Diabetes Detection Using Principal Component Analysis and Neural Networks

R. Haritha[1](✉), D. Sureshbabu[2], and P. Sammulal[3]

[1] Computer Science Engineering, Sree Chaitanya Institute of Technological Sciences, Karimnagar, India
haritha.ravula@gmail.com

[2] Computer Science and Engineering, Kakatiya Government College, Hanamkonda, India
sureshd123@gmail.com

[3] Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, India
sammulalporika@gmail.com

Abstract. Data mining is a growing discipline in the medical field that aims to extract knowledge relevant large amounts of data. It uses tools from statistics, artificial intelligence, and optimization techniques, etc. This paper present the detection of diabetes on the basis of data taken form UCI repository (PIMA), with help of neural network and principal component analysis. Data training and testing perform according to k fold verification and NN based approach yields 99% of accuracy. Further PCA NN approach is proposed for dimension reduction techniques and it gives accuracy 98.7% marginally low from NN based approach.

Keywords: UCI · NN · PCA · PIMA

1 Introduction

1.1 Expert System

An expert system is a decision support tool capable of reproducing the cognitive mechanisms of an expert or of a group of experts. It consists of 3 parts; a fact base; a rule base and an inference engine. So, an expert system is a software able to answer questions by reasoning from facts and known rules [1]. The fields of interrogation to be filled by the user can be more or less numerous and the concepts displayed in these fields will be treated by the search engine of the software by using the Boolean language “AND”, “OR”, “NOT”.

1.2 Diabetes Diagnostic Assistance

Types of diabetes: There are several types of diabetes in the world. The most famous are:

K. C. Santosh and R. S. Hegadi (Eds.): RTIP2R 2018, CCIS 1036, pp. 270–285, 2019.
https://doi.org/10.1007/978-981-13-9184-2_24

(1) *Type 1 diabetes* (Known as "insulin-dependent diabetes") usually affects children and adults under 30/40 (young subjects). The onset of this type of diabetes is brutal. Affected individuals lose weight due to their diabetes and are treated for life with insulin therapy.
(2) *Type 2 diabetes* (Known as "non-insulin-dependent diabetes") is also called diabetes of maturity and in contrast to type 1 diabetes appears very insidiously in people generally older (typically >40 years) and overweight [2].

Type 2 diabetes is much more prevalent than type 1 diabetes. It accounts for almost 90% of diabetes worldwide. Reports of type 2 diabetes in children - formerly rare - are on the rise. In some countries, diabetes accounts for almost half of new cases diagnosed in children and adolescents [2].

1.3 Classification Systems

These are systems based on classification methods. These classification methods are intended to identify the classes to which objects belong from certain descriptive features. These types of systems apply to a large number of human activities and are particularly suited to the problem of automated decision-making [3, 4]. The use of computers for the realization of this classification becomes more and more frequent. Even though the expert's decision is the most important factor in the diagnosis, classification systems provide substantial assistance as they reduce errors due to fatigue and the time required for diagnosis. As part of the classification, Artificial Neural Networks (ANN) have been widely proven in the scientific and industrial community [5–12].

1.4 Problem Located in Classification Systems

The presence of redundant attributes or highly noisy attributes in the databases, performance of the system may decline. This extreme situation risks leading to a classification without any real interest for the user [13]. This requires the use of variable selection techniques which aim to select or extract an optimal subset of the most relevant characteristics or parameters to make a better learning and ensure a good performance of the classification system.

2 Proposed Methodology

2.1 System Model

Diabetes is increasing internationally and is fourth among chronic diseases in our country and around the world. Its causes are complex but are largely due to rapid increases in the incidence of overweight, obesity and sedentary lifestyle. Although diabetes has serious consequences for the human body, a large proportion of its cases and complications could be prevented by good glycaemic control and early diagnosis. This requires the use of a diagnostic support system to facilitate decision-making and minimize uncertainty about the current or future state of the patient (Fig. 1)

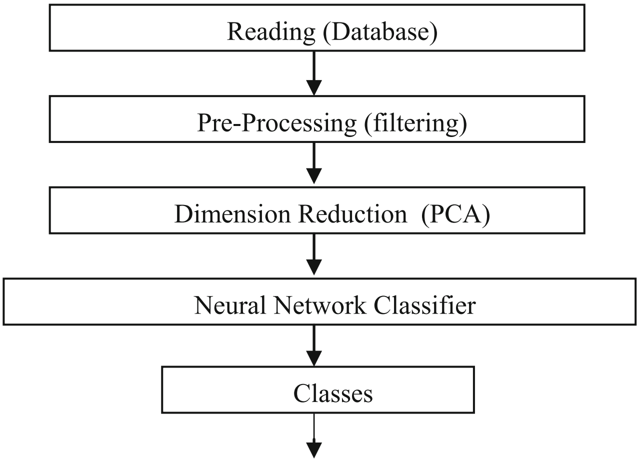

Reading (Database)
Pre-Processing (filtering)
Dimension Reduction (PCA)
Neural Network Classifier
Classes

Fig. 1. Generalized architecture of diabetic detection using neural network

2.2 Dimension Reduction

The PCA is a method of exploratory data analysis: from a set of N observations characterized by m initial quantitative variables, we try to condense the representation of the data while conserving at best their global organization. For this, we represent the data on k new variables (the principal components, k < m) obtained as linear combinations of the initial variables, keeping as much variance as possible.

The purpose of principal component analysis (PCA) is to transform an X matrix of p variables in another Y matrix of wrong virtual variables ordered from highest to lowest variance.

Let n be patterns of dimension p, that is [14]:

$$X^{*} = \left[x_{ij}\right]_{i=1,p}^{j=1,n} \tag{1}$$

We can also represent the information "centered" on its average, and obtain its average and variance:

$$X^{*} = \left[x_{ij} - \bar{x}_{i}\right]_{i=1,p}^{j=1,n} \tag{2}$$

$$\bar{x} = \frac{1}{n}\sum\nolimits_{j=1}^{n} x_{ij} \tag{3}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum\nolimits_{j=1}^{n} \left(x_{ij} - \bar{x}_{i}\right)^{2}} \tag{4}$$

$$Z = \left[\frac{x_{ij} - \bar{x}_{i}}{\sigma_{i}}\right]_{i=1,p}^{j=1,n} \tag{5}$$

The covariance matrix is defined, as:

$$S = \frac{XX^T}{n-1} = \left(S_{ij}\right)_{i=1,p}^{j=1,p} \tag{6}$$

Fulfilling the matrix S the following properties:

- S is a symmetric matrix.
- S is defined as NO negative (non-negative eigen values).
- The trace of S is equal to the Inertia of the points with respect to the origin.

The first factorial axis U_1 relative to the points under study is when this axis maximizes the inertia explained.

$$F_{kj} = Z_j^t U_k \tag{7}$$

$$F_k = \left\{F_{kj}\right\}^{j=1,n} = Z\bar{U}_k \tag{8}$$

Thus the selection of factorial axes is carried out in order of "relevance" in terms of contribution of information (eigenvalues), so each axis that is determined must contribute less and less information.

2.3 Classification

Neural networks have been widely used in the field of classification because of the simplicity of their reasoning and their learning performance inspired by human reasoning. To make a good learning, the model of the neural network will be chosen in an experimental way. It depends on the number of hidden layers, neurons in each layer, number of inputs and outputs. To have good results, the chosen model can have a very complex architecture. It possible to select the set of most relevant variables to make the right learning and at the same time to choose the best model of the neural network while ensuring its simplicity.

Artificial Neural Networks

A network of artificial neurons is a model of calculation whose design is very schematically inspired by the functioning of biological neurons. The neural networks are generally optimized by probabilistic learning methods. They are placed on the one hand in the family of statistical applications, and on the other hand in the family of artificial intelligence methods to which they provide a perceptual mechanism independent of the programmer's own ideas.

The Neuron: The Perceptron of Rosenblatt [15] transposes the behaviour of neurons into an equation integrating the great principles observed in nature. From a stimulus represented by an observation vector $x \in R_n$, each component x_i, Neural networks and variable selection $i \in [1, n]$ is multiplied by a connection weight w_i. The sum of these weighted inputs is then made by adding a bias θ (or activation threshold). For reasons of convenience of notation, this bias is transformed into a new output neuron 1 with a weight link w_0 such that $\theta = w_0$. Finally, the sum (or activation state) is passed in a non-linear staircase function f (or threshold function). In the end the output of a neuron is written in a synthetic way (Eq. (9)):

$$y = f\left(\sum\nolimits_{i=-w_i x_i}^{n}\right) \tag{9}$$

Perceptron alone can be seen as a discrimination function between two classes for a classification problem. It partitions the input space into two regions with a linear decision boundary. With well-studied weights, this linear surface can represent logical functions like AND, OR, and NOT. The Perceptron cannot simulate the exclusive OR (XOR) because in this case, the decision surface is nonlinear. Criticism of the case of XOR by [16] has also provoked a temporary but historic disaffection for the Perceptron. Although a computer neuron is not a perfect model of its biological version, it remains none the less close experimentally observed phenomena. Moreover, aggregated in network, the yet limited capacities of the artificial neuron produce very interesting results as well at the purely functional level as at the level of the modelization. Several algorithms have been proposed to determine w_i weights, starting with the experiment-based [15] method, the least squares technique [17] and finally the gradient-descent techniques as in the case of Multi-Layer Perceptron (MLP).

Computer units are no longer called Perceptron but more simply neurons or nodes. In addition to the layered topology, the main difference with the version of [15] is the use of differentiable and nonlinear activation functions such as the sigmoid, also called the logistic function.

The idea of such a topology is old and it took a number of years to see the appearance of algorithms to calculate the weight of such a network in particular because of the introduction of hidden layers. Proposed for the first time by [18] in 1994, the use of the retro propagation of the error gradient in multi-layered systems will again be brought to the forefront in 1986 by [19], and simultaneously, under a similar name, at [20] during his paper.

These networks are often fully connected, which means that each neuron in a layer i is connected to all the neurons in the i + 1 layer. On the other hand, in a classical scheme, the neurons of the same layer are never connected to each other.

MLPs are essentially used for two purposes: partitioning a shape space for classification problems and approximating functions. Unlike the Perceptron of [15], the MLP can represent any Boolean function with n variables, although some may require an exponential number of neurons in the hidden layers. Due to the non-linearity of the sigmoid as an activation function, the separation boundaries are better adapted to each class in the case of a classification problem. This property is also found in the case of the function approximation that produces continuous and smooth curves at a time.

MLPs have interesting mathematical properties. Many of them are valid for networks with only two hidden layers, which testifies to the potential power of MLPs. It should be noted that these properties are rarely constructive in the sense or that it is shown that a certain number of neurons is sufficient to perform a task, the property gives no information on the topology to choose in order to solve the problem. The majority of the properties are proven without the assumption of the use of the sigmoid, it is enough simply that the function of activation is bounded (raised and minuted), increasing and continuous.

The difficulty of using this network lies in the fact that it is necessary to determine its topology, it is a question of defining the number of neurons of the different layers as well as their interconnections.

If the number of hidden neurons is too small, the learning algorithm will not be able to construct an intermediate representation of the problem that is linearly separable and some of the examples will not be learned correctly. Conversely, if this number is too high, there is a risk of learning the problem by heart: the network perfectly recognizes the learning examples but will give poor results on new data that it did not see during the study.

Learning: The best-known approach for learning a MLP is the gradient descent technique. Indeed, the use of differentiable activation functions makes it possible to use this technique that is both simple to implement and above all very computationally efficient.

We will use in the rest of this section the following notations:

- P the number of shapes in the learning base.
- x_p, $p \in [1, P]$ the form $n^{\circ}p$ of the learning base.
- L the number of network layers (including the input and output layer).
- N_l, $l \in [0, L-1]$ the number of neurons in layer $n^{\circ}l$.
- $o_{l,j}$, the calculated output of neuron $n^{\circ}j$ in layer $n^{\circ}l$.
- $d_j(x_p)$ the component $n^{\circ}j$ of the expected output for the form x_p.
- $w_{l,j,i}$ the weight of the connection between neuron $n^{\circ}j$ in layer $l-1$ and neuron $n^{\circ}j$ in layer l.
- f is the activation function.

The output of any neuron is given by (10):

$$o_{l,j} = f\left(\sum\nolimits_{i=0}^{N_{l-1}} w_{l,j,i} O_{l-1,i}\right) \tag{10}$$

The cost function E to be minimized in the case of a learning is a measure of the error between the desired output for a shape and the output calculated by the network. The error on a form p is generally quantified by a quadratic error $E_p(w)$:

$$E_p(w) = \frac{1}{2}\sum\nolimits_{q=1}^{N_L} \left(O_{L,q}(x_p) - d_q(x_p)\right)^2 \tag{11}$$

The error for the set of forms $E_p(w)$ is therefore:

$$E(w) = \sum\nolimits_{p=1}^{P} E_p(w) \tag{12}$$

The problem boils down to:

$$\min E(w) \tag{13}$$

To solve this type of problem, a classic optimization technique resulting from operational research consists in determining by successive iterations the values of the parameter w. It consists of using an existing point w_0 and making it move in the direction of the anti-gradient. The new point obtained by the translation $w \rightarrow w + \mu x_p \sigma_p$ has a smaller value for the objective function. The parameter μ is a positive step called in this case no learning and σ is the gradient of the error. The translation operation is repeated until a satisfactory solution is obtained. By using the retro propagation of the gradient of the error, the summary of the progress of the method is given by (Algorithm 1).

```
    Algorithm 1
    Learning a MLP by gradient back propagation
1:  Random initialization of network weights

2:  repeat for each sample of the
 learning base do - Propagate the
 sample in the network

    - Calculates error on the output layer

    - Propagation of the error on the
 lower layers - Weight adjustment

    end
    Update the total error until Stop criterion
```

Although the error is minimized locally, the technique converges to a minimum and gives good practical results. In most cases, few problems due to local minima are encountered. However, it persists two problems that we encounter in a real application which are on the one hand the slowness of the convergence if μ is badly chosen and on the other hand the possible risk of converging towards a local and not global minimum of the surface error.

The main defect of this method is a relatively long convergence time which depends on different parameters such as the initialization at the instant t = 0 of the synaptic weights or the initial value of the parameter μ. Nevertheless, it gives good experimental results.

In an implementation of the error propagation retroactivity algorithm, it is also difficult to determine when the weight adjustment of the MLP should be completed. Several stop criteria are used: the iterations stop when the norm of the gradient is close to zero (the weights then vary only very little), or else as soon as the error at the exit is below a certain threshold.

The first criterion is more interesting mathematically because it corresponds to the stabilization of the solution in a minimum, the second is closer to real (interpretable) criteria of good correlation between calculated solution and expected solution. In the latter case, if the problem studied concerns a classification task, we can consider that the learning ends when all the forms are classified, which makes it possible to dispense with the determination of the error rate not to be used exceed.

In practice, we go from this last stopping criterion to a second which takes into account a maximum number of iterations not to be crossed. Indeed, it is not guaranteed that the network can classify all forms, even with an infinite number of iterations. The combination of the two conditions makes it possible to obtain a correct solution in a reasonable time.

3 Simulation Results

3.1 PIMA Database

The tests of the proposed method are carried out on the basis of Pima Indians Diabetes data [22]. The dataset was chosen from the UCI repository that conducts a study of 768 Pima Indian women (500 non-diabetic 268 Diabetics). These same women who stopped their migrations in Arizona (USA) adopting a Westernized way of life develop diabetes in almost 50% of cases. The diagnosis is a variable binary value "class" that allows to know if the patient shows signs of diabetes according to the criteria of the World Health Organization. The eight clinical descriptors are:

1. Npreg: number of pregnancies.
2. Glu: concentration of plasma glucose.
3. BP: diastolic blood pressure, (mmHg).
4. SKIN: triceps skin fold thickness, (mm).
5. Insulin: insulin dose, (mu U/ml).
6. BMI: body mass index, (weight in kg/(height in m2).
7. DPF: Diabetes pedigree function (heredity).
8. Age: age (Year).

3.2 Analysis of Database Data

Table 1 contains information on the parameters taken into consideration.

Table 1. Descriptor information in the database

Attribute name	Min/Max	Standard deviation	Segregated
Npreg	0/17	3.37	17
Glu	0/199	31.973	136
BP	0/122	19.356	47
Skin	0/99	15.952	51
Insu	0/846	115.244	186
Bmi	0/67.1	7.884	248
Ped	0.078/2.42	0.331	517
Age	21/81	11.76	52

3.3 *Evaluation Criteria*

Data classification performance was evaluated by calculating true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs), percent sensitivity (S_e), specificity (S_p) and the classification rate (TC), their respective definitions are as follows:

- *VP*: diabetic classified diabetic.
- *VN*: non-diabetic classified non-diabetic.
- *FP*: non-diabetic classified diabetic.
- *FN*: diabetic classified as non-diabetic.

Sensitivity is the ability to give a positive result when the disease is present. It is calculated by:

$$S_e = \frac{VP}{VP + FN} \tag{14}$$

Specificity is the ability to give a negative result when the disease is absent. It is calculated by:

$$S_p = \frac{VN}{VN + FP} \tag{15}$$

Classification rate is the percentage of correctly classified examples. It is calculated by:

$$TC = \frac{VP + VN}{VN + VN + FP + FN} \tag{16}$$

3.4 Results and Interpretation

Experimentation 1

In the first experiment we used a multilayer perceptron with the topology [8: 6: 1] which has as input the eight parameters of the base PIMA: with a learning step = 0.5 (Fig. 2).

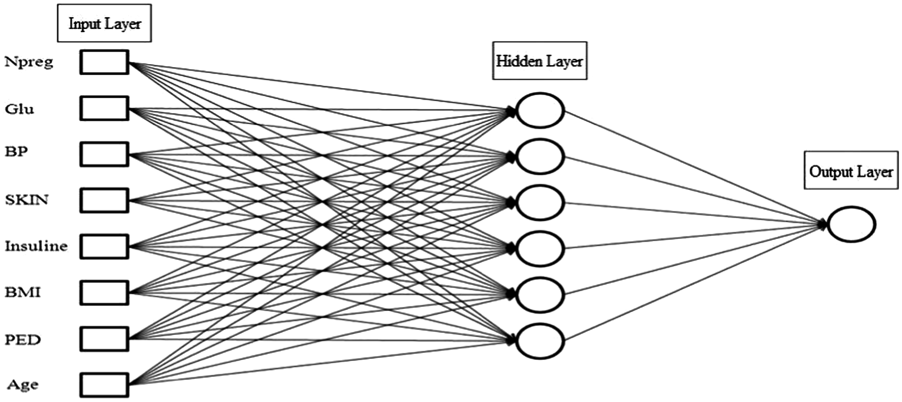


Fig. 2. Architecture used in experimentation

The results of this experiment show that the specificity of the system is very high which means that the system has made a good apprenticeship for the negative data. So when a patient is non-diabetic our model detects it very successfully.

On the other hand the sensitivity of the system is very weak which means that the system has made a bad recognition of the positive data. So many diabetic patients have been recognized as non-diabetic. This can generate a major risk for the health of the patient.

With these performances, we can say that the model gave an average classification rate and a good specificity. On the other hand it gave a weak sensitivity. What remains a disadvantage to study (Fig. 3).

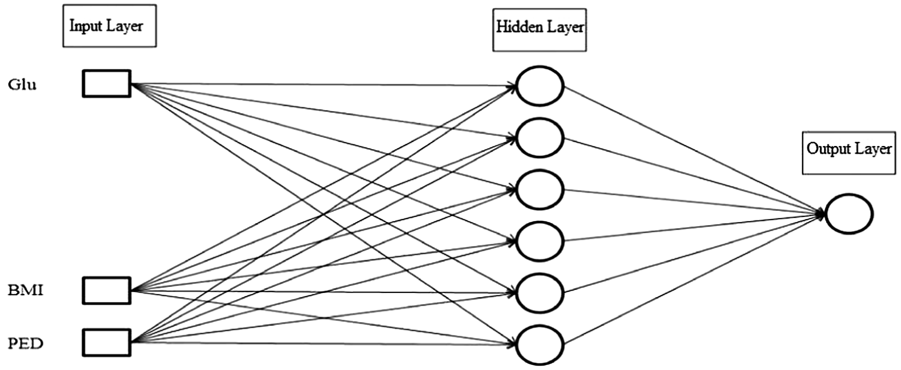


Fig. 3. Confusion matrix plot for NN based method

Here the confusion matrix of Neural network shows that out of 78 there are 77 outputs gives correct result while rest 1 gives wrong output, so accuracy count for Non-diabetic is 98.7%.

Now out of 229 cases of Diabetic 227 gives correct result while rest 2 gives wrong result, so accuracy count for Diabetic is 99.1%. So exact accuracy count drawn for both Diabetic and Non-Diabetic are 99.0%.

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{1}{1 + 77} = 1.3\%$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \frac{77}{1 + 77} = 98.7\%$$

$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{227}{227 + 2} = 99.1\%$$

$$\text{Miss} = \frac{\text{FN}}{\text{TP} + \text{FN}} = \frac{2}{227 + 2} = 0.9\%$$

$$\text{Rate Error} = \frac{\text{FN + FP}}{\text{TOTAL}} = \frac{2 + 1}{307} = 1.0\%$$

$$\text{Rate Accuracy} = \frac{\text{TN + TP}}{\text{TOTAL}} = \frac{77 + 227}{307} = 99.0\%$$

Experiment 2:

In order to improve the performance of the previous experiment, we applied the PCA method to the same architecture of Experiment 1 [8: 6: 1] with a learning step = 0.5 and a maximum of iteration = 200.

This method has selected the variables (Glu: Glucose, BMI: mass, PED: Heredity) as variables that have the discriminating power between the two classes. Coming back to the nature of diabetes disease we can confirm that this method has actually found the most relevant variables: that is why the change in glucose concentration is the most used parameter for the diagnosis of this disease and if we back to the causes of diabetes, we note that the majority of diabetic men suffer from the problem of overweight and that the genetic factor is responsible for most cases of pancreas failures which is the main abnormality of a development of a diabetes (Fig. 4).

Here the confusion matrix of PCA-NN shows that out of 36 there are 36 outputs gives correct result while rest none of the gives wrong output, so accuracy count for Non-diabetic is 100%. Now out of 113 cases of Diabetic 111 gives correct result while rest 2 gives wrong result, so accuracy count for Diabetic is 98.2%. So exact accuracy count drawn for both Diabetic and Non-Diabetic are 98.7%.

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{0}{0 + 36} = 0\%$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \frac{36}{1 + 36} = 100\%$$

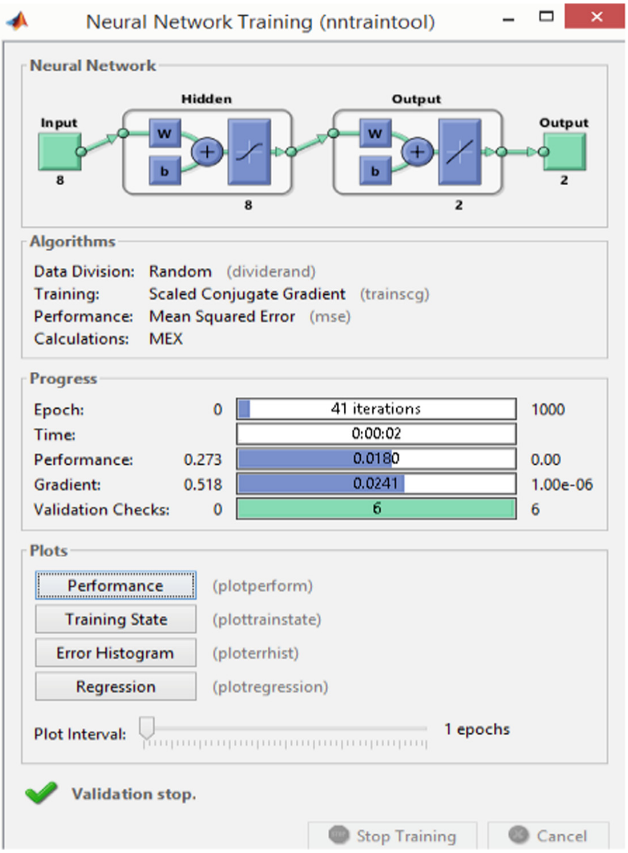
$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{111}{111 + 2} = 98.2\%$$

$$\text{Miss} = \frac{\text{FN}}{\text{TP} + \text{FN}} = \frac{2}{111 + 2} = 1.8\%$$

$$\text{Rate Error} = \frac{\text{FN + FP}}{\text{TOTAL}} = \frac{2 + 0}{149} = 1.3\%$$

$$\text{Rate Accuracy} = \frac{\text{TN + TP}}{\text{TOTAL}} = \frac{36 + 118}{149} = 98.7\%$$

We also note a slight decrease in its specificity. Table 2 summarizes the results obtained by the two experiments:


Neural Network Training (nntraintool)
Neural Network
Input
8
Hidden
W
b
8
Output
W
b
2
Output
2

Algorithms

Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	41 iterations	1000
Time:		0:00:02	
Performance:	0.273	0.0180	0.00
Gradient:	0.518	0.0241	1.00e-06
Validation Checks:	0	6	6

Plots
Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Plot Interval:
1 epochs

✓ Validation stop.

Fig. 4. Confusion matrix plot for PCA-NN based method

Table 2. Performance table

Method	Number of variables	Error rate	Specificity	Accuracy
NN (Proposed)	8	1%	%	99.0%
PCA+NN (Propose)	3	1.3%	100%	98.7%

Comparing the results of the two experiments, we note that the proposed method decreased the number of input variables by more than 60% and that the number of connections decreased by more than 55%, which gave a good optimization of the architecture of the model and a strong improvement of the classifier performances (Fig. 5).

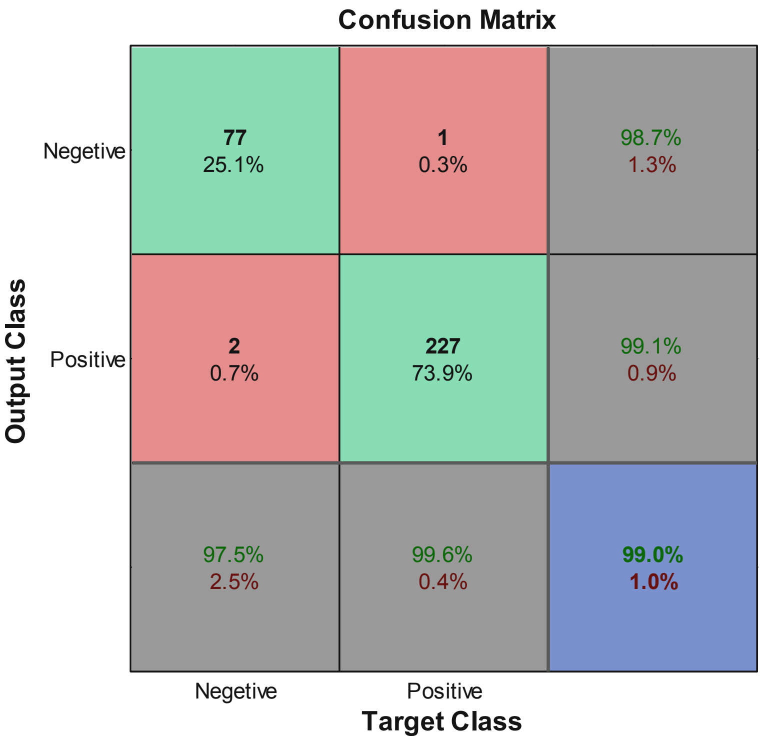


Fig. 5. Architecture found after the application of the OCD method

3.5 Comparison with Works of Literature

In order to situate the performance of the proposed approach, we carried out a comparative study between the results obtained and those of the work already done in this field (studied in the state of the art) with the PIMA database. Table 3 summarizes the comparison with the other works:

Table 3. Comparison table with literature works

Name of the method	Type of method	Number of variables	Accuracy
FCBF+SVM [23]	Filter	4	77.99%
GR+RBF [24]	Filter	5	86.46%
GR+MLP [24]	Filter	5	78.21%
CAFS+MLP [25]	Wrapper	6	76.18%
GA+SVM [26]	Wrapper	4	81.50%
TS1+MLP [27]	Wrapper	4	79.55%
OCD+MLP	Embedded	3	83.59%
NN (Proposed)	Neural network	8	99.0%
PCA+NN (Propose)	PCA and neural network	3	98.7%

After several researches on this problematic we can confirm that the method proposed is the first method of the type Embedded applied on the PIMA database and the Number of variables found by this method is the smallest number of variables selected so far. We also note that the classification rate is the best among the methods that use the MLPs and even among the other methods except in the work of Karegowda et al. [24] they found a better rate but using the RBF as a classifier. From the Table 3, it is clear that the proposed work outperforms other works (Figs. 6 and 7).

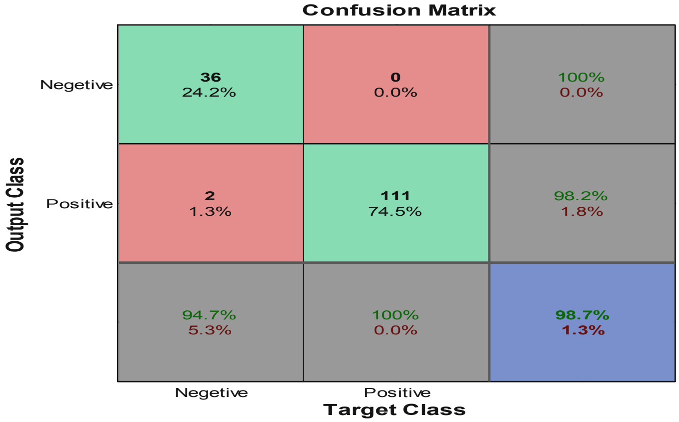


Fig. 6. Neural network training

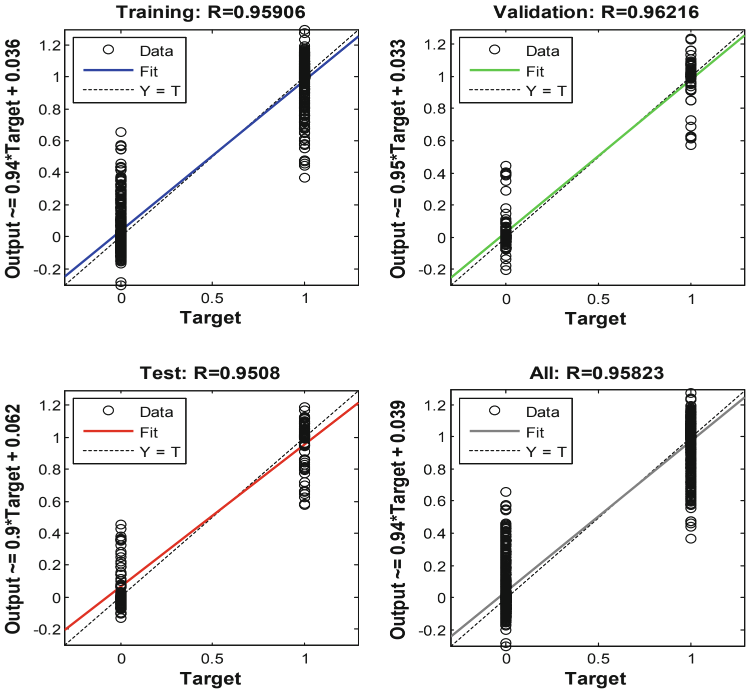


Fig. 7. Output

4 Conclusion

We presented a variable selection method that confirmed its performance in the test performed and that gave an interesting improvement of the model error, sensitivity and classification rate while optimizing its architecture. After comparing the results obtained with work in the literature, we noticed that the results found are comparable or better than the other results. This method selected the variables (Glu: Glucose, BMI: mass, PED: Heredity) as the most relevant variables to perform a better classification. The results obtained after the use of this method are very promising and are well located among the work already done in this area which confirms the rigor of the contribution proposed for the resolution of our problem. In the future, we plan to ensure the interpretability of the results of the model by integrating the notion of fuzzy with the classifier. We also want to generalize this modest application to all types of diseases in order to integrate it into the future in a system of diagnostic assistance applicable in a hospital or a medical office.

Reference

1. Zeki, T.S., Malakooti, M.V., Ataeipoor, Y., Tabibi, S.T.: An expert system for diabetes diagnosis. Am. Acad. Sch. Res. J. **4**(5), 1 (2012)
2. American Diabetes Association: Classification and diagnosis of diabetes. Diabetes Care **38** (Suppl. 1), S8–S16 (2015)
3. Lavery, L.A., Armstrong, D.G., Murdoch, D.P., Peters, E.J., Lipsky, B.A.: Validation of the infectious diseases society of America's diabetic foot infection classification system. Clin. Infect. Dis. **44**(4), 562–565 (2007)
4. American Diabetes Association: Standards of medical care in diabetes—2014. Diabetes Care **37**(Suppl. 1), S14–S80 (2014)
5. Amato, F., et al.: Artificial neural networks in medical diagnosis. J. Appl. Biomed. **11**(2), 47–58 (2013)
6. Jayalakshmi, T., Santhakumaran, A.: A novel classification method for diagnosis of diabetes mellitus using artificial neural networks. In: 2010 International Conference on Data Storage and Data Engineering (DSDE), pp. 159–163. IEEE, February 2010
7. Ahmadlou, M., Adeli, H.: Enhanced probabilistic neural network with local decision circles: a robust classifier. Integr. Comput.-Aided Eng. **17**(3), 197–210 (2010)
8. Karegowda, A.G., Manjunath, A.S., Jayaram, M.A.: Application of genetic algorithm optimized neural network connection weights for medical diagnosis of pima Indians diabetes. Int. J. Soft Comput. **2**(2), 15–23 (2011)
9. Iyer, A., Jeyalatha, S. Sumbaly, R.: Diagnosis of diabetes using classification mining techniques, arXiv preprint (2015). arXiv:1502.03774
10. Durairaj, M., Kalaiselvi, G.: Prediction of diabetes using soft computing techniques-a survey. Int. J. Sci. Technol. Res. **4**(3), 190–192 (2015)
11. Erkaymaz, O., Ozer, M.: Impact of small-world network topology on the conventional artificial neural network for the diagnosis of diabetes. Chaos, Solitons Fractals **83**, 178–185 (2016)
12. Erkaymaz, O., Ozer, M., Perc, M.: Performance of small-world feedforward neural networks for the diagnosis of diabetes. Appl. Math. Comput. **311**, 22–28 (2017)

13. Mahajan, A., Kumar, S. Bansal, R.: Diagnosis of diabetes mellitus using PCA and genetically optimized neural network. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 334–338. IEEE, May 2017
14. Mangathayaru, N., Mathura Bai, B., Srikanth, P.: Clustering and classification of effective diabetes diagnosis: computational intelligence techniques using PCA with kNN. In: Satapathy, S.C., Joshi, A. (eds.) ICTIS 2017. SIST, vol. 83, pp. 426–440. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-63673-3_52
15. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**(6), 386 (1958)
16. Minsky, M., Papert, S.A., Bottou, L.: Perceptrons: An Introduction to Computational Geometry. MIT press, Cambridge (2017)
17. Widrow, B., Hoff, M.E.: Adaptive switching circuits, No. TR-1553–1. Stanford Univ CA Stanford Electronics Labs (1960)
18. Werbos, P.J.: The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting, vol. 1. Wiley, Hoboken (1994)
19. Rumelhalt, D.E.: Learning internal representations by error propagation. Parallel Distrib. process. **1**, 318–362 (1986)
20. Le Cun, Y.: Learning process in an asymmetric threshold network. In: Bienenstock, E., Soulié, F.F., Weisbuch, G. (eds.) Disordered Systems and Biological Organization. NATO ASI Series (Series F: Computer and Systems Sciences), vol. 20, pp. 233–240. Springer, Heidelberg (1986). https://doi.org/10.1007/978-3-642-82657-3_24
21. Tetko, I.V., Livingstone, D.J., Luik, A.I.: Neural network studies. 1. comparison of overfitting and overtraining. J. Chem. Inf. Comput. Sci. **35**(5), 826–833 (1995)
22. Pima, A.F., Asuncion, A.: Pima Indians Diabetes Data Set. UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2010). http://archive.ics.uci.edu/ml
23. Balakrishnan, S., Narayanaswamy, R.: Feature selection using fcbf in type ii diabetes databases. Int. J. Comput. Internet Manag. **17**(1), 50–58 (2009)
24. Karegowda, A.G., Manjunath, A.S., Jayaram, M.A.: Comparative study of attribute selection using gain ratio and correlation based feature selection. Int. J. Inf. Technol. Knowl. Manag. **2** (2), 271–277 (2010)
25. Kabir, M.M., Islam, M.M., Murase, K.: A new wrapper feature selection approach using neural network. Neurocomputing **73**(16–18), 3273–3283 (2010)
26. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Syst. Appl. **31**(2), 231–240 (2006)
27. Wang, Y., Li, L., Ni, J., Huang, S.: Feature selection using tabu search with long-term memories and probabilistic neural networks. Pattern Recogn. Lett. **30**(7), 661–670 (2009)