Alsalamah et al., Prensa Med Argent 2016, 102:6
DOI:10.4172/lpma.1000235

La Prensa MedicaArgentina

Research Article

Prediction of Coronary Artery Disease Using a Combination of Methods for Training Radial Basis Function Networks

Alsalamah M*, Amin S and Palade V

Abstract

Cardiovascular disease (CAD) is among the most prevalent diseases around the world; nevertheless, its diagnosis requires highly qualified medical staff (e.g., cardiologists) because of the many variables involved in the process. Due to diagnostic complexity and the limited number of available qualified staff, the development of smart systems that could automate the diagnostic process is paramount. This paper investigates two systems in order to achieve this goal. The first system proposes the application of a data fusion with Kalman filtering in diagnosing CAD as well as in the prediction of the need to conduct a Coronary Artery Bypass Graft (CABG) in patients identified as having CAD. The second system, which is based on a combination of Particle Swarm Optimization (PSO) and a Gravitational Search Algorithm (GSA), is also proposed. Patient data was gathered from King Abdullah Medical City in Saudi Arabia, and a statistical analysis was conducted to explore the relationship between an array of variables and CAD. After identifying pertinent variables for diagnosis, some learning algorithms (e.g., Kalman Filtering, Particle Swarm Optimization and Gravitational Search Algorithm) were applied to the collected data sets to train the system for predicting the diseased condition. The main aim of this paper is to identify the underlying functional relationship between the medical patient records and the medical diagnosis in the datasets in order to predict the presence or absence of the disease for new patients. This work takes a novel approach by using different neural networks training algorithms, e.g., Quasi Newton and Scaled Conjugate Gradient (SCG) with several activation functions on an extended Kalman filter.

Keywords

Extended kalman filter; Coronary artery disease; Radial basis function networks; Particle swarm optimization; Gravitational search algorithm

Introduction

Modern lifestyle habits have significantly increased incidents of cardiovascular disease. Qualified staff available for disease diagnosis in this medical area remains limited and, therefore, are under increased pressure. Fortunately, the diagnosis of complex diseases has become much easier due to progress in computing technologies and artificial intelligence using patient information and the manifestation of their symptoms.

Coronary Artery Disease (CAD) is a complex condition of artery blockage with high mortality figures [1]. The prevalence of Coronary Artery Disease (CAD) is increasing across the globe with high costs for governments and other healthcare stakeholders. In addition to financial pressures, CAD frequently results in mortality and is one of the world's most prevalent causes of death.

A host of factors are used in the diagnosis of CAD, including patient blood pressure, cholesterol level, sugar levels, high BMI (overweight/obese), physical inactivity, unhealthy eating and smoking [2]. Other factors, such as age, gender, and family history of heart disease, are also likely risk factors for CAD [3]. With so many factors involved, detection is challenging because it requires identifying and interpreting the symptoms, risk factors and the patient's medical history. This study was based on real patients in the Saudi Arabia population - in King Abdullah Medical City. Variables are known to have a relationship with CAD were considered and data collected.

When a patient shows symptoms of heart disease, several tests must be immediately done by a doctor (most often an experienced cardiologist) to diagnose for coronary artery disease and prescribe the appropriate treatment regime [4]. This process is generally highly laborious and resource intensive, which makes the diagnosis and treatment very expensive. For this reason, amplifying new smart systems, which facilitate this process, is an urgent priority.

This research investigates the application of a Kalman Filtering (KF) for diagnosing coronary artery disease using two training algorithms for the prediction of the need to conduct a Coronary Artery Bypass Graft (CABG) in patients identified as having CAD. A second method used is training Radial Basis Function Networks by using a hybrid of Particle Swarm Optimization (PSO) and a Gravitational Search Algorithm (GSA) to solve the CAD prediction problems. Here, the GSA and PSO algorithms are employed as new training methods for a Radial Basis Function Network in order to investigate the efficiency of these algorithms. The derivation of the Kalman filter is involved in the data fusion algorithm, which simplifies the recursive calculation of the CAD status and the Coronary Artery Bypass Graft (CABG) requirement (i.e., the two factors of interest). This process uses a combination of knowledge/observations/measurements from patients, predictions from models and considers the inherent noise in the observations/measurements. Non-linear measurements were also involved in the prediction process; as a result, an extended variant of the KF (Extended Kalman Filter – EKF) was also applied in the research. The strength of the EKF is its ability to implement non-linear models [5], making it an ideal candidate for neural network training.

Most applications of EKF training for neural networks have been for time-series predictions [6,7]. Time-series constraints on the data can be eliminated by using a Radial Basis Function (RBF) neural network architecture designed for classification. Our approach shows the use of EKF with various training algorithms used to train Radial Basis Function Neural Networks for CAD prediction.

Studies show that Coronary Artery Disease (CAD) is increasing across the globe, requiring excessive resources to be used by healthcare

***Corresponding author:** Mashail Alsalamah, Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry CV1 5FB, United Kingdom, E-mail: alsalam2@coventry.ac.uk

Received: October 18, 2016 **Accepted:** November 07, 2016 **Published:** November 14, 2016

stakeholders in trying to manage the disease [1,8-17]. This research could help develop for more efficient management strategies for the disease.

The paper is structured as follows. The literature review describes the previous work done in the detection of CAD. The use of Kalman Filtering for this is presented in Section 2, followed by a discussion of the methodology and the principles of this novel approach of using the Extended Kalman filtering, Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA) for radial basis functions training. The next section presents the procedure for testing and verification of the patient dataset. Section 4 presents the results and the analysis of the tests conducted using several combinations of different training algorithms. The final section is a summary of the work and provides some recommendations for future work.

Background

The correct diagnosis of Coronary Artery Disease (CAD) depends, along with other factors, on a patient's blood pressure, cholesterol and sugar levels. Being overweight, physical inactivity, unhealthy eating and smoking tobacco, are all risk factors for CAD. A family history of heart disease also increases the risk for developing CAD. If a patient is at high risk for heart disease or already has symptoms, a doctor can use several tests to diagnose CAD. However, this process is always complicated, expensive and requires expertise.

CAD is caused by the build-up of plaque in the walls of the arteries that supply blood to the heart (coronary arteries) and other areas of the thoracic region of the body. This plaque consists of cholesterol and other substances that are deposited in the arterial wall [2]. The build-up of plaque inside the arteries results in stenosis or a narrowing of the arteries over time, which could ultimately cause partial or total blockage of blood flow – a condition known as atherosclerosis. Atherosclerosis makes it harder for blood to flow and, when the heart muscle receives less than enough blood to function, there is resultant chest pain or discomfort around the thoracic region called Angina Pectoris – a common symptom of CAD. Over time, CAD weakens the heart muscle and can lead to heart failure; an acute condition where the heart is unable to pump blood the way that it should. An initial sign of this is irregular heartbeat/heart rhythm called arrhythmia [2,3].

To improve diagnostic efficiency, automatic Computer-Assisted Detection tools have been applied to the diagnosis process in recent years. Linear and logistic regression models are frequently used [8-12]. Other commonly used predictive models are the Linear Discriminant Analysis, K-nearest Neighbour Classifier, Artificial Neural Network and the Support Vector Machine [13-16]. These models have, to some extent, been shown to have good predictive value. For example, Mandal showed that the prediction accuracy of training and test sets of Linear Discriminant Analysis could be as high as 90.6% and 72.7%, respectively. While Heydari showed that Artificial Neural Networks can produce accuracy as high as 81.2% on a test set. In spite of these reasonable results, there are limitations in most learning algorithms. For example, the Linear Regression and the Linear Discriminant Analysis are both linear techniques that cannot be extended to non-linear modalities (variables) which are requisite for a proper diagnosis of CAD. As a way of circumventing this issue, researchers started to apply more complex models - such as the combination of a Support Vector Machine with a Radial Basis Function (RBF) Kernel, Support Vector Machines optimized by particle swarm optimization or other forms of integration of two individual approaches to generate better non-linear techniques [16,17-19]. These combinations improved the prediction accuracy on the training and the test sets to as high as 96.9% [13].

Despite this improvement in the performance of the newer and more complex models, the cross-validated prediction accuracy of these models still needs to be improved, with [16] reporting a cross-validated prediction accuracy using a Support Vector Machine and RBF hybrid non-linear technique that could only get as high as 92.67%. There is room for improvement. One important way of improving the process, which is yet to be well exploited, is to enhance the quality of the data sets used for training.

Measures of an algorithms' performance, such as validation error, are affected by variations in the data. A potential candidate for improving data quality is the Matrix Completion, a process that adds to the number of entries to the data that contains some unknown/missing values. Research indicates the Matrix Completion could greatly enhance the accuracy of prediction [20-22].

Furthermore, application of the hybrid Extended Kalman Filter (EKF) in the diagnosis of CAD shows potential for enhancing accuracy. The Kalman Filter (KF) is a well-established estimation theory that has been in existence since the 1960s. Though initially designed to provide recursive solutions through linear optimal filtering for estimating desired parameters, the extended version of the filter (i.e., Extended Kalman Filter – EKF) has the capability to handle non-linear systems/conditions [23]. This learning algorithm has been used in diverse research realms, and has shown excellent results in terms of prediction accuracy [24,25]. Nonetheless, the EKF's alluring capabilities are yet to be explored in the area of Coronary Artery Disease prediction. This is most likely due to the lack of awareness about its existence, as the majority of researchers in this realm of research and beyond are more familiar with the KF, which can only provide a recursive solution through linear means [26,27].

Methodology

RBF Neural Network models are a suitable method for classification problems, such as in the detection of diseases (e.g., occurrence of CAD). In order to address classification problems, several steps are required. This section provides a brief on the methodology steps we followed, as shown in Figure 1.

Statistical analysis

Before being applied to the RBF network, the data set has to be pre-processed to identify a relationship between existence/non-existence of CAD based on many variables. The current study was based on a sample of the Saudi Arabian population, in King Abdullah Medical City, with the objective to identify the relationship between CAD and other variables; namely demographic variables such as age, gender, occupation, physical variables (height, weight), smoking habits and medical history among others.

Data analysis: There were 60 variables and 688 observations in the data set collected. This was initially analyzed by means of frequency distributions and graphs in order to understand the general nature of the data and to determine the optimal statistical model to test the hypotheses. Logistic regression was the main analysis used to test the hypotheses. As CAD would be the target variable (dependent variable) of the logistic regression model, the observations with missing CAD status data were removed from further analysis. CAD was measured on a dichotomous scale, with the two categories being mutually exclusive - satisfying the prior assumptions of logistic regression.

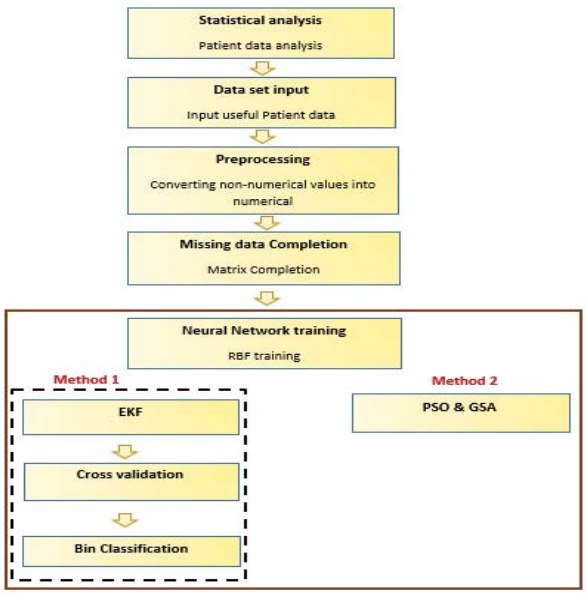

Figure 1: Methodology overview.

There were 59 independent variables in the data. The frequency distributions of each of them were studied to discover if further modifications would be required to fit the model, and to eliminate data entry errors. Mismatched entries were found, and were considered as "no information" and tagged as 0 in required cases. The distribution was made of independent categorical variables, after cleaning up the data. Among the 687 observations, 402 came from males and 117 had a history of stroke. 124 people were smokers; whereas, 197 were previous smokers.

There were a few continuous variables in the data. Their range, central tendency and dispersion were studied to ensure proper generation of those variables. These variables were be used directly with the logistic model, as they are sufficient for the elementary assumptions of a logistic regression model. However, the model results would only give directional overview – such as "if Blood Urea Nitrogen (BUN) increases, the chances of CAD also increases". On the other hand, if they could be transformed into categorical variables, the model results would provide strategic overview, for example: "People having BUN within 60-80 have more chances to have CAD".

Considering that most of these continuous variables are very important in the medical context; they were transformed into "to be applicable, no medical variable can have a value 0." Hence, all 0 values were considered as "No Information". Following is a brief description of how these variables were transformed into categorical variables, after the classification of the distribution of medical variables had been made (Table 1).

Statistical methodology: The data was prepared for the model and the Information Value (IV) of each variable was calculated, which helped to eliminate variables from the model. IV is a measure equivalent to correlation analysis. But, unlike correlation, it works for only categorical variables. IV indicates the predictive power of the variable.

The second test required for variable elimination is checking multi co-linearity using the Variance Inflation Factor (VIF). If the value of the VIF for any variable is higher than 3, the variable is likely to be correlated with any of the other variables, and will have adverse impact on the model results. The original dataset was used for this operation to extract the affected attributes (Tables 2 and 3).

The variables highlighted (in bold font) will not be used in the final model. The Compute BMI and the BMI group variables are correlated (r=0.9) and removing any one of them will help. Similarly, Systolic HTN and Diastolic HTN are correlated (R =0.961). Hence, we can keep any one of them (Table 4).

The next step was to build the logistic regression model, with CAD as the target variable.

The overall model concordance us 84.7%, which indicates that the model predicted 84.7% of observations correctly and is statistically good. Any concordance value >60% is considered good.

The pseudo R-square value of the model is 0.559 which is moderate. The higher the pseudo R-sq is the better the model, with R-square ranging from 0 to 1 (Table 5).

B is the coefficient of the variable. SE is the standard error of the variable. Wald is the chi-sq value that determines the significance of the variable - a higher chi-sq means a more significant variable. The df is the degrees of freedom of that variable. Sig. is the p-value - the lower the p- value, the higher the significance. EXP (B) is the impact of the variable on the target. The variables that have p-values < 0.1 are statistically significant at the 10% level of significance. At the end of the analysis and under the consultant super vision by Dr Osma from King Abdullah Medical City, 21 effected attributes have been highlighted and assigned in the diagnosis of CAD (Table 6).

Summary of analysis results

The variables that significantly impacted CAD diagnosis are (Table 7):

- Amlodipine: those who have taken Amlodipine have a higher risk of developing CAD
- Enoxaparinclexame: those who have taken Enoxaparinclexame have a risk of developing CAD.
- HF: those who have reported 'yes' to HF have a higher risk of developing CAD.
- Rosuvastatin: those who have taken Rosuvastatin have a higher risk of developing CAD.
- Smoking: smokers who have taken Rosuvastatin have a higher risk of developing CAD.
- Stroke: people who have a history of stroke have a higher risk of developing CAD.
- Age: people between 26-40 years are in the low-risk zone of CAD
- Weight: weight overall is a significant factor associated with CAD, but no particular age-group has been identified as more/less risk prone.
- BMI: people who have perfect weight are at much lesser risk than underweight or overweight people.
- HDL: unlike overall cholesterol level and bad cholesterol LDL, good cholesterol HDL is a significant factor of CAD. People having lower HDL, <59 mg have a higher risk of developing CAD.
- FBS: whoever has above normal FBS are in a high risk zone of

Table 1: Distribution of independent variables (Continuous).

Variable	Mean	Median	Min	Max	Standard deviation
BGC	79.87	0	0	999	122.54
BUN	10.23	0	0	999	39.62
CH	133.60	135	0	340	60.60
ComputeBMI	28.28	27.72	0	114.06	9.93
FBS	37.14	0	0	430	67.35
HB	8.41	11.5	0	26.4	6.57
HDL	35.61	35	0	346	21.35
Hight	159.74	163	0	999	54.94
LDL	87.16	89	0	552	47.58
PPBS	38.84	0	0	1174	97.10
RBC	3.22	4.3	0	85	4.51
TG	121.11	112	0	722	78.31
WBC	5.10	5.6	0	96.5	5.59
Weight	76.05	74	0	999	43.03
Age	51.70	53	0	999	40.08
diastolicHTN	53.09	76	0	114	38.80
systoilcHTN	86.69	120	0	200	63.34
timeofexcercise	14.23	0	0	600	42.56

Table 2: IV Values.

Information Value	Predictive Power
< 0.02	useless for prediction
0.02 to 0.1	Weak predictor
0.1 to 0.3	Medium predictor
0.3 to 0.5	Strong predictor
>0.5	Suspicious or too good to be true

Note: The variables with low IV will not be used in the model

developing CAD.

- PPBS: people having a slightly higher measurement on PPBS are at higher risk of developing CAD than those who have very high or normal measurements
- BUN: BUN overall is a cause of CAD, but no significant measurement group is identified as being at high risk.

Data set classification

There were three datasets used in this research. However, the main application was the CAD dataset, collected from King Abdullah Medical City. Below, a brief description for each dataset is provided:

- The King Abdullah Medical City hospital data is obtained from the history of different patients, which came from the evaluation of CAD related diseases.
- In order to evaluate the system with different data, we used two other kinds of data set, which are:
- Cancer dataset: from The Neural Network Toolbox in MATLAB. It has 8 input attributes and 2 outputs and 1000 records in the dataset.
- Simple class dataset: from the Neural Network Toolbox in MATLAB. It has 2 inputs attributes and 4 outputs and 1000 records.

Pre-processing

Data requires pre-processing before being fed to the RBF network for CAD detection. Two steps were taken: digitization and missing data completion.

Digitization: Most of the data were non-numerical. Furthermore, the data contained values with large statistical anomalies. Also, it was necessary to convert the data with non-numerical format into proper numbers, so it could be used for the training and testing of the neural network.

The following steps were taken:

- Encoding all the input data;
- Logical values: (Yes) or (No) were encoded to 1 or 0 respectively;
- Values with multiple possibilities were encoded by scaled numbers in the range of 1 to 9;
- Values which were already numbers remained so,
- Encoding output data so as to enlist all possible outputs. These were then converted to multiple single digit binary outputs, as shown in block 1 in Figure 2.

Matrix completion for missing data: Due to several entry deficiencies in recording, interview or manual entry, patient data can contain some anomalies - in certain cases missing several or one of the major contributing fields (e.g., high level of missing data, which is inherent in the sets of data often used in CAD diagnosis/prediction as these data sets come from multiple sources, e.g., oral interviews, doctors' examinations and technical measurements with different instruments).

Consequently, the estimation process can be disturbed. To avoid this, it would seem the patient's information should be discarded. However, patient information is generally confidential and cannot be discarded by the source. In this case, the use of Matrix Completion technique seems very useful to improve prediction efficiency. Although this has not provided a high percentage of accuracy, it has helped to use patient information with less missing field numbers. Thus, higher confidence in the Matrix Completion results was considered in the study.

In this work, the Exact Matrix Completion via Convex Optimization method was selected to improve the data quality and availability, based on its ability to evaluate the sparse matrices using the convex optimization problem.

Training the RBF neural network

RBF networks, because of their classification capability, are a good candidate for training with non-linear data. An RBF network consists

Table 3: VIF of the variables.

Coefficients[a]			
Model		**Collinearity Statistics**	
		Tolerance	**VIF**
	Smoking	.563	1.777
	age	.761	1.313
	gender	.619	1.614
	Weight	.415	2.410
	ComputeBMI	.180	5.545
	BMIGroup	**.166**	**6.026**
	Hight	.523	1.914
	typeofsmoking	.476	2.102
	dursmoking	.763	1.310
	Squitting	.479	2.087
	NOciggateD	.587	1.704
	Exercise	.607	1.648
	timeofexcercise	.656	1.524
	000000000000	.500	2.000
	systoilcHTN	**.051**	**19.633**
	diastolicHTN	**.054**	**18.460**
	DM	.449	2.229
	BGC	.542	1.844
	PDM	.523	1.911
	PCVD	.553	1.808
	ObeseR	.759	1.318
	Anemia	.757	1.321
	Stroke	.733	1.365
	HF	.917	1.090
	Amlodipine	.911	1.097
	Enoxaparinclexame	.700	1.429
	Asprin	.353	2.834
	Atrovastatin	.586	1.706
	Cerivastatin	.476	2.100
	Fluvastatin	**.014**	**72.731**
	Pitavastatin	.871	1.147
	Pravastatin	**.008**	**126.381**
	Rosuvastatin	**.049**	**20.235**
	Clopidogreal	**.175**	**5.699**
	Pantoprazole	**.190**	**5.251**
	Nitroglycerin	**.037**	**26.764**
	perindoprilarginine	**.214**	**4.675**
	Angiography	**.238**	**4.210**
	LDL	.529	1.890
	CH	.403	2.482
	HDL	.704	1.421
	TG	.733	1.365
	FBS	.731	1.368
	PPBS	.712	1.405
	WBC	.314	3.181
	HB	.287	3.490
	RBC	.723	1.383
	BUN	.932	1.073
	Albumin	.939	1.065
a. Dependent Variable: CAD			

of three layers: namely the input layer, the hidden layer and the output layer. The input layer broadcasts the coordinates of the input vector to each of the units in the hidden layer. Each unit in the hidden layer then produces an activation based on the associated radial basis function. Finally, each unit in the output layer computes a linear combination of the activations of the hidden units. How a RBF network reacts to a given input stimulus is completely determined by the activation functions associated with the hidden units and the weights associated with the links between the hidden layer and the output layer. In the RBF networks constructed with the proposed learning algorithm, each activation function associated with the hidden unit was built either using TPS or R4RlogR as an activation function.

Table 4: Concordance values.

Observed		**Predicted**		
		CAD		**Percentage Correct**
		0	**1**	
CAD	0	449	39	92.0
	1	66	133	66.8
Overall Percentage				84.7

Table 5: Model Summary.

Model Summary			
Step	**-2 Log likelihood**	**Cox & Snell R Square**	**Nagelkerke R Square**
1	485.719*	.391	.559

Note: The pseudo R-square value that determines the goodness of fit of the logistic model

Table 6: Goodness of fit – Hosmer-Lemeshow.

Hosmer and Lemeshow Test			
Step	**Chi-square**	**df**	**Sig.**
1	4.348	8	.824

Note: This is another test of goodness of fit. The bigger the value of Sig. (Significance), the better the model is. 0< significance <1.

Table 7: Top 10 other diagnosis outcomes when CAD is present.

Other Diagnosis	**Count**
MI	140
unstable angina	27
essential primary hypertension	9
atrial fibrillation	10
Cardiomyopathy	33
Angina	22
angina pectoris	13
Arrhythmia	40
Chest pain	43
dilated cardiomyopathy	65

In this work, two methods were used for training. The first method used the Extended Kalman Filter for the learning procedure, and it used different training algorithms, such as Quasi Newton and Scaled Conjugate Gradient (SCG). The second prediction method is PSOGSA - the Gravitational Search Algorithm (GSA) - which is a novel heuristic optimization method based on the law of gravity and mass interactions.

Method 1: Extended kalman filtering

The Kalman Filter gain is a time-varying gain matrix. The matrices used were:

- Auto-covariance matrix (for lag zero) of the estimation error of the corrected estimate:
- Auto-covariance matrix (for lag zero) of the estimation error of the predicted estimate:
- The transition matrix A of a linearized model of the original nonlinear model was calculated with the most recent state

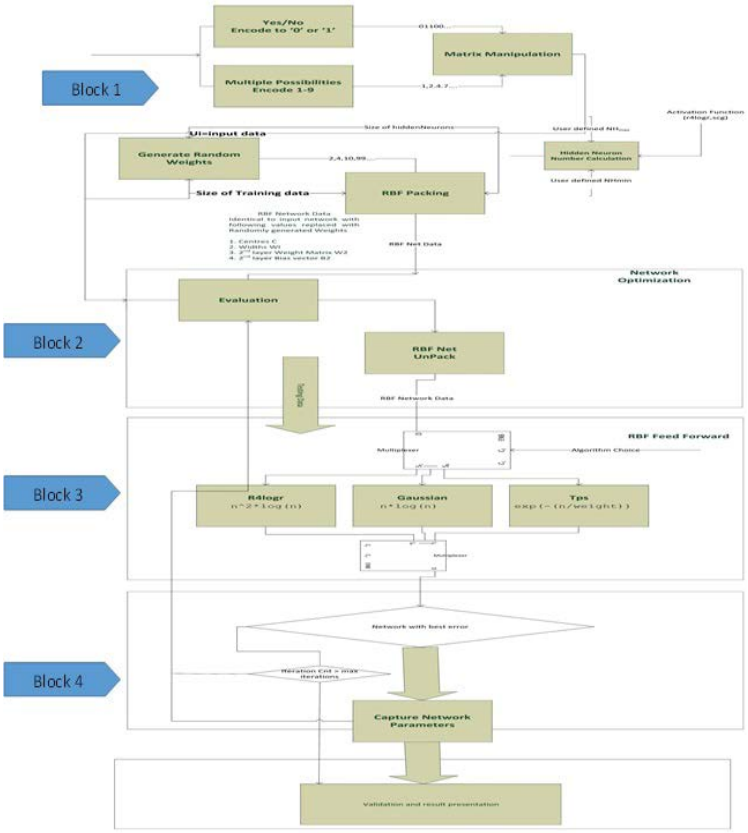

Figure 2: Flow of the RBF network buildup to obtain the CAD diagnosis result.

estimate, which was assumed to be the corrected estimate xc (k):

- Given the continuous-time nonlinear process model. Linearize it at the operating point to obtain A.
- Then calculate A = A discrete as the discretized version of A continuous. Forward method of discretization in manual calculations can be used; however in this case MATLAB is used to discretize this function.

EKF Implementation for current work: The first step was the derivation of the equations on which EKF neural networks training algorithm are based. A neural networks (NN) can be described as a non-linear discretized system w_{K+1}.

$$w_{k+1} = w_k + \omega_k$$

Where w_K is the weight vector? And w_{K+1} computation: for k=1,2…, compute state estimate prpogation

The second equation, known as the observation or measurement equation, represents the network's desired response vector y_k as a nonlinear function

$$y_k = h_k(w_k, u_k, v_{k-1}) + v_k$$

Where h_k is the derivative matrix, w_K is the weight vector; u_k is an input training pattern and v_{k-1} is the recurrent node activations v_k from the previous time step for GRBFs.

The input vector u_k, the weight parameter vector w_k, and, for recurrent networks, the recurrent node activations v_k; this equation was augmented by random measurement noise n_k.

The noise (measured) n_k is given as:

$$E[v_k v_l^T] = \delta_{k,l} R_k$$

Where R_k is the covariance noise matrix

Similarly the process noise v_k is given as:

$$E[\omega_k \omega_l^T] = \delta_{k,l} Q_k$$

Where Q_k is the covariance matrix of the process noise

The training problem using Kalman filter theory can now be described as finding the minimum mean-squared error estimate of the state 'w' using all observed data so far. We assumed network architecture with M weights and no output nodes and cost function components.

The EKF solution to the training problem is given by the following recursion:

$$A_k = [R_k + H_k^T P_k H_k]^{-1}$$

$$K_k = P_k H_k A_k$$

$$\overline{W}_{k+1} = \overline{W} + K_k \varepsilon_k$$

$$\varepsilon_k = \overline{y}_k + y_k$$

The vector w_k represents the estimate of the state (i.e., weights) of the system at update step k. This estimate is a function of the Kalman gain matrix K_k and the error vector.

$$\varepsilon_k = \overline{y}_k + y_k$$

Where Y_k is the target vector y_k and is the network's output vector for the kth presentation of a training pattern.

The Kalman gain matrix is a function of the approximate error covariance matrix P_k, a matrix of derivatives of the network's outputs with respect to all trainable weight parameters H_k, and a global scaling matrix A_k. The matrix H_k may be computed via static backpropagation or backpropagation through time for feedforward and recurrent networks, respectively.

The scaling matrix A_k is a function of the measurement noise covariance matrix R_x, as well as of the matrices H_k and P_k. Finally, the approximate error covariance matrix P_k evolves recursively with the weight vector estimate; this matrix encodes second derivative information about the training problem, and is augmented by the covariance matrix of the process noise Q_k.

This algorithm attempts to find weight values that minimize the sum of squared error.

$$\sum_k \varepsilon_k^T \varepsilon_k$$

Note that the algorithm requires that the measurement and process noise covariance matrices, R_k and Q_k be specified for all training instances. Similarly, the approximate error covariance matrix P_k must be initialized at the beginning of training.

Generalized Extend Kalman Filter (GEKF) training was carried out in a sequential fashion as shown in the signal flow diagram below.

We describe one training step as follows:

- An input training pattern u_k was propagated through the network to produce an output vectory y_k. Note that the forward propagation is a function of the recurrent node activations v_k from the previous time step for GRBFs. The error vector j_k is computed in this step as well.
- The derivative matrix H_k was obtained by back propagation. In this case, there was a separate backpropagation for each component of the output vector y_k and the backpropagation phase involved a time history of recurrent node activations for GRBFs.
- The Kalman gain matrix was computed as a function of the derivative matrix H_k, the approximate error covariance matrix P_k, and the measurement covariance noise matrix R_k. Note that this step included the computation of the global scaling matrix A_k.
- The network weight vector was updated using the Kalman gain matrix K_k, the error vector J_k, and the current values of the weight vector W_k.
- The approximate error covariance matrix was updated using the Kalman gain matrix K_k, the derivative matrix H_k, and the current values of the approximate error covariance matrix P_k. Although not shown, this step also included the augmentation of the error covariance matrix by the covariance matrix of the process noise Q_k, the procedures shown in block 2, 3 and 4 in Figure 2.

Calculation of kalman filter gain: The Kalman filter gain K_k was calculated as follows and is shown in block 2, 3 and 4 in Figure 2:

- The initial step, and the operations here were executed only once. The initial value was set to some guessed value (matrix), e.g., to the identity matrix (of proper dimension)
- Calculation of the Kalman Gain
- Calculation of auto-covariance of corrected state estimate error
- Auto-covariance of corrected state estimate error
- Calculation of auto-covariance of the next time step of predicted state estimate error
- Auto-covariance of predicted state estimate error.

Quasi-newton: The Quasi-Newton method is an extension of the Newton optimization algorithm. Unlike the Newton method, which calculates the Hessen of the function (which is complex, resource heavy and approximation is used between the steps), Quasi-Newton quickly optimizes and is simpler to implement, calculating the minima of the function iteratively. The training of the network is based on different options; these options are adjustable once the network evolves to its optimized set of weights and centres. Quasi-newton not only uses the function but its gradient to find the minima of the function.

One of the optimization goals is to achieve the required performance (in our case, the MSE) then the training shall stop. However, in certain cases, because of the large amount of variance in the input, achieving this goal is not possible; therefore, there shall be a way to stop the algorithm at a reasonable time. The condition on which the training must be stopped is any one or all of the following:

- The maximum number of set repetitions is reached.
- The maximum amount of time is exceeded.
- Performance is minimized to the goal.
- The performance gradient falls below minimum gradient.
- Validation performance has increased more than maximum fail times since the last time it decreased (when using validation) and shown in block 3 in Figure 2.

Scaled Conjugate Gradient (SCG): SCG is a supervised learning algorithm for feedforward neural networks, which is a member of the class of conjugate gradient methods.

The SCG is one of the conjugate gradient methods; however it adapts the search direction and step size more carefully, determined by the second order approximation. The other three conjugate gradient algorithms require a line search for the every iteration, which is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. The SCG training algorithm was developed to avoid this time-consuming

line search, thus significantly reducing the number of computations performed in each iteration, although it may require more iterations to converge than the other conjugate gradient algorithms. The storage requirements for the SCG algorithm are about the same as those of CGF and shown in block 3 in Figure 2.

Procedure: The main theme of this classification solution is to optimize the network and then validate the results. This is mainly divided into three steps: i.e. train, test and validate.

The available data will be divided into training and testing data. This way several models trained on the training set will be available to be applied on the test set. The best result on the test set based on these several trained models is then considered as the optimal simulation. This procedure is also detailed in Figure 2.

Cross validation: The discussed above can introduce a bias towards a particular data set. Therefore, the data set can be portioned and swapped as testing and training sets to negate this bias and also to calculate an average based on these different partitions.

The data set in this system has been divided into three sections: training, testing and validation as following:

- Training set is used to build the model. This contains a set of data, the pre-classified target and predictor variables.
- Testing set is used to evaluate how well the model does with data outside the training set. The test set contains the pre-classified results data but they are not used when the test set data is run through the model till the end, when the pre-classified data are compared against the model results. The model is adjusted to minimize the error on the training set.
- Validation set is used to evaluate the adjusted model in step 2, where again, the validation set data is run against the adjusted model and results compared to unseen pre-classified data.

One of the optimization goal is to achieve the required performance (in our case is the MSE) then the training shall stop. However in certain cases because of the large amount of variance in the input this is not possible to achieve this goal, then there shall be a way to stop the algorithm at a reasonable time. Therefore, the conditions on which the training must stop can be one of those cases; the performance is minimized to the goal. Or Validation performance has increased more than maximum fail times since the last time it decreased (when using validation).

Bin classification: Bin classification is the method used to evaluate the correctness of the algorithm. Since this is a classification problem, hence the regression analysis is not too suitable for this work. The error percentage is calculated based on the expected results and the actual results. The expected results, as explained before, are obtained from the current calculations of the feedforward RBF network.

The calculated values from the network have similar rows and columns as the actual output target data. The requirement for bin classification and error calculation is that there shall be a minimum of two columns of expected results.

Method 2: Particle Swarm Optimization Gravitational Search Algorithm (PSOGSA)

In Feed forward Neural Networks, the minimum error can be found by the best combination of connection weights and biases during the learning process. However, most of the time, the feed forward networks converge to the local minimum, and not to the global minimum, thus learning algorithms lead the feed forward networks towards local minima and not global minima. There have been several training algorithms used for FNNs. In addition, there have been several heuristic algorithms used to train FNNs, which include SA (Simulated Annealing), GA (Genetic Algorithms), and Particle swarm optimization (PSO), Magnetic Optimization Algorithms etc. The SA and GA attempt to achieve the global minimization, but their convergence rate is very slow. However Mirjalili suggested using a hybrid combination of a Particle Swarm Optimization and Gravitational Search Algorithm rather than the FNNs. The basic idea of PSOGSA is to combine the ability for obtaining the global best in PSO with the local search capability of GSA, potentially, a very good candidate for FNN training. Three different experiments were set to test and compare the PSOGSA with simple PSO and GSA algorithms with varying numbers of hidden nodes (3 to 7 in one case and then up to 30 for another experiment). These experiments included a parity check, iris detection and a function approximation problem. The outcome of the experiments clearly indicated that the MSE (mean square error) of PSOGSA is better than using PSO or GSA separately. This proved that PSOGSA can resolve the problem of local minima and also enhance the convergence speed. Thus, in this research the PSOGSA was used to train RBF Networks.

In PSOGSA, an initial population was generated and randomly initialized as in the case of all GAs. This was done to evaluate whether each agent could be considered as a best solution. After initialization, the gravitational force, gravitational constant, and resultant forces among agents were calculated using equations given respectively. After that, the accelerations of particles were defined, and in the each iteration, the best solution so far was updated. After calculating the accelerations and updating the best solution, the velocities of all agents were calculated. Finally, the positions of agents were updated. The process of updating velocities and positions was stopped when meeting an end criterion. The basic idea of PSOGSA is to combine the ability for social thinking (gbest) in PSO with the local search capability of GSA, as shown in Figures 3 and 4.

Implementation of PSOGSA

1) Initial Parameters for PSO to set up the classifier:

- Number of particles
- Maximum number of iterations.
- Inertia weight.
- Max inertia weight.
- Min inertia weight.

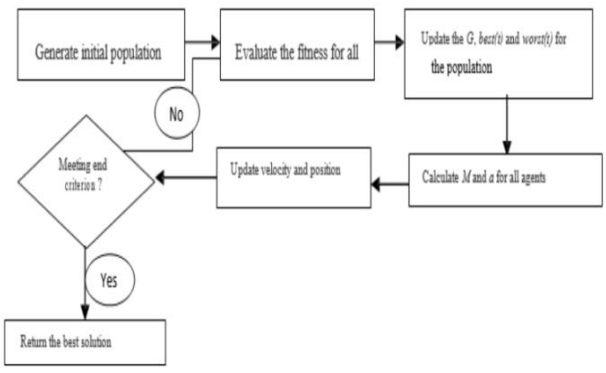

Figure 3: General steps of the gravitational search algorithm.

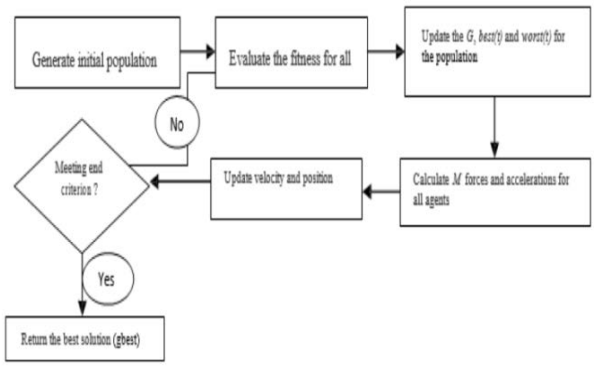

Figure 4: Steps of PSOGSA.

- Velocity vector.
- Position vector.
- Convergence vector.

2) Initialize gBestScore:

3) Calculate MSE:

- Calculate value using RBFN.
- Update Fitness value.
- Update gbest and pbest.

4) Update the w (weight) of PSO
5) Update velocity of particles
6) Update position of particles

Results and Analysis

Each data set was used for the first method, with Extended Kalman Filtering, and having three different combinations: by swapping the training, testing and validation sets. In this experiment, the two different training algorithms (Quasi-Newton & SCG) were used with the different combinations of the data sets to evaluate the performance of the training algorithms. The accuracy for the three datasets was good. The prediction probability of this combination resulted in an accuracy of about 92%, using cross validation for the CAD dataset and above 95% in the two other datasets using cross validation. The second method that used PSOGSA has provided above 83% accuracy with the cancer dataset and about 92% with the CAD dataset using cross validation with varying the number of hidden nodes.

Cancer data set

Figure 5 provides the results on the Cancer Data set tested on the similar combinations of training algorithms and activation functions. There are slight differences in accuracy by using different training and activation algorithms. This can be summarized as follows:

The dataset has been divided randomly into three sub-datasets, and each sub-dataset divided into training, testing and validation for 100 iterations.

Table 8 shows the details of Figure 5, including training errors, testing errors, validation error and the best iteration in each training algorithm in the command window for each set:

The following observations were made:

- The accuracy of the combination Quasi New and R4R for the three sub datasets (cross validation) was: 97.4, 97.4 and 96.4.
- The accuracy of the combination Quasi New and TPS for the three sub datasets (cross validation) was: 94.4, 99.4 and 98.4.
- The accuracy of the combination SCG and R4R for the three sub datasets (cross validation) was: 95.4, 93. 9 and 85.4.
- The accuracy of the combination SCG and TPS for the three sub datasets (cross validation) was: 96.9, 97.4 and 95.9.
- The combination of Quasi-newton and TPS produced the best and most consistent set of training and activation algorithms; even with a change in data sets, the validation error did not increase from 5.53%.
- The best testing performance of Data set 2 is likely due to fewer missing and erroneous data, hence keeping the test errors at a maximum of 4.00% for all the combinations. With a Quasi-Newton training algorithm, the testing error is limited to 1.00%.
- The SCG and the R4logR combination produced the highest number of iterations or processing time with the best result. The combination was computationally slow to converge; however, if we look at the validation results, this was second best out of the four combinations used.
- The training error is still relatively low for all of the combinations; however the Quasi-newton and R4logR combination provides the best in terms of training error.

* We can reference the differences between the validation errors from each dataset (CAD, Cancer and simple class to the nature of the data and the number of the input attributes.

Simple data set

Figure 6 provides the results of the Simple Data set taken from the MATLAB Neural network and tested on the similar combinations of Training algorithms and activation functions for reference to the hospital data set results. The difference is that this is more of a linear

The following observations were made:

- The accuracy of the combination Quasi New and R4R for the three sub datasets (cross validation) was: 99.6, 97.2 and 98.2, sequentially
- The accuracy of the combination Quasi New and TPS for the three sub datasets (cross validation) was: 97.8, 98.2 and 98.4, sequentially.
- The accuracy of the combination SCG and R4R for the three sub datasets (cross validation) was: 91.4, 98.8 and 99.4, sequentially.
- The accuracy of the combination SCG and TPS for the three sub datasets (cross validation) was: 96, 98.8 and 98.8, sequentially.
- The combination of Quasi New and R4R produce the best and most consistent set of training and activation algorithms, with a validation error 0.0%4.
- The best testing performance is of the Data set, which is likely due to fewer missing and erroneous data; hence, keeping the test errors at only 2.09% for all the combinations.
- The SCG and the R4Rlog combination produced the highest number of iterations or processing time, with the best result. The combination was computationally slow to converge; however, if we look at the validation results this is second best out of the four combinations used.
- Overall the combination of Quasi New and R4R performed the best and took less iteration to converge.

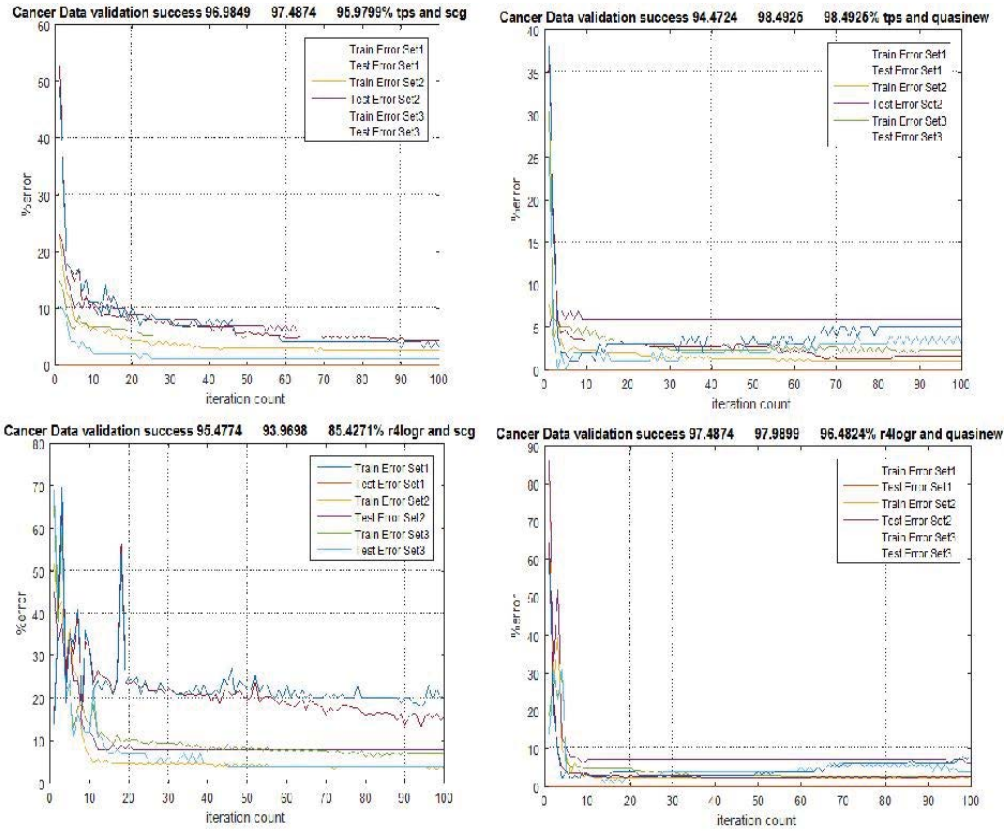

Figure 5: Cancer Data Set with three Combinations of data sets and 4 different training and activation functions.

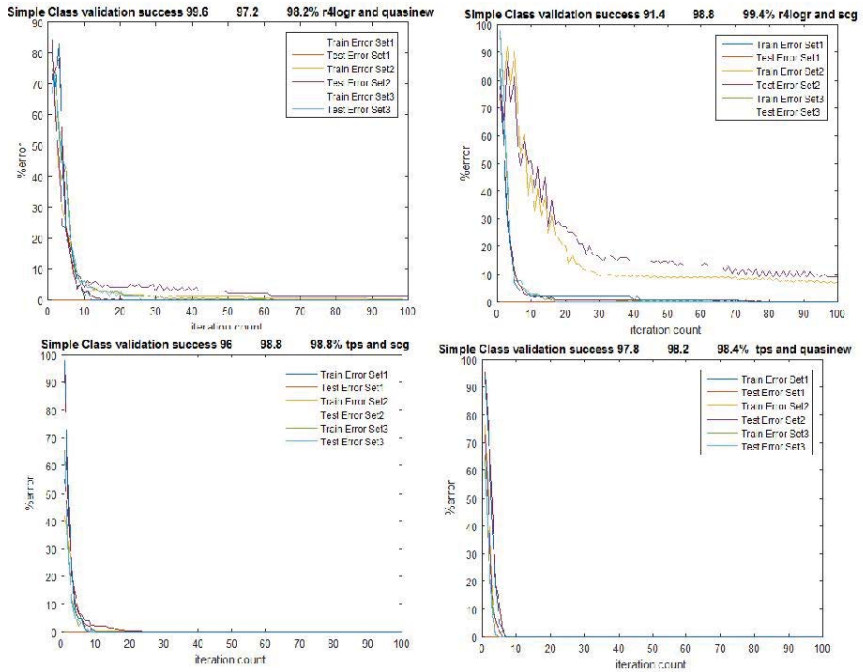

Figure 6: Simple Data Set with three Combinations of data and 4 different training and activation functions.

With matrix completion

The detailed results of the four different combinations are given in Figure 7 and Table 9, including training errors, testing errors, validation error and the best iteration in each training algorithm for each of the three sets after Matrix completion.

The following observations were made:

- The accuracy of the combination Quasi New and R4R for the three sub datasets (cross validation) was: 93.0, 91.4 and 91.9, sequentially.
- The accuracy of the combination Quasi New and TPS for the three sub datasets (cross validation) was: 93.0, 92.5 and 91.9, sequentially.
- The accuracy of the combination SCG and R4R for the three sub datasets (cross validation) was: 91.9, 91.9 and 89.3, sequentially.
- The accuracy of the combination SCG and TPS for the three sub datasets (cross validation) was: 93.0, 92.5 and 91.9, sequentially.
- The combination of SCG and TPS produced the best and most consistent set of training and activation algorithms; even with change in the data sets, the validation error did not increase from 6.95%, except the second set, which produced a result of 8.02%.
- The best testing performance is of the Data set 1; the possible reason is fewer missing and erroneous data, hence keeping the test errors at only 3.00% for all the combinations.
- The SCG and the R4Rlog combination produced the highest number of iterations or processing time, with the best result. This combination was a computationally slow to converge; however, if we look at the validation results, this is second best out of the four combinations used.
- The training error is still relatively high for all of the combinations; however the Quasi Newton and R4RlogR combination provides the best in terms of training error.
- The highest validation error reached was 10.40% as compared to the non-matrix completed set; however, in general the results found improved by 0.4 to 0.5%, which is quite significant at the top end of predicting a correct result.
- Overall the combination of SCG and R4RlogR performed the best. The only downside is that it takes far more iterations to converge; however, this is an offline classification application, thus in terms of real-time performance this is not a consideration (Table 10).

PSOGSA

Run of experiment with changing number of hidden nodes, two datasets, Cancer data set and Saudi Arabia hospital data (CAD)

Data Set: Cancer_dataset

Inputs: 9x1000

Targets: 2x1000

Figure 8 shows the RBFN with PSOGSA algorithm trained very quickly due to the exponential decrease in the mean square error (MSE). After 30 iterations, the network was well trained (MSE < 0.3), and it continued to improve marginally after that point, yielding a good classification rate of 83.21% overall with 20 hidden units, the RBFN with PSOGSA proved to be very successful with this type of input (Figure 9).

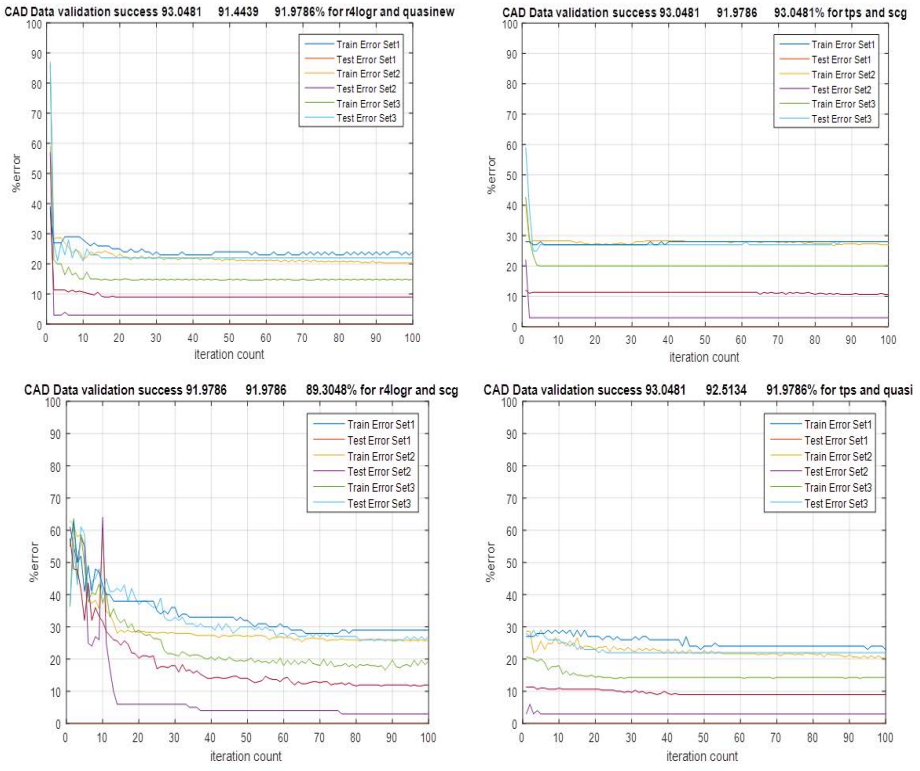

Figure 7: Three Combinations of data sets and 4 different training and activation functions with Matrix Completion.

Table 8: Summary of the best result obtained from Figure 11.

Combinations of training algorithms & activation functions		Best results of different data sets														
			Set No 1					Set No 2						Set No 3		
Training Algo	Act Fcn	Train Error %	Test Error%	Valid Error%	Accur %	Best iter	Train Error%	Test Error%	Valid Error%	Accur %	Best iter	Train Error %	Test Error %	Valid Error %	Accur %	Best iter
Quasi New	R4R	2.67	6.00	2.51	97.4	9	4.67	1.00	2.01	97.4	14	5	2	3.52	96.4	4
Quasi New	Tps	7.67	5.00	5.53	94.4	1	5.00	1.00	1.51	99.4	16	4	1	1.51	98.4	6
SCG	R4R	5.67	8.00	4.52	95.4	12	9.00	4.00	6.03	93.9	45	15.33	14	14.57	85.4	1
SCG	Tps	3.00	6.00	3.02	96.9	41	5.67	1.00	2.51	97.4	22	4.67	3	4.02	95.9	90

Table 9: Summary of the best results obtained from Figure 12.

Combinations of training algorithms & activation functions		Best results of different data sets														
			Set No 1					Set No 2						Set No 3		
Training Algo	Act Fcn	Train Error %	Test Error%	Valid Error%	Accur %	Iter no	Train Error%	Test Error%	Valid Error%	Accur %	Iter no	Train Error %	Test Error %	Valid Error %	Accur %	Best iter
Quasi New	R4R	2.67	3.00	0. 04	99.6	7	3.67	4.00	2.08	97.2	7	3	2.09	1.02	98.2	6
Quasi New	Tps	3.57	4.00	2.02	97.8	5	3.00	4.00	1.08	98.2	21	4	3.00	1.04	98.4	11
SCG	R4R	4.67	9.00	8.06	91.4	9	6.04	3.00	1.02	98.8	13	3.35	5.00	0.06	99.4	20
SCG	Tps	3.00	5.00	4.0	96	4	4.00	2.00	1.02	98.8	17	3.00	2.09	1.02	98.8	17

Table 10: Summary of the best result obtained from Figure 13

		Best results of different data sets														
			Set No 1					Set No 2						Set No 3		
Training Algo	Act Fcn	Train Error %	Test Error%	Valid Error%	Accur %	Iter no	Train Error%	Test Error%	Valid Error%	Accur %	Iter no	Train Error %	Test Error %	Valid Error %	Accur %	Best iter
Quasi New	R4R	28.33	3.00	6.95	93.0	2	20.00	21.00	8.56	91.4	3	9.00	23.0	8.02	91.9	29
Quasi New	Tps	28.67	3.00	10.70	93.0	1	14.67	22.00	7.49	92.5	21	9.00	23.0	8.02	91.9	49
SCG	R4R	26.00	3.00	8.02	91.9	76	11.33	27.00	8.02	91.9	81	12.33	28.0	10.70	89.3	66
SCG	Tps	27.67	3.00	6.95	93.0	2	23.67	25.00	8.02	92.5	3	11.33	27.0	6.95	91.9	3

Data Set: CAD_dataset

Inputs: 19x687

Targets: 2x687

In the CAD dataset, the RBFPSOGSA algorithm trains very quickly due to the exponential decrease in the Mean Square Error (MSE) After 40 iterations, the network was well trained (MSE < 0.15) and it continued to improve marginally after that point, yielding a good classification rate of 92.55%

Regarding the simple dataset, this algorithm did not react very well. Nonetheless, it could be inferred that this is likely due to the number of feature inputs.

Conclusion

This work addressed the prediction of Coronary Artery Disease by using patient information as a set of data to feed the RBF neural network. Two methods were used for training the RBF neural networks: Extended Kalman Filtering (EKF) and Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA). We found that both methods perform significantly differently on different subsets of the training and validation data. The prediction probability

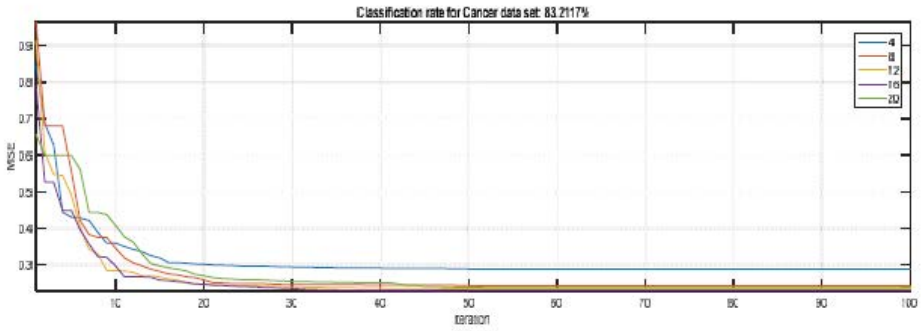

Figure 8: Classification Error for PSOGSA with varying Hidden nodes – Cancer data set.

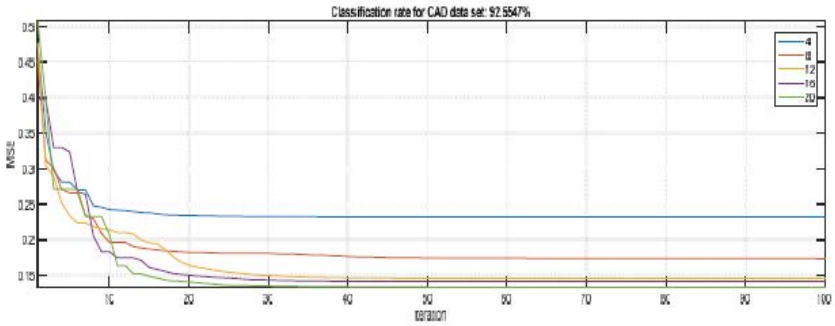

Figure 9: Classification Error for PSOGSA with varying Hidden nodes – CAD data set.

of this combination resulted in an accuracy of about 92%, using cross validation. We also noticed that the classifier has different outcomes for predicting data in different classes. This suggests the possibility that an ensemble of classifiers trained on different parts of the dataset might result in greater performance to meet the need of each sub-dataset. Results obtained on the cancer data set were better than the Hospital Data set, proving the efficiency of the different combinations (two training algorithms with two activation functions); since the Cancer data set was complete and there were no missing or erroneous data, the different combinations of testing, training and validation sets did not have a significant difference.

In supervised machine learning, the aim is to identify the relationship between some inputs and response data for regression or classification problems. For example, we are given demographic and historical medical data of the patients as the inputs along with the diagnosis data on a particular disease as the response. The paper attempted to find the underlying functional relationship between the medical observations and diagnosis data in order to predict the presence or absence of the disease from new medical data.

Acknowledgments

The authors are thankful to King Abdullah Medical City in Saudi Arabia for providing the patient data that was used in this study.

References

1. WHO (2015) WHO World Health Organization.
2. National Institute of Health (2015) What is Coronary Heart Disease. 03 October 2015.
3. British Heart Foundation (2015) Risk factors.
4. Wan DM (2000) The unscented Kalman filter for nonlinear estimation. IEEE conference.
5. Julier KU (1997) New extension of the Kalman filter to nonlinear systems. International Society for Optics and Photonics.
6. Okutani YJS (1984) Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research 18: 1-11.
7. Genders TS (2011) A clinical prediction rule for the diagnosis of coronary artery disease: validation, updating, and extension. European Heart Journal 32: 1316-1330.
8. Genders TS (2012) Prediction model to estimate presence of coronary artery disease: retrospective pooled analysis of existing cohorts. British Medical Journal 344: 1-13.
9. Yamada HD (1997) Review of Studies using multivariable analysis of clinical and exercise test data to predict angiographic coronary artery disease. Progress in Cardiovascular Diseases 39: 457-481.
10. Pryor RR (1983) Estimating the likelihood of significant coronary artery disease. American Journal of Medicine 75: 771-780.
11. Diamond GA, Forrester JS (1979) Analysis of probability as an aid in the clinical diagnosis of coronary-artery disease. N Engl J Med 300: 1350-1358.
12. Kalman R (1960) A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME. Journal of Basic Engineering 83: 35-45.
13. Heydari NZ (2012) Comparison of Artificial Neural Networks with Logistic Regression for Detection of Obesity. Journal of medical systems 36: 2449-2454.
14. Hongzong BF (2007) Support vector machines classification for discriminating coronary heart disease patients from non-coronary heart disease. West Indian Medical Journal 56: 451-457.
15. Hedeshi AM (2014) Coronary Artery Disease Detection Using a Fuzzy-Boosting PSO Approach. Computational Intelligence and Neuroscience 6: 11-15.

16. Mandal SN (2012) Accurate prediction of coronary artery disease using reliable diagnosis system. Journal of Medical Systems 36: 3353-3373.
17. Karabulut IT (2012) Effective Diagnosis of Coronary Artery Disease Using The Rotation Forest Ensemble Method. Journal of Medical Systems 36: 3011-3018.
18. OH S (2010) Matrix Completion: Fundamental Limits and Efficient Algorithms. California: Stanford University.
19. Wu TT, Lange K (2015) Matrix Completion Discriminant Analysis. Comput Stat Data Anal 92: 115-125.
20. Arif AM (2010) Detection and Localization of Myocardial Infarction using K-nearest Neighbor Classifier. Journal of Medical Systems 36: 279-289.
21. Comak E (2010) A Biomedical Decision Support System Using LS-SVM Classifier with an Efficient and New Parameter Regularization Procedure for Diagnosis of Heart Valve Diseases. Journal of Medical Systems 36: 549-556.
22. Møller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural networks 6: 525-533.
23. Haykin S (2001) Kalman Filtering and Neural Networks. John Wiley and Sons.
24. Sepasi LB (2014) Improved extended kalman filter for state of charge estimation of battery pack. Journal of Power Sources 255: 368-376.
25. Sun XM (2008) Extended Kalman Filter for Estimation of Parameters in Nonlinear State-Space Models of Biochemical Networks. PLoS ONE 3: 1-13.
26. Van Ooyen BN (1992) Improving the convergence of the back-propagation algorithm. Neural Networks 5: 465-471.
27. Mirjalili S (2010) A new Hybrid PSOGSA Algorithm for Function Optimization. International Conference on Computer Information Application.

Author Affiliations

Top

[1]Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry CV1 5FB, United Kingdom