www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 9, September 2014 Page No. 8439-8444

Diabetes Prediction in Women Based On Soft Computing Techniques

S.Karthikeyeni, S.Saranya
Associate Professor,
Department of Computer Science (P.G.), Kongu Arts and Science College, Erode
karthikeyeni@gmail.com
Research Scholar
Kongu Arts and Science College, Erode
hemassaranya@gmail.com

ABSTRACT
Diabetes Mellitus, generally called as High Sugar Problem is a metabolic disorder described by chronic hyperglycemia enhancing rapidly. It is a polygenic disease characterized by abnormal high glucose in the blood. Statistics say that 90 to 95 % of the World Diabetics have Type 2 Diabetes. The consequences of diabetes may be macro vascular complication or micro vascular complication. This proposes a new model for prediction of complications developing due to Diabetes Mellitus. This system compares the performance of various soft computing techniques such as Artificial Neural Networks, Support Vector Machine, Particle Swarm Optimization and Genetic Algorithms for prediction of diabetes disease.
Keywords: ANN, Support Vector Machine, Particle Swarm Optimization, Genetic Algorithm.

1. INTRODUCTION

Diabetes mellitus is characterized by abnormally high levels of sugar (glucose) in the blood. When the amount of glucose in the blood increases, e.g., after a meal, it triggers the release of the hormone insulin from the pancreas. Insulin stimulates muscle and fat cells to remove glucose from the blood and stimulates the liver to metabolize glucose, causing the blood sugar level to decrease to normal levels. In people with diabetes, blood sugar levels remain high. This may be because insulin is not being produced at all, is not made at sufficient levels, or is not as effective as it should be. The most common forms of diabetes are type 1 diabetes (5%), which is an autoimmune disorder, and type 2 diabetes (95%), which is associated with obesity. Gestational diabetes is a form of diabetes that occurs in pregnancy, and other forms of diabetes are very rare and are caused by a single gene mutation. Also, there have been many computerized methods proposed for diagnosis of diabetes. All these methods have some input values which would be the result of different tests that should be carried out in hospitals. A woman who was atleast 21 years old was tested for diabetes according to World Health Organization criteria. The objective of this data set is diagnosis of diabetes for woman. Based on personal data such as age, number of times pregnant, and the results of medical examinations e.g. blood pressure, body mass index, result of glucose tolerance test, etc., try to decide whether diabetes is positive or not.

Back Propagation algorithm is used for diagnosis. Back propagation is a systematic method for training multi layer artificial neural network. It is a multilayer feed forward network that used extent gradient-decent based back propagation rule. Multi-layer feed forward network trained by using extent gradient-descent based back propagation algorithm is limited to search for a suitable set of weights in an apriority fixed network topology. Particle Swarm Optimization finds the optimal positions, to be represented as a point in an n-dimensional solution space. SVM finds the optimal hyperplane that separates clusters of vector. GA is an optimization techniques inspired by natural selection and natural genetics. The system is developed using Matlab.

2. RELATED WORKS

Data Mining is the process of extracting knowledge hidden from large volumes of raw data. The knowledge must be new, not obvious, and one must be able to use it. It is the process of analyzing data from different perspectives and summarizing it into useful information that can be used to increase revenue, cut costs or both.

2.1 Data Mining Techniques

Data mining is a multi-disciplinary field; data mining adopted its techniques from many research areas such as statics, machine learning, database system, neural networks, rough sets, and visualization.

The most commonly used techniques include artificial neural networks, decision trees, and the nearest-neighbor method. Each of these techniques analyzes data in different ways:

- **Artificial neural networks** are non-linear, predictive models that learn through training. Although they are powerful predictive modeling techniques, some of the power comes at the expense of ease of use and deployment.
- **Decision trees** are tree-shaped structures that represent decision sets. These decisions generate rules, which then are used to classify data. Decision trees are the favored technique for building understandable models.
- **The nearest-neighbor method** classifies dataset records based on similar data in a historical dataset.

Data mining tasks can be classified into two categories:

1. Descriptive tasks will look at some given data and find patterns in it.
2. Predictive tasks look at historical data to predict what will happen in the future.

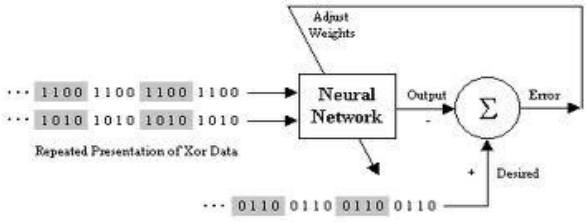
Artificial Neural Network Techniques

The most frequently used neural network techniques for forecasting applications are (i) Feed forward neural network and (ii) Recurrent neural networks[1].

2.2 Artificial Neural Network

An Artificial Neural Network (ANN), usually called Neural Network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. An input is presented to the neural network and a corresponding desired or target response is set at the output using supervised training. An error is composed from the difference between the desired response and the system output. This error information is feedback to the system and adjusts the system parameters in a systematic fashion using the learning rule. The process is repeated until the performance is acceptable.

Artificial Neural Networks [3] are the second best way to implement a solution" motivated by the simplicity of their design and because of their universality, only shadowed by the traditional design obtained by studying the physics of the problem. At present, artificial neural networks are emerging as the technology of choice for many applications, such as pattern recognition, prediction, system identification, and control.



Computation Model of ANN

2.2.1 Learning through back propagation

Supervised learning is carried out through back propagation, a generalization of the least mean squares algorithm in the linear perceptron. The error function to fix the attention on one of the weights, wij whose associated edge points from the i-th to the j-th node in the network. The backpropagation step computes the gradient of E with respect to this input, i.e.,$\partial E/\partial O_i w_{ij}$. Since in the backpropagation step O_i is treated as a constant

$$\frac{\partial E}{\partial w_{ij}} = o_i \frac{\partial E}{\partial o_i w_{ij}}$$

each node i,the output o_i of the node in the feed-forward step. The cumulative result of the backward computation in the backpropagation step up to this node.

The backpropagated error at the j-th node by δ_j,to express the partial derivative of E with respect to w_{ij} as

$$\frac{\partial E}{\partial w_{ij}} = O_i \delta_j$$

Using gradient descent by adding to each weight w_{ij} the increment

$$\Delta w_{ij} = -\gamma o_i \delta_j$$

This correction step is needed to transform the backpropagation algorithm into a learning method for neural networks.

2.2.2 Training of artificial neural networks

A neural network has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to **'train' the neural network** by feeding it teaching patterns and letting it change its weights according to some learning rule. The different learning situations are categorized as:

- Supervised learning or Associative learning
- Unsupervised learning or Self-organization
- Reinforcement Learning

2.2.3 Modifying patterns of connectivity of Neural Networks

Learning paradigms, supervised learning and unsupervised learning result in an adjustment of the weights of the connections between units, according to some modification rule. Virtually all learning rules can be considered as a variant of the Hebbian learning rule [5]. The basic idea is that if two units' **j** and **k** are

active simultaneously, their interconnection must be strengthened.

If **j** receives input from **k**, the simplest version of Hebbian learning prescribes to modify the weight w_{jk} with

$$\Delta\omega_{jk} = yy_jy_k$$

where y is a positive constant of proportionality representing the learning rate. Another common rule uses, not the actual activation of unit ***k*** but the difference between the actual and desired activation of **k.** The weights are adjusted as

$$\Delta\omega_{jk} = y\, y_j(d_k - y_k)$$

in which d_k is the desired activation provided by a teacher. This is often called the Widrow-Hoff rule or the delta rule.

2.3 Support vector machine

. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on in addition performing linear classification[1].

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a heuristic technique suited for search of optimal solutions and based on the concept of swarm. The Particle Swarm Optimization algorithm is a biologically-inspired algorithm motivated by a social analogy. PSO ability to effectively face classification of multiclass database instances is examined. Data classification probably represents the most commonly applied supervised data mining technique. It consists in generating, starting from a set of learning examples labeled into classes, a mapping from instances to classes which can then be used to classify new instances. In fact, when in a multi-dimensional space a class prototype is represented by a centroid, the classification can be seen as the problem of finding the optimal positions of all the class centroids [2].

2.5 Genetic Algorithm

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic[4]. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. Genetic Algorithms are a family of computational models inspired by evolution. An implementation of a genetic algorithm begins with a population of random chromosomes. The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

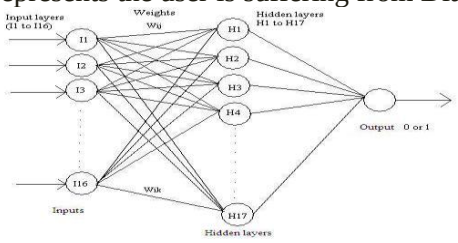
1. Selection rules select the individuals, called parents that contribute to the population at the next generation.
2. Crossover rules combine two parents to form children for the next generation.
3. Mutation rules apply random changes to individual parents to form children.

The new population generated undergoes the further selection, crossover and mutation till the termination criterion is not satisfied. Convergence of the genetic algorithm depends on the various criterions like fitness value achieved or number of generations [5].

3. PROPOSED WORK

3.1 Artificial Neural Network Architecture

The network architecture refers to the number of layers, the number of nodes in each layer, and the number of hidden layers in the network. In the proposed system, 16 inputs have been chosen for the diagnosis (Fig.). So there will be 16 input layers for the network. N+1 hidden layer have been designed for the network. The output layer of the network has one neuron which gives the value either 0 or 1. The value 0 represents the user is not affected with diabetes and the value 1 represents the user is suffering from Diabetes.



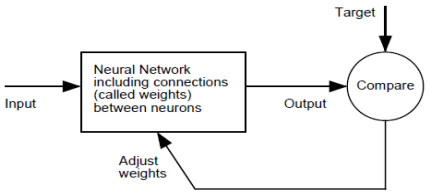
Neural Network Architecture

3.1.1 Back Propagation Algorithm

Back propagation is a systematic method for training multi layer artificial neural network. It is a multi-layer feed forward network that uses extent gradient-descent based back propagation rule. During the training phase, the training data is fed to the input layer. The data is propagated through the hidden layer and then to the output layer, which is called forward pass. During this phase, each node in the hidden layer gets input from all the input layer nodes, which are then multiplied with the randomly assigned weights and then summed up. Similarly the output layer node receives inputs from all nodes of the hidden layer, which are then multiplied with the randomly assigned weights and then summed up. This forms the output of the output layer. The output values are now compared with the target value, and the error between the output value and the target output value is calculated and it is propagated back to the hidden layer, which is called backward pass. This error is used to update all the weights between the input and hidden layers as well as the weights between hidden and output layers. The training continues until the error is minimized.

The main idea behind using neural networks for problem solving is their ability to learn from "past" data and to generalize when responding to new input data. But there are other approaches to using neural networks, such as analyzing neural networks and extracting rules and explicit knowledge, and inserting rules into connectionist architectures for the purpose of approximate reasoning. Neural networks are composed

of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network. Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training.



Adjust weights to Neural Network

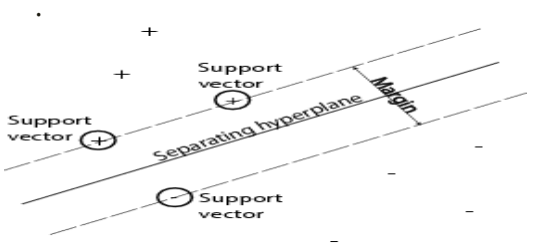
The Pseudo Code of the Back Propagation is shown below

```
Initialize all weights and biases in network
While terminating condition is not satisfied
{
for each training sample X in samples
{
// Propagate the inputs forward
 for each hidden or output layer unit j
 {
Ij = S i wijOi + q
Oj = 1 / 1+e –Ij
// compute the output of each unit j.
 // Back propagate the errors
 for each unit j in the output layer
 Err j = O j (1- Oj) (T j – O j);
// Compute the error
 for each unit j in the hidden layers, from the last to the first hidden layers
 Err j = O j (1- Oj) Sk Err k w jk;
//Compute the error wrt the next higher layer, k
 for each weight w ij in network {
 D w ij = (l)Err j Oi; // weight increment
 wij = wij + D w ij; } // weight update
 for each bias q j in network {
 D q j = (l)Err j; // bias increment
 q j = q j + D q j; } // bias update
   }
}
```

Pseudo Code of Back Propagation

3.2 Support Vector Machine

SVMs are sets of supervised learning methods whose training technique permits to represent complex non linear functions. The characteristic parameters of the system are determined solving a quadratic convex optimization problem. That SVM has the "mathematical advantage" of having a global minimum, it ensures that the resulting parameters are actually the best that can be found for the problem, given the particular training set. The Pseudo Code of the Support Vector Machine is shown below



Maximum margin hyperplanes for SVM

Training an SVM Classifier

Train an SVM classifier with the `svmtrain` function. The most common syntax is:

svmstruct=svmtrain(data,groups,kernel function,rbf);

The inputs are:

Data : Matrix of data points, where each row is one observation, and each column is one feature.

Groups: Column vector with each row corresponding to the value of the corresponding row in `data`. `groups` should have only two types of entries. So `groups` can have logical entries, or can be a double vector or cell array with two values.

Kernel_Function: The default value of 'linear' separates the data by a hyperplane. The value 'rbf' uses a Gaussian radial basis function. The resulting structure, SVMstruct, contains the optimized parameters from the SVM algorithm, enabling you to classify new data.The Pseudo Code of the Support Vector Machine is shown below

Require: ***X*** and ***y*** loaded with training labeled data, ***α*** <= 0 or ***α*** <= partially trained
SVM
 C <= some value (10 for example)
 repeat
 for all *{**xi**, yi}, {**xj**, yj}* **do**
 Optimize *αi* and *αj*
 end for
 until no changes in ***α*** or other resource constraint criteria met
Ensure: Retain only the support vectors (***αi*** > 0)

Pseudo Code for Support Vector Machine

3.3 Particle Swarm Optimization

PSO is one of the evolutionary computation techniques which developed from the social behavior of swarm in nature. It can determine the solution iteratively for an objective function from the searching area. The evaluation of PSO algorithm is accomplished by depends on the moment of each particle and the swarm collaboration. Depending on the best knowledge and experience of swarm, each particle starts to move randomly from their initial position. The particles are attracted toward the location of the

current global best position and the personal best position. The basic algorithm step can be explained in three stages which are described as follow,

(i) Evaluating the fitness value of each particle.

(ii) Updating local and global best fitness and positions.

(iii) Updating the velocity and the position of each particle.The Pseudo Code of the Particle Swarm Optimization is shown below

1. Begin
2. randomly initialize particles swarm
3. while (the stopping criterion is not met)
4. evaluate fitness of particles
5. for n = 1 to number of particles
6. find *pbest*
7. find *gbest*
8. for d=1 to number of dimension of particle
9. update the position of particles by Eq. (1)-(2)
10. next d
11. next n
12. update the inertia weight value by Eq.(3)
13. next generation until stopping criterion is met
14. End

Pseudo Code for Particle Swarm Optimization

Velocity and Fitness Calculation

Step 1: The velocity and position of all particles are randomly set to within pre-defined ranges

Step 2: Velocity updating– At each iteration, the velocities of all particles are updated according to

$$v_i=w.v_i+c_1r_1(p_{i,best}-p_i)+c_2r_2(g_{i,best}-p_i) \quad (1)$$

where p_i and v_i are the position and velocity of particle *i* respectively.

p_i,*best* and g_i,*best* is the position with the 'best' objective value found so far by particle *i* and the entire population respectively.*w* is a inertia weight. r_1 and r_2 are random variables in the range [0,1]. c_1 and c_2 are factors controlling the related weighting of corresponding terms. The random variables help the PSO with the ability of stochastic searching.

Step 3: Position updating– The positions of all particles are updated according to

$$p_i=p_i+v_i \quad (2)$$

After updating, p_i should be checked and limited to the allowed range.

Step 4: Memory updating– Update $p_{i,best}$ and $g_{i,best}$ when condition is met

$$p_{i,best}=p_i$$
$$g_{i,best}=g_i \quad (3)$$

Where *f(x)* is the objective function to be optimized.

Step 5: Stopping condition– The algorithm repeats steps 2 to 4 until certain stopping conditions are met, such as a pre-defined number of iterations.

Once stopped, the algorithm reports the values of g_{best} and $f_{(gbest)}$ as its solution.PSO utilizes several searching points and the searching points gradually get close to the global optimal point using its p_{best} and g_{best}. Initial positions of p_{best} and g_{best} are different. However, using thee different direction of p_{best} and g_{best}, all agents gradually get close to the global optimum.

3.4 Genetic Algorithm

A GA is a stochastic general search method. It proceeds in an iterative manner by generating new populations of individuals from the old ones. Every individual is the encoded (binary, real, etc.) version of a tentative solution. The canonical algorithm applies stochastic operators such as selection, crossover, and mutation on an initially random population in order to compute a new population. In generational GAs all the population is replaced with new individuals. The total process is described as follows:

1- Generate randomly an initial population
2- Evaluate this population using the fitness function
3- Apply genetic operators such selection, crossover, and mutation
4- Turn the process "Evaluation Crossover mutation" until reaching the stopped criteria fixed in prior.

The Pseudo Code of the Genetic Algorithm is shown below

t = 0;

The initial population *P*(*t*) is randomly chosen; Repeat Evaluate each chromosome;

For *i* = 1 to *n* do

Randomly choose a chromosome *c* from the current population; If there are chromosomes in the mating region of *c*:

i. Use proportional selection to choose a chromosome fo crossover;
ii. Crossover takes place and a descendant is obtained;
iii. The descendant replaces the worst chromosome in its own mating region with respect to the fitness function provided the descendant is better. Else, if there are no Chromosomes in the mating region of *c*, mutation is applied to *c*. The descendant replaces the parent chromosome only if it is better. Merging is applied to all chromosomes; *t* = *t* + 1;Until (stop condition)

Pseudo Code for Genetic Algorithm

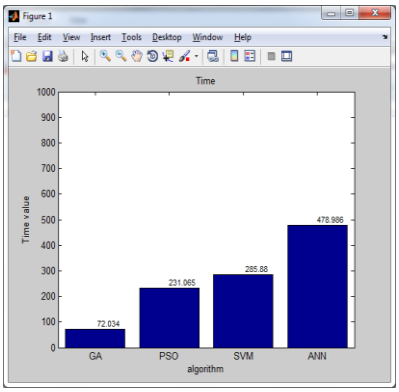
4. RESULTS AND DISCUSSION

4.1 Diabetes Dataset

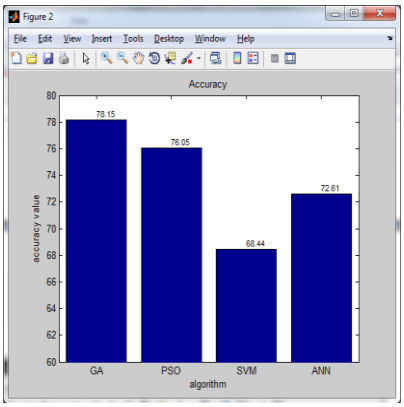
Data sets used for the proposed system are Diabetes for women. The diabetics dataset contains information such as number of times pregnant, plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/ (height in m)^2), Diabetes pedigree function and Age (years). This dataset is mainly based on women's pregnancy time. The classification experiments are conducted on the diabetes dataset. A total of 768 cases are available in diabetes dataset. 14 patients had a glucose level of 0, 11 patients had a body mass index of 0, 43 others had a diastolic pressure of 0, 170 others had successive rate of water level of 0, and 165 others

had serum insulin levels of 0. After deleting these cases there were 366 Cases with no missing values (140 tested positive cases and 226 tested negative). The data set was divided into training and testing using 55-45 ratio.

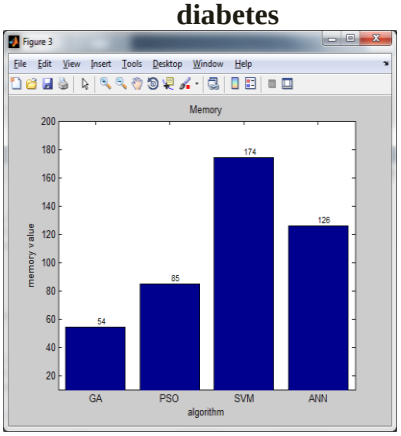
Time Slot of Soft Computing Techniques for diabetes



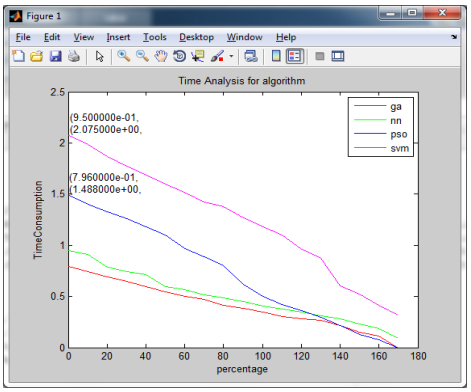
Accuracy of Soft Computing Techniques for diabetes



Memory Slot of Soft Computing Techniques for diabetes



Time Slot Chart in Milliseconds



5. CONCLUSION AND FUTUREWORKS

The system produces an accuracy level of 68% to 80% by the comparison of algorithms. The experimental results show that the Genetic Algorithm technique is superior to Neural Network, Support Vector Machine and Particle Swarm Optimization. Genetic Algorithm technique provides high level of accuracy and occupies less memory in short period of time. The system produces feasible solution for the diagnosis of diabetics. The system can be enhanced with the following features in future, besides the issue of the time constant for the past history of the input variables, the other significant factors that influence the glucose metabolism, like weight, genes and blood content as well as the physiological parameters such as heart rate and skin impedance can be integrated in the modeling of the neural network, support vector machine, particle swarm optimization, genetic algorithm.

6. REFERENCES

1. file:///F:/svm.htm#ref24
2. file:///G:/Introduction_to_Particle_Swarm_Optimization.htm
3. Fortin, V., Ouarda, T. B. M. J. & Bobée, B. (1997) Comment on "The use of artificial neural networks for the prediction of water quality parameters" (ed. by H. R. Maier & G. C. Dandy). Wat Resour. Res. 33, 2423-2424.
4. HIAN CHYE KOH , School of Business, SIM University, Singapore.
5. Yongjian Fu, "Data Mining :Tasks, Techniques, and Applications".