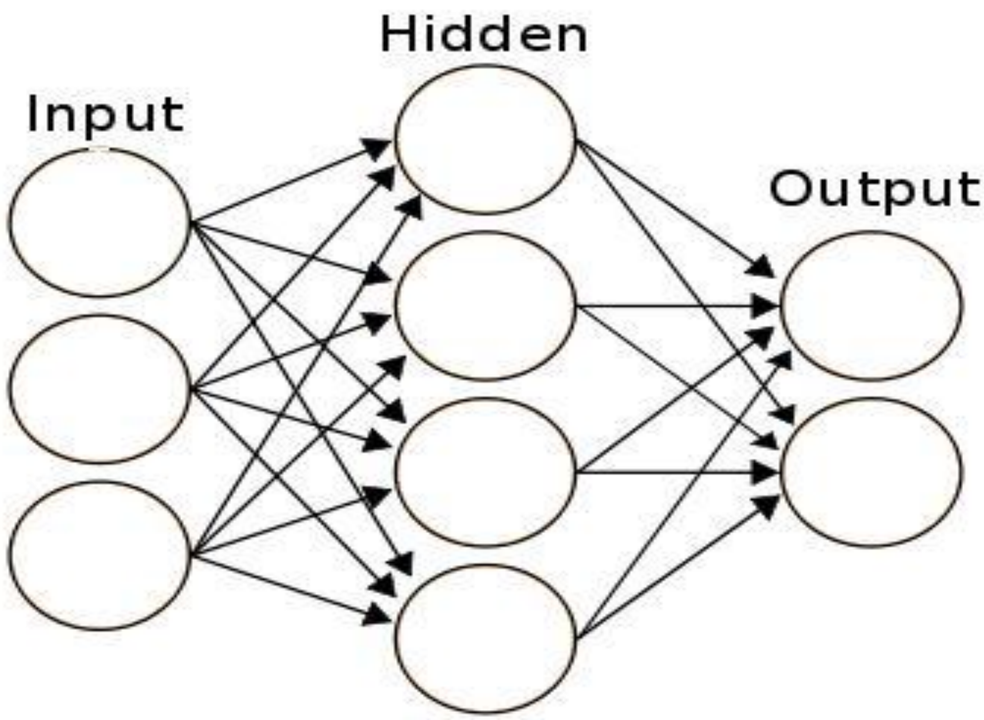


Fig. 2.2 Multi-Layer perceptron architecture (Lakshmi and Subadra 2013)

To classify nonlinear problems multilayer perceptions are used. Early studies have demonstrated that in most cases a single hidden layer is sufficient (Carlo 2007 & Krenker et al. 2011).

2.7.2 Learning in Artificial Neural Networks

An Artificial neural network learns by adjusting its connection weights and biases (free parameters) to minimize the cost function for unsupervised learning or reducing the difference between the outputs it produces and the corresponding target outputs. Learning can be supervised in which case the dataset used for training the ANN model is labeled with the correct answers. The main task in this type learning is therefore to adjust the model's free parameters to produce answers as close to the desired output as possible. It is analogous to learning under the supervision of a teacher. Learning can also be unsupervised, whereby there exists no external teacher but a cost function guides the network in identifying patterns within a given dataset.

2.7.2.1 Supervised Learning

This learning approach attempts to find the input-output relationship based on a labeled dataset. The input data is fed into the learning system and the corresponding output generated. The generated output is then compared with the labeled or target output to find the deviation error. The error is thereafter fed back into the learning system so that the free parameters are adjusted accordingly to reduce it. The process continues until a certain threshold difference between the generated output and the desired output is attained. Figure 2.1 below illustrates the process.

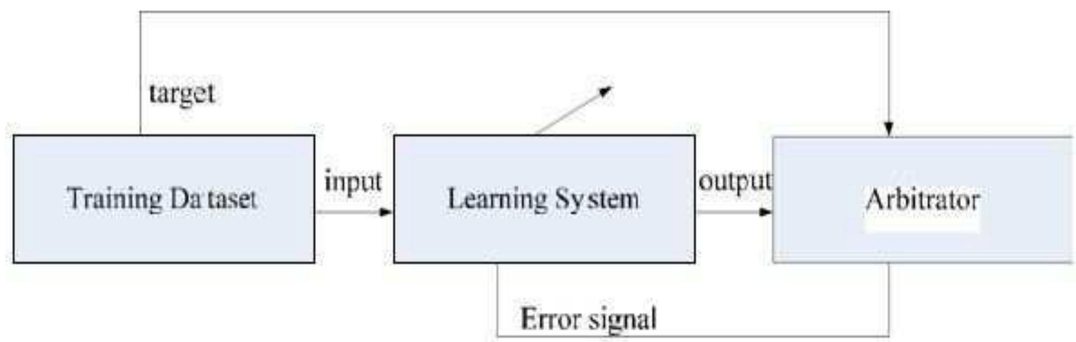


Fig 2.3 Supervised Learning Process (Kumar 2013)

2.7.2.2 Unsupervised learning

Unsupervised learning is a technique in machine learning in which the free parameters of an Artificial Neural Network are set based on a given dataset and a cost function which needs to be minimized or maximized. The cost function is determined by the task formulation. Unsupervised learning is most applicable to estimation problems domain including but not limited to statistical modeling, compression, blind source separation and clustering (Krenker et al. 2011).

The key objective in unsupervised learning is to find out how the data is structured. Clustering is one of the most common types of unsupervised learning. An attempt is made to categorize data into different classes from their similarity. One of the most common applications is the Self-organizing maps

2.8 Neural Network Training Algorithms

As pointed out in ***section 2.4*** above neural networks outperform other machine learning and statistical techniques in diagnosing type-2 diabetes mellitus. However, the performance of ANN is greatly dependent on the algorithm used to train the network. Back propagation (BP) algorithm is the most commonly used algorithm for determining weight combination in ANN models due to its power and simplicity (Lakshmi and Subadra. 2013), however, studies have shown that the algorithm does not guarantee reaching a global minimum in multimodal problems therefore negatively impacting on its generalization ability (Shankaracharya et al. 2011 & Shanker 1996). To address this problem researchers have tried using other algorithms such as genetic algorithms (GA), however, its greedy approach of dropping the dismally performing chromosomes through the deployment of the survival for the fittest principle make them lose previous experience which could otherwise be handy in building a better solution. Furthermore, GA assumes that a crossover between two best chromosomes delivers better performance which is not usually the case making them less suited to this task at hand. Particle Swarm optimizations (PSO) promises better results in such problems as opposed to GA, previous knowledge is kept and individuals who go beyond the optima boundaries are simply brought back instead of being dropped completely retaining previous knowledge. The social component of the PSO attracts the particles towards the global minima (Eberhart & Kennedy 1995) enhancing the system's stability.

2.8.1 Back propagation algorithm

In this algorithm network weights are first initialized to small numbers for example between –1.0 and +1.0. The input vector is then applied to the network during the forward pass in which the output is calculated. Most often the produced output is completely different from the target output owing to the randomness of the initial weights. An error (target-actual) is then computed and used mathematically to change the weights such that the subsequent error gets smaller. Otherwise stated, the outputs are made to be as close as possible to the desired output. The process is repeated until the minimal error threshold is attained.

2.8.2 Particle Swarm Optimization

This is a nonlinear optimization algorithm originally designed by James Kennedy and Russell Eberhart (Kennedy & Eberhart 1995). The algorithm is modeled on a coordinated group behavior such as flocks of birds, school of fish and a swarm of insects. Since its advent, it has been widely tested and proven to be efficient in finding optimal or near optimal solutions in large search spaces (Anthony & Gerry 2000). Each individual (Particle) has a virtual current position in the search space, a current velocity and the best found individual position as well as the best found group position. Each particle's virtual position represents a possible solution to a minimization or a maximization problem.

The algorithm is iterative and in each iteration, every particle moves to a new location possibly representing a better solution. The equation below is used to update the particle's new position:

$$\mathbf{v}_{(t+1)} = (w * \mathbf{v}_{(t)}) + (c1 * rd1 * (\mathbf{p}_{(t)} - \mathbf{x}_{(t)}) + (c2 * rd2 * (\mathbf{g}_{(t)} - \mathbf{x}_{(t)}) \quad (1)$$

$$\mathbf{x}_{(t+1)} = \mathbf{x}_{(t)} + \mathbf{v}_{(t+1)} \quad (2)$$

where in equation (1), $v_{(t+1)}$ is the new velocity vector, w is the inertia, $\mathbf{v}_{(t)}$ is the current velocity, c1 is the cognitive weight, rd1 is a random variable greater than or equal to zero but strictly less than 1, $\mathbf{p}_{(t)}$ is a vector of the best position ever reached by the current particle, $\mathbf{x}_{(t)}$ is the current particle position, C2 is the social weight, rd2 is a random variable greater than or equal to zero but strictly less than 1, $\mathbf{g}_{(t)}$ is a vector of the best position ever reached by all the particles.

In equation (2), $\mathbf{x}_{(t+1)}$ represents the new particle position.

The update process continues until a certain objective function value is attained. In this case a certain minimum value of the mean square value is reached or maximum numbers of epochs are exhausted in case of no convergence. It promises many advantages over other optimization algorithms used to train Artificial Neural Networks including: -

1. Capability of solving optimization problems with non – continuous solution domain
2. No restriction on the transfer function since they can even be used with non-differentiable transfer functions
3. The inherent explosive nature of PSO enables them to escape the local minima
4. They don't lose some individuals who at one point don't perform well on the fitness function implying that no experience is lost during the training process.

Though they suffer from the curse of dimensionality, the disadvantage does not affect this study since there is no large dimensionality.

2.9 Proposed training model

To improve the accuracy of the ANN model, PSO algorithm is used to explore the global minima while BP algorithm has been used to exploit the global minimum. This approach increases the chances of locating the global minimum and with excellent exploitative ability of BP algorithm, the best point of the global minimum will be determined bettering the performance of the network.

2.9.1 Model rationale

The main reason for choosing ANN as explained in the earlier sections lies in the high predictive ability compared to other machine learning techniques in medical diagnosis (Kazemnejad et al. 2010; Shankaracharya et al. 2011; Mansour 2013; Kamruzzaman 2006). Their main short coming of converging at a local minimum on multimodal problem when trained by the back propagation algorithm has been eluded by first exploring the solution curve using better explorers of such nature of solution, Particle Swarm Optimization algorithms, thereafter the back propagation algorithm is deployed for its best exploitative ability to exploit the explored global minimum so that the network settles at the best global minimum position.

CHAPTER THREE

METHODOLOGY

3.1 Introductions

This study aims at building an ANN system capable for identifying pre-diabetics and undiagnosed diabetics so that appropriate measures are taken either to avoid or delay the onset of type-2 diabetes for people with impaired glucose tolerance and for those who are already diabetic appropriate blood sugar control measures administered to avoid the onset of type-2 diabetes related complications such as blindness, lower limb amputation, kidney failure and even death. As already indicated in the introduction section the prevalence of diabetes in Kenya is high and is still growing therefore there is a need to find out some of the predisposing factors and caution the affected individuals.

3.2 Initial survey

First diabetic public awareness survey was contacted to find out whether the members of the public are aware of the causes of diabetes, the risk of them becoming diabetic, the government's strategy to curb diabetes high prevalence as well as the individual's role in preventing himself from becoming diabetic. During this survey randomly selected members of the public within Trans Nzoia County were interviewed orally to find out answers to the above questions.

Another interview with diabetes experts within Trans Nzoia County was also contacted in which the experts were interviewed with the aim of establishing the nature of diabetes records in the respective health facilities, the procedures employed in dealing with new cases of diabetes in order to establish a data collection plan.

3.3 Data collection and pre-processing

Before the start of the process, a survey was carried on various diabetes clinic dataset within Trans Nzoia County to establish the nature of the data to be collected. Though many hospitals are computerized, data stored in electronic form lacked important fields such as patients' height, ethnicity, weight, blood pressure among others. The only data stored in most of these clinics included patients' name, admission date, their age, diagnosis results, prescriptions and the treatment offered. Although this information is significant to hospital administration, most of the computer systems having been established to improve revenue collections in hospitals were not suitable for research purposes which meant that the only possible option was to collect data from the patients' manual files. Though a bit cumbersome a significant amount of data was collected, however, most files had positive cases since negative cases are not captured in diabetes clinics. The only negative cases present were for those people whose random sugar level indicated that they were suffering from diabetes or were pre-diabetic but on contacting a more accurate fasting level sugar test, the results indicated otherwise.

In addition, there was a lot of missing fields such as waist circumference, waist to hip ratio, whether the subjects take alcohol or not, whether they smoke or not, the family diabetes history, whether the patient has suffered from Gestational diabetes before among others. In some cases, computer systems were absent and due to incompetence and fatigue among the medical staff in the facilities, scanty patients' data was recorded making the collection process impossible.

This being a qualitative research whose main focus is pattern extraction, another approach was designed to collect more negative cases. Any patient who was directed to test for random blood sugar level test from the labs was taken to the diabetes clinic where other details such as waist circumference, blood pressure e.tc relevant to the study were recorded irrespective of the results.

Known diabetic cases and pregnant women were excluded from the study because some women experience gestational diabetes mellitus which disappears after delivery therefore the high blood sugar could wrongly be classified as type 2 diabetics. Known cases were excluded since after a subject knowing his or her diabetes status he would try to control it making some patterns not to come out clearly.

3.4 Data preprocessing and data encoding

The collected data was passed through various data processing steps before finally being used in the study.

3.4.1 Missing data

Missing data was calculated from related fields of the same record for example where there existed Body Mass Index (BMI) and weight but height is missing the height could be derived from the square root of the quotient of weight and BMI.

In cases where the missing data could not be derived from other fields, the whole record was dropped. However, where there was adverse missing of data in a particular field, the whole field was dropped instead. As a result, fields such as waist circumference were dropped from the dataset.

3.4.2 Data classification

After executing the above deletion, the data was then divided into three groups: training, validation and test data. The training dataset was used to train the dataset, validation set used to test the performance of different model parameters while the test data was used to test the final model for accuracy, sensitivity and specificity. The Microsoft Excel randBetween function was used to randomly categorize the dataset into the three sets. For every record a random number from 0 to 10 was generated. Since the training set was to take 70% of the data and 15% both for validation data and test data, records which were assigned values in the range of 0 to 7 were classified as training, while those greater than 7 but less than or equal to 8.5 classified as the validation set while the remaining records were classified as the test set. It resulted into 233 training set records, 37 records of the validation set and 77 records of the test set totaling to 347 records.

3.4.3 Feature selection

After classifying the dataset, the training data was subjected to R decision tree software to select the best predicting attributes. The concept of information gain was therefore used to select the best distinguishing explanatory variables. Variables that appeared in the resultant tree were considered to be good discriminators while those left out were considered trivial discriminators and were thus dropped from the dataset.

3.4.4 Encoding of categorical data

Ethnicity, gender and the target variable (i.e. negative (-) and positive (+)) are categorical variables. In order to make all of them to be equally treated, binary encoding was applied as follows: -

Gender was represented by two input neurons, one for male and another one for the female gender. The male gender neuron is turned only when the instance has a male gender while the female neuron is only turned on when the gender is female. A neural was deemed to be on when its output is set to “1” and off when its output is set to “0”.

The same technique was used to represent positive and negative target variable values where a “1” represented a positive and a “0” represented a negative.

On ethnicity, n-1 binary numbers were used to represent the present ethnic communities where n represents the total number of different ethnicities available.

3.4.5 Scaling and normalization of numerical data

Numeric data such as systole and diastole blood pressure, age, weight and BMI were scaled down to the range [0, 1) using the formula: -

Scaled_ValueX = (x-Minx)/ (MaxX-Minx)

Where,

x is the value being scaled, **Minx** – is the least value in the range and **MaxX** is the highest possible value in the range and Scaled_ValueX is the new value to be used in the model.

The above formula is also used to detect the outliers and therefore remove them from the dataset.

3.5 Diabetes tests

Table 3.1 below provides the test criteria according to the World Health Organization standards (World Health Organization [WHO] 2006).

Table 3.1: Diabetes testing criteria

Subject type	Random Blood Sugar Level	Fasting Blood sugar level
Diabetics	>11mmol/l	≥7.0mmol/l (126mg/dl)
Pre-diabetics	>7.8 mmol/l and <11mmol/l	6.1 to 6.9mmol/l or (110mg/dl to 125mg/dl)
Normal	3.5 mmol/l to 7.8 mmol/l	3.5 mmol/l to 5.6 mmol/l

NB: The random blood sugar test is not a certain method of testing diabetes since the test can be done when the subject has just eaten high glycemic index food temporarily elevating blood sugar levels. Therefore after a subject testing positive with RBS s/he is advised to do fasting plasma test for confirmation.

3.6 Model building

The model was built via a prototype. The prototype was tested for various model parameters to find the best parameter combinations.

3.6.1 Initial model building

The selected features above formed the inputs of the neural network ensemble with the input layer transmitting the input to the hidden neurons directly without any manipulation. A single hidden layer with a hyperbolic tangent transfer function was used. The outputs from the hidden layer were served as input to the output layer neurons. The hyperbolic tangent activation function was used for both hidden and output layer neurons. The pattern [10] represented a positive case while the pattern [01] represented a negative case. After convergence or exhaustion of maximum set epochs, the model used the current weight set as initial weights for the BP algorithm to further tune the global minima. The models prediction ability was subjected to a variety of sets of inputs variables selected on the strength of their closeness to the root of the classification tree above with the aim of improving on the ensemble's generalization ability. .

3.7 Analysis techniques

The performance of the network was mainly evaluated on accuracy, sensitivity and specificity. An epoch verses mean square error graph containing training curve, test curve and validation curve was plotted and to avoid over fitting the training stopped when test error curve was at the lowest level.

3.8 System requirement analysis

The system analysis was carefully done considering pertinent issues such as the various components of the system, the targeted system users, system accessibility and the purpose of the system. Several techniques were used including but not limited to analyzing interviews results, empirical methods and reuse of ideas gained from previous research studies.

3.8.1 Analyzing Interview results

Several interviews were done throughout the research process to answer various survey questions. They included interviews with the medical experts as well as interviews with the members of the public.

3.8.1.1 Interviews with medical experts

Several oral interviews were carried with various medical officers within Tran Nzoia County mainly to: -

i. Determine the nature of the medical records and the level of computerization in health facilities within the county
ii. The various attribute types used for testing diabetes
iii. The structure of hospital administration
iv. The inflow rate of diabetes patients

It was important to know the nature of medical records so that proper data collection methods could be employed. Electronic data make the data collection easier, faster and reduces errors. You just collect the data, validate it and reuse it but when the data is in manual files data collection becomes a challenge since you need to manually go through the manual files extract the relevant data, validate and finally transform it into the required electronic format. This requires more time and the data so collected is likely to contain some human errors.

3.8.1.2 Interviews with the members of public

The main aim of the study was to investigate the public awareness of diabetes mellitus, their level of education, and their level of computer literacy. The collected data informed the type of system interface used and the effectiveness of the system as a tool to facilitate diabetes awareness.

Randomly selected members of the public within Trans Nzoia County were orally asked simple questions about diabetes such as whether they have heard of the disease, the causes of the disease, measures that can be taken to avoid the disease, whether they have any of their relatives affected by the disease, whether they know of the Kenyan government's plan on Diabetes, Education standard and lastly whether they are computer literate or not. Only individuals aged 18 years of age and above were considered during the interview.

3.8.2 Analysis through empirical methods

Artificial neural network's performance greatly relies on the nature of the problem and model parameters used in the design. Some of the most important parameters include number and nature of the inputs, the number of hidden layer neurons, the activation function and the training algorithm employed. Unfortunately there is no universally accepted direct way of deriving most of these parameters; most of them are arrived at through a trial and error method.

Early studies have shown that the number of layers doesn't affect the performance of Artificial Neural Networks and that a single hidden layer is sufficient for a MLP, therefore only one single hidden layer was considered in this study. The main concern was the number of hidden layer neurons to be used in the model.

To achieve this, a prototype of the neural network trained with PSO was used. The maximum number of epochs was set to 10,000, the number of particles set to 20, the exit error set to 0.065 while the both the social weight and cognitive weights were set to 1.49445.

Other parameters arrived at after experimental investigations include the number of PSO particles, cognitive and social weight values.

3.9 System Design

The figure below shows the overview of the system components and how they relate to each other. Input data from the user is first normalized then encoded with appropriate technique as outlined in the data processing section. The output of which is fed to the ANN input layer for classification. The output so produced is encoded back to human understandable format.

NB: This is a simplified version with a few neurons for clarity purposes.

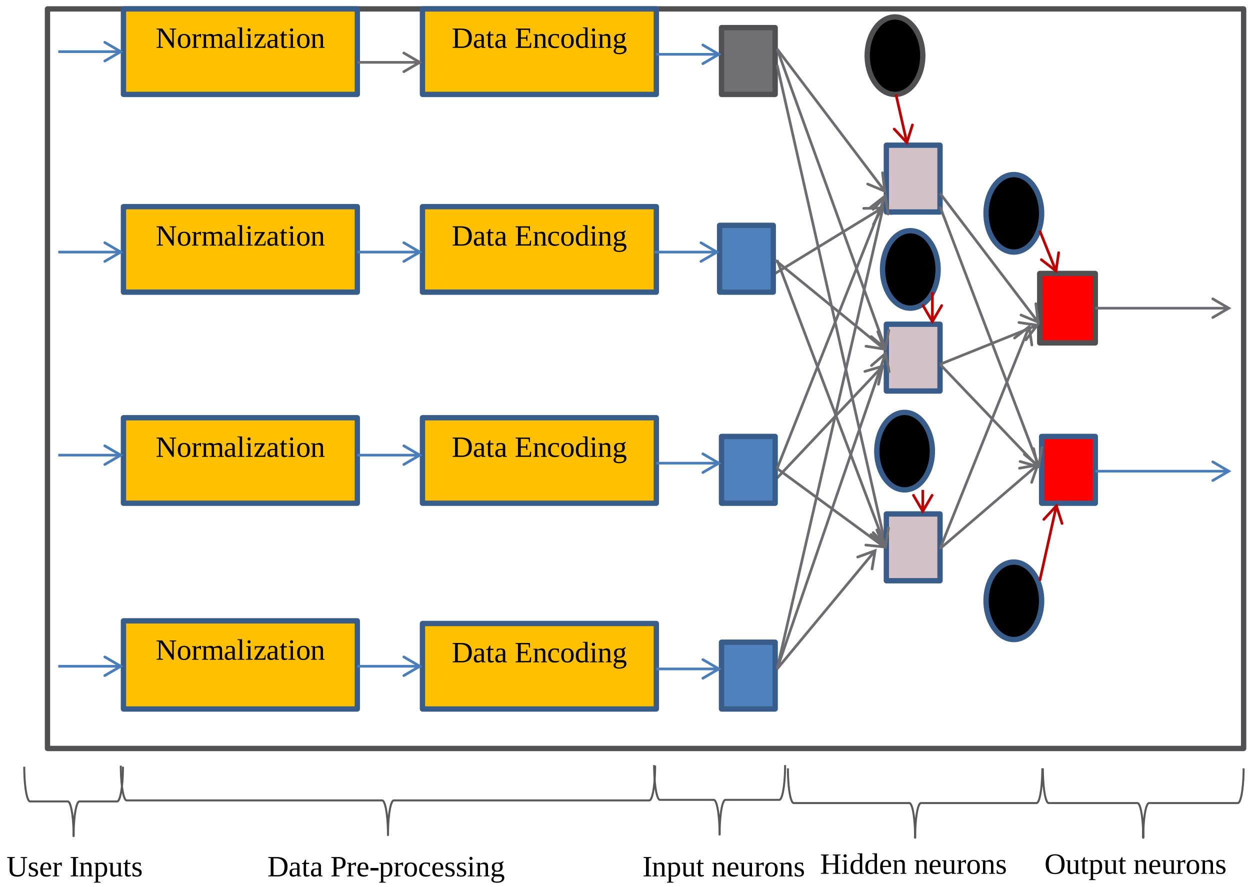


Fig. 3.1: System component layout

CHAPTER FOUR

IMPLEMENTATION, RESULTS AND DISCUSSION

The user inputs diabetes attributes through a graphical user interface shown below.

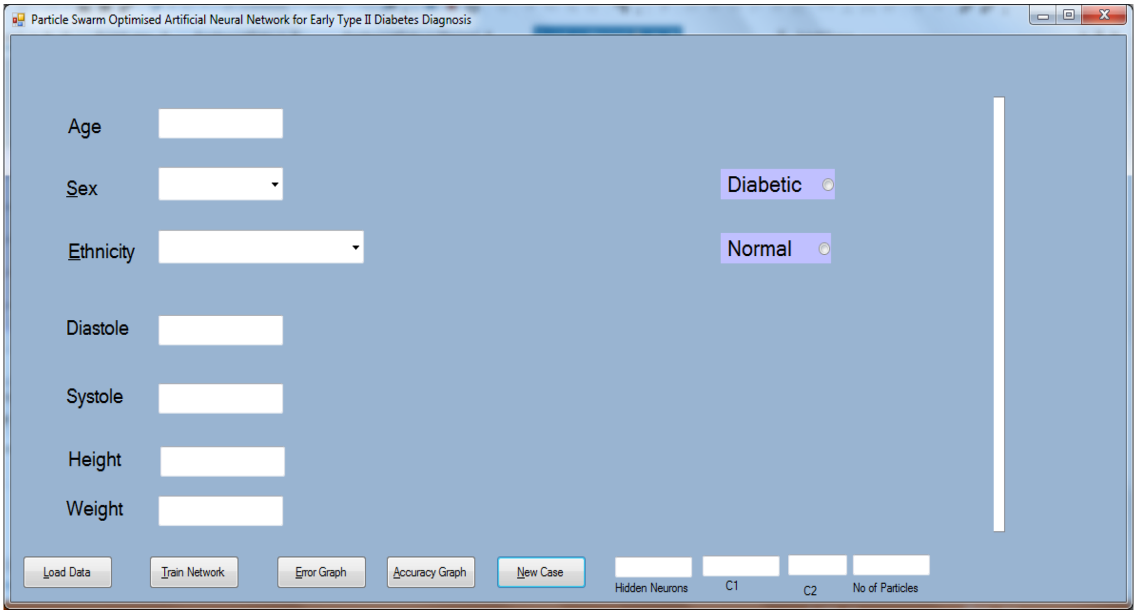


Fig. 4.1: User input layout.

The inputs are then normalized and encoded by various methods as explained in the data preprocessing section. The preprocessing outputs are then fed into the neural network model for processing.

4.1 System processing flow

The figure below shows the overview of the system's process flow

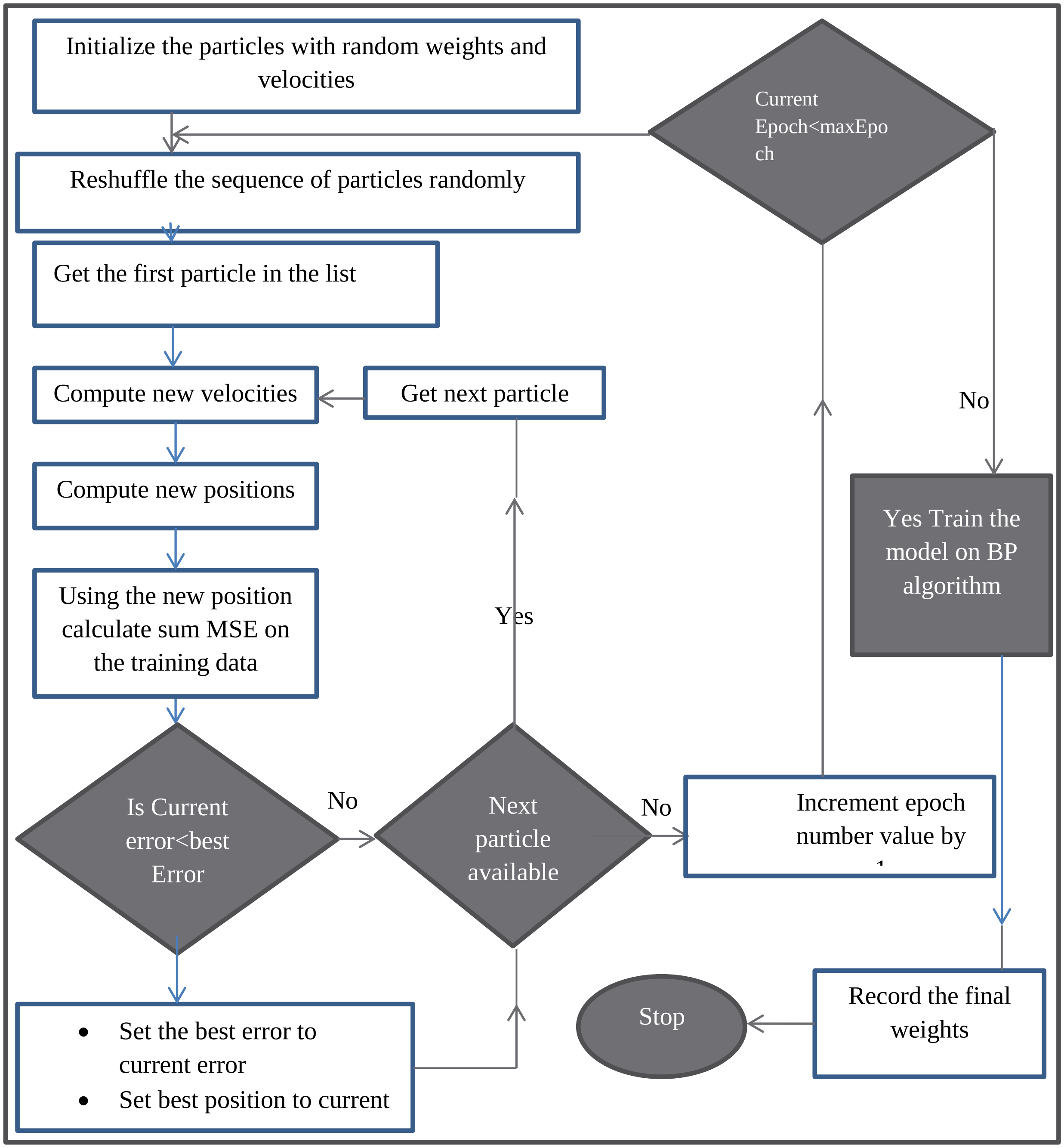


Fig. 4.2: Overall system flow chart

As illustrated in fig. 4.2 above, the model is first trained using PSO algorithm to locate the global minimum then latter tuned using the powerful back propagation algorithm to further improve on accuracy.

4.4 System parameters

After experimenting with the prototype, the following parameter values were found to be optimal for the network and therefore used to implement the system.

- **The social and cognitive weights –** this value determines the rate of convergence of the network. The best performing value for both cases was found to be 1.49445
- **Number of PSO particle-** the number of particles was set to 20, the smallest number of particles that show stable high generalization ability and as early studies have shown higher number of particles slow the model (Xiaohui & Eberhart 2002).
- **Input neurons –** the number of input neurons was set to 12 based on the best generalizing variables chosen
- **Hidden layer –** the network has one hidden layer with 11 neurons
- **Output neurons** – two neurons are used for the output layer.
- **The activation function**

  The input layer neurons feed forward the input signals without changing it to the hidden layer. However, both the hidden layer neurons and the output layer neurons do some simple computation to get their outputs. One type of computation that is common among most MLP is the summation of products of input signals and their associated weights. However, for the output to have some meaning in nonlinear problems, the sum has to be passed through a squashing function. Such a function is referred to as an activation function. There are several function used for this purpose.

  The table below shows some of the common activation functions.

Table 4.1: Some common activation functions (Lakshmi PK et al. 2013)

Name	Function
Binary Threshold	$y=1; x \geq 0$ $y=0; x<0$
Bipolar Threshold	$y=1; x\geq 0$ $y= -1; x<0$
Sigmoid	$y = 1/(1 + e^{-x})$
Linear	$y=x$
Saturating linear	$y=1; x\geq 0$ $y=0;x<0$ $y=x;0\leq x\leq 1$
Symmetric saturating Linear	$y=1; x>1$ $y= -1; x< -1$ $y=x; -1\leq x\leq 1$
Hyperbolic Tangent Sigmoid	$y = (e^{x} – e^{-x})/(e^{x} + e^{-x})$
Positive linear	$y=x;x\geq 0$ $y=0;x<0$

The choice of a particular activation function depends on the application problem at hand and the training algorithm the network uses. For back propagation algorithm only differentiable activation function are suitable (James 2013), limiting the functions to only sigmoid and hyperbolic tangent functions. The linear function is used for linearly separable problems. Sometimes the implementation platform restricts the type of activation function used. Some hardware for implementing ANN don't support some activation functions (Lakshmi et al. 2013). Some studies have indicated that TANH – TANH activation function combination for both hidden layer neurons and output layer neurons outperform the other differentiable functions (Bekir & Vehbi, n.d), therefore in this study the TANH – TANH activation function were chosen.

4.5 The Training process

Figure 4.2 above outlined the training process. Each input variable is represented by an input neuron. Each component of a split input variables like gender and ethnicity is represented by a neuron that makes the gender neuron to have two input neurons with the pattern [10] representing a male and [01] representing a female.

There are 12 input neurons, a single hidden layer with 11 neurons and two output neurons. Each hidden and output layer neuron has a bias connected. An output pattern [10] represents a positive while [01] represents a negative.

The network is fully connected with 167 weights (12*11 + 11*2) and 13 biases (13+2).

Both the hidden and the output layer neurons have hyperbolic tangent as their activation function. Figure 4.3 below illustrates the structure of a hidden layer neuron.

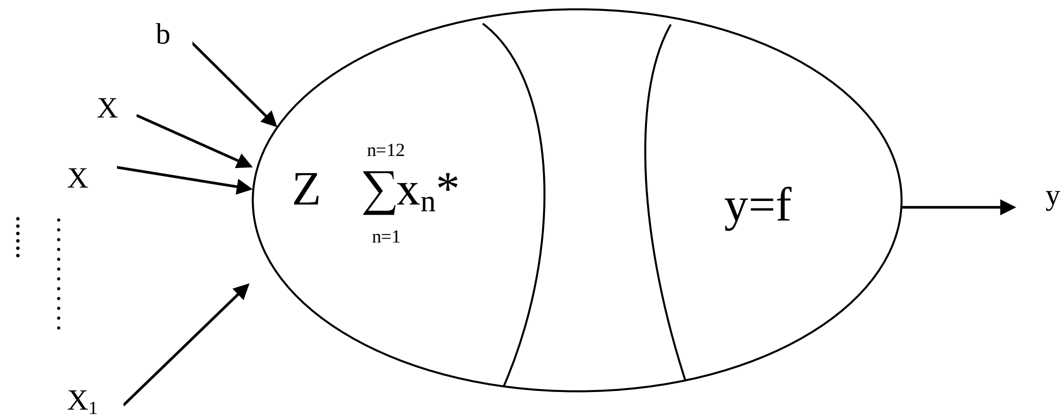


Fig 4.3: A hidden or output layer neuron structure

Fig. 4.3 above shows the hidden or output layer neuron structure, b is the neuron's bias and Z is the product sum of the inputs and their associated weights plus the bias. Z is then fed into the function f (.) to produce the neuron output y. f (.) is the hyperbolic tangent function for both hidden and output layer.

Each particle of the PSO has 137 dimensions describing the particle's position which are equivalent to the weights of the neurons. The objective of the training process is therefore to find the best weights combinations that minimize the mean squared error of the whole neuron network model.

The following equations was used to update the particle position

1. **V** (t+1) = w*v (t) + c1*r1*(**pBest (t) - x(t)**) + c2*r2*(**gBest (t) -x(t)**)
2. **X (t+1)= x(t) + v (t +1)**

Where,

In equation 1**,** **V** (t+1) is the new velocity containing, w is the inertia weight, **v** (t) is the current velocity, c1 is a constant in this set up it has been set to 0.729, r1 is a random component in the range [0,1), **pBest (t)** is the best position ever reached by the individual neuron, **x(t)** is the current particle position, c2=c1, r2 is a random component in the range [0,1), **gBest**(t) is the best position reached by the group members during the training. Equation 2 is used to calculate the new position, **X (t+1).** Vectors **v**, **gBest**, **pBest** and **x** have 137 components which represent the neuron weights and connected biases.

The maximum number of epochs is set to 100,000 and the exit Error set to 0.065 meaning that training can continue up to 100,000 epochs or when the MSE is equal or less than 0.065.

First the network is trained using PSO algorithm and on convergence or after 100,000 epochs it is subjected to a back propagation algorithm which tunes the network's global minimum further increasing its accuracy.

4.6 Implementation platform

There are many factors that need to be taken into account when deciding which platform to implement such a system. In this study accessibility, cost, implementation flexibility and the nature of functionalities to be implemented were considered. A decision had to be made on whether to use hardware verses software platform, whether to use neural network specialized software development tools or use the general programming languages. In the subsequent sub-sections of this section, the pros and cons for each of these platforms are discussed.

4.6.1 On chip implementation

The defining characteristics of ANN include high parallelism, dynamic adaptive ability and modularity, therefore a suitable platform for building this system ought to support this characteristics. Though the other characteristics can easily be implemented on both on chip and off chip platforms, it is difficult to implement parallelism on off chip platform. A lot of hardware has been devised to unveil this property in ANN implementation and have proved to perform better than their counterparts implemented on sequential computers. However, though such systems bring in speed, have a direct link to the biological implementation of the network and have proved to perform better on specialized problems such as image processing, speech synthesis and analysis, pattern recognition among others (Lakshimi et al. 2013) they suffer the following shortcomings:

1. Distribution channels are limited to physical since such systems cannot be distributed through other channels as the Internet
2. Implementation of some activation functions which involve division or exponential operator is a challenge (Lakshimi et al. 2013) making it difficult to implement hyperbolic tangent function or the sigmoid function.
3. Model update may require replacement of some hardware components which is costly and cumbersome
4. It is also expensive and cumbersome to experiment new algorithms on such platforms

4.6.2 Off-chip implementation

In this platform, the ANN is implemented on normal computers whose hardware has no special neural network related hardware, meaning they mainly use software to model the system. While such an implementation suffers from lack of inherent parallelism as the one witnessed in the on chip implementation, the following advantages are witnessed: -

1. Many and flexible distribution channels such as the Internet
2. They can easily be used to test new training algorithm
3. All activation functions can be implemented with ease
4. It is easy to make updates to models already in the market
5. With ubiquity of computers and high computer literacy it is easy to train users to use such systems

4.6.3 Off-chip implementation through general programming languages

There are two ways of off-chip implementation using software. One can either use general purpose software or use special software development tools such as MATLAB and WEKA tools. Though it is faster to develop ANN using the specialized software, there is lack of flexibility (James 2013) and therefore some people decide to use general purpose programming software so that they can have maximum control of their programs.

In this study, the model was built using C# general programming language from scratch. Appendix A outlines some of the important source code segments.

After evaluation of the above advantages and disadvantages of each approach and given that the software is supposed to reach as many people as possible, general programming language software implementation proved to more advantageous than hardware. The model does not have very high dimensionality, can take advantage of ubiquity of computers and the level of computer literacy.

4.2 Results

4.2.1 Preliminary survey results

The purpose of this study was to come up with an early warning system for pre-diabetics and undiagnosed diabetics to enable them seek the necessary medical attention either to prevent or delay the onset of diabetes for pre-diabetic cases or be able to manage the disease to avoid debilitating effects of diabetes complications. The system so developed was also to be used as an awareness campaign tool. Therefore to start with a public interview was carried out on Trans Nzoia County residents.

It emerged that only 48% of the respondents understood the causes of diabetes mellitus while 52% didn't know the causes of diabetes and therefore don't know how they can protect themselves from the disease. 32% of the respondents had affected people in their families indicating that the disease prevalence is actually high. None of the respondents understood the government's strategy in curbing the disease. These figures are worrying given that most of the respondents were educated; 36% had university first degrees, 48% had college diplomas, 8% were secondary school graduates and 8% were primary school drop outs. However, with 92% of the respondents being computer literate and with an understanding of how to get the simple predictor variables in this study is a clear indication that this system impacts positively towards diabetes awareness and a moderately simple system interface was sufficient

The survey indicated that a big percentage of the public though they have heard of the disease, don't know the causes of the disease, ways to avoid it and even the government plans on how to curb it. The table below summarizes the findings. A positive response implies yes response while negative implies no response.

Table 4.2: Preliminary survey results

Question	Number of (+) responses	Number of negative (-) responses	Totals
Aware of diabetes	12	13	25
With diabetes relatives	8	17	25
Aware of Govt's plan on DM	0	25	25
Computer literate	23	2	25
With post primary education	23	2	25

4.2.2 R i386 3.1.2 tree

When the dataset was subjected to R i386 3.1.2 tree tool ethnicity, both diastolic and systolic blood pressure levels and age emerged to be the best distinguishing factors appearing ahead of other known factors like family history which did not appear in the decision tree and BMI. The figure below shows the resulting tree diagram.

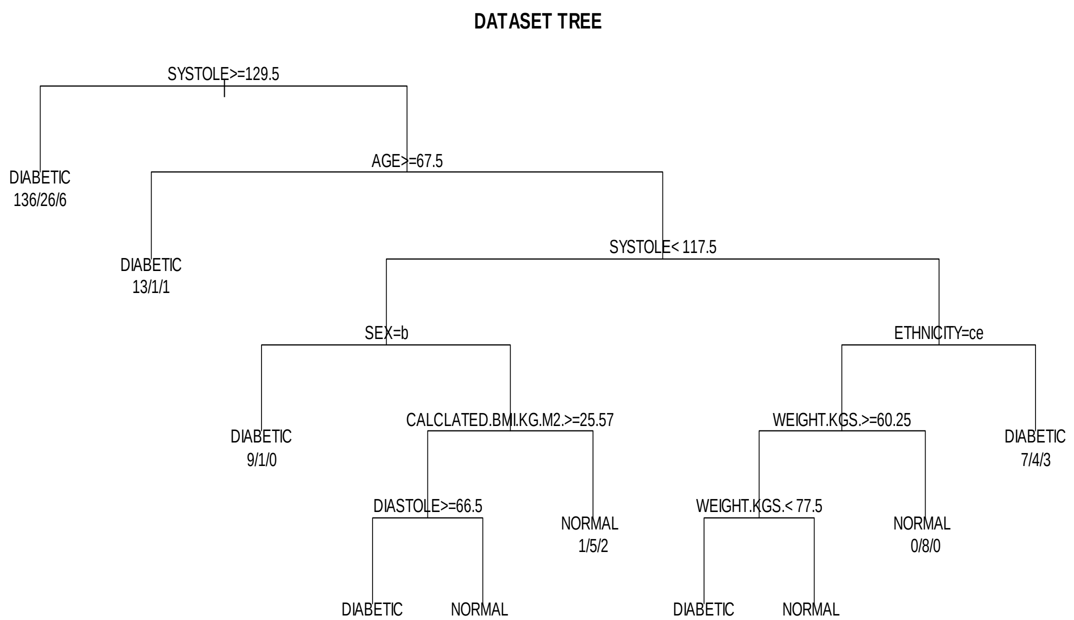


Fig. 4.4: *Predictor variable ranking by the R Decision tree*

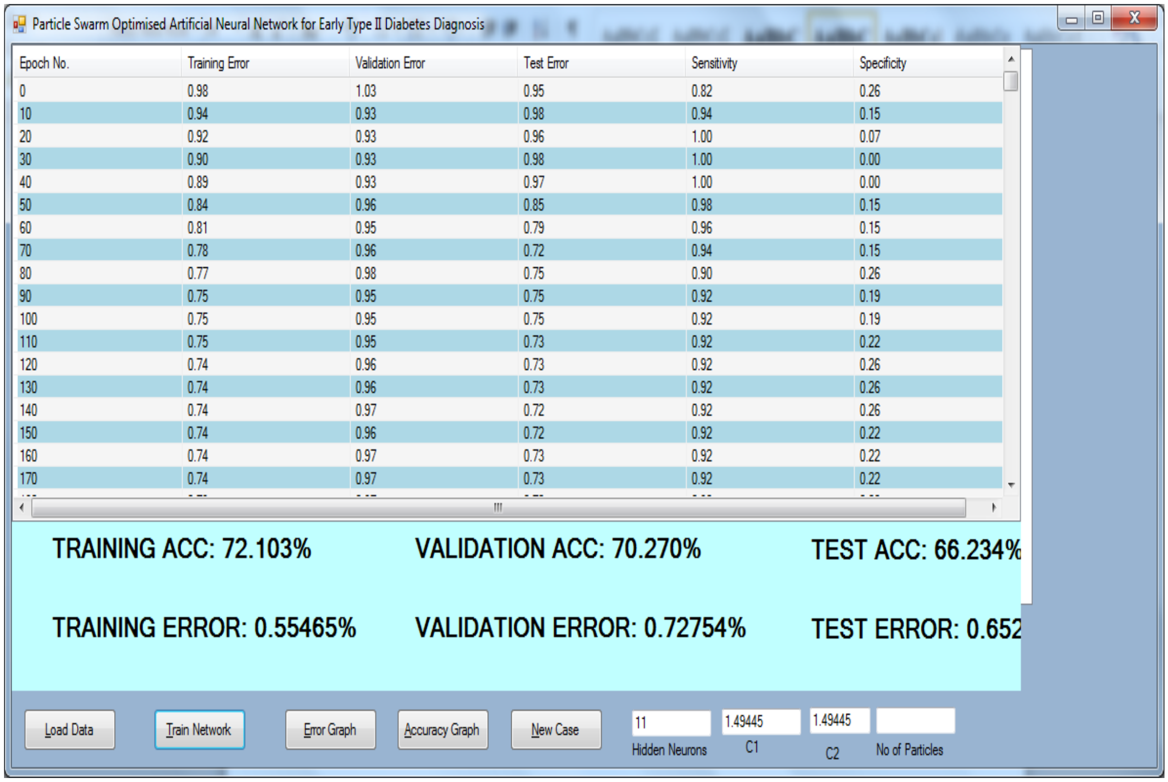
4.2.3 The network's performance in relation to the number of hidden layer neurons

The network's performance in terms of speed and accuracy was found to depend on the number of hidden layer neurons. Table 4.3 illustrates this dependence with the best accuracy of 71.62% on test data and 71.67% witnessed with 20 hidden layer neurons.

Table 4.3 *Number of hidden layer neurons vs. accuracy*

ANN architecture	Training Accuracy	Test Accuracy
12-1-2	73.77	59.46
12-2-2	74.25	58.11
12-3-2	73.82	59.46
12-4-2	73.82	59.46
12-5-2	75.11	59.46
12-6-2	73.39	59.46
12-7-2	73.39	59.46
12-8-2	76.61	59.46
12-9-2	73.82	59.46
12-10-2	75.54	58.11
12-11-2	71.67	71.62
12-12-2	74.25	59.46
12-13-2	74.25	58.11
12-14-2	74.25	59.46
12-15-2	74.89	62.16
12-16-2	72.53	56.76
12-17-2	73.82	59.46
12-18-2	73.82	59.46
12-19-2	73.61	59.46
12-20-2	67.52	59.46

The training time also increased with increase in the number of hidden layer neurons. The best values of social and cognitive constants emerged to be 1.49445. The network attained the best generalization of 70.27% when trained with 20 particles as illustrated by the graph below. The best generalization ability on validation dataset was attained with 11 hidden layer neurons.



Epoch No.	Training Error	Validation Error	Test Error	Sensitivity	Specificity
0	0.98	1.03	0.95	0.82	0.26
10	0.94	0.93	0.98	0.94	0.15
20	0.92	0.93	0.96	1.00	0.07
30	0.90	0.93	0.98	1.00	0.00
40	0.89	0.93	0.97	1.00	0.00
50	0.84	0.96	0.85	0.98	0.15
60	0.81	0.95	0.79	0.96	0.15
70	0.78	0.96	0.72	0.94	0.15
80	0.77	0.98	0.75	0.90	0.26
90	0.75	0.95	0.75	0.92	0.19
100	0.75	0.95	0.75	0.92	0.19
110	0.75	0.95	0.73	0.92	0.22
120	0.74	0.96	0.73	0.92	0.26
130	0.74	0.96	0.73	0.92	0.26
140	0.74	0.97	0.72	0.92	0.26
150	0.74	0.96	0.72	0.92	0.22
160	0.74	0.97	0.73	0.92	0.22
170	0.74	0.97	0.73	0.92	0.22

Fig. 4.5 An image of hidden layer neuron investigating prototype

4.2.4 Values of social and cognitive weights

Just like the number of hidden layer neurons the above values were found through experimentation on a prototype. Table 4.4A and 4.4B below summarizes the results obtained after five runs per every value.

Table 4.4A: Mean accuracy vs. social and cognitive weights

Value of C1 and C2	Training Accuracy	Validation Accuracy
1.0	73.820	59.459
1.1	60.515	51.351
1.2	84.120	62.162
1.3	73.820	59.459
1.4	73.820	59.459
1.5	72.103	70.270
1.6	75.107	59.459

Table 4.4B: Mean accuracy vs. social and cognitive weights

Value of C1 and C2	Training Accuracy	Validation Accuracy
1.7	73.820	59.461
1.8	60.565	52.351
1.9	73.820	59.461
141	74.249	56.757
1.42	73.820	59.459
1.49445	71.103	70.270
1.51	72.961	62.162

The best performance is attained when both values are set to either 1.49445.

4.2.5 Number of PSO particles vs. Accuracy

With the above system parameter settings, the prototype was also subjected to a series of experiments to test the number of particles that are enough to train the network. Five runs were made. During the first run the number of epochs was set to 1,000 and an increment of 2,000 epochs added to each subsequent run. An average of the five runs per number of particles used for both training accuracy and validation accuracy were taken and plotted on the graph below. The generalization ability of the network stabilizes with 20 and above particles

Fig. 4.6: Number of PSO particles vs. Accuracy

As clearly illustrated by the graph above, 20 and above particles are sufficient for good generalization of the model.

4.2.6 The overall results

In the overall the network attained 66.23% accuracy on test data and sensitivity of 70.23% against specificity of 64.24%.

4.3 Discussion

The results of this study confirms that an early warning system with sufficient accuracy can be built for undiagnosed diabetics and pre-diabetics based on simple questions known to an average individuals and that requires less calculation just as it was in the case of Heikes et al (2008) using machine learning techniques. Contrary to Heikes et al (2008) this study based its investigation on residents of Trans Nzoia county of Kenya while the former used the US population. However, there are a lot of similarities on best predicting variables from both studies. In Heikes et al (2008) age, ethnicity, weight and blood pressure appeared to be playing the biggest responsibility in discriminating the affected individuals from normal subjects and the same has been confirmed by this study. Heikes et al (2008) built their system on decision trees while this study has been built on Artificial Neural Networks. The Heikes et al (2008) tool attained a sensitivity of 88% and specificity of 75% while this study realized a sensitivity of 70.23% and specificity of 64.24% the lower values of specificity and sensitivity in this study compared to the former may be due to the fact that the former used a more validated dataset tested by several professionals.

Just as in the Public Health of England [PHE] (2014) where it was demonstrated that ethnicity strongly contributes to the degree of an individual's predisposition to diabetes, this study also tend to show similar results with members of the Kikuyu community of Kenya appearing to be more predisposed to diabetes than other communities. 41.80% of the positive cases were members of the community. This is interesting given that members of the Kikuyu community are not the most dominant community in the area and the sample taken was based on the natural inflow of patients referred by various health officers to test their blood sugar levels implying that there was no prejudice in the sample selection procedure. This study did not find out the exact reasons for this high predisposition of the Kikuyu community, however, in Public Health of England [PHE] (2014) South Asian people living in England were found to be more predisposed to diabetes due to significant accumulation of more metabolically active fat in the abdomen and around the waist than their counterparts from the European population.

32% of the respondents had at least one of their family members suffering from diabetes mellitus an indication that the disease prevalence is actually high among the residents of Trans Nzoia County.

The study has also shown that there is low public awareness on diabetes with 52% of the responds unaware of its causes. More worrying is the fact that most of the responds were fairly educated, 92% having at least secondary level of education implying that they are in a good position to understand the causes of the disease. This was also observed in the Ministry of Health and Sanitation [MHS] (2010) with the lack of public awareness being attributed to inadequate systems in our Primary Health Care (PHC) geared towards tackling non communicable diseases. It is also worrying for this high level of unawareness to still exist four years after the ministry of health launching its 2010 -2015 strategic plan on curbing diabetes mellitus. It tends to show that the plan has not been effectively implemented emphasized by 100% of the respondents being unaware of the government's plans on controlling diabetes mellitus in Kenya.

The ability of decision trees to rank explanatory variables based on their discriminating power with the best classifiers appearing near the root of the tree while those with little discriminatory ability appearing near the leaves of the tree or not appearing in the tree structure at all has been exploited in this study to determine the best predictors with systolic blood pressure, age and ethnicity topping the list. Though Artificial Neuron Networks have shown better predicting results than decision trees (Krenker et al. 2011) their black box nature of prediction makes it difficult for human beings to interpret the results and hence the incorporation of the decision tree. However, such architecture makes the system not to change the predicting variables with future trend changes therefore limiting the systems lifespan.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

In conclusion, best predicting variables for diabetes mellitus have been identified upon which an early warning system for type-2 diabetes has been developed using Particle Swarm Optimized ANN. The system has demonstrated reasonable accuracy of 66.23%, a specificity of 64.24 and sensitivity of 70.23% on test data which makes it usable for Trans Nzoia County residents and the entire Kenyan population by extension. These are acceptable accuracy ranges given that this is not a gold standard test but a mere early warning system (Kelly et al. 2007)

The study has shown that some communities are more predisposed to diabetes than others therefore individuals from such communities need to be more cautious by probably engaging in physical activities, eating healthy diet and regularly seeking medical checkups so as to avoid suffering from the disease. This system might also be handy for such individuals and the entire public, instead of making regular visits to health facilities they can still just use the system in their homes, work places, social places and any other places they can access computers. This will make as many people as possible to be aware of their diabetes status and therefore take necessary actions either to delay its onset or prevent it completely. For those affected already will seek proper medical attention to avoid the associated complications. To enable easy access, the system can be placed on the internet from where it can be easily downloaded.

As indicated in the literature review, most health care centers are ill equipped to deal with diabetes due to lack of necessary facilities and competent personnel. This system can be used to make health workers more effective and efficient therefore boosting service delivery. Instead of doing numerous resource and time consuming laboratory tests, the system can be utilized therefore making the process faster and less costly because individuals unlikely to be suffering from the disease will be unveiled by the system hence will not undergo further testing. Laboratory resources will only be used on people whose result as per the system is positive. Since the explanatory variables used in this study are not complex and can be obtained by an average individual, less skilled personnel can be deployed to assist the diabetic specialists with the aid of the system.

To the academic community such a system needs to be a seed to a series of similar systems. These researches can explore aspects the study didn't manage to address such as alternative ways of finding best predicting variables other than using the decision trees which will make the process more automated, accurate and responsive to the changing trends. This is because as already demonstrated in the literature review and the discussion sections the ability of various predicting variables appears to be changing, therefore if such changes occur in future, the early warning systems should be able to adapt themselves to such changes.

Instead of using decision trees to determine the best predicting variables to advise the public on the type-2 diabetes risk factors, a rule extraction algorithm such as the one used in Kamruzzaman et al (2006) could be used making the system to produce more easily comprehensible explanations. In the above study the continuous outputs of the hidden layer neurons were discretized and rules extracted from the compact ANNs by probing the discretized activation values. The resulting explanation is likely to be more accurate than that of a decision tree because just as it has been indicated by the above study and others in the literature review, ANN are better classifiers than decision trees therefore insights drawn directly from the models are likely to be more accurate than those drawn from the decision trees.

REFERENCES

1. Abdulbari, B, Mahmoud, Z & Ammar, A 2005, ‘Genetics, Obesity, and Environmental Risk Factors Associated with Type 2 Diabetes’, *Croat Medical Journal*, Vol. 46. No. 2 pp. 302-307
2. Anthony, KA 2006, ‘Understanding Diagnostic Test 1: Sensitivity, Specificity and Predictive Values’
3. Anthony, O & Gerry, D 2000, ‘Adapting Particle Swarm Optimization to Dynamic Environment’
4. Asha, G, Karaganda, AS & Manjunath, MA 2011, ‘Application of genetic algorithm optimized neural network connection weights for medical diagnosis of PIMA Indians diabetes’, *International Journal on Soft Computing* Vol. 2. No.2
5. Bekir, K & Vehbi, O n.d, ‘Performance Analysis of Various Activation Functions Generalized MLP Architectures of Neural Networks’, *International Journal of Artificial Intelligence And Expert Systems (IJAE)*, Vol. 1 No .4
6. Carlo, U 2007, ‘Multilayer ANN and back propagation’
7. Diptam, D, Argha, R & Choudhury K 2013, ‘Training Artificial Neural Network Using Particle Swarm Optimization’, *International Journal of Research in Computer Science and Software Engineering*, Vol. 3 No. 3, pp. 430 – 434
8. Elaine, B 2006, ‘Introduction to Sensitivity and Specificity of Diagnostic Tests’, *School of Nursing and Academic Division of Midwifery*
9. Hamman, FR 1992, ‘Genetic and Environmental Determinants of Non-Insulin-Dependent Diabetes Mellitus’, *Diabetes/Metabolism Reviews*, Vol. 8, No. 4
10. Heikes, KE, Eddy, DM, Arondekar, B & Schlessinger L 2008, ‘Diabetes risk calculator: a simple tool for detecting undiagnosed diabetes and pre-diabetes’, *Diabetes Care*, 31:1040-1045
11. *http://healthservices.uonbi.ac.ke/node/842*
12. International Diabetes Federation 2007, Diabetes Atlas 3rd Edition, IDF
13. International Diabetes Federation 2011, Diabetes Atlas 4rd Edition, IDF
14. James, M 2013, ‘Neural Network Back propagation Using C#’, *Visual Studio Magazine*

15. James, M 2013, ‘Neural Network Training with Particle Swarm Optimization’, *Visual Studio Magazine*
16. Jou, WL, Cheng, YC, Hung, YL, Fen, YC, Mei YW, Tsai, R, Chen, HY, Hwang, JJ & LEE, MC 2009, ‘Cross-Sectional Validation of Diabetes Risk Scores for Predicting Diabetes, Metabolic Syndrome, and Chronic Kidney Disease in Taiwanese’, *Diabetes Care*, Vol. 32, No. 12
17. Kamruzzaman, SM & Monirul, IM 2006, ‘An Algorithm to Extract Rules from Artificial Neural Networks for Medical Diagnosis Problems’, *International Journal of Information Technology,* Vol. 12 No. 8
18. Kazemnejad, A, Batvandi, Z & Faradmal, J 2010, ‘Comparison of artificial neural network and binary logistic regression for determination of impaired glucose tolerance/diabetes’, *Eastern Mediterranean Health Journal,* Vol. 16 No. 6. pp. 615 – 620
19. Kelly, HZ, O’Malley, J & Mauri, L 2007, ‘Receiver-Operating Characteristic Analysis for Evaluation Diagnosis Tests and Predictive Models’, *Statistical Primer for Cardiovascular Research,* pp. 654-657
20. Kennedy, J & Eberhart, R 1995, ‘Particle Swarm Optimization’, *IEEE service Center*
21. Krenker, A, Bester, J & Kos, A 2011, ‘Introduction to the Artificial Neural Networks’, available at: *www.intechopen.com*
22. Kumar, S & Kumaravel, A 2013, ‘Diabetes Diagnosis using Artificial Neural Network’, *International Journal of Engineering Sciences & Research Technology, pp.* 1642-1644
23. Lakshmi, PK & Subadra, M 2013, ‘A Survey on FPGA based MLP Realization for On-chip Learning*’, International Journal of Scientific & Engineering Research, Vol. 4, No.1*
24. Mansour, R, Zandkarimi, E & Amirhossein, H 2013, ‘Comparison of Artificial Neural Network, Logistic Regression and Discriminant Analysis Efficiency in Determining Risk Factors of Type 2 Diabetes’, *World Applied Sciences Journal,* Vol. 23 No. 11, pp. 1522 – 1529
25. Ministry of Health and Sanitation 2010, ‘Kenya National Diabetes Strategy 2010 – 2015’, MHS

26. Pandya, V 2005, 'A HANDEL-C implementation of the Back-Propagation Algorithm on Field Programmable Gate Arrays', *University of Guelph*
27. Public Health England 2014, 'Adult obesity and type 2 diabetes', PHE
28. Rajul, P, Annie, M, Shefali, P, Chandra, S & Ravi, T 2008, 'Understanding and Using Sensitivity and Predictive Values', *Indian Ophthalmic Journal*
29. Shankaracharya, Devang, O, Subir, S & Ambarish, SV 2011, 'Computational Intelligence in Early Diabetes Diagnosis: A Review', *Review of Diabetic Studies*, Vol. 7. No. 4.
30. Tiffany, LEJ 2013, 'Diabetes Mellitus: the increasing burden of disease in Kenya', *South Sudan Medical Journal*, Vol. 6. No 3
31. Van, TJ, Van, HTW, Pearson, P & Wijmenga, C 2001, 'Defining the Genetic Contribution of Type 2 Diabetes Mellitus'
32. World Health Organization 2006, 'Definition and Diagnosis of Diabetes Mellitus and Intermediate Hyperglycemia', WHO, *Report of a WHO/IDF Consultation*
33. Xiaohui, H & Eberhart, R 2002, 'Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization', **Available at**: www.swarmintelligence.org/papers/SCI2002Constrained.pdf

APPENDIX

Appendix A

Source code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

public class NeuralNetwork
    {

        private int numInput;
        private int numHidden;
        private int numOutput;
        private double[] inputs;
        private double[][] ihWeights; // input-hidden
        private double[] hBiases;
        private double[] hOutputs;
        private double[][] hoWeights; // hidden-output
        private double[] oBiases;
        private double[] outputs;


        public NeuralNetwork(int numInput, int numHidden, int numOutput)
        {

            this.numInput = numInput;
            this.numHidden = numHidden;
            this.numOutput = numOutput;
            this.inputs = new double[numInput];
            this.ihWeights = MakeMatrix(numInput, numHidden);
            this.hBiases = new double[numHidden];
            this.hOutputs = new double[numHidden];
            this.hoWeights = MakeMatrix(numHidden, numOutput);
            this.oBiases = new double[numOutput];
            this.outputs = new double[numOutput];
        }
```

```
private static double[][] MakeMatrix(int rows, int cols)
{
   double[][] result = new double[rows][];
   for (int r = 0; r < result.Length; ++r)
      result[r] = new double[cols];
   return result;
}



public void SetWeights(double[] weights)
{

   int numWeights = (numInput * numHidden) + (numHidden * numOutput)
       + numHidden + numOutput;
   if (weights.Length != numWeights)
      throw new Exception("Bad weights array length: ");

   int k = 0;

   for (int i = 0; i < numInput; ++i)
      for (int j = 0; j < numHidden; ++j)
         ihWeights[i][j] = weights[k++];
   for (int i = 0; i < numHidden; ++i)
      hBiases[i] = weights[k++];
   for (int i = 0; i < numHidden; ++i)
      for (int j = 0; j < numOutput; ++j)
         hoWeights[i][j] = weights[k++];
   for (int i = 0; i < numOutput; ++i)
      oBiases[i] = weights[k++];
}

public double[] GetWeights()
{

   int numWeights = (numInput * numHidden) + (numHidden * numOutput)
     + numHidden + numOutput;
   double[] result = new double[numWeights];
   int k = 0;
   for (int i = 0; i < ihWeights.Length; ++i)
      for (int j = 0; j < ihWeights[0].Length; ++j)
         result[k++] = ihWeights[i][j];
   for (int i = 0; i < hBiases.Length; ++i)
      result[k++] = hBiases[i];
   for (int i = 0; i < hoWeights.Length; ++i)
      for (int j = 0; j < hoWeights[0].Length; ++j)
```

```
            result[k++] = hoWeights[i][j];
    for (int i = 0; i < oBiases.Length; ++i)
        result[k++] = oBiases[i];
    return result;
}

// ----------------------------------------------------------------------------------

public double[] ComputeOutputs(double[] xValues)
{
    if (xValues.Length != numInput)
        throw new Exception("Bad xValues array length");

    double[] hSums = new double[numHidden];
    double[] oSums = new double[numOutput];

    for (int i = 0; i < xValues.Length; ++i)
        this.inputs[i] = xValues[i];

    for (int j = 0; j < numHidden; ++j)
        for (int i = 0; i < numInput; ++i)
            hSums[j] += this.inputs[i] * this.ihWeights[i][j];

    for (int i = 0; i < numHidden; ++i)
        hSums[i] += this.hBiases[i];

    for (int i = 0; i < numHidden; ++i)
        this.hOutputs[i] = HyperTanFunction(hSums[i]);

    for (int j = 0; j < numOutput; ++j)
        for (int i = 0; i < numHidden; ++i)
            oSums[j] += hOutputs[i] * hoWeights[i][j];

    for (int i = 0; i < numOutput; ++i)
        oSums[i] += oBiases[i];

    double[] output = new double[numOutput];
     for(int i=0; i<numOutput; i++)
         output[i] = HyperTanFunction(oSums[i]);

    double[] retResult = new double[numOutput];
    Array.Copy(output, retResult, output.Length);
    return retResult;
}
private static double HyperTanFunction(double x)
{
```

```
            if (x < -20.0) return -1.0;
            else if (x > 20.0) return 1.0;
            else return Math.Tanh(x);
        }



public double[] Train(double[][] trainData, int numParticles, int maxEpochs,
            double exitError, double probDeath)
        {

            Random rnd = new Random(16);

            int numWeights = (this.numInput * this.numHidden) +
                      (this.numHidden * this.numOutput) +
              this.numHidden + this.numOutput;


            int epoch = 0;
            double minX = -10.0;
            double maxX = 10.0;
            double w = 0.729;

            double r1;

            double r2;

            Particle[] swarm = new Particle[numParticles];

            double[] bestGlobalPosition = new double[numWeights];
            double bestGlobalError = double.MaxValue;


            for (int i = 0; i < swarm.Length; ++i)
            {
                double[] randomPos = new double[numWeights];
                for (int j = 0; j < randomPos.Length; ++j)
                {
                    randomPos[j] = rnd.NextDouble() * (maxX - minX)  + minX;
                }

                double error = MeanSquaredError(trainData, randomPos);
                double[] ranVelocity = new double[numWeights];

                for (int j = 0; j < ranVelocity.Length; ++j)
                {
                    double lo = 0.1 * minX;
```

```
      double hi = 0.1 * maxX;
      ranVelocity[j] = (hi - lo) * rnd.NextDouble() + lo;

    }
    swarm[i] = new Particle(randomPos, error, ranVelocity, randomPos, error);

    if (swarm[i].error < bestGlobalError)
    {
      bestGlobalError = swarm[i].error;
      swarm[i].position.CopyTo(bestGlobalPosition, 0);
    }
  }

  int[] sequence = new int[numParticles];
  for (int i = 0; i < sequence.Length; ++i)
    sequence[i] = i;

  while (epoch < maxEpochs)
  {
    if (bestGlobalError < exitError) break;

    double[] newVelocity = new double[numWeights
    double[] newPosition = new double[numWeights];
    double newError;

    Shuffle(sequence, rnd);

    for (int pi = 0; pi < swarm.Length; ++pi)
    {
      int i = sequence[pi];
      Particle currP = swarm[i];

      for (int j = 0; j < currP.velocity.Length; ++j
       {
        r1 = rnd.NextDouble();
        r2 = rnd.NextDouble();


        newVelocity[j] = (w * currP.velocity[j]) +
          (c1 * r1 * (currP.bestPosition[j] - currP.position[j])) +
          (c2 * r2 * (bestGlobalPosition[j] - currP.position[j]));
       }

      newVelocity.CopyTo(currP.velocity, 0);

     for (int j = 0; j < currP.position.Length; ++j)
```

```
}

    {
        newPosition[j] = currP.position[j] + newVelocity[j];
        if (newPosition[j] < minX)
            newPosition[j] = minX;
        else if (newPosition[j] > maxX)
            newPosition[j] = maxX;
    }

    newPosition.CopyTo(currP.position, 0);

    newError = MeanSquaredError(trainData, newPosition);
    currP.error = newError;

    if (newError < currP.bestError)
      {
        newPosition.CopyTo(currP.bestPosition, 0);
        currP.bestError = newError;
       }

    if (newError < bestGlobalError)
       {
        newPosition.CopyTo(bestGlobalPosition, 0);
        bestGlobalError = newError;
        BestTrainingEpoch = epoch;
        }



    double die = rnd.NextDouble();
    if (die < probDeath)
       {

        for (int j = 0; j < currP.position.Length; ++j)
        currP.position[j] = rnd.NextDouble() * (maxX - minX)  + minX;
        currP.error = MeanSquaredError(trainData, currP.position);
        currP.position.CopyTo(currP.bestPosition, 0);
        currP.bestError = currP.error;

        if (currP.error < bestGlobalError)
        {
            bestGlobalError = currP.error;
            currP.position.CopyTo(bestGlobalPosition, 0);
        }
    }

}
```

```
        errLog[epoch][0]=this.MeanSquaredError(trainData,bestGlobalPosition);
        eccLog[epoch][0] = this.Accuracy(trainData);
        errLog[epoch][1] =    this.MeanSquaredError
                              (validationData,bestGlobalPosition);
        eccLog[epoch][1] = this.Accuracy(validationData);

        errLog[epoch][2] = this.MeanSquaredError(testData, bestGlobalPosition);
        eccLog[epoch][2] = this.Accuracy(testData);
        errLog[epoch][3] = this.Sensitivity(testData);
        errLog[epoch][4] = this.Specificity(testData);



      ++epoch;

    }
    this.SetWeights(bestGlobalPosition);
    double[] retResult = new double[numWeights];
    Array.Copy(bestGlobalPosition, retResult, retResult.Length);
    return retResult;

}

private static void Shuffle(int[] sequence, Random rnd)
{
    for (int i = 0; i < sequence.Length; ++i)
    {
      int r = rnd.Next(i, sequence.Length);
      int tmp = sequence[r];
      sequence[r] = sequence[i];
      sequence[i] = tmp;
    }
}

public double MeanSquaredError(double[][] trainData, double[] weights)
{
    this.SetWeights(weights);

    double[] xValues = new double[numInput];
    double[] tValues = new double[numOutput];
    double sumSquaredError = 0.0;
    for (int i = 0; i < trainData.Length; ++i)
    {
      Array.Copy(trainData[i], xValues, numInput);
      Array.Copy(trainData[i], numInput, tValues, 0, numOutput);
      double[] yValues = this.ComputeOutputs(xValues);
```

```
        for (int j = 0; j < yValues.Length; ++j)
        sumSquaredError += ((yValues[j] - tValues[j])
                    * (yValues[j] - Values[j]));
    }
    return sumSquaredError / trainData.Length;
}



public double Accuracy(double[][] testData)
{

    int numCorrect = 0;
    int numWrong = 0;
    double[] xValues = new double[numInput];
    double[] tValues = new double[numOutput];
    double[] yValues;

    for (int i = 0; i < testData.Length; ++i)
    {
                Array.Copy(testData[i], xValues, numInput);
                Array.Copy(testData[i], numInput, tValues, 0, numOutput);
        yValues = this.ComputeOutputs(xValues);
        int maxIndex = MaxIndex(yValues);
        if (tValues[maxIndex] == 1.0)
            ++numCorrect;
        else
            ++numWrong;
    }
    return (numCorrect * 1.0) / (numCorrect + numWrong);
 }

public double Sensitivity(double[][] testData)
{

    double Positives = 0;
    double numTruePositive = 0;
    double SensitivityValue = 0;
    double[] xValues = new double[numInput];
    double[] tValues = new double[numOutput];
    double[] yValues;
    for (int i = 0; i < testData.Length; ++i)
    {
         Array.Copy(testData[i], xValues, numInput);
         Array.Copy(testData[i], numInput, tValues, 0, numOutput);
         yValues = this.ComputeOutputs(xValues);
```

```
            int intMaxIndexY = MaxIndex(yValues);
            int intMaxIndexT = MaxIndex(tValues);


            if (intMaxIndexT == 0)
            {
                ++Positives;
                if (intMaxIndexY == 0)
                {
                    ++numTruePositive;
                }

            }

            SensitivityValue = numTruePositive / Positives;
        }               return (SensitivityValue);

    }
    //
    public double Specificity(double[][] testData)
    {

        double Negatives = 0;
        double numTrueNegative = 0;
        double SpecificityValue = 0;
        double[] xValues = new double[numInput];
        double[] tValues = new double[numOutput];
        double[] yValues;
        for (int i = 0; i < testData.Length; ++i)
        {
             Array.Copy(testData[i], xValues, numInput);
             Array.Copy(testData[i], numInput, tValues, 0, numOutput);
            yValues = this.ComputeOutputs(xValues);
            int intMaxIndexY = MaxIndex(yValues);
            int intMaxIndexT = MaxIndex(tValues);


            if (intMaxIndexT == 1)
            {
                ++Negatives;
                if (intMaxIndexY == 1)
                {
                    ++numTrueNegative;
                }

            }
```

```
            SpecificityValue = numTrueNegative / Negatives;
          }

          return (SpecificityValue);
        }

        private static int MaxIndex(double[] vector)
        {
                        int bigIndex = 0;
          double biggestVal = vector[0];
          for (int i = 0; i < vector.Length; ++i)
          {
            if (vector[i] > biggestVal)
            {
              biggestVal = vector[i];
              bigIndex = i;
            }
          }
          return bigIndex;
        }

      }
      public class Particle
      {
        public double[] position;
        public double error;
        public double[] velocity;
        public double[] bestPosition;
        public double bestError;

        public Particle(double[] position, double error, double[] velocity,
         double[] bestPosition, double bestError)
        {
          this.position = new double[position.Length];
          position.CopyTo(this.position, 0);
          this.error = error;
          this.velocity = new double[velocity.Length];
          velocity.CopyTo(this.velocity, 0);
          this.bestPosition = new double[bestPosition.Length];
          bestPosition.CopyTo(this.bestPosition, 0);
          this.bestError = bestError;

         }
}
```