Available online at www.sciencedirect.com
SCIENCE DIRECT
Engineering Applications of Artificial Intelligence 17 (2004) 567–576
Engineering Applications of ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Detection of electrocardiographic changes in partial epileptic patients using Lyapunov exponents with multilayer perceptron neural networks

Elif Derya Übeyli[a], İnan Güler[b,*]

[a]Department of Electrical and Electronics Engineering, Faculty of Engineering, TOBB Ekonomi ve Teknoloji Üniversitesi, Söğütözü, Ankara, Turkey
[b]Department of Electronics and Computer Education, Faculty of Technical Education, Gazi University, 06500 Teknikokullar, Ankara, Turkey

Received 14 October 2003; received in revised form 19 April 2004; accepted 16 July 2004
Available online 11 September 2004

Abstract

In this study, a new approach based on the consideration that electrocardiogram (ECG) signals are chaotic signals was presented for detection of electrocardiographic changes in patients with partial epilepsy. This consideration was tested successfully using the nonlinear dynamics tools, like the computation of Lyapunov exponents. Multilayer perceptron neural network (MLPNN) architectures were formulated and used as basis for detection of electrocardiographic changes in patients with partial epilepsy. Two types of ECG beats (normal and partial epilepsy) were obtained from the MIT-BIH database. The computed Lyapunov exponents of the ECG signals were used as inputs of the MLPNNs trained with backpropagation, delta-bar-delta, extended delta-bar-delta, quick propagation, and Levenberg–Marquardt algorithms. The performances of the MLPNN classifiers were evaluated in terms of training performance and classification accuracies. Receiver operating characteristic (ROC) curves were used to assess the performance of the detection process. The results confirmed that the proposed MLPNN trained with the Levenberg–Marquardt algorithm has potential in detecting the electrocardiographic changes in patients with partial epilepsy.

Keywords: Electrocardiographic changes; Partial epilepsy; Chaotic signal; Lyapunov exponents; Multilayer perceptron neural network (MLPNN); Training algorithms; Levenberg–Marquardt algorithm

1. Introduction

The electrocardiogram (ECG) signal is the recording of the bioelectrical and biomechanical activities of the cardiac system. It provides valuable information about the functional aspects of the heart and cardiovascular system (Saxena et al., 2002; Foo et al., 2002; Maglaveras et al., 1998). Epileptic seizures are associated with several changes in autonomic functions, which may lead to cardiovascular, respiratory, gastrointestinal, cutaneous, and urinary manifestations. Cardiovascular changes have received the most attention, because of their possible contribution to sudden unexplained death in epilepsy patients. The ECG should be reviewed for high-risk cardiac abnormalities during epileptic seizures. A change in heart rate can be used as an extra clinical sign and can be very informative with respect to the first manifestation of the epileptic discharge (Leutmezer et al., 2003; Zijlmans et al., 2002; Tomson et al., 1998; Opherk et al., 2002; Rocamora et al., 2003).

Conventional methods of monitoring and diagnosing electrocardiographic changes rely on detecting the presence of particular signal features by a human observer. Due to large number of patients in intensive care units and the need for continuous observation of

*Corresponding author. Tel.: 90-312-212-3976; fax: +90-312-212-0059.
E-mail address: iguler@gazi.edu.tr (İ. Güler).

0952-1976/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2004.07.002

such conditions, several techniques for automated electrocardiographic changes detection have been developed in the past 10 years to attempt to solve this problem. Such techniques work by transforming the mostly qualitative diagnostic criteria into a more objective quantitative signal feature classification problem. The techniques have been used to address this problem such as the analysis of ECG signals for detection of electrocardiographic changes using the autocorrelation function, frequency domain features, time frequency analysis, and wavelet transform (Saxena et al., 2002; Foo et al., 2002; Maglaveras et al., 1998, Sternickel, 2002; Kundu et al., 2000; Nugent et al., 1999; Simon and Eswaran, 1997; Addison et al., 2000). Even though fairly good results have been obtained using such techniques, they seem to provide only a limited amount of information about the signal because they ignore the underlying nonlinear signal dynamics (Owis et al., 2002). In recent years, there has been an increasing interest in applying techniques from the domains of nonlinear analysis and chaos theory in studying the behavior of a dynamical system from an experimental time series such as ECG signals (Owis et al., 2002; Casaleggio and Braiotta, 1997; Silipo et al., 1998; Govindan et al., 1998; Fell et al., 2000). The purpose of these studies is to determine whether dynamical measures especially Lyapunov exponents can serve as clinically useful parameters. However, these approaches require a signal with very high resolution and very low noise level to be able to use the small scale changes in these parameters that would take place as a result of variabilities in the signals.

The present study was conducted with the purpose of answering the question of whether nonlinear measures such as Lyapunov exponents are capable of discriminating ECG signals belonging to the normal and the partial epilepsy subjects. The computation of Lyapunov exponents was the basis for the automatic detection of electrocardiographic changes in patients with partial epilepsy. More specifically, the ECG signals were modeled using multilayer perceptron neural networks (MLPNNs). The computed Lyapunov exponents defining the behavior of the ECG signals were used as inputs of the MLPNNs. The MLPNNs presented in this study were trained, cross validated and tested with the computed Lyapunov exponents of the ECG signals obtained from normal subjects and subjects suffering from partial epilepsy. The presented MLPNNs were trained with backpropagation (BP), delta-bar-delta (DBD), extended delta-bar-delta (EDBD), quick propagation (QP) algorithms. In order to improve convergence rate, the MLPNN trained with the Levenberg–Marquardt algorithm. The correct classification rates and convergence rate of the neural network model trained with the Levenberg–Marquardt algorithm presented in this study were found to be high.

2. Materials and method

Decision making was performed in two stages: feature extraction by computing Lyapunov exponents (128 Lyapunov exponents selected as neural network inputs) and classification using the MLPNNs trained with the BP, DBD, EDBD, QP, and Levenberg–Marquardt algorithms. The ECG signals from the MIT-BIH database (MIT-BIH Database, 2003) were used to train and test the proposed MLPNNs. A rectangular window, which was formed by 256 discrete data, was selected so that it contained a single ECG beat. For two classes (normal and partial epilepsy) training and test sets were formed by 360 vectors (180 vectors from each class) of 128 dimensions (Lyapunov exponents).

2.1. Lyapunov exponents

Lyapunov exponents are a quantitative measure for distinguishing among the various types of orbits based upon their sensitive dependence on the initial conditions, and are used to determine the stability of any steady-state behavior, including chaotic solutions. The reason why chaotic systems show aperiodic dynamics is that phase space trajectories that have nearly identical initial states will separate from each other at an exponentially increasing rate captured by the so-called Lyapunov exponent (Haykin and Li, 1995; Casdagli, 1989; Eckmann et al., 1986; Abarbanel et al., 1991). This is defined as follows. Consider two (usually the nearest) neighboring points in phase space at time 0 and at time t, distances of the points in the ith direction being $\delta x_i(0)$ and $\delta x_i(t)$, respectively. The Lyapunov exponent is then defined by the average growth rate λ_i of the initial distance

$$\frac{\|\delta x_i(t)\|}{\|\delta x_i(0)\|} = 2^{\lambda_i t}(t \to \infty) \quad \text{or}$$

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \log_2 \frac{\|\delta x_i(t)\|}{\|\delta x_i(0)\|}. \tag{1}$$

The existence of a positive Lyapunov exponent indicates chaos (Haykin and Li, 1995; Casdagli, 1989; Eckmann et al., 1986; Abarbanel et al., 1991). This shows that any neighboring points with infinitesimal differences at the initial state abruptly separate from each other in the ith direction. In other words, even if the initial states are close, the final states are much different. This phenomenon is sometimes called sensitive dependence on initial conditions. Numerous methods for calculating the Lyapunov exponents have been developed during the past decade (Abarbanel et al., 1993). Generally, the Lyapunov exponents can be estimated either from the equations of motion of the dynamic system (if it is known) (Benettin et al., 1980), or from the observed time series (Packard et al., 1980). The latter is what is of interest due to its direct relation to the work in this paper. The idea is based on the well-known technique of state space reconstruction with delay coordinates to build a system with Lyapunov exponents identical to that of the original system from which our measurements have been observed. Generally, Lyapunov exponents can be extracted from observed signals in two different ways. The first is based on the idea of following the time-evolution of nearby points in the state space (Wolf et al., 1985). This method provides an estimation of the largest Lyapunov exponent only. The second method is based on the estimation of local Jacobi matrices (Sano and Sawada, 1985) and is capable of estimating all the Lyapunov exponents. Vectors of all the Lyapunov exponents for particular systems are often called their Lyapunov spectra.

2.2. *Artificial neural networks*

Artificial neural networks (ANNs) may be defined as structures comprised of densely interconnected adaptive simple processing elements (neurons) that are capable of performing massively parallel computations for data processing and knowledge representation. ANNs can be trained to recognize patterns and the nonlinear models developed during training allow neural networks to generalize their conclusions and to make application to patterns not previously encountered (Haykin, 1994; Basheer and Hajmeer, 2000). The MLPNNs, which have features such as the ability to learn and generalize, smaller training set requirements, fast operation, ease of implementation and therefore most commonly used neural network arhitectures, have been adapted for detection of electrocardiographic changes in patients with partial epilepsy. As shown in Fig. 1, a MLPNN consists of (i) an input layer with neurons representing input variables to the problem, (ii) an output layer with neurons representing the dependent variables (what is being modeled), and (iii) one or more hidden layers containing neurons to help capture the nonlinearity in the data.

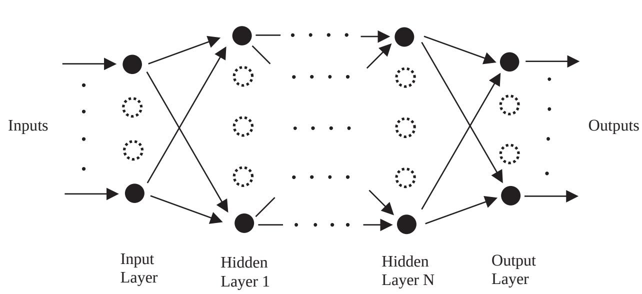

Fig. 1. MLPNN topology.

2.2.1. *Multilayer perceptron neural networks*

Presently the most widely used ANN type is a MLPNN which has been playing a central role in applications of neural networks. The MLPNN is a nonparametric technique for performing a wide variety of detection and estimation tasks (Haykin, 1994; Basheer and Hajmeer, 2000; Chaudhuri and Bhattacharya, 2000). In the MLPNN, each neuron j in the hidden layer sums its input signals x_i after multiplying them by the strengths of the respective connection weights w_{ji} and computes its output y_j as a function of the sum:

$$y_j = f\left(\sum w_{ji} x_i\right), \quad (2)$$

where f is activation function that is necessary to transform the weighted sum of all signals impinging onto a neuron. The activation function (f) can be a simple threshold function, or a sigmoidal, hyperbolic tangent, or radial basis function.

The sum of squared differences between the desired and actual values of the output neurons E is defined as

$$E = \frac{1}{2} \sum_j (y_{dj} - y_j)^2, \quad (3)$$

where y_{dj} is the desired value of output neuron j and y_j is the actual output of that neuron. Each weight w_{ji} is adjusted to reduce E as rapidly as possible. How w_{ji} is adjusted depends on the training algorithm adopted (Haykin, 1994; Basheer and Hajmeer, 2000; Chaudhuri and Bhattacharya, 2000).

Training algorithms are an integral part of ANN model development. An appropriate topology may still fail to give a better model, unless trained by a suitable training algorithm. A good training algorithm will shorten the training time, while achieving a better accuracy. Therefore, training process is an important characteristic of the ANNs, whereby representative examples of the knowledge are iteratively presented to the network, so that it can integrate this knowledge within its structure. There are a number of training algorithms used to train a MLPNN and a frequently used one is called the BP training algorithm (Haykin, 1994; Basheer and Hajmeer, 2000; Chaudhuri and Bhattacharya, 2000). The BP algorithm, which is based on searching an error surface using gradient descent for points with minimum error, is relatively easy to implement. However, the BP algorithm has some problems for many applications. The algorithm is not guaranteed to find the global minimum of the error function since gradient descent may get stuck in local minima, where it may remain indefinitely. In addition to this, long training sessions are often required in order to find an acceptable weight solution because of the well-known difficulties inherent in gradient descent optimization. Therefore, a lot of variations to improve the convergence of the BP were proposed such as DBD, EDBD, QP (Rumelhart et al., 1986; Jacobs, 1988; Minai and Williams, 1990; Fahlman, 1988). Optimization methods such as second-order methods (conjugate gradient, quasi-Newton, Levenberg–Marquardt) have also been used for ANN training in recent years. The Levenberg–Marquardt algorithm combines the best features of the Gauss–Newton technique and the steepest-descent algorithm, but avoids many of their limitations. In particular, it generally does not suffer from the problem of slow convergence (Hagan and Menhaj, 1994; Battiti, 1992). A number of researchers have carried out comparative studies of MLPNN training algorithms (Chan, 1990; Sidani and Sidani, 1994; Hannan and Bishop, 1997). The results of the studies have illustrated that the relative performance of algorithms depends on the problem being used. Therefore, in this study the MLPNNs were trained with the BP, DBD, EDBD, QP, and Levenberg–Marquardt algorithms.

2.2.2. Levenberg–Marquardt algorithm

ANN training is usually formulated as a nonlinear least-squares problem. Essentially, the Levenberg–Marquardt algorithm is a least-squares estimation algorithm based on the maximum neighborhood idea. Let $E(\mathbf{w})$ be an objective error function made up of m individual error terms $e_i^2(\mathbf{w})$ as follows:

$$E(\mathbf{w}) = \sum_{i=1}^{m} e_i^2(\mathbf{w}) = \|f(\mathbf{w})\|^2, \tag{4}$$

where $e_i^2(\mathbf{w}) = (\mathbf{y}_{di} - \mathbf{y}_i)^2$ and $\mathbf{y}_{di}$ is the desired value of output neuron i, $\mathbf{y}_i$ is the actual output of that neuron.

It is assumed that function $f(\cdot)$ and its Jacobian J are known at point $\mathbf{w}$. The aim of the Levenberg–Marquardt algorithm is to compute the weight vector $\mathbf{w}$ such that $E(\mathbf{w})$ is minimum. Using the Levenberg–Marquardt algorithm, a new weight vector $\mathbf{w}_{k+1}$ can be obtained from the previous weight vector $\mathbf{w}_k$ as follows:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \delta\mathbf{w}_k, \tag{5}$$

where $\delta\mathbf{w}_k$ is defined as

$$\delta\mathbf{w}_k = -(J_k^{\mathrm{T}} f(\mathbf{w}_k))(J_k^{\mathrm{T}} J_k + \gamma\mathbf{I})^{-1}. \tag{6}$$

In Eq. (6), J_k is the Jacobian of f evaluated at $\mathbf{w}_k$, γ is the Marquardt parameter, $\mathbf{I}$ is the identity matrix (Hagan and Menhaj, 1994; Battiti, 1992).

3. Results and discussion

3.1. Feature extraction by computing Lyapunov exponents

In the present study, the technique used in the computation of Lyapunov exponents was related with the Jacobi-based algorithms. The Lyapunov exponents of the typical segment of ECG signals obtained from a normal subject and a subject with partial epilepsy are given in Figs. 2 and 3, respectively. It can be noted that the Lyapunov exponents of the typical segment of ECG signals obtained from normal subject differ significantly from the Lyapunov exponents of the typical segment of ECG signals obtained from subject with partial epilepsy. As it is seen from Figs. 2 and 3 there are positive Lyapunov exponents, which confirm the chaotic nature of the ECG signals obtained from normal subjects and subjects with partial epilepsy. The largest Lyapunov exponents of the 40 segments of the ECG signals belonging to normal subjects and subjects with partial epilepsy (20 segments from each class) are shown in Fig. 4. From Fig. 4 one can see that there is a significant

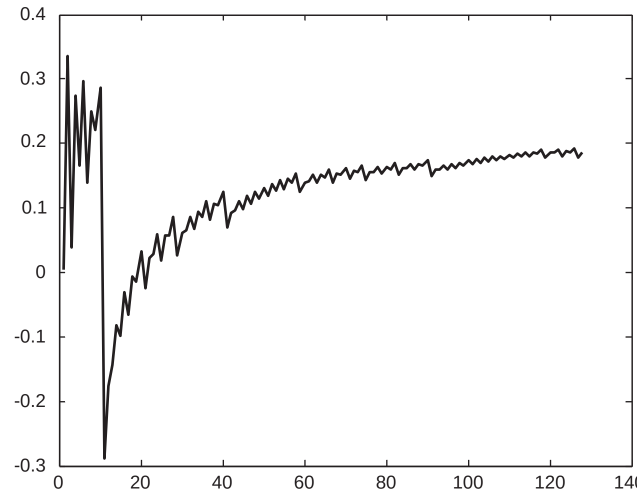

Fig. 2. Lyapunov exponents of a typical normal ECG beat.

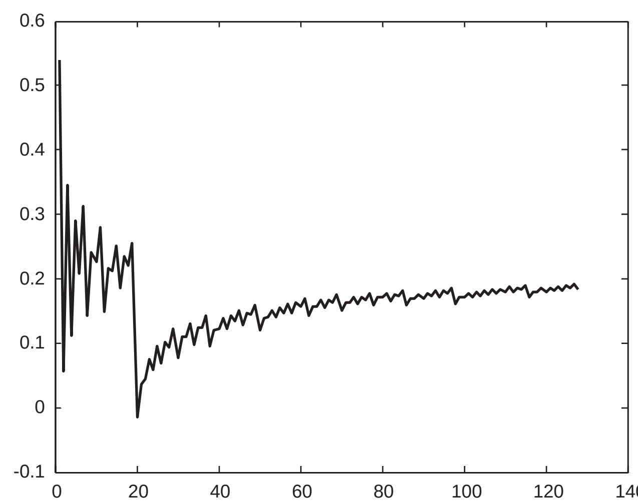

Fig. 3. Lyapunov exponents of a typical partial epilepsy ECG beat.

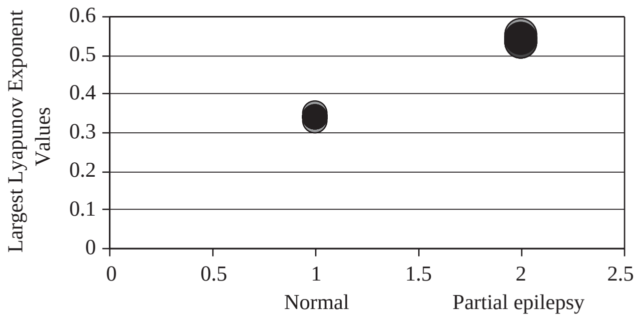

Fig. 4. Largest Lyapunov exponents of the 40 segments of the ECG signals belonging to normal subjects and subjects with partial epilepsy (20 segments from each class).

increase in the largest Lyapunov exponent values of the ECG signals obtained from subjects with partial epilepsy comparing with the largest Lyapunov exponent values of the ECG signals obtained from normal subjects. The Lyapunov exponents were computed using MATLAB software package.

3.2. Application of MLPNNs to ECG signals

ANN architectures are derived by trial and error and the complexity of the neural network is characterized by the number of hidden layers. There is no general rule for selection of appropriate number of hidden layers. A neural network with a small number of neurons may not be sufficiently powerful to model a complex function. On the other hand, a neural network with too many neurons may lead to overfitting the training sets and lose its ability to generalize which is the main desired characteristic of a neural network. The most popular approach to finding the optimal number of hidden layers is by trial and error. In the present study, after several trials it was seen that two hidden layered network achieved the task in high accuracy. The most suitable network configuration found was 10 neurons for each hidden layer. In the hidden layers and the output layer, sigmoidal function was used, which introduced two important properties. First, the sigmoid is nonlinear, allowing the network to perform complex mappings of input to output vector spaces, and secondly it is continuous and differentiable, which allows the gradient of the error to be used in updating the weights. The MLPNNs were trained by using the BP, DBD, EDBD, QP, and Levenberg–Marquardt algorithms. For the Levenberg–Marquardt algorithm, the Marquardt parameter (γ) was set to 0.01. The MLPNNs were implemented by using MATLAB software package (MATLAB version 6.0 with neural networks toolbox).

Selection of the ANN inputs is the most important component of designing the neural network based on pattern classification since even the best classifier will perform poorly if the inputs are not selected well. Input selection has two meanings: (1) which components of a pattern, or (2) which set of inputs best represent a given pattern. Since the Lyapunov exponents contain a significant amount of information about the signal, the computed Lyapunov exponents (128 Lyapunov exponents) of the ECG signals of each subject were used as the MLPNNs inputs.

The adequate functioning of ANN depends on the sizes of the training set and test set. In this study, training and test sets were formed by 360 vectors (180 vectors from each class) of 128 dimensions (Lyapunov exponents). The 160 vectors (80 vectors from each class) of 128 dimensions were used for training and the 200 vectors (100 vectors from each class) of 128 dimensions were used for testing. A practical way to find a point of better generalization is to use a small percentage (around 20%) of the training set for cross validation. For obtaining a better network generalization 32 vectors (16 vectors from each class) of training set, which were selected randomly, were used as cross-validation set. Beside this, in order to enhance the generalization capability of the MLPNNs, the training and the test sets were formed by data obtained from different patients. For both of the beat types, waveform variations were observed among the vectors belonging to the same class.

The outputs of the MLPNNs were represented by unit basis vectors:

[0 1] = normal beat.
[1 0] = partial epilepsy beat.

3.3. Performance analysis of MLPNNs

The MLPNNs were trained with the training set, cross validated with the cross-validation set, and checked with the test set. In this study, performance analysis of the MLPNNs is examined in two parts: training performance and testing performance.

3.3.1. Training performance of MLPNNs

The training set provided to the MLPNNs was representative of the whole space of concern so that the trained MLPNNs had the ability of generalization. In training, a representative training set with examples was presented iteratively to the MLPNNs and the output activations were calculated using the MLPNNs weights. An error term, based on the difference between the output of MLPNNs and desired output, was then propagated back through the MLPNNs to calculate changes of the interconnection weights. The square difference between the output of MLPNNs and the desired output over training iterations was plotted for observing how well the MLPNNs were trained. The curve of the mean square error (MSE) versus iteration is the training curve. In general, it is known that a network with enough weights will always learn the training set

Table 1
The values of minimum and final MSE during training and cross validation

MLPNN trained with different algorithms	Number of epochs		Minimum MSE	
	Training	Cross validation	Training	Cross validation
BP	4600	4600	0.004589	0.005472
DBD	3500	3500	0.001563	0.002058
EDBD	2700	2700	0.000911	0.000985
QP	1600	1600	0.000458	0.000561
Levenberg–Marquardt	500	500	0.000275	0.000341

better as the number of iterations is increased. However, this decrease in the training set error is not always coupled to better performance in the test. When the network is trained too much, the network memorizes the training patterns and does not generalize well. The training holds the key to an accurate solution, so the criterion to stop training must be very well described. Cross validation is a highly recommended criterion for stopping the training of a network. In this study, when the error in the cross validation increased, the training was stopped because the point of best generalization had been reached. The values of minimum MSE and final MSE of the MLPNNs trained with five different training algorithms during training and cross validation are given in Table 1. As it is seen from Table 1, MSE curve of the MLPNN trained with the Levenberg–Marquardt algorithm is converging to a small constant approximately zero in 500 epochs and the BP, DBD, EDBD, QP algorithms have poor convergence rates comparing with the Levenberg–Marquardt algorithm. In Fig. 5, the error of the MLPNN trained with the Levenberg–Marquardt algorithm in training set and the cross-validation set is shown on the same graph.

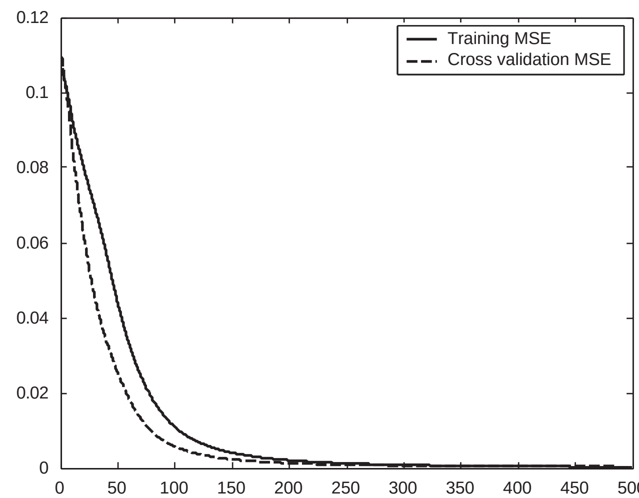

Fig. 5. Training and cross-validation MSE curves of the MLPNN trained with the Levenberg–Marquardt algorithm.

3.3.2. Testing performance of MLPNNs

The trained MLPNNs were tested using the test set formed by 200 vectors (100 vectors from each class) of 128 dimensions. The MLPNNs applied their past experience to test data and produced a solution based on the training and topology of the MLPNNs. The evaluation of testing performance of the MLPNNs was performed by assessment of classification results, the values of statistical parameters, receiver operating characteristic (ROC) curves analysis, and performance evaluation parameters.

In classification, the aim is to assign the input patterns to one of several classes, usually represented by outputs restricted to lie in the range from 0 to 1, so that they represent the probability of class membership. While the classification is carried out, a specific pattern is assigned to a specific class according to the characteristic features selected for it. In this study, there were two classes: normal beat and partial epilepsy beat. Classification results of the MLPNNs trained with five different algorithms were displayed by a confusion matrix. The confusion matrix showing the classification results of the MLPNNs is given below.

Confusion matrix

MLPNN trained with different algorithms	Output/desired	Result (normal beat)	Result (partial epilepsy beat)
BP	Result (normal beat)	89	12
	Result (partial epilepsy beat)	11	88
DBD	Result (normal beat)	90	11
	Result (partial epilepsy beat)	10	89
EDBD	Result (normal beat)	90	9

	Result (partial epilepsy beat)	10	91
QP	Result (normal beat)	93	6
	Result (partial epilepsy beat)	7	94
Levenberg–Marquardt	Result (normal beat)	98	3
	Result (partial epilepsy beat)	2	97

According to the confusion matrix, 2 normal beats were classified incorrectly by the MLPNN trained with the Levenberg–Marquardt algorithm as partial epilepsy beats and 3 partial epilepsy beats were classified as normal beats. According to the confusion matrix, the classification results of the MLPNN trained with the Levenberg–Marquardt algorithm are better than that of the other MLPNNs. The test performance of the MLPNNs was determined by the computation of the following statistical parameters:

Specificity: *number of correct classified normal beats/number of total normal beats.*

Sensitivity: *number of correct classified partial epilepsy beats/number of total partial epilepsy beats.*

Total classification accuracy: *number of correct classified beats/number of total beats.*

The values of these statistical parameters are given in Table 2. As it is seen from Table 2, the MLPNN trained with the Levenberg–Marquardt algorithm classified normal beats and partial epilepsy beats with the accuracy of 98.00% and 97.00%, respectively. The normal beats and partial epilepsy beats were classified with the accuracy of 97.50%. According to Table 2, the correct classification rates of the MLPNN trained with the Levenberg–Marquardt algorithm for normal beats and partial epilepsy beats are higher than that of the other MLPNNs.

The performance of a test can be evaluated by plotting a ROC curve for the test. ROC curves make it possible to highlight the performance of each MLPNN in terms of probability of detection of partial epilepsy beats. For a given result obtained by a classifier system, four possible alternatives exist that describe the nature of the result: (i) true positive (TP), (ii) false positive (FP), (iii) true negative (TN), and (iv) false negative (FN) (Zweig and Campbell, 1993). In this study, a TP decision occurred when the positive detection of the MLPNN coincided with a positive detection of the physician. A FP decision occurred when the MLPNN made a positive detection that did not agree with the physician. A TN decision occurred when both the MLPNN and the physician suggested the absence of a positive detection. A FN decision occured when the MLPNN made a negative detection that did not agree with the physician. ROC curves which are seen in Fig. 6 represent the performances of the MLPNNs on the test file. The performances of tests depend on the shapes of the ROC curves. A good test is one for which sensitivity rises rapidly and 1-specificity hardly increases at all until sensitivity becomes high. Fig. 6 shows that the performance of the MLPNN trained with the Levenberg–Marquardt algorithm is very high comparing with the performances of the other MLPNNs.

The difference between the output of the network and the desired response is referred to as the error and can be measured in different ways. In this study, MSE, mean absolute error (MAE), minimum absolute error, maximum absolute error, and correlation coefficient (r) were used for measuring error of the MLPNNs. The sizes of MSE and MAE can be used to determine how well the

Table 2
The values of statistical parameters

MLPNN trained with different algorithms	Statistical parameters		
	Specificity (%)	Sensitivity (%)	Accuracy (%)
BP	89.00	88.00	88.50
DBD	90.00	89.00	89.50
EDBD	90.00	91.00	90.50
QP	93.00	94.00	93.50
Levenberg–Marquardt	98.00	97.00	97.50

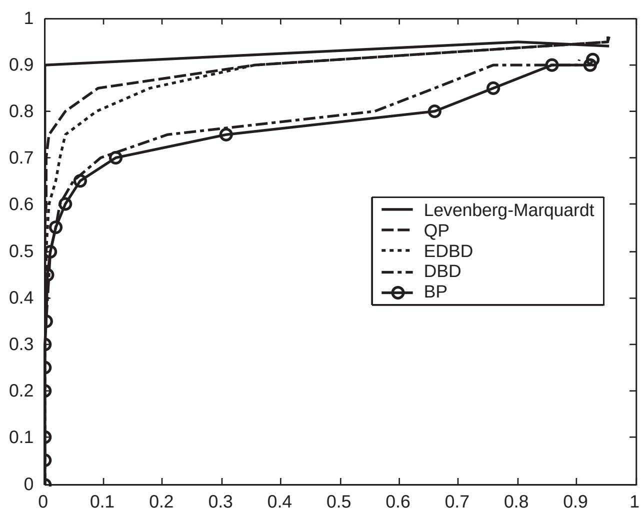

Fig. 6. ROC curves.

Table 3
The values of performance evaluation parameters during test process

MLPNN trained with different algorithms	Performance	Result (normal beats)	Result (partial epilepsy beats)
BP	MSE	0.001225	0.001032
	MAE	0.083526	0.084569
	Minimum absolute error	0.019561	0.020364
	Maximum absolute error	0.527162	0.596328
	r	0.915236	0.918952
DBD	MSE	0.000962	0.000987
	MAE	0.069852	0.075523
	Minimum absolute error	0.009856	0.017325
	Maximum absolute error	0.445269	0.452632
	r	0.932056	0.926989
EDBD	MSE	0.000895	0.000783
	MAE	0.051622	0.050732
	Minimum absolute error	0.009162	0.009089
	Maximum absolute error	0.342563	0.332756
	r	0.945233	0.950227
QP	MSE	0.000728	0.000652
	MAE	0.045891	0.039251
	Minimum absolute error	0.008963	0.007751
	Maximum absolute error	0.326258	0.285623
	r	0.966230	0.968956
Levenberg–Marquardt	MSE	0.000297	0.000514
	MAE	0.012533	0.015743
	Minimum absolute error	0.005207	0.007428
	Maximum absolute error	0.112586	0.156397
	r	0.982586	0.972041

network output fits the desired output, but they may not reflect whether the two sets of data move in the same direction. The correlation coefficient solves this problem. The correlation coefficient is limited within the range [−1,1]. When $r = 1$ there is a perfect positive linear correlation between network output and desired output, which means that they vary by the same amount. When $r = -1$ there is a perfectly linear negative correlation between network output and desired output, that means they vary in opposite ways. When $r = 0$ there is no correlation between network output and desired output. Intermediate values describe partial correlations. The values of performance evaluation parameters during test process of the presented MLPNNs are given for normal and partial epilepsy beats in Table 3. The classification results, the values of statistical parameters, ROC curves, and performance evaluation parameters indicated that the MLPNN trained with the Levenberg–Marquardt algorithm was the most efficient algorithm for detecting electrocardiographic changes in patients with partial epilepsy.

4. Conclusion

This paper presented a new application of detecting electrocardiographic changes in patients with partial epilepsy using Lyapunov exponents. Toward achieving this aim, the ECG signals were considered as chaotic signals and this consideration was tested successfully using the nonlinear dynamics tools in the literature, like the computation of Lyapunov exponents. This was the basis for the automatic detection of electrocardiographic changes in patients with partial epilepsy. More specifically, the ECG signals were modelled using MLPNNs. The MLPNNs trained with the BP, DBD, EDBD, QP, and Levenberg–Marquardt algorithms were used to detect electrocardiographic changes in patients with partial epilepsy. The MLPNNs were trained, cross validated and tested with the computed Lyapunov exponents of the ECG signals obtained from normal subjects and subjects suffering from partial epilepsy. The classification results, the values of statistical parameters, ROC curves, and performance evaluation parameters were used for evaluating the classifiers. The classifications of normal beats and partial epilepsy beats, performed by the MLPNN trained with the Levenberg–Marquardt algorithm, were done with the accuracy of 98.00% and 97.00%, respectively. We therefore have concluded that the proposed classifier trained with the Levenberg–Marquardt algorithm can be used in detecting electrocardiographic changes in patients with partial epilepsy.

Acknowledgments

This study has been supported by the State Planning Organization of Turkey (Project no: 2003 K 120470, Project name: Biomedical signal acquisition, processing and imaging).

References

Abarbanel, H.D.I., Brown, R., Kennel, M.B., 1991. Lyapunov exponents in chaotic systems: their importance and their evaluation using observed data. International Journal of Modern Physics B 5 (9), 1347–1375.

Abarbanel, H.D.I., Brown, R., Sidorowich, J.J., Tsimring, L.S., 1993. The analysis of observed chaotic data in physical systems. Reviews of Modern Physics 65 (4), 1331–1392.

Addison, P.S., Watson, J.N., Clegg, G.R., Holzer, M., Sterz, F., Robertson, C.E., 2000. Evaluating arrhythmias in ECG signals using wavelet transforms. IEEE Engineering in Medicine and Biology 19 (5), 104–109.

Basheer, I.A., Hajmeer, M., 2000. Artificial neural networks: fundamentals, computing, design, and application. Journal of Microbiological Methods 43, 3–31.

Battiti, R., 1992. First- and second-order methods for learning: between steepest descent and Newton's method. Neural Computation 4, 141–166.

Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.-M., 1980. Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Meccanica 15 (1), 9–30.

Casaleggio, A., Braiotta, S., 1997. Estimation of Lyapunov exponents of ECG time series—the influence of parameters. Chaos, Solitons & Fractals 8 (10), 1591–1599.

Casdagli, M., 1989. Nonlinear prediction of chaotic time series. Physica D 35 (3), 335–356.

Chan, L-W., 1990. Efficacy of different learning algorithms of the back propagation network. IEEE Region 10 Conference on Computer and Communication Systems, vol. 1, Hong Kong, 24–27 September, pp. 23–27.

Chaudhuri, B.B., Bhattacharya, U., 2000. Efficient training and improved performance of multilayer perceptron in pattern classification. Neurocomputing 34, 11–27.

Eckmann, J.-P., Kamphorst, S.O., Ruelle, D., Ciliberto, S., 1986. Liapunov exponents from time series. Physical Review A 34 (6), 4971–4979.

Fahlman, S.E., 1988. An empirical study of learning speed in backpropagation networks. Computer Science Technical Report, CMU-CS-88-162. Carnegie Mellon University, Pittsburgh.

Fell, J., Mann, K., Röschke, J., Gopinathan, M.S., 2000. Nonlinear analysis of continuous ECG during sleep II. Dynamical measures. Biological Cybernetics 82, 485–491.

Foo, S.Y., Stuart, G., Harvey, B., Meyer-Baese, A., 2002. Neural network-based EKG pattern recognition. Engineering Applications of Artificial Intelligence 15, 253–260.

Govindan, R.B., Narayanan, K., Gopinathan, M.S., 1998. On the evidence of deterministic chaos in ECG: Surrogate and predictability analysis. Chaos 8 (2), 495–502.

Hagan, M.T., Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks 5 (6), 989–993.

Hannan, J.M., Bishop, J.M., 1997. A comparison of fast training algorithms over two real problems. IEE Fifth International Conference on Artificial Neural Networks, Conference Publication No. 440, Cambridge UK, 7–9 July, pp. 1–6.

Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan, New York.

Haykin, S., Li, X.B., 1995. Detection of signals in chaos. Proceedings of the IEEE 83 (1), 95–122.

Jacobs, R.A., 1988. Increased rate of convergence through learning rate adaptation. Neural Networks 1, 295–307.

Kundu, M., Nasipuri, M., Basu, D.K., 2000. Knowledge-based ECG interpretation: a critical review. Pattern Recognition 33, 351–373.

Leutmezer, F., Schernthaner, C., Lurger, S., Pötzelberger, K., Baumgartner, C., 2003. Electrocardiographic changes at the onset of epileptic seizures. Epilepsia 44 (3), 348–354.

Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C., Strintzis, M., 1998. ECG pattern recognition and classification using non-linear transformations and neural networks: a review. International Journal of Medical Informatics 52, 191–208.

Minai, A.A., Williams, R.D., 1990. Back-propagation heuristics: a study of the extended delta-bar-delta algorithm. Proceedings of International Joint Conference on Neural Networks, vol. 1, San Diego, California, 17–21 June, pp. 595–600.

MIT-BIH Database, 2003. Available from Massachusetts Institute of Technology, 77 Massachusetts Avenue, Room E25-505A, Cambridge, MA 02139, USA.

Nugent, C.D., Webb, J.A.C., Black, N.D., Wright, G.T.H., McIntrye, M., 1999. An intelligent framework for the classification of the 12-lead ECG. Artificial Intelligence in Medicine 16, 205–222.

Opherk, C., Coromilas, J., Hirsch, L.J., 2002. Heart rate and EKG changes in 102 seizures: analysis of influencing factors. Epilepsy Research 52, 117–127.

Owis, M.I., Abou-Zied, A.H., Youssef, A.-B.M., Kadah, Y.M., 2002. Study of features based on nonlinear dynamical modeling in ECG arrhytmia detection and classification. IEEE Transactions on Biomedical Engineering 49 (7), 733–736.

Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S., 1980. Geometry from a time series. Physical Review Letters 45 (9), 712–716.

Rocamora, R., Kurthen, M., Lickfett, L., von Oertzen, J., Elger, C.E., 2003. Cardiac asystole in epilepsy: clinical and neurophysiologic features. Epilepsia 44 (2), 179–185.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Sano, M., Sawada, Y., 1985. Measurement of the Lyapunov spectrum from a chaotic time series. Physical Review Letters 55 (10), 1082–1085.

Saxena, S.C., Kumar, V., Hamde, S.T., 2002. Feature extraction from ECG signals using wavelet transforms for disease diagnostics. International Journal of Systems Science 33 (13), 1073–1085.

Sidani, A., Sidani, T., 1994. A comprehensive study of the back-propagation algorithm and modifications. IEEE Conference Record, Orlando FL USA, 29–31 March, pp. 80–84,

Silipo, R., Deco, G., Vergassola, R., Bartsch, H., 1998. Dynamics extraction in multivariate biomedical time series. Biological Cybernetics 79, 15–27.

Simon, B.P., Eswaran, C., 1997. An ECG classifier designed using modified decision based neural networks. Computers and Biomedical Research 30, 257–272.

Sternickel, K., 2002. Automatic pattern recognition in ECG time series. Computer Methods and Programs in Biomedicine 68, 109–115.

Tomson, T., Ericson, M., Ihrman, C., Lindblad, L.E., 1998. Heart rate variability in patients with epilepsy. Epilepsy Research 30, 77–83.

Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A., 1985. Determining Lyapunov exponents from a time series. Physica D 16 (3), 285–317.

Zijlmans, M., Flanagan, D., Gotman, J., 2002. Heart rate changes and ECG abnormalities during epileptic seizures: prevalence and definition of an objective clinical sign. Epilepsia 43 (8), 847–854.

Zweig, M.H., Campbell, G., 1993. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clinical Chemistry 39 (4), 561–577.

İnan Güler graduated from Erciyes University in 1981. He took his M.S. degree from Middle East Technical University in 1985, and his Ph.D. degree from İstanbul Technical University in 1990, all in Electronic Engineering. He is a professor at Gazi University where he is Head of Department. His interest areas include biomedical systems, biomedical signal processing, biomedical instrumentation, electronic circuit design, neural networks, and artificial intelligence. He has written more than 100 articles on biomedical engineering.

Elif Derya Übeyli graduated from Çukurova University in 1996. She took her M.S. degree in 1998, all in electronic engineering. She took her Ph.D. degree from Gazi University, electronics and computer technology. She is an instructor at the Department of Electrical and Electronics Engineering at TOBB Economics and Technology University. Her interest areas are biomedical signal processing, neural networks, and artificial intelligence.