International Archives of Medical Research
Volume 1, No. 1, pp.28-42, 2011.

ORIGINAL RESEARCH PAPER

Forecasting Diabetes Mellitus with Biometric Measurements

Emrullah Acar[1], Mehmet Siraç Özerdem[2,*] and Veysi Akpolat[3]

[1] Batman University, Faculty of Engineering, Batman / Turkey, emrullah.acar@batman.edu.tr
[2] Dicle University, Faculty of Engineering, Diyarbakir / Turkey, sozerdem@dicle.edu.tr
[3] Dicle University, Faculty of Medicine, Diyarbakir / Turkey, vakpolat@dicle.edu.tr

Abstract

Forecasting diabetes mellitus with biometric measurements is presented in this paper. Multilayer perceptron, Elman, ART1 Neural Networks, K-Nearest Neighbor (k- NN) and Support Vector Machine (SVM) methods were used in diabetes mellitus forecast system. The result of this study will provide alternative solutions to the medical staff in determining whether someone has diabets or not which is much easier rather than presently doing a blood test. The feedforward and feedback neural networks, K-Nearest Neighbour (k -NN) and Support Vector Machine (SVM) classifiers have been chosen for learning and testing of 768 data where 268 of them are diagnosed with diabetes. For forecasting system, 8 different biometric measurements were used. These parameters are; number of times pregnant, plasma glucose concentration, blood pressure, triceps skin fold thickness, serum insulin, body mass index, diabetes pedigree function and age. Different structures of networks were tested and the results are compared in terms of testing performance for each network model. The main purpose of this study is to forecast whether someone has diabetes or not. Finally, the best performance was observed as 87.06% in the LS-SVM model structure.

Keywords: *Forecast, Diabetes Mellitüs, Artificial Neural Networks (ANNs), K-Nearest Neighbor (k- NN), Support Vector Machine (SVM)*

[*] Corresponding author: E-mail: siracozerdem@gmail.com, Tel.: +90 412 2488001 (ext. 3540), Fax: +90 412 248 8218

Introduction

DIABETES is at this time one of the greatest growing global health causes as it is fourth largest occasion decease intercontinental. It is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels. The incidence and prevalence of diabetes are escalating especially developing and newly industrialized nations. More than 220 million people worldwide have diabetes. More than 80% of diabetes deaths occur in low-and middle-income countries. The WHO’s Global strategy on diet, physical activity and health complements WHO's diabetes work by focusing on population-wide approaches to promote healthy diet and regular physical activity, thereby reducing the growing global problem of overweight and obesity [1].

Therefore, forecasting whether someone has diabetes or not is very important.

Artificial Neural Network (ANN) techniques are non-linear statistical data modeling or decision making tools.They can be used to model complex relationships between inputs and outputs or to find patterns in data. In pattern recognition, the k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. A Support Vector Machine (SVM) is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. These methods were used for forecasting system in this paper.

In our present work the problem of forecasting diabetes has been treated as binary classification problem, (i.e; the persons who are diabetic falls in a class 1 and who are non-diabetic falls in a class 0). Feedforward Networks (i.e, Perceptron and Multilayered Perceptron) and Recurrent / feedback Networks (i.e, ART Model and Elman Model), K-Nearest Neighbor (k- NN) and Support Vector Machine (SVM) classifiers are used as classifiers. Finally, their testing performances are compared to decide which model of network has best performance value. In this study, we used complete system architecture for forcasting of diabetes mellitus with biometric measurements.

Diabetes Mellitus

Diabetes mellitus is a group of metabolic diseases characterized by high blood sugar (glucose) levels, that result from defects in insulin secretion, or action, or both.With diabetes, your body cannot properly use the energy from the food you eat. When this energy transfer breaks down, the cells are damaged. Since the cells cannot take in glucose, the amount of glucose in your blood increases. Too much glucose in the blood is called "high blood sugar" or diabetes. There are two major forms of diabetes: type 1 and type 2. Type 1 of diabetes accounts for 5% to 10% of all cases of diabetes. Although it may occur at any age, type 1 diabetes usually begins early in life-during childhood or the teenage years. Type 1 diabetes occurs because the cells in the pancreas that make insulin are damaged. People with type 1 diabetes make little or no insulin to control their blood sugar levels, which means they must take insulin to stay alive. Type 2 is the most common form of diabetes. It is usually diagnosed in people older than 30, but it can occur in children and young adults. People with type 2 diabetes can produce insulin, but it is either not enough or the body does not use it properly. Blood sugar levels usually can be controlled with diet and exercise. In a mild form, type 2 diabetes can go undetected for many years. If left untreated for too long though, it can lead to serious medical problems, including heart and blood vessel disease [2].

The data collection, the forecasting methods, architecture of forecasting system and performance criteria are introduced in the following stage.

Materials and Methods

Data Collections

In this paper, the dataset collected from the Pima Indian population near Phoenix Arizona was used and consisting of 768 persons records. 268 of which were diagnosed with diabetes and rest were non-diabetes (normal) [3].

The biometric parameters considered in our present work to forecast diabetes. These parameters are; number of times pregnant, Plasma Glucose Concentration at 2 Hours in an Oral Glucose Tolerance Test (GTIT), Diastolic Blood Pressure (mm Hg), Triceps Skin Fold Thickness (mm), 2-Hour Serum Insulin Uh/ml), Body Mass Index (Weight in kg / (Height in inc)), Diabetes Pedigree Function and Age (years) which are used as inputs to the forecasting system. 567 (# diabetes, # normal) and 201 (# diabetes, # normal) patterns were selected randomly for the training and testing set, respectively.

Artificial Neural Networks

Artificial neural networks are a computational tool, based on the properties of biological neural systems. They are powerful tools for modeling, especially when the underlying data relationship is unknown. ANNs can identify and learn correlated patterns between input data sets and corresponding target values. After training, ANNs can be used to predict the outcome of new independent input data. ANNs imitate the learning process of the human brain and can process problems involving non-linear and complex data even if the data are imprecise and noisy. An ANN is a computational structure that is inspired by observed process in natural networks of biological neurons in the brain. It consists of simple computational units called neurons, which are highly interconnected. ANNs have become the focus of much attention, largely because of their wide range of applicability and the ease with which they can treat complicated problems. ANNs are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementations of nonlinear static or dynamic systems. A very important feature of these networks is their adaptive nature, where “learning by example” replaces “programming” in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available. ANNs are now being increasingly recognized in the area of classification and prediction, where regression model and other related statistical techniques have traditionally been employed [4]. In this paper, perceptron, multilayer perceptron, elman and ART1 network structures were used in forecasting system.

a) Perceptron

The Perceptron is a single layer neural network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector . Perceptron may have continuous valued inputs. Its activation is determined by equation:

$$a = w^T u + \theta \tag{1}$$

Moreover, its output function is:

$$f(a) = \begin{cases} +1 & for\ \ 0 \le a \\ -1 & for\ \ a < 0 \end{cases} \tag{2}$$

having value either +1 or -1.

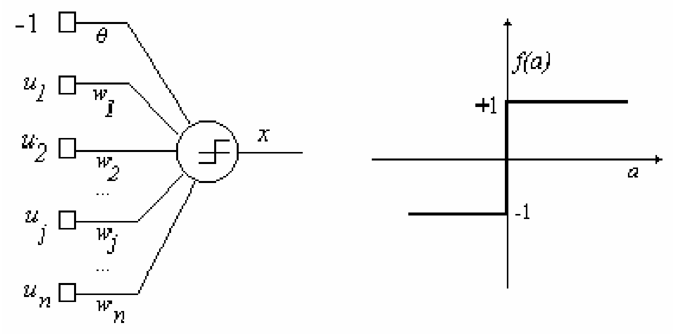

Figure 1. Single perceptron network

Now, consider such a perceptron in N dimensional space (Figure 1), the equation

$$w^T u + \theta = 0 \tag{3}$$

that is

$$w_1 u_1 + w_2 u_2 + ... + w_N u_N + \theta = 0 \tag{4}$$

defines a hyperplane. This hyperplane divides the input space into two parts such that at one side, the perceptron has output value +1, and in the other side, it is -1 [5]. A perceptron can be used to decide whether an input vector belongs to one of the two classes, say classes A and B. The decision rule may be set as to respond as class A if the output is +1 and as class B if the output is -1. The perceptron forms two decision regions separated by the hyperplane. The equation of the boundary hyperplane depends on the connection weights and threshold.

b) Multilayer Perceptron

Multilayer perceptrons (MLPs) are feedforward neural networks trained with the standard backpropagation algorithm. They are supervised networks so they require a desired response to be trained. They learn how to transform input data into a desired response, so they are widely used for pattern classification[6]. In a multilayer perceptron (MLP) topology, neurons are grouped into distinct layers as depicted in Figure 2. Output of each layer is connected to input of nodes in the following layer. That is, inputs of the first layer (input layer) are the inputs of the hidden layer. In addition, the outputs of the last layer are the output of the network.

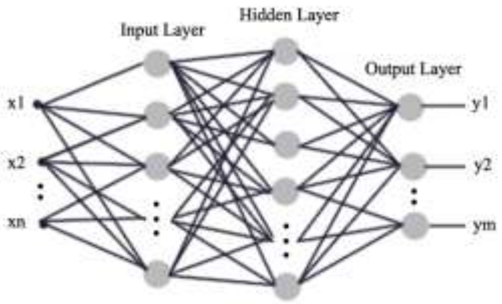

Figure 2. MLP Network with one hidden layer

A MLP is especially useful for approximating a classification function that maps input vector (x1,x2, ... xn) to one or more classes C1,C2,...Cm.

By optimizing weights and thresholds for all nodes, the network can represent a wide range of classification functions [7]. The MLP architecture with the back propagation (BP) learning algorithm, which has become the most popular in engineering applications, was used in this study. BP algorithm is based on minimization of the quadratic cost function by tuning the network parameters. The mean square error (MSE) is considered as a network performance for a training set. Parameters which minimize this cost function are determined. The averaged square error is given by equations (5) and (6) [8].

$$e_j(n) = d_j(n) - y_j(n) \tag{5}$$

$$\varepsilon(n) = \frac{1}{2P}\sum_{j\in C}\sum_{n=1}^{P} e_j^2(n) \tag{6}$$

In these equations e, n, d, y, P and C indicate error signal at the output, iteration number, desired output, generated output by network, total number of patterns contained in the training set and number of neurons at output layer respectively.

The adjustment of synaptic weights is given by equation (7).

$$\Delta w_{ji}(n) = \alpha\Delta w_{ji}(n-1) + \eta\delta_j(n)y_i(n) \tag{7}$$

For the weights between hidden and output layers

$$\delta_j(n) = e_j(n)\varphi'(\sum_{i=0}^{m} w_{ji}(n)y_i(n)) \tag{8}$$

For the weights between input layer and hidden layers.

$$\delta_j(n) = \varphi'(\sum_{i=0}^{m} w_{ji}(n)y_i(n))\sum_{k}\delta_k(n)w_{kj}(n) \tag{9}$$

where, η and α indicates learning-rate and momentum, respectively [8].

c) The ELMAN Network

Elman Networks are a form of recurrent Neural Networks which have connections from their hidden layer back to a special copy layer. This means that the function learnt by the network can be based on the current inputs plus a record of the previous state(s) and outputs of the network.This feedback path allows Elman networks to learn to recognise and generate temporal patterns as well as spatial patterns [10]. The Elman network is a recurrent neural network of simple architecture which can be trained with the backpropagation (BP) algorithm.A general architecture of Elman network is shown in Figure 3. As shown in Fig 3, the first layer is the input layer and the inputs are weighted with the weight function. Besides the output layer, all the other layers have recurrent links [10]. The state units of the Elman network can memorize all the feed inputs such that the outputs of the network depend upon the current input as well as the previous inputs. At each time step, the values of the hidden layer units are copied to the state layer, and this information can be stored for future use.

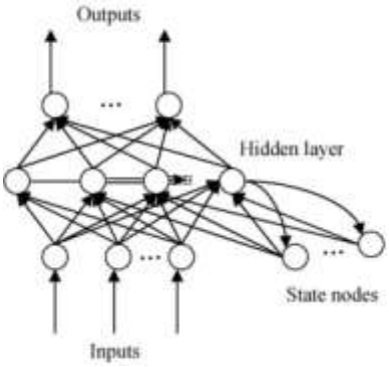

Figure 3. The general architecture of the Elman network

The network can be expressed by a state-space model. In the state-space model, the time series is modeled as a linear transformation of a time-dependent state [10]:

$$\vec{x}(t) = C\vec{s}(t) + \vec{\varepsilon}(t). \tag{10}$$

where $\vec{x}$ is the function of the time variable t, $\vec{s}$ is the state vector, C is the transformation matrix $\vec{\varepsilon}(t)$ is the noise. This time-dependent state vector can be modeled as:

$$\vec{s}(t) = As(t-1) + B\eta(t). \tag{11}$$

where A and B are the coefficient matrices. $\eta(t)$ is the noise. When the states are dependent on the past sequence vector, the state vector can be expressed as:

$$\vec{s}(t) = A\vec{s}(t-1) + D\vec{x}(t-1) \tag{12}$$

where D is the coefficient matrix. The above models are linear statespace models. In the Elman network, we have activation functions in the hidden layer with the state vector:

$$\vec{s}(t) = \delta(A\vec{s}(t-1) + D\vec{x}(t-1)) \tag{13}$$

where σ(·) represents the application of the activation function.

d) Adaptive Rezonance Theory Network (ART1)

Adaptive resonance theory is a cognitive and neural theory about how the brain develops and learns to recognize and recall objects and events throughout life[11]. The ART1 simplified model consists of two layers of binary neurons (with values 1 and 0), called F1 (the comparison layer) and F2 (the recognition layer) as shown in Fig 4.

Each neuron in F1 is connected to all neurons in F2 via the continuous-valued forward long term memory (LTM) Wf, and vice versa via the binary-valued backward LTM Wb. The other modules are gain 1 and 2 (G1 and G2), and a reset module. Each neuron in the comparison layer receives three inputs: a component of the input pattern, a component of the feedback pattern, and a gain G1.

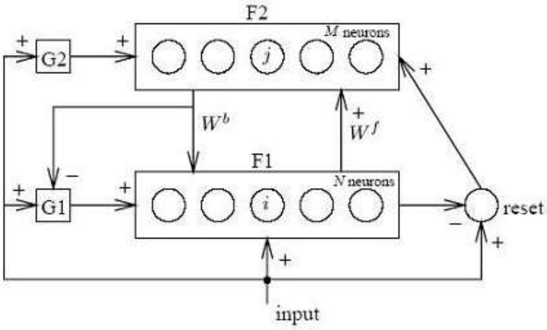

Figure 4. The general architecture of the ART1 Network

A neuron outputs a 1 if and only if at least three of these inputs are high: the 'two-thirds rule.' The neurons in the recognition layer each compute the inner product of their incoming (continuous-valued) weights and the pattern sent over these connections. The winning neuron then inhibits all the other neurons via lateral inhibition. Gain 2 is the logical 'or' of all the elements in the input pattern x. Gain 1 equals gain 2, except when the feedback pattern from F2 contains any 1; then it is forced to zero. Finally, the reset signal is sent to the active neuron in F2 if the input vector x and the output of F1 differ by more than some vigilance level.The network starts by clamping the input at F1. Because the output of F2 is zero, G1 and G2 are both on and the output of F1 matches its input. The pattern is sent to F2, and in F2 one neuron becomes active. This signal is then sent back over the backward LTM, which reproduces a binary pattern at F1. Gain 1 is inhibited, and only the neurons in F1 which receive a 'one' from both x and F2 remain active. If there is a substantial mismatch between the two patterns, the reset signal will inhibit the neuron in F2 and the process is repeated [12]. The procedure of the implemented algorithm is given below.

Step 1. Initialisation:

$$w_{ji}{}^{b}(0)=1$$
$$w_{ij}{}^{f}(0)=\frac{1}{1+N} \qquad (14)$$

where N is the number of neurons in F1, M the number of neurons in F2, 0 ≤ i<N and 0 ≤ j<M. Also, choose the vigilance threshold ρ, 0 ≤ ρ ≤ 1;

Step 2. Apply the new input pattern x:
Step 3. compute the activation values y0 of the neurons in F2:

$$y_i{}'=\sum_{j=1}^{N} w_{ij}{}^{f}(t)x_i \qquad (15)$$

Step 4. select the winning neuron k (0 ≤ k <M):
Step 5. vigilance test: if

$$\frac{w_k{}^{b}(t).x}{x.x}>p \qquad (16)$$

where . denotes inner product, go to step 7, else go to step 6. Note that ${w_k}^b.x$ essentially is the inner product $x^*.x$ which will be large if x^* and x near to each other;

Step 6. neuron k is disabled from further activity. Go to step 3;
Step 7. Set for all l, 0 ≤ l < N:

$$\begin{aligned} &{w_{kl}}^b(t+1) = {w_{kl}}^b(t)x_l \\ &{w_{lk}}^f(t+1) = \frac{{w_{kl}}^b(t)x_l}{1/2+\sum_{i=1}^{N}{w_{ki}}^b(t)x_i} \end{aligned} \tag{17}$$

Step 8. re-enable all neurons in F2 and go to step 2.

e) K- Nearest Neighbor Classifier (k-NN)

The well-known k-NN approach to classification has proven successful in many applications. In this method, we measure the distance from a test set item to each of the training set items, noting the k training set items that are nearest.

We then classify the test set item by whichever class is most common among those k “nearest neighbors,” letting each neighbor “vote.” (In case of ties, we have chosen to include all training set items no farther away than the kth nearest neighbor, so in this case there will be more than k voters.) A number of investigators have considered the question of how best to measure distance: approaches have included global metrics ,local metrics ,metrics that are specific to the problem and so on. By far the most common metric, though, has been Euclidean distance, under which the distance between two points xr and xs, say, is given by the square root of the (possibly weighted) sum of the squared distances over each co-ordinate.Although generalizations are possible, we use the simple form:

$$d(x_r, x_s) = \left[\sum_{i=1}^{p} c_i (x_{ri} - x_{si})^2\right]^{1/2} \tag{18}$$

In ordinary Euclidean distance, the weights c_i, $i = 1, \ldots, p$ are all equal to 1.However, experience suggests that two relatedsteps can improve classi&cation accuracy.First, we might expect some of the measurements to be irrelevant to the problem.Naturally we hope to be able to give weights of zero to these irrelevant columns. This echoes, of course, the variable selection problem that appears in almost every statistical model.

Second, relevant variables may be measuring similar quantities on quite different scales. Under these circumstances, it seems obvious that reducing each of the variables to a common scale may help k-NN classification by preventing one of the measurements from dominating all the others. A third problem that needs to be tackled is that of selecting the best value of k, the number of neighbors to be considered [13].

f) Support Vector Machine Classifier (SVM)

Support vector machine (SVM) classifier, motivated by results of statistical learning theory,is one of the most effective machine learning algorithms for many complex binary classification problems.Given the training set $T = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\ \} \in (x \times y)^l$,when the penalty factor C and kernel function K(.,.) are selected properly, we can construct a function

$$g(x) = \sum_{i:x \in x_+} aiK(x, x_i) - \sum_{i:x \in X_-} aiK(x, xi) + b, \tag{19}$$

where the non-negative weights a_i and b are computed during training by solving a convex quadratic programming. In order toestimate the probability of an unlabeled input x belonging to the positive class, $P(Y = 1|x)$, we map the value g(x) to the probability by

$$P_r(y = 1|x) = P_{A,B}(g(x)) \equiv 1/[1 + \exp(A * g(x) + B] \tag{20}$$

where A and B are then obtained by solving the optimization problem.

$$\underbrace{\min}_{z=(A,B)} f(z) = -\sum_{i=1}^{l} (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \tag{21}$$

$$s.t. \quad t_i = \begin{cases} (N_+ + 1)/(N_+ + 2) & if\ y_i = +1 \\ 1/(N_- + 2) & if\ y_i = -1 \end{cases} \tag{22}$$

$$pi = P_{(A,B)}(g(x_i)), \quad i = 1, ..., l \tag{23}$$

where N_+ and N_- , respectively, represent the number of positiveand negative points in training set. Then the label of the new input x is assigned to be positive if the posterior probability is greater than a threshold, otherwise negative, i.e [14].

$$f(x) = \begin{cases} +1 & if\ P_r(y = 1|x) > threshold, \\ -1 & otherwise, \end{cases} \tag{24}$$

where 1 corresponds to diabetic, whereas -1 corresponds to non-diabatic.

Architecture of Forecasting System

In this study, 8 biometric parameters have been chosen for forecasting diabetes mellitus system. The inputs described below are considered significant risk factors for diabetes sufferer. The input biometric variables with their ranges chosen are:

x1. Number of times pregnant [0, 17]
x2. Plasma Glucose Concentration at 2 Hours in an Oral Glucose Tolerance Test (GTIT) [0, 199]
x3. Diastolic Blood Pressure (mm Hg) [0, 122]
x4. Triceps Skin Fold Thickness (mm) [0, 99]
x5. 2-Hour Serum Insulin (Uh/ml) [0, 846]
x6. Body Mss Index (Weight in kg / (Height in in)) [0, 50]

x7. Diabetes Pedigree Function [0, 1.78]
x8. Age (years) [21, 67]

k-nearest neighbor (k-NN) method, support vector machine (SVM) and four different types of neural networks such as perceptron, MLP, Elman and ART1 were used for forecasting system. Each network has different structures are trained to determine the network having highest performance. The constructed system for forecasting the diabetes mellitus with biometric measurements is shown in Fig. 5. Figure 5 shows the system architecture having eight inputs, and two outputs (diabetes/normal person). For perceptron, MLP and Elman networks, the input variables are normalized in the range of [0, 1] as a pre-process. Since ART1 is used only with binary digits, the input variables converted into binary form as a pre-process. An example of convertion for 0.32 input value: round(0.32) → 0.3*10 → 3 digit is 1 ≡ [1110000000]. So, each input has 10 digits for ART1 network.

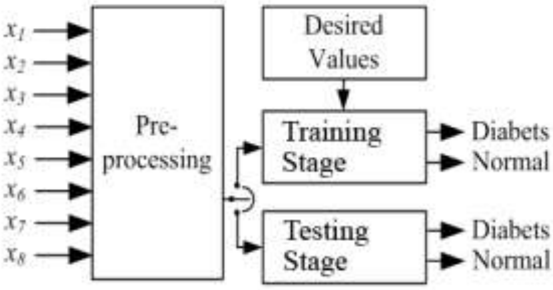

Figure 5. Forecasting system of diabetes mellitus

Similarly,the input variables are normalized in the range of [0,1] as a pre-process for k-NN and SVM classifiers.

Performance Criteria

The data set has 768 patterns and 268 patterns of which were diagnosed with diabetes. The entire sample of 768 was first randomly seperated into a training and test sample, with 567 cases for training and 201 cases for test set. For training the ANN, k-NN and SVM model, the normalized data were fed into the network and these models were trained till the networks obtained sufficiently small error target. It was observed that during the training period for the ANN networks, the MSE decreased with increasing number of iterations [11]. In the testing stage, the network was tested by 201 remaining data selected randomly. This process was repeated for each ANN, k-NN and SVM models to obtain the highest network performance. Then training and testing performance values of each model were tabulated. The performance of each network was calculated as shown in eqn (25).

$$performance = \frac{\#of\ correctly\ forecasted\ pattern}{Totally\ \#of\ tested\ pattern}.100 \tag{25}$$

Errors occurring at the learning and testing stages are called the root-mean squared (RMS) defined as follows

$$RMS = ((1/p)\sum_j \left|t_j - o_j\right|^2)^{1/2} \tag{26}$$

In the equation, p, t and o indicate total number of pattern, desired output and generated output by network, respectively.

Performance values were evaluated for each network model and RMS values were calculated for each ANN network structure to find out which network model having the highest performance with lowest RMS value.

Result

In this paper, we present the results of forecasting system based on ANNs having different architectures, k-NN and LS-SVM. Since the ANNs with one single layer have best performance observed in many applications, one single layer was used for the ANNs in this study.

Each structure of ANN networks has 8 input nodes (Since, each input has 10 digits, 80 input nodes were used in the input layer for the ART1 network) in the input layer and 1 output node in the output layer. The problem is a binary classification. That is, the output here is either 0 or 1, where 0 indicates non-diabetic and 1 as diabetic.

The performances of ANNs such as perceptron (Table I), MLP (Table II), the Elman (Table III), ART1 (Table IV); K-Nearest Neighbor (Table V) and Least Squares Support Vector Machine (Table VI) are calculated.

Each structure was trained with different learning methods for the forecasting system. It was observed that all the results of testing stages for the trained system have at least 70% performance except the ART1 network.

As shown in the Table 1, Table 2 and Table 3, the observed performance of the ANNs structures have the range in [70.15 82.10]. MLP with 8-20-2 structure has the highest performance was observed among these three structures.

Table 1. The performances of different structures of perceptron. The weights are randomly initialized in the range (0,1). Learning factor is 0.9.

Perceptron Structure (Epoch)	# of Correctly Detected Patterns	Performance	RMS
5	162 / 201	80.59 %	0.44
25	163 / 201	81.01 %	0.43
50	151 / 201	75.12 %	0.49
75	158 / 201	78.65 %	0.46
100	161 / 201	80.09 %	0.44
150	141 / 201	70.15 %	0.54
200	150 / 201	74.62 %	0.50
250	163 / 201	81.09%	0.43

As shown in the Table 4, the observed performance of the ART1 ANN structure has the range in [59.20 62.68]. The ANN with three different vigilance threshold (ρ) values was trained. It was observed that ART1 NN has lower performances when it was compared with the other structures.

As shown in the Table 5, the observed performance of the k-NN classifier structure have the range in [74.13 79.10]. The number of nearest neighbors (k) was taken in the range [2 20] for classify the data and the highest performance was observed with k=18. Figure 6 shows the performance of k-

NN classifier versus number of nearest neighbors used in classification.

As shown in the Table 6, the observed performance of the LS-SVM model was 87.06%. This performance was the best among the all analysis.

Table 2. The performances of different structures of MLP. The weights are randomly initialized in the range (0,1). Learning factor is 0.9 and momentum is 0.95. Activation function is sigmoid for the layers.

MLP Structure	# of Correctly Detected Patterns	Performance	RMS
8-5-1	158 / 201	78.60 %	0.37
8-10-1	151 / 201	75.12 %	0.39
8-15-1	149 / 201	74.13 %	0.40
8-20-1	165 / 201	82.10 %	0.38
8-25-1	151 / 201	75.12 %	0.39
8-30-1	151 / 201	75.14 %	0.39
8-40-1	153 / 201	76.61 %	0.43

Table 3. The performances of different structures of Elman. Activation function is log-sigmoid for the layers.

Elman Structure	# of Correctly Detected Patterns	Performance	RMS
8-5-1	154 / 201	76.61 %	0.38
8-10-1	149 / 201	74.13 %	0.39
8-15-1	153 / 201	76.11 %	0.38
8-20-1	148 / 201	73.63 %	0.38
8-25-1	149 / 201	74.13 %	0.39
8-30-1	153 / 201	76.11 %	0.38
8-40-1	149 / 201	74.32 %	0.39

Table 4. The performance of ART1.

ρ	# of Correctly Detected Patterns	Performance (%)	RMS
0.9	126 / 201	62.68	0.61
0.7	124 / 201	61.69	0.62
0.5	119 / 201	59.20	0.62

The statistical value, namely, RMS, is within acceptable ranges which meet the integrity of the ANNs testing stages.

Table 5. The performance of k- NN Classifier

The number of nearest neighbors (k)	Misclass	Performance (%)
2	0.25	74.13
4	0.22	77.11
6	0.23	76.62
8	0.23	76.62
10	0.24	76.12
12	0.25	75.62
14	0.25	75.62
16	0.22	77.61
18	0.20	79.10
20	0.23	77.61

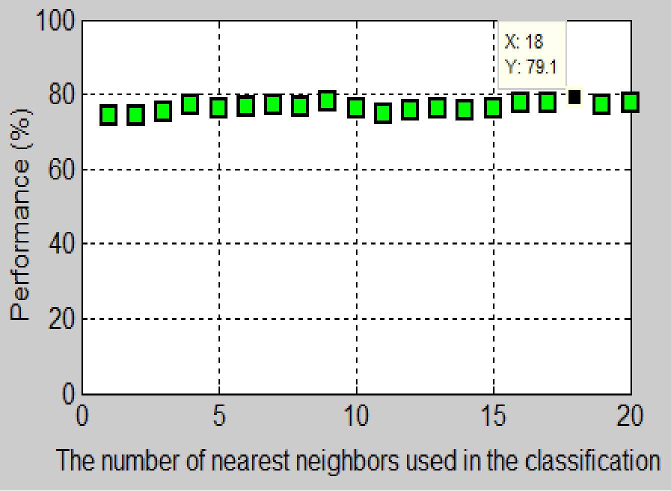

Figure 6. Performance based k-NN Architecture

Table 6. The Performance of LS-SVM Classifier

[gamma, sig2]	optimization routine	cost function	Performance (%)	RMS
4.18, 12.04	simplex	Leave one out ls-svm	87,06	0.38

Since the best performance was observed in the LS-SVM model, the corresponding ROC curve was created shown on Figure 7. The ROC curve is a fundamental tool for diagnostic test evaluation. In a ROC curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (Specificity) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC curve is a measure of how well a parameter can distinguish between two diagnostic groups (diabets/normal). That is, accuracy is measured by the area under the ROC curve. Since the area under the ROC curve determined as 0.83913 which is in the range [0.80 0.90], accuracy of forecasting would be considered to be "good".

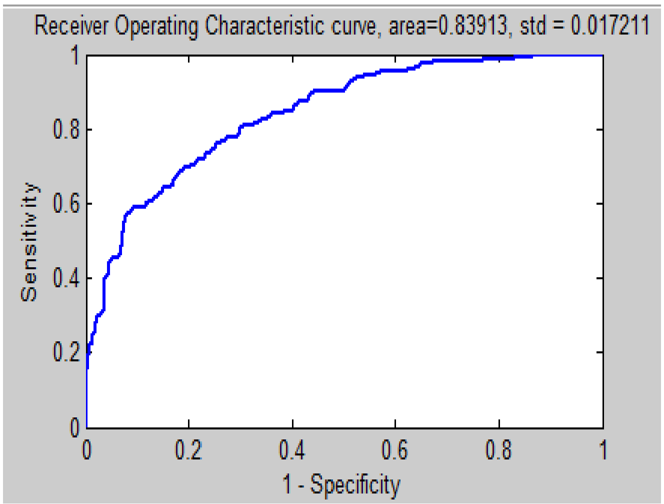

Figure 7. ROC curve of the Diabetes Mellitus.

Conclusion

Forecasting diabetes mellitus with biometric measurements is presented in this paper. Multilayer perceptron, Elman, ART1 Neural Networks, K-Nearest Neighbor (k- NN) and Support Vector Machine (SVM) methods were used in diabetes mellitus forecast system. In this paper, the problem of forecasting diabetes has been treated as binary classification problem, (i.e; the persons who are diabetic falls in a class 1 and who are non-diabetic falls in a class 0). For the system, 8 biometric measurements were used as inputs, which are number of times pregnant, plasma glucose concentration, blood pressure, triceps skin fold thickness, serum insulin, body mass index, diabetes pedigree function and age. During the learning stages of forecasting system, it was observed that each system having different learning method has high performance such as above 90% to learn the patterns. However, the same performance could not be observed in the testing stages. Although MLP neural network with different structures had high performance, the best performance was observed in the LS-SVM classifier as 87.06%.

References

1. World Health Organization, Media Centre, Diabetes

2. Texas Heart Institute, Heart Information Center, Diabetes Mellitus

3. Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C. and Johannes, R. S. (1988) Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications in Medical Care (Washington, 1988), ed. R. A. Greenes, pp. 261–265. Los Alamitos, CA: IEEE Computer Society Press.

4. Artifical Neural Networks, GIRISH KUMAR JHA, Indian Agricultural Research Inst., PUSA, New Delhi-110 012

5. Frank Rosenblatt. (1958). "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." Psychological Review, 65(6).

6. http://www.nd.com/definitions/mlp.htm

7. Rudjer Boskovic Institute, Neural Networks tutorials

8. Zurada, J.M., 1992. Introduction to Artificial Neural Networks. West Publishing Company, pp. 423–426.

9. University of Ulster, School of Computing and Intelligent Systems, Computational Intelligence Course, Recurrent Networks and Neural Network Applications Tutorial.

10. S.I. Ao and V. Paladea, “Ensemble of Elman neural networks and support vector machines for reverse engineering of gene regulatory networks”, Applied Soft Computing, Vol 11, Issue 2, March 2011, Pages 1718-1726

11. Encyclopedia of Cognitive Science Book, Adaptive Resonance Theory, Stephen Grossberg

12. Principe, J.C. "Artificial Neural Networks"(extract).The Electrical Engineering HandbookEd. Richard C. Dorf Boca Raton: CRC Press LLC, 2000.

13. Samuel E. Buttrey, and Ciril Karo." Using k-nearest-neighbor classification in the leaves of a tree." Computational Statistics & Data Analysis 40 (2002) 27 – 37. Received1 May 2001; accepted 1 October 2001.

14. Yan Xu a,b, Xiao-Bo Wang a, Jun Ding b, Ling-Yun Wu c, Nai-Yang Deng a," Lysine acetylation sites prediction using an ensemble of support vector machine classifiers", Journal of Theoretical Biology 264 (2010) 130–135.