

ECG Decision Support System based on feedforward Neural Networks

Hela Lassoued*, Raouf Ketata and Slim Yacoub

National Institute of Applied Sciences and Technology, INSAT, Centre Urbain Nord BP. 676, 1080 Tunis Cedex, Tunisia.

*E-mail: laswed.hela@gmail.com.

This paper was edited by Suresh Annamalai.

Received for publication September 12, 2018.

Abstract

The success of an Electrocardiogram (ECG) Decision Support System (DSS) requires the use of an optimum machine learning approach. For this purpose, this paper investigates the use of three feedforward neural networks; the Multilayer Perceptron (MLP), the Radial Basic Function Network (RBF), and the Probabilistic Neural Network (PNN) for recognition of normal and abnormal heartbeats. Feature sets were based on ECG morphology and Discrete Wavelet Transformer (DWT) coefficients. Then, a correlation between features was applied. After that, networks were configured and consequently used for the ECG classification. Next, with respect to the performance criteria fixed by the DSS users, a comparative study between them was deduced. Results show that for classifying the MIT-BIH arrhythmia database signals, the RBF (ACC = 99.9%) was retained as the most accurate network, the PNN (Tr_time = 0.070 s) as the rapidest network in the training stage and the MLP (Test_time = 0.096 s) as the rapidest network in testing stage.

Keywords

Machine learning, Decision Support System, ECG, Arrhythmia classification, Neural Networks.

Today, making a good decision from an amount of data has become a hard mission in various fields such as industrial, economic, medical, and many others (Brans and Mareschal, 1994). Therefore, for few decades, computers have been used to develop powerful Decision Support Systems (DSS) (Ponomariov et al., 2017) which resolve complex models and deliver fast and accurate results. For example, in medical area, the use of DSS has become a requirement since doctors have to predict immediately the appropriate medical decision corresponding to a huge quantity of heterogeneous data. In fact, the doctors are frequently analyzing long waveforms such as the Encephalograms (EEG), the Electromyograms (EMG), the Electrocardiograms (ECG), and otherwise (Reilly and Lee, 2010; Isa et al., 2013). However, up to now, there have been doctors who are identifying manually the irregularities. Consequently, this approach is considered as a limited method since it is tedious, time consuming, and sometimes uncertain. As a result, the percentage

of death in numerous medical services is constantly rising, mainly in Cardio Vascular Service where cardiologists usually make decision especially through analyzing ECG signals (De Chazel et al., 2004; Celin and Vasanth, 2017). In fact, any disorder in its characteristics or any change in its morphological pattern is an indication of cardiac arrhythmia. Thus, the development of DSS supports them to identify these abnormalities and subsequently to apply the appropriate treatment. Hence, many computer researchers have proposed several Machine Learning (ML) approaches for building DSS. These approaches are categorized in artificial intelligence area where they apply a self-learning without being explicitly programmed (Kotsiantis et al., 2007). These ML approaches are recently executed to classify ECG abnormalities (Sonawane et al., 2013). So, among the most widely used ML classification approaches, we found the Fuzzy Logic (Ozbay et al., 2006), the Neuro-Fuzzy Networks (Benali et al., 2010; Haihua et al., 2015), the Support Vector Machine

(Kohli et al., 2010), the Genetic Algorithms (Martis and Chakraborty, 2011), and the Artificial Neural Networks (ANN) (Che Soh et al., 2014). By the way, referring to literature, ANNs are among the most implemented on ECG arrhythmia classification (Kelwade and Salankar, 2015; Berkaya et al., 2018), since they are consistent in giving accurate results (Silipo and Marchesi, 1998).

There are different types of ANNs, such as feedforward backpropagation neural networks (Matul Imah et al., 2013; Erkaymaz et al., 2017; Savalia et al., 2017), recurrent networks (Rather et al., 2015), convolutional neural networks (Kiranyaz et al., 2015), Hopfield networks (Li et al., 2016), etc. Even so, selecting the appropriate ANN for a classification task is still a research topic.

On this subject, we focus on the preparation of a DSS, to classify ECG recordings into normal and pathological class. Accordingly, three different feedforward ANNs, which are the Multilayer Perceptron (MLP) (Rafiq et al., 2001), the Radial Basic Function Network (RBF) (Zhao and Li, 2017), and the Probabilistic Neural Network (PNN) (Mao et al., 2000) are applied. We aim mostly to prepare the DSS to choose automatically the appropriate ANN structure, with respect to the users' criteria. Consequently, while taking into account the accuracy (ACC), the time classification (Time_response), and the Mean Square Error (MSE), we have evaluated for each ANN, its optimum structure (MLP_opt, RBF_opt, and PNN_opt). Then, a comparative study between them is reached. After that, according to the demanded users' criteria, the DSS chooses the appropriate optimum ANN structure.

This paper is organized as follows. In "Materials and methods" section, the proposed arrhythmia classification methodology is described. In this section, we begin by presenting the arrhythmia database MIT-BIH. Then, we present the feature extraction as well as the used feedforward artificial neural networks. Next, we detail the experiments and results in "Experiments and results" section. In "Discussion" section, a comparative study is discussed. Finally, a conclusion is reached.

Materials and methods

Overview

In order to prepare a DSS which classifies ECG signals into normal and abnormal class, we have followed the entire methodology described in the following block diagram (Figure 1). The block diagram includes two main stages which are Pre-processing and Neural Network arrhythmia classification. The first stage deals with the MIT-BIH ECG recordings pre-processing (Silva

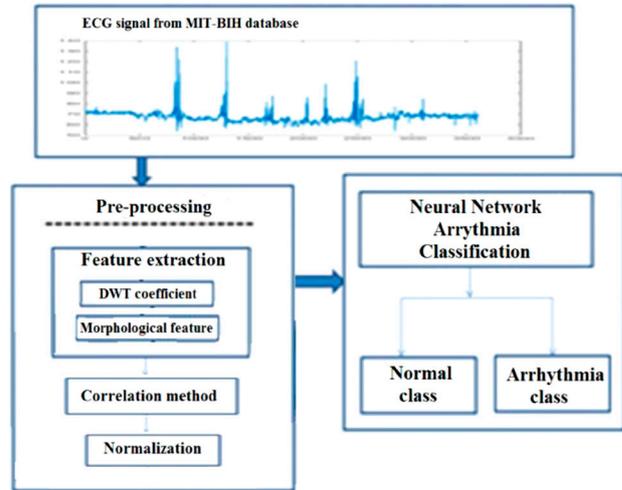


Figure 1: Block diagram of an ECG arrhythmia classification topic.

and Moody, 2014). In fact, Discrete Wavelet Transformer (DWT) coefficients and morphological features were extracted. Then, a correlation method is applied to select the most pertinent features. After that, the selected features are normalized and scaled to identify the input dataset. However, the second stage consists on the Neural Network Arrhythmia classification. This stage is based on the use of three feedforward ANN, as machine learning classifier, which are the MLP, the RBF, and the Probabilistic PNN.

In this work, we focus mostly on the second stage. Therefore, according to the flow chart described in Figure 2, the input dataset is divided into two datasets used for training and testing the ANNs. The division respects the Association for the Advancement of Medical Instrumentation (AAMI) recommendations (De Chazel et al., 2004). In addition, according to Figure 2, structural and parametric studies are done in order to select the three optimum structures (MLP_opt, RBF_opt, and PNN_opt). Therefore, during these two studies, each ANN is trained and tested many times. Thus, several structures are obtained. Among them, the optimum one is which achieves the highest ACC, the fast Time_response, and the lowest MSE. But, when the obtained accuracies are equals, the ANN structure which has the fast Time_response is selected as the optimum one. Whereas, when both of the accuracies and time responses are equals, the optimum ANN structure is one which has the lowest MSE. After obtaining the optimum structures (MLP_opt, RBF_opt, and PNN_opt), a comparison study between them is investigated, according to the number of hidden neurons, ACC, Time_response, and MSE. Then, DSS chooses one of the optimum structures

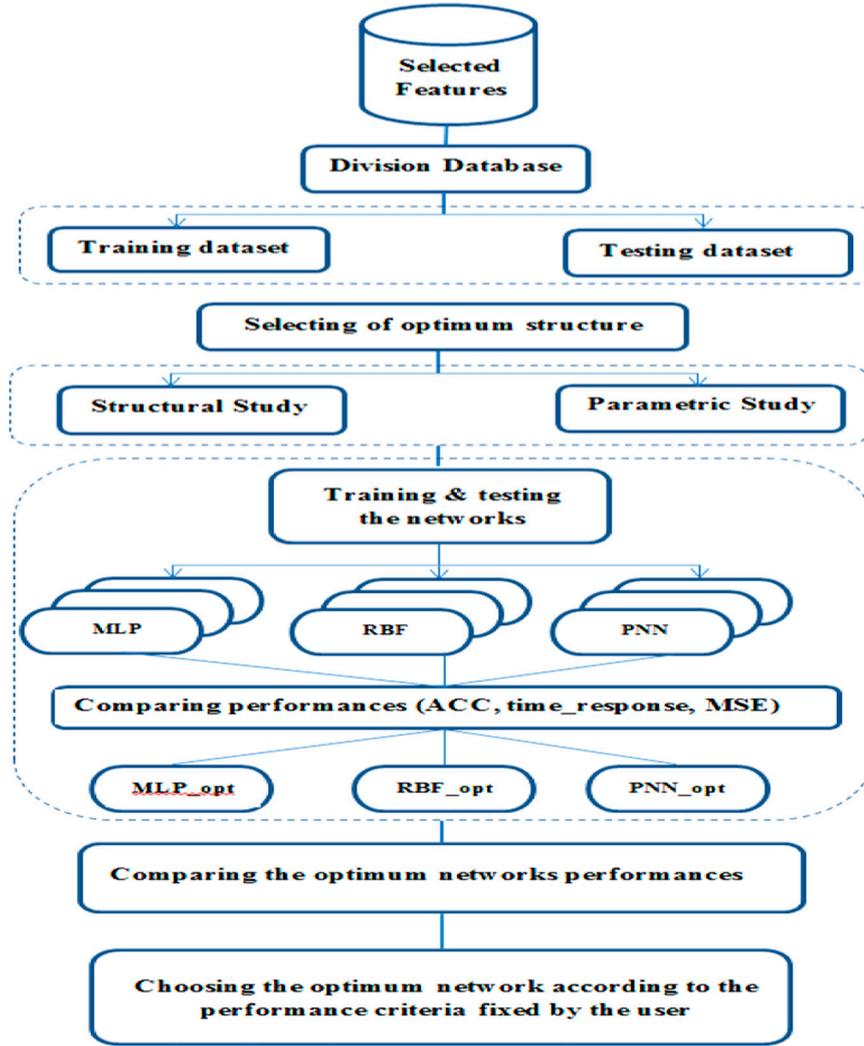


Figure 2: Flow chart of the proposed work.

with respect to the users' performance criteria. In fact, users might demand the DSS either to choose the ANN structure where decision time has got priority over decision accuracy, either to choose the most accurate ANN and no matter how rapid the network, or to choose automatically the structure without any requirements. In this case, the DSS chooses the most accurate network.

The evaluation performances, used to evaluate the three ANNs, are described in the following equations:

$$ACC = \frac{NCC}{N} \quad (1)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (Target - Output) \quad (2)$$

$$Time_response = Tr_time + Test_time \quad (3)$$

where NCC is the number of correctly classified inputs vector, N is the number of inputs vector, Tr_time is the training time, and $Test_time$ is the testing time.

MITBIH arrhythmia database

The standard arrhythmia database MIT-BIH was used in this study. It contains 48 records of heartbeats at 360Hz for approximately 30min of 47 different patients. Each record has two ECG leads (lead A and lead B) which are depending on the electrodes configuration on the patient's body. The database includes approximately 109,000 beat labels. In addition,

it contains 25 normal records, 19 abnormal records, and 4 paced beats records (102, 104, 107, and 217) which were excluded in this study. Therefore, we have only treated the first minute of 44 ECG samples where those with arrhythmia can be interpreted as “1” and the other without arrhythmia can be interpreted as “0” (Silva and Moody, 2014).

Feature extraction

The classification stage of ECG waveforms requires the extraction of the corresponding features (Alfarhan et al., 2017). In fact, the ECG waveform, as it is shown in Figure 3, consists of three basic waves P, QRS, and T waves and sometimes U wave. Indeed, P wave arises due to the atrial depolarization, QRS complex represents the ventricular depolarization, and T wave arises due to the repolarization of the ventricle (De Chazel et al., 2004).

In this study, morphological and time frequency features are extracted (Lassoued and Ketata, 2017; Lassoued and Ketata, 2018). On the first hand, we have applied two MATLAB toolbox which are the waveform Database software Package (WFDB toolbox) (Silva and Moody, 2014) and the ECG-kit (Demski and Llamedo, 2016) to extract the morphological characteristics. Thus, we have identified for each ECG recording, peak values (P, Q, R, S and T), the standard deviation of time duration between waves (P-R, P-T, S-T, Q-T, Q-R-S, and R-R) and the mean and the standard deviation of the heart rate values (Lassoued and Ketata, 2017; Lassoued and Ketata, 2018). On the second hand, we have returned, for each ECG signal, the details and approximations coefficients of the discrete wavelet transformer (DWT) of the mother wavelet Daubechiesb6 (db6), with eight decompositions. Accordingly, we have considered their arithmetic mean, variance, and standard deviation. Finally, we have collected for each ECG record, a total of 62 features consisting of 48 DWT coefficients and 14 morphological characteristics (Lassoued and Ketata, 2017; Lassoued and Ketata, 2018).

Feature normalization

Although normalization is not necessary for MLP and RBF networks, it is recommended for PNN. So, for a fair comparison between the three different ANNs (MLP, RBF, and PNN), we have applied the WEKA Min-Max normalization function to standardize all features at the same level (Yadav et al., 2014; Jain et al., 2017), as it is shown in Equation (3). By applying this equation, all the component of the input feature vector are normalized and rescaled in the interval [0, 1]:

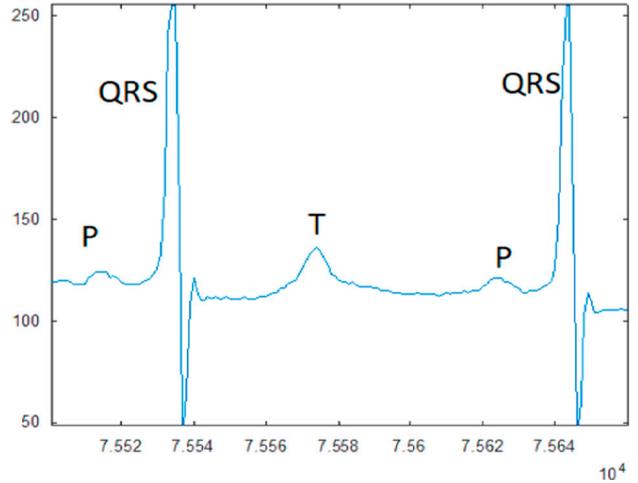


Figure 3: Normal ECG recording.

$$X_{nor} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{4}$$

where X_{min} and X_{max} are the minimum and the maximum values for one feature, respectively. X_{nor} is the normalized feature vector and X is the original feature vector.

Feature selection

The ACC of the classifier depends on several factors, such as the quality and the size of the input feature vectors. Thus, feature selection stage is required in order to use the most pertinent features and also to reduce the feature vectors size. For this purpose, we have applied the WEKA selection function “Correlation Attribute Eval” (Savic et al., 2017). This function evaluates features by measuring the correlation between the feature and the corresponding class (normal or abnormal). In this study, we have considered only the attributes having a correlation ranking up to 0.300. Therefore, the size of the input matrix becomes (10 × 44) instead of (62 × 44). Table 1 describes the degree of correlation for the 10 selected features.

Feed forward artificial neural networks

The ANN is inspired from the biological neural networks functioning. They are distinguished by their ability to identify nonlinear relationships between data, without the need to understand the nature of the phenomena (Dalvi et al., 2016). Regarding the feed forward ANN, they are familiar by the aptitude

Table 1. Selected attributes.

Selected Features	Max-Q	Std-a6	Std-a5	Std-a7	Std-a4	Std-a3	Std-a2	Std-a1	std-a8	Moy-d8
Correlation rank	0.350	0.326	0.318	0.314	0.311	0.310	0.307	0.306	0.304	0.300

Max- Q: Maximum of peak Q in ECG signal; Std- a_i : Standard deviation of approximation number i ($i \in [1-8]$); Moy- d_8 : Average of detail number 8. Min: Minimum value, Max: Maximum value, Std: Standard deviation.

of their neurons in the input layer to feed their output, through the hidden layer, until the output layer. In fact, whatever the feed forward ANN, the number of neurons in the input layers is adjusted to the number of variables in the input vector. The number of neurons in the output layer is appropriate for the number of classes. The number of neurons in hidden layer depends on the classification problem.

The following equation defines how the neurons compute their outputs:

$$O = \sum_{i=1}^N w_i x_i \quad (5)$$

where O is the neuron's output, N is the number of inputs, x_j , w_j are the input variable and its corresponding weight, respectively.

In this work, we have focused on three feedforward neural networks, which are the MLP, RBF, and PNN. In the following sub sections, each ANN is explained in details.

MLP network

The MLP consists of an input layer, one or more hidden layers followed by an output layer. MLP uses the backpropagation algorithm for training data. In fact, it begins the training with random weight values and calculates the output value from a set of records for which the expected output value is known. The weights are transferred to the next level via the activation functions. The below equation shows how the output of the MLP (Y_i) can be computed (Khan and Jabbar, 2009):

$$Y_i = \sum_{j=1}^m f(w_{ij} x_j + b_i) \quad (6)$$

where w_{ij} is the corresponding weight for the input j in the layer l , m is the number of inputs in the

previous layer, f is the activation function, and Y_i is the output of the layer. Figure 4 illustrates a MLP network structure.

RBF

Radial basis function network consists also of three layers: an input layer, one hidden radial basis layer, and an output linear layer. It is known by its hidden layer which contains a set of radial basic functions, which are symmetrical and centered at each input feature. They are similar to normal distribution curves. Each neuron in the hidden layer computes the distance between the input vector and its own center. The calculated distance is transformed using the basis function and the result is the output from the neuron. Then, this later is multiplied by a weighting value and fed into the output layer. Finally, the output layer with its linear activation function acts to sum the outputs of the previous layer and yields a final output value for each class (Seshagiri and Khalil, 2000). Figure 5 illustrates a RBF network structure.

The output neuron $y_k(x)$ is as follows:

$$y_k(x) = \sum_{j=1}^l w_{kj} \varphi(\|x - \mu_j\|, \delta_j) \quad (7)$$

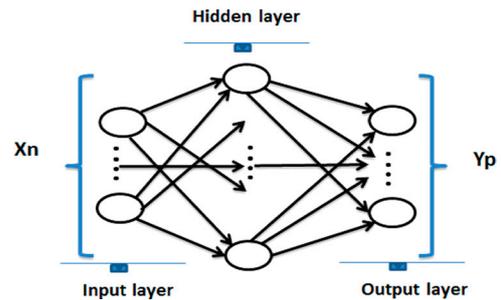


Figure 4: MLP Network structure.

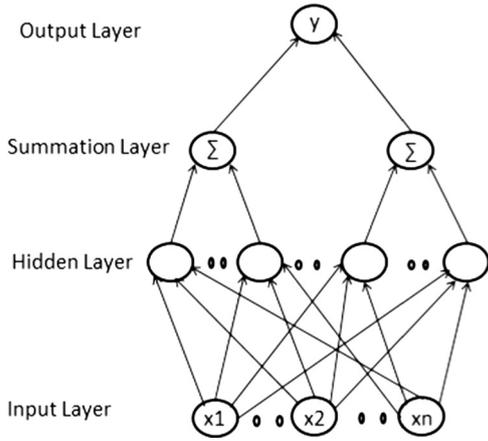


Figure 5: RBF Network structure.

where y_k is the output for the class k , w_{kj} is the associated weigh for the result of the hidden neurons j corresponding to the class k , φ is the radial function, (μ, δ) are its center and its width, respectively, l is the number of hidden neurons, x is a vector of input data.

Accordingly, to configure the RBF network, the choice of the radial basis function as well as the estimation of its centers and its widths is the main challenge.

PNN

PNN consists of four layers: an input layer, one hidden layer, a summation layer, and an output layer. The input layer contains N neurons corresponding to N features. The second layer consists of Gaussian functions for each exemplar feature vector. The third layer consists of the summation layer which sums the output of the hidden layer neurons belonging to the same class. The output of each summation unit is proportional to the Probability Density Function (PDF) of the corresponding class. The fourth layer consists of a competitive output layer that picks the maximum value of its inputs and returns the corresponding class associated to this maximum.

The PNN network for K classes is defined as it is shown in the below equation (Farhidzadeh, 2015):

$$y_j(x) = \frac{1}{n_j} \sum_{i=1}^{n_j} \exp\left(-\frac{(\|x_{j,i} - x\|)^2}{2\sigma^2}\right) \quad (8)$$

The PNN classifies x into k class when the following condition is true:

$$y_k(x) > y_j(x) \quad (9)$$

where n_j represents the number of features in class j , $j \in 6 [1, \dots, M]$, M is the number of classes, $\|x_{j,i} - x\|^2$ is calculated as the sum of squares, x is any input feature vector, $x_{j,i}$ is the i feature of a training feature vector belonging to class j , σ is the smoothing parameter (Spread) which has to be selected in order to minimize the estimated PDF error. Figure 6 illustrates a PNN network structure.

Similarities and difference between networks

Table 2 presents a comparison study which illustrates the similarities and the differences between the three networks (MLP, RBF, and PNN), regarding structural and parametric criteria.

Experiments and results

All of the experiments were conducted by MATLAB R2015.a, on a 64-bit Windows 10 with 64 GB memory and 3.4 GHz I7 CPU. Indeed, we have performed an arrhythmia classification task using three feedforward neural networks (MLP, RBF, and PNN). For each network, we have prepared its own configuration in order to obtain its optimum structure. Then, a comparative study between the three obtained optimum structures (MLP_opt, RBF_opt, and PNN_opt) is done.

MLP configuration

In this section, we aim to configure the optimum structure MLP_opt. Therefore, structural and parametric studies are prepared.

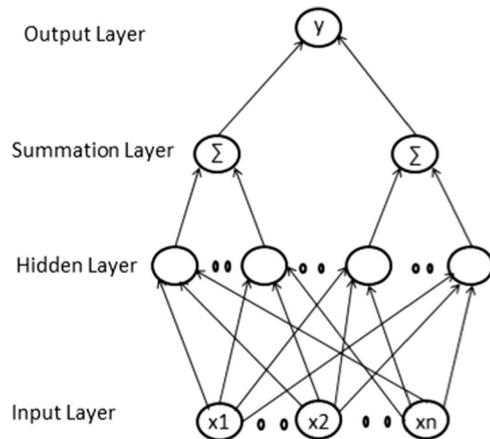


Figure 6: PNN Network structure.

Table 2. MLP, RBF, and PNN Feed_Forward neural networks.

Criteria types	Criterion	MLP	RBF	PNN
Structural	Architecture	An input layer, one or more hidden layer and an output layer	An input layer, one hidden layer and an output layer	An input layer, one hidden layer, a summation layer and an output layer
	Activation function	The activation function is non-linear (sigmoid, log-sigmoid, tan-sigmoid.)	The activation function is a radial basis function which computes the Euclidian distance of the input vector and its weights	The activation function is based on the probability density function
	Number of hidden neurons	no defined principle for determining the number of neurons	no defined principle for determining the number of neurons	The number is equal to the number of instance
	Output layer	The final layer uses the activation function before linearly combining it	The final layer doesn't use activation function, it rather linearly combines the output of the previous neuron	The final layer is a competitive output layer. It picks the maximum of the computed probabilities
	Training Process	Backpropagation training algorithms	Backpropagation or clustering algorithms	There is no computation of weights. The Bayesian decision rule
Parametric	Parameters	Momentum factor, learning rate, parameters according to the training algorithm	Number of centers, spread of radial function	Spread value of the probability density function

MLP structure

In the structural study, first, we have evaluated the impact of the number of hidden layers as well as that of neurons per each hidden layer. Second, we have studied the impact of the learning algorithm. Whereas for the transfer function, after a test and trial study, we have selected the tangent sigmoid and linear activation functions for the hidden and output layers, respectively.

- Hidden layer

In order to evaluate the performances of the MLP, we have varied the number of hidden layers from 1 to 2 and the number of neurons in the hidden layer from 5 to 25. The results are shown in Table 3.

First, by using the MLP with a single hidden layer, the highest ACC reached an ACC of 86.4%. Whereas, by adding a second hidden layer, the most accurate MLP achieves 81.8%. Therefore, it is not necessary to use an MLP with two hidden layers in this classification task. Second, when the number of neurons in the single hidden layer is small, for example 5, the ACC (72.7%) is the lowest and the MSE (0.026) is the highest. Thus, the ACC increases with the increasing number to some extent. Indeed, when the number reaches 15 neurons in the single hidden layer, the ACC is stacked in 81.8% and does not improve anymore.

Third, we have concluded that as long as the network has less number of hidden neurons, the network is considered fast. In fact, by adding neurons in the hidden layer, the learning time increases (0.221 s up to 0.406 s) and the test time decreases

Table 3. MLP performance using different number of hidden layer.

N_HL	NO/HL		ACC(%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
	H1	H2					
1	5	0	72.7	0.221	0.113	0.334	0.026
	10	0	86.4	0.222	0.094	0.316	0.024
	15	0	81.8	0.380	0.099	0.479	0.021
	20	0	81.8	0.390	0.073	0.463	0.016
	25	0	81.8	0.406	0.071	0.477	0.013
2	10	5	81.8	0.315	0.238	0.553	0.019
	10	10	81.8	0.319	0.253	0.572	0.021
	10	15	72.7	0.383	0.281	0.664	0.018
	10	20	72.7	0.423	0.317	0.740	0.017

N_HL: number of hidden layer; NO/HL: Number of neurons per hidden layer; H₁: First hidden layer; H₂: Second hidden layer.

(0.026s up to 0.013s). As a result, we have kept the MLP structure (10, 10, 1) which has 10 neurons in the input layer, 10 hidden neurons, and a neurons in the output layer. By using this structure, we have obtained 86.4% as ACC, a MSE equals to 0.094, a Tr_time equals to (0.222 s), and a test_time equals to (0.024 s).

- Training algorithm

In this section, we have used two categories of learning algorithms based on the principle of backpropagation to evaluate the performances of the saved MLP structure (10, 10, 1). Indeed, the first category uses Jacobian derivatives algorithms which include Levenberg–Marquardt backpropagation (trainlm) and Bayesian regularization backpropagation (trainbr) algorithms. While the second category uses gradient derivatives which contains several algorithms. However, in this study, we have chosen some of them which are the Scaled Conjugate Gradient backpropagation (trainscg), Gradient Descent with Adaptive learning rate backpropagation (traingda), Resilient Backpropagation (trainrp), Gradient Descent with momentum and Adaptive learning rate backpropagation (traingdx), BFGS quasi-Newton backpropagation (trainbfg), Conjugate gradient backpropagation with Powell-Beale restarts (traincgb). The obtained results are shown in Table 4.

Regarding ACC, the MLP network achieves the highest ACC of 86.4% by applying the following algorithms: “trainlm,” “trainscg,” “trainrp,” and “trainbfg.” However, concerning time response, the Jacobian derivatives algorithms are faster than those of the gradient derivatives. In fact, the slowest learning time is given by the algorithm trainLM (0.222 s). In addition, relating to efficiency, the gradient derivatives algorithms generate more efficient results in terms of MSE (0.073) compared to the results obtained using the Jacobian derivatives algorithms (0.094). As a result, we have chosen the trainrp algorithm since it generates a maximum ACC (86.4%), a minimum MSE value (0.073), a Tr_time (0.294 s), and a Test_time (0.096 s).

MLP parameters

At this point, we have to determine the values of the parameters corresponding to the saved MLP structure which applies the training algorithm trainrp. In fact, trainrp uses the coefficient Δ_Inc to increment the weight change and the coefficient Δ_Dec to decrement the change of weight. Therefore, we have to adjust the learning rate Lr, Δ_Inc and Δ_Dec . So, in this study, we have fixed 1.2 and 0.5 as the values of Δ_Inc and Δ_Dec , respectively. However, we have modified the values of the learning rate Lr in the range of [0, 1] as it is described in Table 5.

Table 4. MLP performance with different learning algorithms.

Learning algorithms types	Learning algorithms	ACC(%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
Jacobian derivatives	trainlm	86.4	0.222	0.024	0.246	0.094
	trainbr	74.5	0.229	0.031	0.260	0.124
Gradient derivatives	trainscg	86.4	0.553	0.355	0.908	0.077
	traingda	81.8	0.416	0.218	0.634	0.078
	trainrp	86.4	0.294	0.096	0.390	0.073
	traingdx	81.8	0.543	0.345	0.888	0.076
	trainbfg	86.4	0.330	0.132	0.462	0.087
	traincgb	81.8	0.316	0.118	0.434	0.070

Referring to Table 5, whatever the Lr, we have obtained the same ACC and approximately the same time response. Therefore, we have chosen the Lr value equals to 0.1, as we have obtained the lowest MSE (0.064). Hence, the MLP_opt is the structure which has a single hidden layer with 10 neurons and applies the trainrp learning algorithm. Its parameters, Lr, Δ_Inc , and Δ_Dec are maintained to 0.1, 1.2, and 0.5, respectively.

RBF network configuration

In this section, we aim to configure the optimum structure RBF_opt. Therefore, structural and parametric studies are analyzed.

RBF structure

In the structural study, on the first side, we have evaluated the impact of the number of hidden neurons in

Table 5. MLP performance by learning rate.

Lr	ACC(%)	Time_response (s)	MSE
0.01	86.4	0.390	0.130
0.1	86.4	0.399	0.064
1	86.4	0.392	0.073

the single hidden layer. On the second side, we have studied the choice of the basis function and its impact on the RBF performances.

- Hidden layer

We have evaluated the RBF network by varying the number of neurons in the hidden layer from 5 to 25 as mentioned in Table 6.

Referring to Table 6, it can be seen that the ACC increases with the increasing number of neurons in the single hidden layer. Therefore, ACC improves from 54.5% up to 99.9%. Moreover, the MSE is becoming closer to zero ($2.411e-30$ to 0.204). However, the Tr_time increases from 0.380s to 0.699. Besides, the Test time is almost the same (on average 0.015) regardless of the number of neurons in the hidden layer. Hence, we have chosen the RBF network which achieves the highest ACC of 99.9% and the lowest MSE ($1.266e-30$) by using 20 hidden neurons.

- RBF activation function

In this section, we have evaluated the saved RBF structure (10, 20, 1) by using several types of radial basic functions, such as the Biharmonic, Multiquadric, Inverse Multiquadric, Thin Spline, and Gaussian (Buhmann, 2000). Table 7 summarizes all the results for the different basic functions.

Based on the results obtained in Table 7, since we have obtained the same ACC as well as the Time_response (on average 0.661 s), we have kept the radial func-

Table 6. RBF performance using different N/HL.

NO/HL	ACC (%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
5	54.5	0.380	0.015	0.395	0.204
10	59.1	0.436	0.013	0.449	0.150
15	81.8	0.501	0.014	0.515	0.080
20	99.9	0.640	0.014	0.654	1.266e-30
25	99.9	0.699	0.017	0.716	2.411e-30

NO/HL: nodes per hidden layer.

Table 7. RBF performance using different basic functions.

RBF functions	ACC(%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
Gaussian	99.9	0.380	0.014	0.654	1.266e-30
Polyharmonic	99.9	0.436	0.051	0.639	1.221e-30
Inverse_multiquadric	99.9	0.501	0.121	0.676	0.250e-30
Multiquadric	99.9	0.640	0.132	0.698	0.891e-30
Biharmonic	99.9	0.699	0.002	0.640	0.891e-30

tion which accords the lowest MSE (0.250e-30). Thus, the Inverse_multiquadratic function is chosen. This basic function is presented in Equation (10) (Farhidzadeh, 2015). Where ϕ is the basic function, μ is the center, δ is the width (or the spread), and x is the input vector.

This function computes the Euclidean distance ($\|.\|$) between the interpolation centers and the input vectors:

$$\varphi(x) = \frac{1}{\sqrt{(\|x - \mu\|)^2 + \delta^2}} \quad (10)$$

RBF parameters

We have chosen the RBF network which has 10 neurons in the input layer, 20 neurons in the hidden layer and 1 neuron in the output layer. This network applies the inverse multi quadratic as the basic function in the hidden layer. Therefore, we have to learn the centers μ and the spreads δ as well as the weights from the hidden to the output layer. Regarding centers μ , they are chosen randomly from the training set. Regarding

weights, they are computed by the backpropagation algorithm. However, we have to learn the spreads δ values. In Table 8, we have summarized the RBF performances by changing the spread values.

We have obtained the same ACC (99.9%), Time_response (0.662s on average), and the same MSE (0.250 e-30), whatever the value of the spread parameter. However, we have obtained different Test_time results. So, by referring to the fastest one (0.121 s), we have chosen the spread value 10. As a result, the optimum structure RBF_opt (10, 20, and 1) uses the inverse quadratic radial functions which its center is calculated randomly, its spread value is 10 and its weights are computed using the backpropagation algorithm. This network achieves the highest ACC (99.9%), a MSE close to zero, and a significant test time (0.121 s).

PNN network configuration

In this section, we aim to configure the optimum structure PNN_opt. Therefore, structural and parametric studies are investigated.

Table 8. RBF performance using different spread values.

Basic function	Spread	ACC(%)	Test_time(s)	Time_response (s)	MSE
Inverse_multiquadric	10	99.9	0.121	0.643	0.250 e-30
	1	99.9	0.128	0.668	0.250 e-30
	0.1	99.9	0.140	0.676	0.250 e-30

Table 9. PNN performance.

NO/HL	ACC(%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
22	79.5	0.081	0.218	0.299	0.162

PNN structure

For the PNN network structure, we have studied the impact of the number of hidden neurons in the single hidden layer. However, for the standard PNN network, the number of neurons in the hidden layer is always equal to the number of examples in the training dataset. Indeed, in this study, the input matrix is of size (10×22). Therefore, the PNN generates 22 neurons in the hidden layer. Results obtained are summarized in Table 9. Actually, we have obtained an overall ACC (79.5%) with a MSE of 0.162. Tr_time is processed for 0.081 s and the Test_time is processed in 0.218 s.

PNN parameters

For the PNN, the spread parameter has to be adjusted. Then, we have simulated the PNN with different range in [0.1, 10] of spread parameter. As a result, the value of spread parameter that ensures the best ACC, the least MSE and the speediest classification time response, is chosen. Results are summarized in Table 10.

Referring to the results obtained in Table 10, we have remarked that whatever the spread value, we have obtained same ACC, MSE, and Test_time. Accordingly, it is clear that the PNN network is not influenced by the choice of the spread parameter. We have chosen the value 1 of the spread parameter, since we have obtained the shortest learning time (0.070s). As a result, the optimum structure PNN_opt (10, 22, and 1) with a spread value is equal to 1. This network achieves the highest ACC (79.5%), a MSE (0.162), and a significant Tr_time (0.070s).

Discussion

In this section, two comparative studies are analyzed. The first study aims to compare the optimum structures (MLP_opt, RBF_opt, and PNN_opt) performances. However, the second study aims to compare the proposed work with other related works.

Comparative study between the three optimum structures

After configuring each type of ANN, three optimum networks (MLP_opt, RBF_opt, and PNN_opt) were obtained. Subsequently, the DSS will select one of them according to the criteria demanded by the users. Indeed, users may demand either an accurate response regardless of the classification Time_response, or a fast response regardless of the decision accuracy. Hence, a comparative study between the three optimum structures is required. Therefore, as it is mentioned in Table 11, a comparison study between them in terms of the number of hidden neurons, ACC, MSE, and Time_response (Tr_time and Test_time) is reached.

Thus, referring to Table 11, all the optimum ANNs have good performances on classifying ECG records into normal and arrhythmia class. However, each network has its advantages and its drawbacks. In the first hand, concerning ACC, it is revealed that RBF network achieves the highest ACC of 99.9% followed by MLP_opt network (86.7%) and PNN (79.5%). In the second hand, regarding the number of hidden nodes, we have obtained 10, 20, and 22,

Table 10. Performances of the PNN network according to the spread parameter.

Spread	ACC(%)	Test_time(s)	Time_response (s)	MSE
0.1	79.5	0.081	0.218	0.162
1	79.5	0.070	0.218	0.162
10	79.5	0.074	0.218	0.162

Table 11. ANNs Performances.

ANN	N0/HL	ACC(%)	Tr_time(s)	Test_time(s)	Time_response (s)	MSE
MLP_opt	10	86.4	0.294	0.096	0.390	0.064
RBF_opt	20	99.9	0.755	0.121	0.876	0.250 e-30
PNN_opt	22	79.5	0.070	0.218	0.288	0.162

for the MLP, RBF, and PNN networks, respectively. Consequently, it is noting that the MLP_opt network is the least complex network among the different classifiers. In the third hand, linking to classification time response, we have analyzed the Tr_time and Test_time for each optimum ANN. Therefore, first, PNN_opt achieves the lowest Tr_time (0.070 s) and the highest Test_time (0.218 s). Then, by using PNN_opt, nearly no time consumed for training but Test_time is long. Second, MLP_opt consumes the shortest Test_time (0.096 s) than the RBF_opt (0.121 s) and the PNN_opt (0.162 s). As a result, the comparison between the three optimum ANNs proves that the most relevant ACC (99.9%) and the lowest MSE (0.250 e-30) are provided by the RBF_opt network. However, PNN_opt applies the rapidest training phase (0.070 s) and MLP generates the shortest testing time (0.096 s).

Hence, according to the users' criteria, the DSS will choose one of the optimum structures. In fact, it will select RBF_opt, if experts want an accurate response regardless of the classification time_response. However, if users want a fast response, DSS will select MLP_opt (Test_time = 0.007 s). Whereas, when users have a huge dataset, it is required to choose PNN_opt since it consumes the lowest learning time (Tr_time = 0.06 s).

Comparative study with related works

In order to evaluate the efficiency of this work, a second comparative study with related works, is prepared. Therefore, Table 12 yields a summary of arrhythmia classification studies using Feed Forward ANN as a machine learning model, to classify ECG signals into two classes (normal and abnormal). Thus, we have compared the classification accuracies of the optimum structures (MLP_opt, RBF_opt, and PNN_opt) with those reported in literature.

Based on the results in Table 12, the classification accuracies are effective to the tune of about 99.9 to 82.5% ACC. Therefore, the obtained accuracies (99.9, 86.5, and 79.5%) are considered acceptable. However, comparing only the accuracies of the feed forward ANN classifiers is not sufficient to make a fair comparison. In fact, it is necessary to highlight several factors, such as the choice of the database and the number of recording analyzed.

Indeed, according to the Table 12, the ECG arrhythmia database MIT_BIH was adopted by all the studies, except (Rai et al., 2013; Savalia et al., 2017) who have added to MIT_BIH recordings the ECG signals from NSR database. It is also found that each study analyzed the different number and recording time of ECG signals. Besides, each study has

Table 12. Comparative study with related works.

ANN	Pre-processing	Test conditions	Division Datasets	ACC(%)
(Abhinav-Vishwa et al., 2011)	R peak	MIT_BIH database of 48 signals of 30min	50% for training and 50% for testing	96.8
(Rai et al., 2013)	Morphological and DWT coefficients	45 ECG signal of 1 min from MIT-BIH database	26 signals for training and 19 for testing	97.8
(Tomar et al., 2013)	Morphological, DWT coefficients, power spectral density and Energy of Periodogram	62 ECG signals of 10 s from MIT-BIH database and Normal Sinus Rhythm (NSR)	Cross validation division (70%, 30%)	98.4
(Savalia et al., 2017)	R peaks, the heart beats/min, the duration of complex QRS	66 ECG signal from MIT_BIH arrhythmia database and NSR database	Cross validation division (70%, 30%).	82.5
(Dalvi et al., 2016)	QRS complex, RR interval and the beat waveform morphology. PCA for feature selection	MIT_BIH database of 48 signals of 30min.	18 ECG records for test dataset and 30 ECG for train dataset	96.9
Proposed work	Morphological and DWT coefficients	44 ECG signals of 1 min. recording from MIT_BIH database	22 signal for training and 22 for testing following the AAMI.I recommendations	99.9

its own methodology. For example, for the pre-processing stage, many approaches are approved for the feature extraction, such as the use of DWT coefficient (Abhinav-Vishwa et al., 2011), morphological features (Buhmann, 2000; Rai et al., 2013; Dalvi et al., 2016; Savalia et al., 2017) and many others. Similarly, for feature selection, many methods are employed like the Principal component analysis (PCA) (Dalvi et al., 2016), the correlation method (our proposed work) and others. However, there are works which do not use any method for feature selection (Buhmann, 2000) and also for normalization (Savalia et al., 2017). Other point, each study has divided the input dataset into two sets, for training and testing, by using different approaches. For our proposed work, we have respected the AAMI recommendations, while (Abhinav-Vishwa et al., 2011) has chosen its own datasets and (Savalia et al., 2017) has applied the cross validation division.

Conclusion

This work is as a guide for the DSS to choose the appropriate ANN. For that, three feed forward ANNs (MLP, RBF, and PNN) were evaluated to classify ECG

signals into normal and arrhythmia class. In fact, we have used the first minute of ECG recording from MIT-BIH database. In addition, we have extracted the morphological and the DWT coefficients features. Then, a correlation method was applied to select the most pertinent features. These later were normalized and scaled to identify the input vector for the three ANNs. After that, for each network, structural and parametric studies were investigated. Consequently, three optimum networks (MLP_opt, RBF_opt, and PNN_opt) were obtained. Then, a comparative study between them is realized. Therefore, it proves that RBF is the most accurate network since it achieved (ACC = 99.9%), while the MLP had the fast testing response as it reached the lowest Test_time (0.096 s) and the PNN had the fast training response since it attained the lowest Tr_time (0.070 s). At the end, on the basis of the acquired ACC, we did a comparative study between the proposed work and other related works in the literature. Hence, although not executing the same ECG arrhythmia classification methodology, the obtained optimum structures (MLP_opt, RBF_opt, and PNN_opt), were considered as acceptable networks with 86.8, 99.9, and 79.1% accuracy, respectively.

Literature Cited

- Abhinav-Vishwa, M.K., Lal, S.D., and Vardwaj, P. 2011. Classification of arrhythmic ECG data using machine learning techniques. *International Journal of Interactive Multimedia and Artificial Intelligence* 1(4): 68–71.
- Alfarhan, K.A., Mashor, M.Y., Saad, A.M., AbdulAzeed, H., and Al-Qaisee, M. 2017. Effects of the window size and feature extraction approach for arrhythmia classification. *Journal of Biomimetics, Biomaterials and Biomedical Engineering* 30: 1–11.
- Benali, R., Dib, N., and Bereksi, F.R. 2010. Cardiac arrhythmia diagnosis using a neuro-fuzzy approach. *Journal of Mechanics in Medicine and Biology* 10(3): 417–29.
- Berkaya, S.K., Uysal, A.K., Gunal, E.S., Ergin, S., Gunal, S., and Gulmezoglu. 2018. A survey on ECG analysis. *Biomedical Signal Processing and Control* 43: 216–35.
- Brans, J.P., and Mareschal, B. 1994. The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decision Support System* 12(4–5): 297–310.
- Buhmann, M.D. 2000. Radial basis functions. *Acta Numerica* 9: 1–38.
- Celin, S., and Vasanth, K. 2017. Survey on the methods for detecting arrhythmias using heart rate signals. *Journal of Pharmaceutical Sciences and Research* 9 p. 183.
- Che Soh, A., Chow, K.K., Mohammad Yusuf, U.K., Ishak, A.J., Hassan, M.K., and Khamis, S. 2014. Development of neural network-based electronic nose for herbs recognition. *International Journal on Smart Sensing & Intelligent Systems* 7(2): 584–609.
- Dalvi, R.D. F., Zago, G.T., and Andreao, R.V. 2016. Heartbeat classification system based on neural networks and dimensionality reduction. *Research on Biomedical Engineering* 32(4): 318–26.
- De Chazel, P., O'Dwyer, M., and Reilly, R.B. 2004. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering* 51(7): 1196–206.
- Demski, A., and Llamedo, S.M. 2016. ECG-kit: a MATLAB toolbox for cardiovascular signal processing. *Journal of Open Research Software* 4(1).
- Erkaymaz, O., Ozer, M., and Perc, M. 2017. Performance of small-world feedforward neural networks for the diagnosis of diabetes. *Applied Mathematics and Computation* 311: 22–8.
- Farhidzadeh, H. 2015. Probabilistic neural network training for semi-supervised classifiers. arXiv preprint arXiv, 1509–01271.
- Haihua, X., Xianchuan, Y., Dan, H., and Sha, D. 2015. Sensitivity analysis of hierarchical hybrid fuzzy-neural network. *International Journal on Smart Sensing & Intelligent Systems* 8(3): 1837–54.
- Isa, S.M., Suryana, M.E., Akbar, M.A., Noviyanto, A., Jatmiko, W., and Arymurthy, A.M. 2013. Performance analysis of ECG signal compression using SPIHT. *International Journal on Smart Sensing & Intelligent Systems* 6 (5).
- Jain, A., Sharma, V., and Sharma, V. 2017. Big data mining using supervised machine learning approaches for Hadoop with Weka distribution. *International Journal of Computational Intelligence Research* 13(8): 2095–111.
- Kelwade, J.P., and Salankar, S.S. 2015. Prediction of cardiac arrhythmia using artificial neural network. *International Journal of Computer Applications* 115(20): 30–35.
- Khan, R.Z., and Jabbar, H. 2009. Training algorithms for supervised machine learning: comparative study. *International Journal of Management & Information Technology* 4(3): 354–60.
- Kiranyaz, S., Ince, T., and Gabbouj, M. 2015. Real-time patient-specific ECG classification by 1D convolutional neural networks. *IEEE Transactions on Biomedical Engineering* 63(3): 664–75.
- Kohli, N., Verma, N.K., and Roy, A. 2010. SVM based methods for arrhythmia classification in ECG. Proceeding of the International Conference on Computer and Communication Technology (ICCCT), India.
- Kotsiantis, S.B., Zaharakis, I., and Pintelas, P. 2007. Supervised machine learning: a review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering* 160: 3–24.
- Lassoued, H., and Ketata, R. 2017. Artificial neural network classifier for heartbeat arrhythmia detection. Proceeding of the International Conference on Automatic and Signal Processing (ATS), Engineering and Technology-PET, March 22–24, Sousse.
- Lassoued, H., and Ketata, R. 2018. ECG multi-class classification using neural network as machine learning model. Proceeding of the International Conference on Advanced Systems and Electric Technologies (IC_ASET), IEEE, March 19–22.
- Li, J., Li, X., Huang, B., and Zhao, L. 2016. Hopfield neural network approach for supervised nonlinear spectral unmixing. *IEEE Geoscience and Remote Sensing Letters* 13(7): 1002–6.
- Mao, K.Z., Tan, K.C., and Ser, W. 2000. Probabilistic neural-network structure determination for pattern classification. *IEEE Transactions on Neural Networks* 11(4): 1009–16.
- Martis, R.J., and Chakraborty, C.H. 2011. Arrhythmia disease diagnosis using neural network, SVM, and genetic algorithm-optimized k-means clustering. *Journal of Mechanics in Medicine and Biology* 11(4): 897–915.
- Matul Imah, E., Jatmiko, W., and Basaruddin, T. 2013. Electrocardiogram for biometrics by using Adaptive

Multilayer Generalized Learning Vector Quantization (AMGLVQ): integrating feature extraction and classification. *International Journal on Smart Sensing & Intelligent Systems* 6(5): 1891–1917.

Ozbay, Y., Ceylan, R., and Karlik, K. 2006. A Fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Computers in Biology and Medicine* 36(4): 376–88.

Ponomariov, V., Chirila, L., Apipie, F.M., Abate, R., Rusu, M., Wu, Z., Liehn, E.A., and Bucur, I. 2017. Artificial intelligence versus doctors' intelligence: a glance on machine learning benefaction in electrocardiography. *Discoveries* 5: 1–9.

Rafiq, M.Y., Bugmann, G., and Easterbrook, D.J. 2001. Neural network design for engineering applications. *Computers & Structures* 79(17): 1541–52.

Rai, H.M., Anurag, T., and Shailja, S. 2013. ECG signal processing for abnormalities detection using multi-resolution wavelet transform and Artificial Neural Network classifier. *Measurement* 46(9): 3238–46.

Rather, A.M., Agarwal, A., and Sastry, V.N. 2015. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications* 42: 3243–1.

Reilly, R.B., and Lee, T.C. 2010. Electrograms (ECG, EEG, EMG, EOG). *Technology and Health Care* 18: 443–58.

Savalia, S., Eder, A., and Vahid, E. 2017. Classification of cardiovascular disease using feature extraction and artificial neural networks. *Journal of Biosciences and Medicines* 5: 64–79.

Savic, M., Kurbalija, V., Ivanovic, M., and Bosnic, Z. 2017. A feature selection method based on feature correlation networks. Proceeding of the International Conference on Model and Data Engineering, Springer, Barcelona.

Seshagiri, S., and Khalil, H.K. 2000. Output feedback control of nonlinear systems using RBF neural networks. *IEEE Transactions on Neural Networks* 11(1): 69–79.

Silipo, R., and Marchesi, C. 1998. Artificial neural networks for automatic ECG analysis. *IEEE Transactions on Signal Processing* 46(5): 1417–25.

Silva, I., and Moody, G.B. 2014. An open-source toolbox for analysing and processing Physionet databases in Matlab and octave. *Journal of open research software* 2(1).

Sonawane, J.S., Patil, D.R., and Thakare, V.S. 2013. Survey on decision support system for heart disease. *International Journal of Advancements in Technology* 4(1): 89–96.

Tomar, M.S., Bandil, M.M. K., and Singh, M.D. 2013. Multi resolution analysis of ECG for arrhythmia using soft-computing Techniques. *Methodology* 3(5): 1663–8.

Yadav, A.R., Anand, R.S., and Dewal, A. M. 2014. Analysis and classification of hardwood species based on Coiflet DWT feature extraction and WEKA workbench. Proceeding of the International Conference on Signal Processing and Integrated Networks (SPIN), Noida India.

Zhao, N., and Li, Z. 2017. Viscosity prediction of different ethylene glycol/water based nanofluids using a RBF neural network. *Applied Science* 7(4), p. 409.