Heart Disease Detection

A Performance comparison using different Machine Learning Algorithms.

Rajesh Thomas,
Dept. of Electrical Engineering,
FAMU-FSU College of Engineering,
2525 Pottsdamer Street,
Tallahassee, FL, USA.
rr15f@my.fsu.edu

Sourindu Chatterjee,
Dept. of Electrical Engineering,
FAMU-FSU College of Engineering,
2525 Pottsdamer Street,
Tallahassee, FL, USA.
sc15@my.fsu.edu

Dr. Simon Foo,
Dept. of Electrical Engineering,
FAMU-FSU College of Engineering,
2525 Pottsdamer Street,
Tallahassee, FL, USA.
sfoo@fsu.edu

***Abstract* — This project utilizes various Machine Learning algorithms to predict the occurrence of Coronary Heart Disease (CHD) in a patient with relatively high accuracy. The data for training the networks is taken from the UCI database. This project also aims at comparing the performance of the various algorithms used, namely, K-Nearest Neighbor (KNN), Backpropagation, Radial Basis Network (RBN) and Decision Trees. Pre-processing is done using Principal Component Analysis (PCA) which enables feature reduction. The work done has led to conclusion that RBN provides the maximum accuracy for the UCI database.**

***Index Terms* — Machine Learning, K-Nearest Neighbor, Backpropagation, Radial Basis Network, Decision Tree, Coronary Heart Disease.**

I. Introduction

Coronary Heart Disease (CHD) is a common heart condition which affects a substantial population of the world [2]. Modern machine learning algorithms combined with the recent advancements in digital hardware technology provides a powerful platform which enables the early detection of heart diseases if the right input data are available. This supplements the conventional process of heart disease detection which is usually done with the help of the diagnostic skill of a doctor [3].

This project uses training data gathered from the UCI database, which is comprehensive list of real world data gathered from patients around the world. The samples for the database used here has been taken from four major health institutions. UCI database is very helpful because it records many attributes that can influence CHD, out of which ten are used in this work. Preprocessing is done on the ten inputs using Principal Component Analysis (PCA) which does feature reduction. The orthogonal projection of the features on the principal axis gives the components. Now the various Machine Intelligence (MI) algorithms are used on the pre-processed data [7]. The algorithms used in this project are K-Nearest Neighbor, Backpropagation, Radial Basis Network and Decision Trees. The accuracy of the predicted data is relatively high. Future works can improve the accuracy and reliability, so that it can be used in practical applications.

II. Pre-Processing

Pre-processing the input data is very crucial in applications like this where lot of input attributes must be considered. The pre-processing technique used here is Principal Component Analysis (PCA). This is used for projecting a noisy data on the most paramount basis to eliminate the noise and reveal abstract data structure. PCA uses an orthogonal transformation to convert correlated features to linearly uncorrelated features (called principal components). This helps in feature reduction is a very important step before feeding the extracted feature dataset to various machine learning algorithms. The ten input features are reduced to six features using PCA which enables the MI algorithms to work more efficiently. The graph in Fig. 1 gives a three dimensional plot of the first three principal components. The red dots represent patients with disease and the blue dots represents people without the disease.

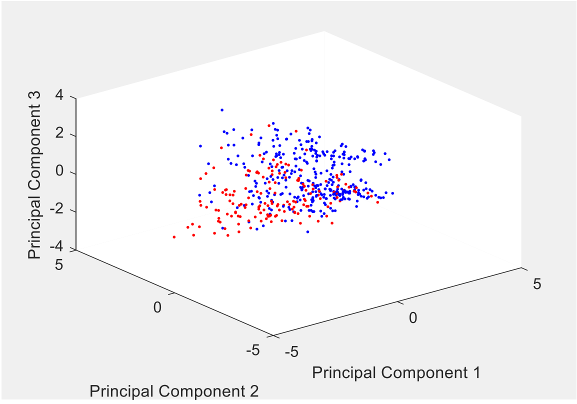

Fig. 1. A sample plot done using first three principal components after PCA

III. Algorithms

This project compares the performance of four different algorithms for detecting coronary heart disease. The pre-processed data is used as input data. The four algorithms used in this work are K-Nearest Algorithm, Backpropagation, Radial Basis Network and Decision Trees. They were chosen because these algorithms are quite prominent in machine leaning and are relatively easy to implement using powerful tools like MatLab.

A. K-Nearest Neighbour

K-Nearest Neighbor (KNN) is quite an easy to implement algorithm which is also conceptually easy to understand. KNN works by checking the 'K' closest features to the new input. The feature with the maximum frequency is selected. Thus KNN can account for local variations in features. There are different methods to find the closest features. The more common ones are using Euclidean distance, Hamming distance, etc. The accuracy of the output also depends on the value of K. K is chosen so that it should neither be too big nor too small.

In two dimensions this can be easily imagined by using a dataset with two input features. One of the features is taken along the X-axis and the other along the Y-axis. A circle is drawn around a new data point so that K values are inside the circle. The class for the new point is chosen as the class that has the greatest frequency among the K values.

Using MatLab, optimization can be done so that the best value of K and the best distance metric can be found for the given training data. The graphs given in Fig. 2. show this.

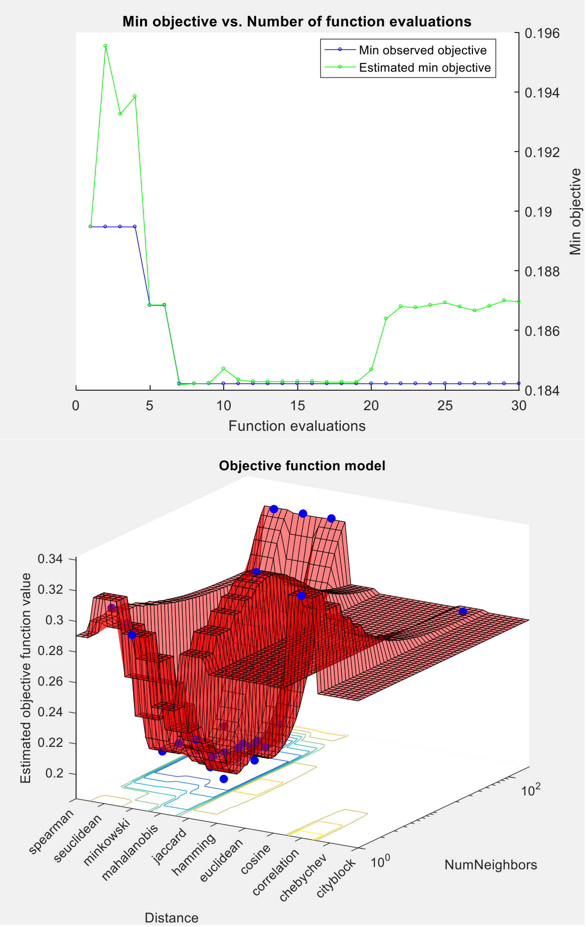

Fig. 2. KNN Optimization Graphs

B. Backpropagation

Backpropagation (also known as Multi-Layer Feed-forward Network) is a very versatile algorithm which can adapt to different scenarios, aptly earning the name 'the workhorse of learning in neural networks'. Backpropagation works by assigning weights to the input vector and the hidden neurons. The weights are changed by calculating the gradient of a loss function with respect to all the weights in the network. The gradient is fed to an optimization function whose output is used to update the weights. Since backpropagation initially needs some outputs for training (to calculate the weights), it is considered as a supervised training method [10]. The figures obtained for training from MatLab is given in Fig. 3.

For this project there are a total of ten input features which are reduced to three using Principal Component Analysis (PCA). Training is done on the inputs and the accuracy is tested using new input data. Optimization is done to find the minimum number of hidden neurons required.

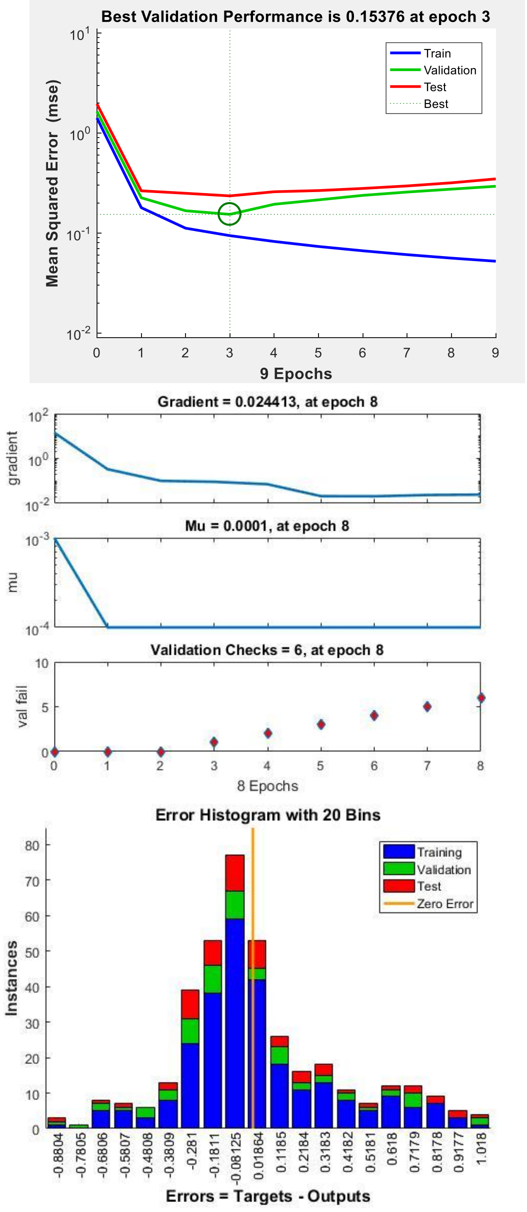

Fig. 3. Backpropagation Optimization Graphs

C. Radial Basis Network

Radial basis network (RBN) works using the concept of Radial Basis Functions (RBF). RBN works by approximating functions using the given data by iterative training. The error on the training data can be reduced to values as small as desired. New neurons can be added until the means square error (MSE) goal is achieved [7]. Thus for a given number of discrete data inputs, a function is made in such a way that it passes through the given inputs as closely as possible.

The function approximation is usually made using the form given in the following equation:

$$y(x) = \sum_{i=1}^{N} w_i \phi(\|x - x_i\|)$$

Where ϕ is a radial basis function which include Gaussian, Multiquadric, Polyharmonic spline, etc., and w_i represents the weights. While implementing as an RBN, ϕ takes the role of activation function. The weights are updated by finding the error of *y(x)*. This is obtained by differentiating *y(x)* with respect to the weights.

The features extracted from PCA is used for training RBN. The training graph obtained is given below in Fig. 4.

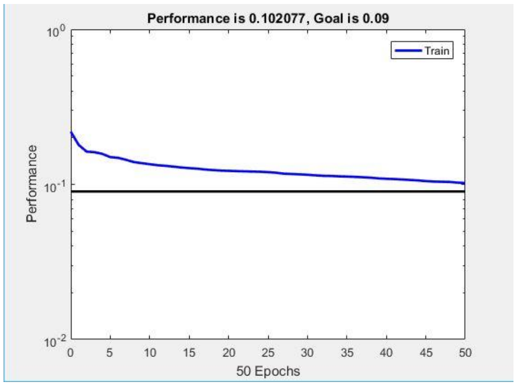

Fig. 4. Error during RBN training.

D. Decision Tree

Decision trees are decision support tools which has a flowchart like structure. The decisions are taken using the relevance of the input features. The relevance is found by calculating the information entropy contained in the features which is often done using the Gini index. The feature with the maximum gain in entropy is considered for splitting the tree.

Training is done by growing the tree based on the entropy of the input data. After the tree has been made, testing is done on new inputs [9]. The optimization graphs are shown below in Fig. 5 and the decision tree in Fig. 6.

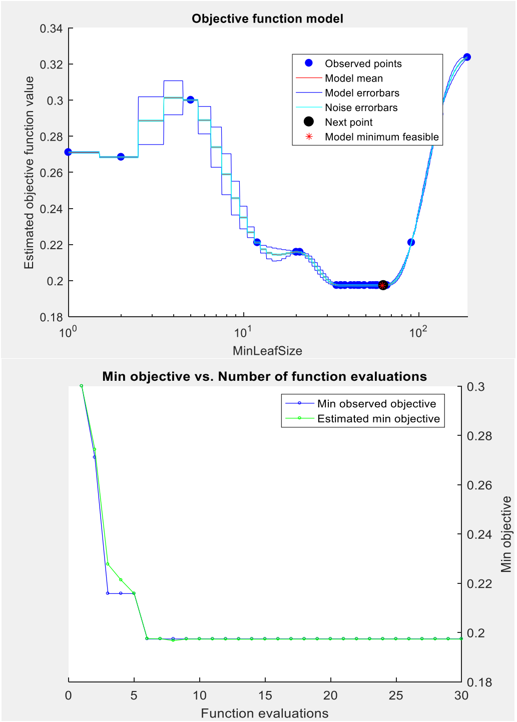

Fig. 5. Optimization graphs for decision tree.

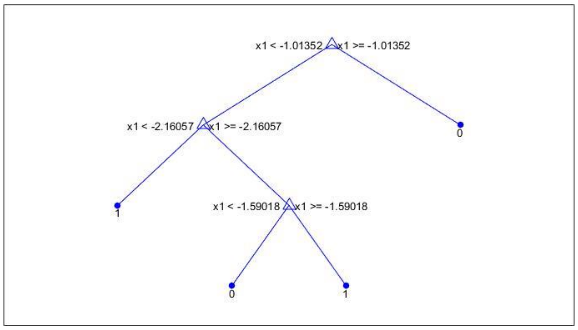

Fig. 6. A sample decision tree grown from the PCA inputs.

IV. Results

This project utilizes real world data from the UCI database. Thus the accuracy obtained was affected by inputs that did not follow a clear pattern. For example, from the input data, it can be seen that some patients suffer from heart disease even if the attributes considered did not clearly show that. Even with these setbacks fairly accurate results could be obtained. Table 1. summarizes the results. The algorithms were trained using the inputs with and without doing pre-processing using PCA. Testing was done using hundred samples of testing data. The testing data was not used for training, which shows that the algorithms used have a high reliability. Thus the methods used could also work with entirely new data.

From the table, we can conclude that Backpropagation has the greatest accuracy. It can also be seen that PCA improves the accuracy compared with the raw input. As already said under pre-processing, this project uses six feature after PCA feature reduction.

TABLE I. ACCURACY OF THE MACHINE LEARNING ALGORITHMS

	KNN	**Back-propagation**	**RBN**	**Decision Tree**
With PCA	85%	87%	85%	80%
Without PCA	75%	68%	69%	73%

V. CONCLUSION

The project goal could be achieved with a relatively high accuracy rate. The proposed machine learning techniques worked well and gave expected results. A faster algorithm can be developed with the existing framework to increase the efficiency. A faster detection and early prediction of Coronary Heart Disease can be achieved since the presence of various wearables (like smart watches and activity trackers) with high computing power and connectivity is becoming more and more prevalent. Thus development of a CHD detection app is a great solution for the future expansion of this project.

ACKNOWLEDGMENT

The authors acknowledge the support and knowledge given by Dr. Simon Foo, who is the professor of Machine Intelligence in the Florida State University. The authors also thank the Department of Electrical and Computer Engineering of the FAMU-FSU College of Engineering for providing the facilities for this project and access to the MatLab software. We also remember with gratitude David Aha and the authors of the UCI database for creating this great resource.

REFERENCES

[1] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. (1988). *UCI Hear Disease Database* [online]. Available FTP: http://mlearn.ics.uci.edu/databases/heart-disease/

[2] Jonathon Shlens, “A Tutorial on Principal Component Analysis”, arXiv preprint arXiv:1404.1100 (2014).

[3] Muhammad Saqlain, Wahid Hussain, Nazar A. Saquib and, Muazzam A. Khan, “Identification of Heart Failure by Using Using Unstructured Data of Cardiac Patients”, 45th International Conference of Parallel Processing Workshops, 2016.

[4] Shweta H. Jambuika, Vipul K. Dabhi, and Harshadkumar B. Prajapati, “Classification of ECG signals using Machine Learning Techniques: A Survey”, ICACEA, 2015.

[5] Mashail Alsalamah, Dr. Saad Amin, and Dr. Jhon Halloran, “Diagnosis of Heart Disease by Using a Radial Basis Function Network Classification Technique on Patients’ Medical Records”, Coventry University, United Kingdom.

[6] C. Kalaiselvi, “Diagnosing of Heart Diseases using Average K-Nearest Neighbor Algorithm of Data Mining”. 3rd International Conference on Computing for Sustainable Global Development, 2016.

[7] MathWorks, Introducing Machine Learning. [PDF]

[8] MathWorks, *Radial Basis Approximation* [online]. Available: https://www.mathworks.com/help/nnet/examples/radial-basis-approximation.html

[9] MathWorks, *Decision Trees* [online]. Available: https://www.mathworks.com/help/stats/classification-trees-and-regression-trees.html

[10] Michael Nielsen (Jan 2016). *How the backpropagation algorithm works* [online]. Available: http://neuralnetworksanddeeplearning.com/chap2.html