Expert Systems with Applications 37 (2010) 7563–7569

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

ECG beat classification using particle swarm optimization and radial basis function neural network

Mehmet Korürek, Berat Doğan *

Faculty of Electrical and Electronics Engineering, Department of Electronics and Communication Engineering, Istanbul Technical University, 34469 Istanbul, Turkey

ARTICLE INFO

Keywords:
ECG
Particle swarm optimization
Radial basis function neural networks
Arrhythmia classification

ABSTRACT

This paper presents a method for electrocardiogram (ECG) beat classification based on particle swarm optimization (PSO) and radial basis function neural network (RBFNN). Six types of beats including Normal Beat, Premature Ventricular Contraction (PVC), Fusion of Ventricular and Normal Beat (F), Atrial Premature Beat (A), Right Bundle Branch Block Beat (R) and Fusion of Paced and Normal Beat (f) are obtained from the MIT-BIH arrhythmia database. Four morphological features are extracted from each beat after the preprocessing of the selected records. For classification stage of the extracted features, a RBFNN structure which is evolved by particle swarm optimization is used. Several experiments are performed over the test set and it is observed that the proposed method classifies ECG beats with a smaller size of network without making any concessions on the classification performance.

1. Introduction

The electrocardiogram is a graphic record of the direction and magnitude of the electrical activity that is generated by the heart. It is the most easily accessible bioelectric signal that provides the doctors with reasonably accurate data regarding the patient heart condition (Ghongade & Ghatol, 2007). Thus, ECG is an effective tool for diagnosis of the heart abnormalities. On the other hand, it is a very time consuming job for doctors to analyze long ECG records. Therefore, many computer based methods have been proposed for automatically diagnosis of the ECG beat abnormalities. The main principle of such methods are based on pattern recognition techniques.

Neural networks are widely used and powerful tools for pattern recognition problems like ECG beat classification. Many methods have been proposed which are based on different neural network structures (Ceylan, Ozbay, & Karlik, 2009; Dokur & Olmez, 2001; Olmez, 1997; Prasad & Sahambi, 2003; Ubeyli, 2009).

A recently proposed neural network structure which is widely used to perform pattern recognition tasks due to its fast learning algorithms is radial basis function neural network (RBFNN). RBFNNs are nonlinear hybrid networks which are typically containing a single hidden layer of neurons. Fig. 1 shows the general structure of the RBFNN. The input layer broadcasts the coordinates of the input vector to each of the nodes in the hidden layer. Each node in the hidden layer then produces an activation based on the associated radial basis function. Finally, each node in the output layer computes a linear combination of the activations of the hidden nodes (Ou, Oyang, & Chen, 2005). The general mathematical form of the output nodes in an RBFNN is as follows:

$$c_j(x) = \sum_{i=1}^{k} \omega_{ji} \|x - \mu_i\|; \sigma_i \tag{1}$$

where $c_j(x)$ is the function corresponding to the jth output unit (class-j) and is a linear combination of k radial basis functions $\phi()$ with center μ_i and bandwidth σ_i. Also, ω_j is the weight vector of class-j and ω_{ji} is the weight corresponding to the jth class and ith center.

In pattern recognition problems usually a Gaussian function is used as the basis function of the RBFNN. So, the Eq. (1) becomes:

$$c_j(x) = \sum_{i=1}^{k} \omega_{ji} \exp\left(\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \tag{2}$$

From the Eq. (2) it can be clearly seen that the output of the RBFNN is dependent to the total number of neurons k, the weights between the output and the hidden layer of the network ω_{ji}, centers of the each neuron μ_i and finally bandwidth of the each neuron σ_i. So the classification performance of the RBFNN lie in determining the correct parameters for the network.

RBFNN can be trained with different methods. One class of the traditional training algorithms for the RBFNN starts with a predetermined network structure. Then the centers and widths of the RBFNN are trained (Qin, Chen, Lui, & Lu, 2005). Several methods have been proposed to find the centers of the RBFNN. These are usually clustering based methods that find center locations

* Corresponding author. Tel.: +90 0212 2853645.
E-mail addresses: korurek@itu.edu.tr (M. Korürek), bdogan@itu.edu.tr (B. Doğan).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.04.087

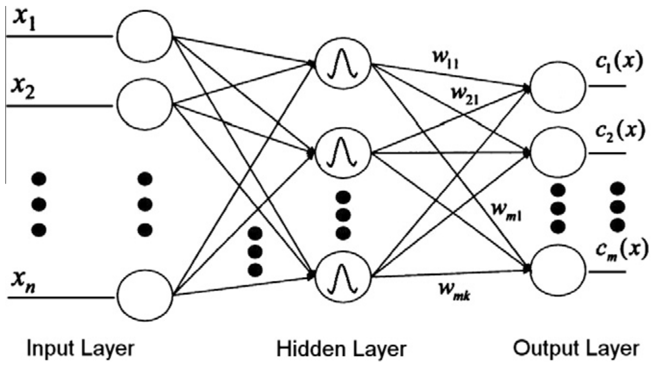

Fig. 1. General structure of the radial basis function neural network.

between the input feature vector locations or directly uses some of the input feature vectors as the centers of the neurons. However, it has been proven that the best center locations may not be necessarily located inside the input feature vectors (Man Wai & Kin Cho, 1998).

The most common algorithms to determine the neuron centers of the RBFNN are the K-Means algorithm and the Kohonen-SOM. Victor Markos et al. used K-Means, Fuzzy c-means and orthogonal least squares (OLS) algorithms to determine the neuron centers of the RBFNN for the diagnosis of the obstructive sleep apnoea syndrome (Victor Markos et al., 2008). Tu Song et al. used self organizing map (SOM) and incremental-SOM (ISOM) to determine the centers of the RBFNN for acoustic fault identification of underwater vehicles (Tu, Ben, Tian, & Zhang, 2008).

Although several studies have been performed, it is a relatively new method to classify ECG beats with RBFNN. Azemi et al. used independent component analysis (ICA) and wavelet transform for the classification of five types of ECG beats with multi-layer perceptron (MLP), RBFNN and k-NN classifier (Azemi et al., 2006). Ghongade and Ghatol used six morphological features for the classification of four types of ECG beats by using MLP, RBFNN and support vector machines (SVM) (Ghongade & Ghatol, 2008). Chudacek et al. extracted 13 morphological features to compare performance of seven methods including RBFNN, on ECG beat classification (Chudacek et al., 2007). In these studies RBFNN is trained with the traditional methods that have been mentioned before. In a more recent study, Melgani and Bazi used particle swarm optimization (PSO) to train support vector machines (SVM) for ECG beat classification (Melgani & Bazi, 2008). After classification phase they compared the proposed method with the SVM-linear, SVM-poly, SVM-rbf, k-NN, RBF, PCA-SVM, PCA-k-NN, and PCA-RBF classifiers. It is observed that the PSO–SVM combination has the best classification performance over nine types of classifiers.

Neural network and evolutionary computation methodologies have each been proven effective in solving certain classes of problems. There has been a significant increase in research and development in the area of applying evolutionary computation (EC) techniques for the purposes of evolving one or more aspects of artificial neural networks. These EC techniques have usually been used to evolve neural network weights, but sometimes have been used to evolve network structure or the network learning algorithms (Kennedy and Eberhart, 2001).

In this paper a recently proposed population based EC technique, particle swarm optimization (PSO), is used to evolve the RBFNN parameters (neuron centers and bandwidth of the each neuron).

2. Methodology

Fig. 2 shows the general block diagram of the proposed system. Six types of beats including, Normal Beat, Premature Ventricular Contraction, Fusion of Ventricular and Normal Beat, Artrial Premature Beat, Right Bundle Branch Block Beat and Fusion of Paced and Normal Beat are selected from the MIT-BIH arrhythmia database. After the preprocessing, R peaks are extracted from the selected records. Then, ECG beats are collected by selecting 128 points from each side of the R peaks. After the normalization process, four morphological features, including RR_t, RR_{t+1} intervals, $QRSh$ and $QRSwidth$ are extracted from the beats. Fig. 3 shows the extracted features. A training set which includes 50 feature vectors for each class is prepared for the training phase of the classification. Then the RBFNN evolved with the PSO algorithm by using the training set. The centers and the bandwidths of the each neuron are determined. Then, several experiments are performed on the test set and the proposed method is compared with the traditional RBFNN training algorithms, like K-Means, Kohonen-SOM. The method is also compared with the k-NN classifier.

2.1. Preprocessing and R peak detection

In the preprocessing stage, ECG signal is filtered and the baseline drift is removed. A low pass linear phase filter is used for noise removal (Korurek & Nizam, 2008). Filter characteristics is shown in the following Eqs. (3)–(7):

$$y(n) = F(b, a, x) \tag{3}$$

$$y(n) = b(1) * x(n) + \ldots + b(nb+1) * x(n-nb) - \ldots - a(na+1) * y(n-na) \tag{4}$$

$$B = [\,1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1\,] \tag{5}$$

$$A = [\,1 \quad -1\,] \tag{6}$$

Then, the z domain equation of the filter can be given with Eq. (7),

$$Y(z) = \frac{1 - z^{-6}}{1 - z^{-1}} X(z) \tag{7}$$

Once the noise is removed from the signal, the baseline correction is performed with a 500 samples wide median fitler.

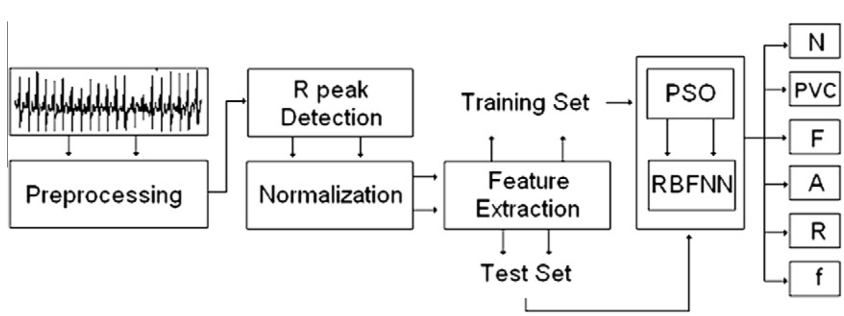

Fig. 2. Block diagram of the proposed system.

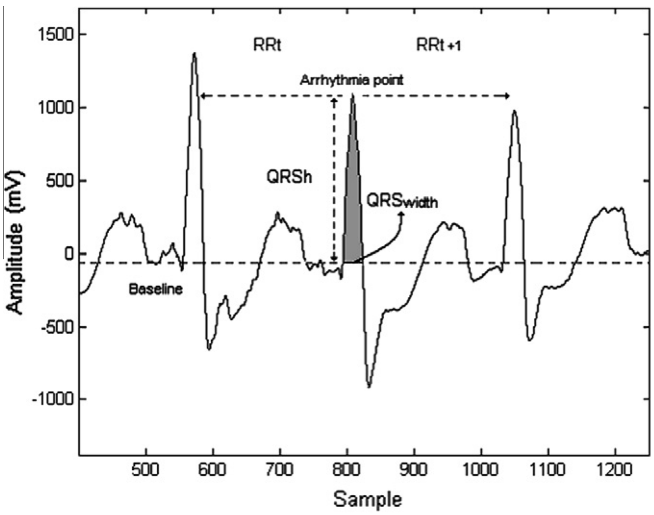

Fig. 3. Extracted morphological features for classification.

After the noise removal and the baseline correction the Pan–Tompkins algorithm (Pan & Tompkins, 1985) is used to detect the R peaks of the ECG beats from the selected records.

2.2. Normalization of the ECG signals

During the normalization stage, feature values of the eight normal ECG beats in the record are used to normalize the selected beat's features. The equations used in the normalization stage are as follows (Korurek & Nizam, 2008):

$$QRSx(t) = \frac{\sum_{i=t-7}^{t} QRSx(i)}{8}; \quad QRSx(t) \in \text{'Normal ECG'} \tag{8}$$

$$RRx(t) = \frac{\sum_{i=t-7}^{t} RRx(i)}{8}; \quad RRx(t) \in \text{'Normal ECG'} \tag{9}$$

where, x represents the feature to be normalized. After the normalization the training and test sets are formed. Fig. 4 shows the distribution of the feature vectors within the training set.

3. Particle swarm optimization

Particle swarm optimization was first introduced by Kennedy and Eberhart (1995). The PSO is a stochastic search through the n-dimensional space of the real numbers.

In PSO, each particle in the swarm represents a point in the solution space. The particles move around the space to find the optimum solution while taking into consideration the best solution (point) visited by the individual and by the whole swarm.

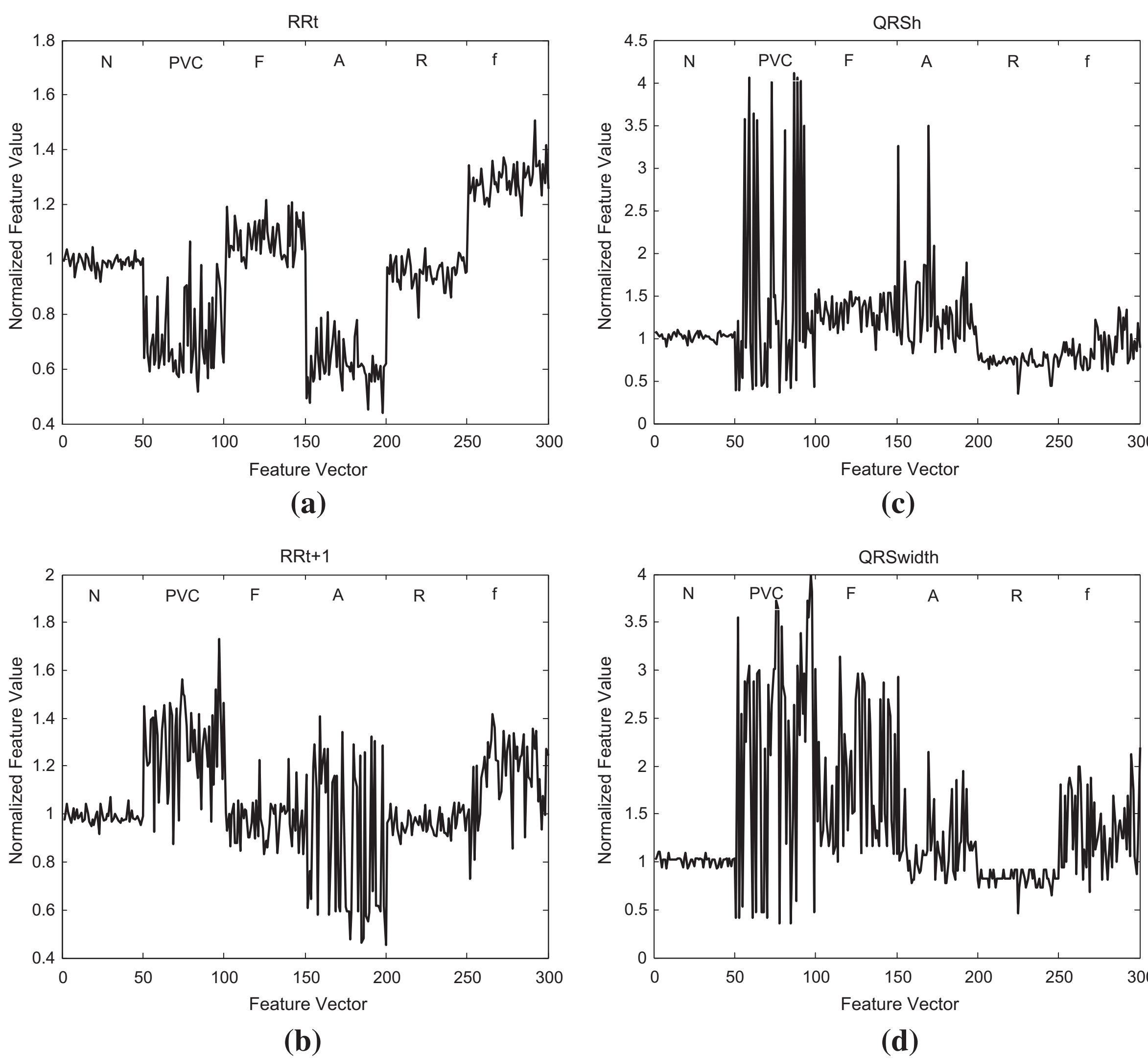

Fig. 4. Distribution of the feature vectors within the training set (a) RR_t, (b) RR_{t+1}, (c) $QRSh$, (d) $QRSwidth$.

The main operators of the PSO algorithm are the velocity and the position of the each particle. In each iteration particles evaluate their positions according to a fitness function. Then the velocity and the position of the each particle are updated according to Eqs. (10) and (11).

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i(t) - x_i(t)) + c_2 r_2 (p_g(t) - x_i(t)) \tag{10}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{11}$$

where, $v_i(t)$ and $v_i(t+1)$ represents the current and next velocity, which controls the movement direction and magnitude of the ith particle and $x_i(t)$ and $x_i(t+1)$ represents the current and next position of the ith particle. ω, inertia weight is a coefficient which controls exploration degree of the search, and $0 < c_1$, $c_2 \leqslant 2$ are the acceleration coefficients which controls the divergence of each particle for each iteration. r_1 and r_2 are uniform random variables which allow stochastic search of the algorithm. Finally, p_i and p_g represent the personal best position for the ith particle and for the whole swarm, respectively.

Since the changes in the velocity are stochastic, a particle can diverge from the solution space. So, a method is implemented to limit the velocity. At each iteration, after the velocity of the ith particle is updated, if the velocity is greater or smaller than from a given $[-v_{\max}, v_{\max}]$ interval, it is limited to $-v_{\max}$ or $v_{\max}$. This prevents the particle to diverge from the solution space. If the solution space boundary can be predicted, the $v_{\max}$ value can be choosen as $v_{\max} = k \times x_{\max}$, $0.1 \leqslant k \leqslant 1.0$ (Corne, Darigo, & Glover, 1999, chap. 25).

Another important parameter, which balances the global search and the local search ability of the algorithm is the ω, inertia weight. For large values of the inertia weight, the global search ability of the algorithm increases. Nevertheless, once the algorithm converges to the optimum solution, it can be considered as a disadvantage to select a large value for the inertia weight. For this reason, the methods which offer to adjust the inertia weight adaptively have been proposed (Shi & Eberhart, 1998, 2001). Since the classification performance is met, in this paper a constant inertia weight is used.

3.1. RBFNN as particles of the PSO

The RBFNN architecture mainly has three parameters: the centers of the neurons, the bandwidth of the each neuron, and the weights between the hidden layer and the output layer of the each network. Once the centers and the bandwidths of the each neuron are determined the weights can be calculated by the pseudo–inverse method. So, in this study only the centers and the bandwidths of a single RBFNN thought as a particle in the solution space. Let *net* be a vector of parameters, than *net* can be represented as,

$$net = [(c_1, \sigma_1), (c_2, \sigma_2), \ldots, (c_n, \sigma_n)] \tag{12}$$

where, c and σ are the centers and the bandwidths of the neurons, respectively and n is the maximum neuron number. For a given set of particles (*nets*), PSO evaluates the fitness of the each particle at each iteration and searches for the optimum network. The fitness of the network is formulated by Eq. (13) (Qin et al., 2005):

$$E = N_t \times \log\left(\frac{1}{N_t}\sum_{i=1}^{N_t} \|y_i - o_i\|^2\right) \tag{13}$$

where N_t is the number of training feature vectors and y_i, o_i are the desired output and the network output for the feature vector i, respectively.

Zheng Qin et al. evolved RBFNN with a similar manner and tested the network for different types of pattern recognition problems (Qin et al., 2005). From this study, it can be clearly seen that this method classifies the given patterns with a relatively small network size without making any concessions on classification performance. This exactly meets the term optimization which aims maximum quality at minimum cost.

The algorithm which is used in this study to evolve RBFNN is as follows (Qin et al., 2005):

1. Initialize swarm of N particles. Each particle defines a network and the associated centers and bandwidths. Set the number of iterations as *MaxIteration*. Set *count* = 0.
2. Decode each particle into a network. Compute the connection weights between the hidden layer and the output of the network by the pseudo–inverse method. Compute the fitness of each particle.
3. Update p_i for each particle and p_g for whole swarm.
4. Update the velocity of each particle according to Eq. (10). Limit the velocity in $[-v_{\max}, v_{\max}]$.
5. Update the position according to the Eq. (11).
6. Set *count* = *count* + 1; if *count* < *MaxIterations*, go to step 2, otherwise terminate the algorithm.

Table 1 lists the parameters which are choosen for PSO algorithm to evolve the RBFNN for ECG beat classification.

Table 1
PSO algorithm parameters.

ω (inertia coefficient)	0.7298
c_1, c_2	1.496
$[-v_{\max}, v_{\max}]$	[−0.1, 0.1]
Swarm size	20
MaxIteration	500

4. Results and discussions

For the given training set the algorithm is initialized and optimum RBFNN structure is found. Ten experiments are performed over the test set (Table 2) with different number of neurons and the results are compared with the K-Means algorithm. Number of clusters are set as the number of neuron centers in K-Means algorithm in order to compare two methods. The comparison of two methods can be seen in Fig. 5. In this figure it can be clearly seen that the proposed method classifies the ECG beats with a smaller size of network without making any concessions on classification performance. The K-Means algorithm needs about 25 cluster centers to reach the classification performance of the proposed method, which only needs 10 neurons. Table 3 lists the classification result of the proposed method for an experiment with 10 neurons.

Another important point is that, the proposed method can classify the ECG beats with an appreciable performance even with only 6 neurons which equals to the total class number. It is thought that, the proposed method achieves this due to its variable neuron bandwidths in addition to the ability of finding optimum neuron

Table 2
Test set.

Beat type	Number of beats	Records
Normal Beat (N)	836	113, 119, 213, 234
Premature Ventricular Contraction (PVC)	605	119, 208, 221, 223, 228, 233
Fusion of Ventricular and Normal Beat (F)	248	208, 213
Atrial Premature Beat (A)	165	209, 220, 222
Rigth Bundle Branch Block Beat (R)	150	212
Fusion of Paced and Normal Beat (f)	130	217

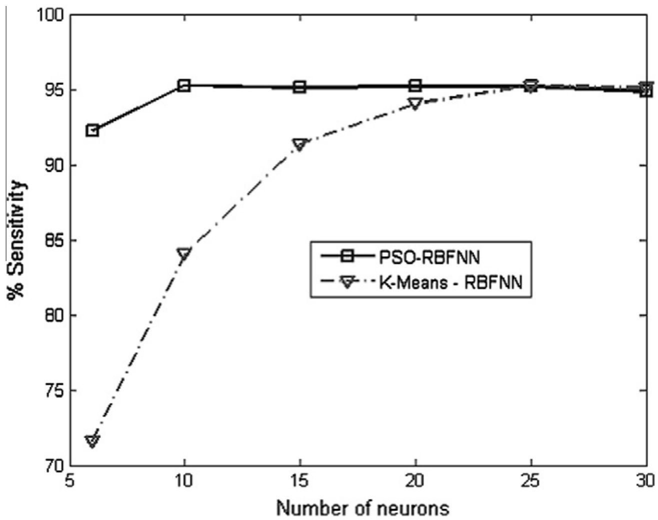

Fig. 5. Comparison of the PSO–RBFNN and K-Means–RBFNN.

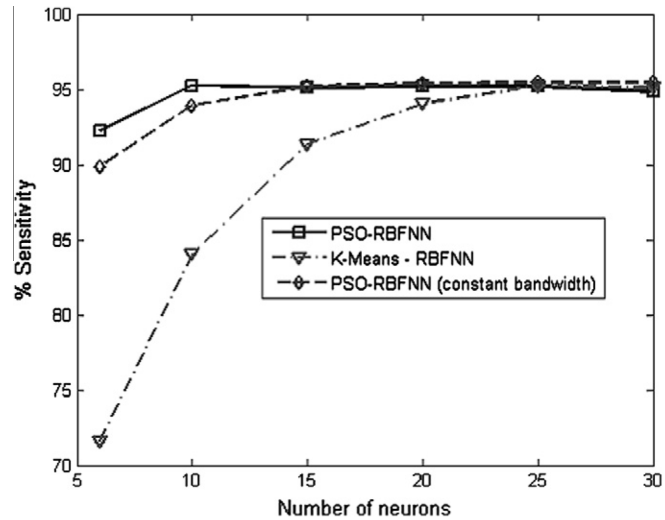

Fig. 6. Comparison of the constant bandwidth PSO–RBFNN with K-Means–RBFNN.

centers. The traditional RBFNN training algorithms generally use a constant bandwidth value for all neurons, which can be given in Eq. (14).

$$\sigma = \frac{d_{\max}}{\sqrt{2k}} \tag{14}$$

where, σ is the bandwidth of a single neuron, $d_{\max}$ is the maximum distance between the two neuron centers and the k is the total neuron number. For the constant bandwidth case only centers of the neurons are determined. So, the Eq. (12) becomes,

$$net = [c_1, c_2, c_3, \ldots, c_n] \tag{15}$$

Since the distribution of the clusters cannot be predicted, it is not logical to use a constant value for the neuron bandwidths, if the neuron number is not enough. To prove this proposal, another experiment is performed with the same training set. The centers of the neurons are again found by the PSO while bandwidths are accepted constant for each neuron and calculated at each iteration by Eq. (14). Note that, the bandwidths vary over time so here the term constant means, the bandwidths are same for each neuron at each iteration and similarly the term variable means, the bandwidths are different for each neuron at each iteration. Then, the results are compared for the constant bandwidth and the variable bandwidth cases.

As proposed, from the Fig. 6 it can be seen that, if the neuron number is not enough and the bandwidths of the neurons are constant, the classification performance of the proposed method decreases.

Although, classification performance of the constant bandwidth PSO–RBFNN decreases for six neurons, it is still better than from the K-Means RBFNN. This lies behind the PSO algorithm's behaviour, which can locate the neuron centers anywhere in the solution space.

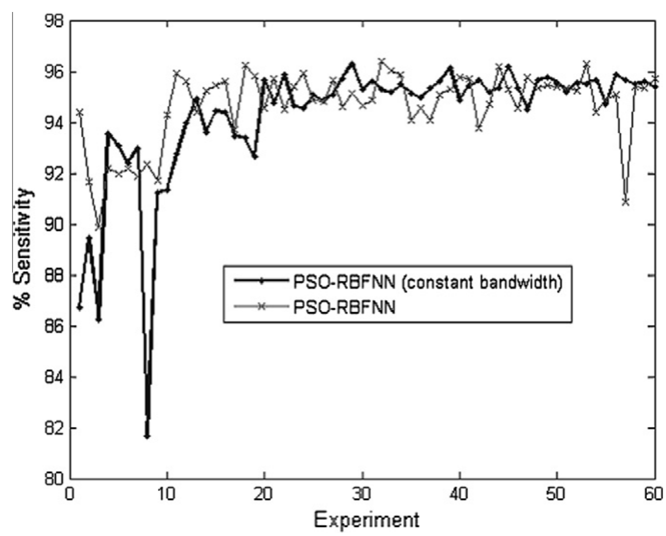

Fig. 7. Comparison of the constant bandwidth case and variable bandwidth case for PSO–RBFNN.

After 15 neurons, the classification performance increases when compared with the variable bandwidth PSO–RBFNN. Fig. 7 shows the detailed classification results for the constant and variable bandwidth cases. Note that, these are the results for the 6, 10, 15, 20, 25 and 30 neuron number and for each neuron number 10 experiment is performed. From this figure it can be seen that, as neuron number increases the constant bandwith PSO–RBFNN

Table 3
Beat-by-beat result of an experiment for 10 neurons with variable bandwidth PSO–RBFNN.

Beat type	N	PVC	F	A	R	f	Sensitivity (%)	Specificity (%)
N	796	0	27	2	11	0	95.215	98.690
PVC	0	577	17	10	1	0	95.371	99.738
F	11	2	232	0	2	1	93.548	97.613
A	3	0	0	161	0	1	97.575	99.390
R	3	1	0	0	146	0	97.333	99.294
f	0	1	1	0	0	128	98.461	99.900
	Average						96.251	99.104

Table 4
Average sensitivity and specificity results for 10 experiments with 10 neurons variable bandwidth PSO–RBFNN.

	N				PVC				F				A				R				f				Se[a] (%)	Sp[b] (%)
	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN		
1	809	27	18	1280	571	34	2	1527	233	15	33	1853	160	5	25	1944	145	5	12	1972	126	4	1	2003	95.94	99.15
2	784	52	17	1281	569	36	5	1524	232	16	68	1818	160	5	11	1958	145	5	14	1970	128	2	1	2003	95.58	98.91
3	770	66	27	1271	572	33	5	1524	232	16	76	1810	150	15	14	1955	145	5	14	1970	128	2	1	2003	94.34	98.71
4	789	47	18	1280	569	36	6	1523	231	17	50	1836	159	6	25	1944	147	3	14	1970	124	6	2	2002	95.22	98.92
5	794	42	18	1280	569	36	2	1527	233	15	63	1823	160	5	12	1957	146	4	12	1972	124	6	1	2003	95.44	98.98
6	797	39	19	1279	568	37	2	1527	233	15	32	1854	160	5	25	1944	146	4	14	1970	125	5	3	2001	95.60	99.10
7	802	34	20	1278	553	52	1	1528	234	14	80	1806	155	10	10	1959	146	4	17	1967	116	14	0	2004	93.70	99.41
8	796	40	17	1281	577	28	4	1525	232	16	45	1841	161	4	12	1957	146	4	14	1970	128	2	2	2002	96.25	99.11
9	779	57	15	1283	576	29	5	1525	234	14	58	1828	159	6	20	1949	147	3	14	1970	127	3	1	2003	95.80	98.94
10	762	74	15	1283	567	38	2	1527	234	14	97	1789	159	6	13	1956	148	2	14	1970	121	9	2	2002	94.55	98.66
Average																									95.24	98.99

[a] Se: sensitivity.
[b] Sp: specificity.

Table 5
Comparison of different classifiers.

Classifier	# of neurons	Sensitivity (%)	Learning time (s)	Classification time (s)
Kohonen–RBFNN	64	93.565	7.940	2.6208
K-Means–RBFNN	25	95.232	0.346	0.4987
PSO–RBFNN	10	95.246	79.768	0.2250
PSO–RBFNN (const. σ)	30	95.469	251.070	0.5847
k-NN	$k = 5$	94.390	–	11.6271
k-NN	$k = 3$	94.989	–	9.9267

gives more stable results per experiment, when compared with the variable case. So, the constant case can be prefered due to its stability and classification performance.

Table 4 lists the average sensitivity and specificty results of 10 experiments which is performed with variable bandwidth PSO–RBFNN with 10 neurons. Sensitivity and specificty can be calculated with the Eqs. (16) and (17), respectively.

$$\text{Sensitivity} = \frac{\text{Number of True Positives}}{\text{Number of True Positives} + \text{Number of False Negatives}} \times 100 \quad (16)$$

$$\text{Specificity} = \frac{\text{Number of True Negatives}}{\text{Number of True Negatives} + \text{Number of False Positives}} \times 100 \quad (17)$$

Table 5 lists the results for the Kohonen–RBFNN and k-NN classifier with the variable bandwidth PSO–RBFNN, constant bandwidth PSO–RBFNN and K-Means–RBFNN classifiers. Here, the highest classification performances are selected for comparison. Except the k-NN classifier, all of the results are average results of 10 experiments.

Note that the proposed method can clasify the whole training set within 0.2250 s which is approximately half of the K-Means–RBFNN's classification time. Experiments are made with an Intel (R) Pentium (R) M 1.86 GHz processor and 2 GB RAM computer. Algorithms are built in MATLAB (R) 7.1.

The proposed method gives the best classification performance with the smallest size of network. By means of additional feature extraction methods the performance of the proposed method can be increased.

References

Azemi, A., Sabzevari, V. R., Khademi, M., Gholizade, H., Kiani, A., Dastgheib, Z. S., (2006). Intelligent arrhythmia detection and classification using ICA. *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2163–2166. doi: 10.1109/IEMBS.2006.259292.

Ceylan, R., Ozbay, Y., & Karlik, B. (2009). A novel approach for classification of ECG arrhythmias: Type-2 fuzzy clustering neural network. *Expert Systems with Applications, 36*(3), 6721–6726.

Chudacek, V., Petrik, M., Georgoulas, G., Cepek, M., Lhotska, L., & Stylios, C., (2007). Comparison of seven approaches for holter ECG clustering and classification. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 3844–3847. doi: 10.1109/IEMBS.2007.4353171.

Corne, D., Darigo, M., & Glover, F. (Eds.). (1999). *New ideas in optimization*. McGraw Hill (pp. 379–387).

Dokur, Z., & Olmez, T. (2001). ECG beat classification by o novel hybrid neural network. *Computer Methods & Programs in Biomedicine, 66*, 167–181.

Ghongade, R., & Ghatol, A. A., (2007). A brief performance evaluation of ECG feature extraction techniques for artificial neural network based classification. *TENCON 2007–2007 IEEE Region 10 Conference*, 1–4. doi: 10.1109/TENCON.2007.4429096.

Ghongade, R., & Ghatol, A., (2008). A robust and reliable ECG pattern classification using QRS morphological features and ANN. *TENCON 2008–2008 IEEE Region 10 Conference*, 1–6. doi: 10.1109/TENCON.2008.4766722.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks* (pp. 1942–1948). NJ: Piscataway.

Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. 1-55860-595-9. Academic Press.

Korurek, M., & Nizam, A. (2008). A new arrhythmia clustering technique based on ant colony optimization. *Journal of Biomedical Informatics, 41*(6), 874–881.

Man Wai Mak, & Kin Wai Cho (1998). Genetic evolution of radial basis function centers for pattern classification. In *Neural networks proceedings, 1998. IEEE world congress on computational intelligence. The 1998 IEEE international joint conference on* (Vol. 1, pp. 669–673).

Melgani, F., & Bazi, Y. (2008). Classification of electrocardiogram signals with support vector machines and particle swarm optimization. *Information Technology in Biomedicine IEEE Transactions on, 12*(5), 667–677.

Olmez, T. (1997). Classification of ECG waveforms by using RCE neural network and genetic algorithms. *Electronics Letters, 33*(18), 1561–1562.

Ou Yu-Yen, Oyang Yen-Jen, & Chen Chien-Yu (2005). A novel radial basis function network classifier with centers set by hierarchical clustering. In *Neural networks, 2005 IJCNN '05. Proceedings 2005 IEEE international joint conference on* (Vol. 3, pp. 1383–1388).

Pan, Jiapu, & Tompkins, Willis J. (1985). A real-time QRS detection algorithm. *Biomedical Engineering IEEE Transactions on, BME-32*(3), 230–236.

Prasad, G. K., & Sahambi, J. S. (2003). Classification of ECG arrhythmias using multi-resolution analysis and neural networks. In *TENCON 2003. Conference on convergent technologies for Asia-Pacific region* (Vol. 1, pp. 227–231).

Qin, Zheng, Chen, Junying, Lui, Yu, & Lu, Jiang (2005). Evolving RBF neural networks for pattern classification. *Lecture Notes in Computer Science, 3801*.

Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *IEEE international conference of evolutionary computation*. Anchorage, Alaska.

Shi,Y., & Eberhart, R. C. (2001). Particle swarm optimization with fuzzy adaptive inertia weight. In *Proceedings of the workshop on particle swarm optimization*. Indianapolis, IN, USA: Purdue School of Engineering and Technology, IUPUI.

Tu Song, Ben Kerong, Tian Liye, & Zhang Linke (2008). Combination of SOM and RBF based on incremental learning for acoustic fault identification of underwater vehicles. In *Image and signal processing. CISP'08. Congress on* (Vol. 4, pp. 38–42).

Ubeyli, E. D. (2009). Combining recurrent neural networks with eigenvector methods for classification of ECG beats. *Digital Signal Processing, 19*(2), 320–329. March.

Victor Markos, J., Hornero, Roberto, Alvarez, Daniel, Campo, Felix Del, Lopez, Miguel, & Zamarron, Carlos (2008). Radial basis function classifiers to help in the diagnosis of the obstructive sleep apnoea syndrome from nocturnal oximetry. *Medical and Biological Engineering and Computing, 46*(4).