ELSEVIER Artificial Intelligence in Medicine 15 (1999) 275–297

Artificial Intelligence in Medicine

Predicting coronary disease risk based on short-term RR interval measurements: a neural network approach

F. Azuaje [a]*, W. Dubitzky [b], P. Lopes [c], N. Black [a], K. Adamson [a,b], X. Wu [b], J.A. White [d]

[a] *Northern Ireland Bio-engineering Centre, University of Ulster, Jordanstown, Co. Antrim BT37 0QB, UK*

[b] *Faculty of Informatics, University of Ulster, Jordanstown, Co. Antrim BT37 0QB, UK*

[c] *Chelsea School Research Centre, University of Brighton, Eastbourne, BN20 7SR, UK*

[d] *Department of Public Health Medicine and Epidemiology, Queen's Medical Centre, Nottingham, NG7 2UH, UK*

Received 19 March 1998; received in revised form 29 June 1998; accepted 18 August 1998

Abstract

Coronary heart disease is a multifactorial disease and it remains the most common cause of death in many countries. *Heart rate variability* has been used for non-invasive measurement of *parasympathetic activity* and prediction of cardiac death. Patterns of heart rate variability associated with *respiratory sinus arrhythmia* have recently been considered as possible indicators of coronary heart disease risk in asymptomatic subjects. The aim of this work is to detect individuals at varying risk of coronary heart disease based on *short-term* heart rate variability measurements under controlled respiration. Artificial neural networks are used to recognise Poincaré-plot-encoded heart rate variability patterns related to coronary heart disease risk. The results indicate a relatively coarse binary representation of Poincaré plots could be superior to an analogue encoding which, in principle, carries more information. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Coronary heart disease; Heart rate variability; Artificial neural networks; Pattern recognition; Data representation

* Corresponding author. Tel.: +44-1232-368941; Fax: +44-1232-366863; E-mail: fj.azuaje@ulst.ac.uk.

PII: S0933-3657(98)00058-X

1. Introduction and rationale

Coronary heart disease is a multifactorial disease and it remains one of the most common causes of death in many countries [7]. Coronary heart disease risk assessment is a difficult and complex task. Parasympathetic activity corresponds to the firing of the parasympathetic nerve of the heart, and it has been established that in a number of cardiovascular diseases this activity is decreased [1]. Heart rate variability has been used as a non-invasive index of parasympathetic activity as well as a possible predictor of mortality and sudden cardiac death [18]. Furthermore, patterns of heart rate variability associated with respiratory sinus arrhythmia (which corresponds to the heart beat variation in synchrony with respiration) have recently been considered as possible indicators of coronary heart disease risk in asymptomatic (symptom-free) subjects [16].

While conventional analysis can be used to measure the respiratory component of heart rate variability, a non-linear quantitative method, the Poincaré plot, has been proposed for the evaluation of respiratory sinus arrhythmia [26]. There are several studies demonstrating that Poincaré plots can represent distinctive patterns that distinguish healthy subjects from patients with heart failure [13].

In addition to conventional methods (such as Fourier Analysis) several statistical models (e.g. multivariate analysis) [23] and artificial intelligence techniques (i.e. neural networks) [24] have been proposed for early detection of individuals at risk.

There have been several approaches for modelling the diagnosis of coronary heart disease [9,20]. For example, attempts to distinguish normal subjects from subjects with heart failure combining chaos theory and neural networks [6] have been made. In contrast with our study, they use long-term electrocardiogram records. Other works have approached the problem by using data obtained directly from the patient (self-applied questionnaires), as in the works of Shen et al. [24]. While Dickhaus and Heinrich [8] have combined neural networks and *wavelets* to identify patients at high risk of developing *ventricular tachycardia*. Thus, the value of supervised neural networks in predicting coronary disease risk is being widely demonstrated.

This paper describes a method that uses a feed-forward neural network to recognise heart disease risk categories based on Poincaré plots originating from short-term (less than ten minutes) electrocardiograms. Different types of input/output pattern representations have been implemented in order to analyse the influence of the feature extraction procedure (pre-processing) on the recognition process. The results of this study indicate that different pre-processing techniques can be used for the interpretation of the Poincaré plots and that there are some possible links between these techniques and the artificial neural network architecture.

The remainder of this paper is organised as follows. Section 2 provides a detailed introduction to pattern recognition and neural networks, as well as the description of the standard back-propagation learning algorithm. Section 3 describes the various experiments that have been conducted for this study, this includes data acquisition, pre-processing and classification experiments. Section 4 presents the results of the experiments. In Section 5 an evaluation criterion is presented which provides a more formal basis for the analysis of the obtained results. Section 6 presents overall conclusions and a brief look at future work.

2. Pattern recognition and artificial neural networks

Pattern recognition covers a wide range of activities in everyday life. Humans have the capability to receive sensory data, identify its source, extract useful information from it and, based on previous experience, make decisions.

Artificial neural networks have arisen from analogies with models of the way that humans might approach pattern recognition tasks, although they have developed a long way from the biological roots, providing a new approach to computing involving massively parallel computation. Haykin gives the following definition of an artificial neural network ([11], p. 2).

A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network through a *learning* process; and
- *Interneurone* connection strengths known as *synaptic weights* are used to store the knowledge.

Many different artificial neural network topologies have been proposed, for example, *feed-forward neural networks*. Feed-forward networks are especially useful

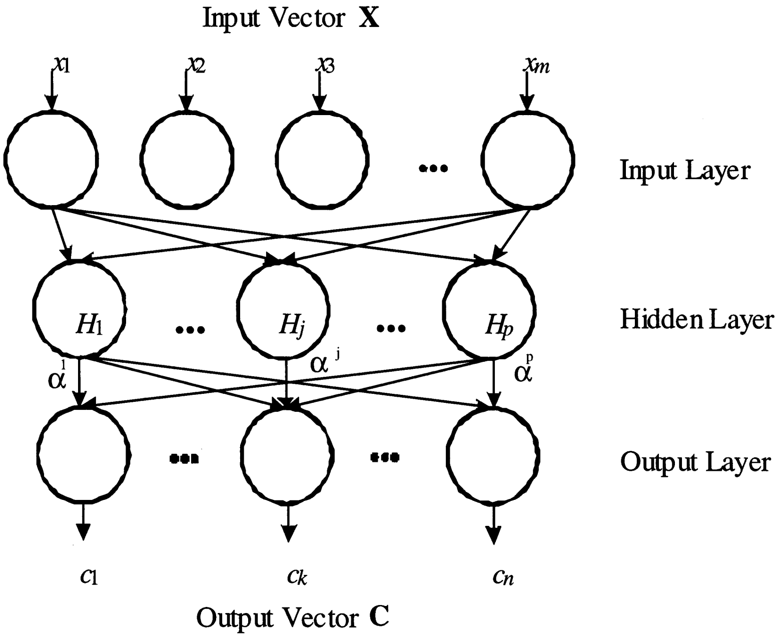

Fig. 1. A 2-layer artificial neural network.

for pattern recognition tasks. Fig. 1 shows a two layer feed-forward network. It consists of an input layer, a hidden layer and an output layer. A *node* (neurone) in the network has a number of inputs and a single output. For example, neurone H_j has x_1, x_2, ... , x_m as its input and α^j as its output. A link in the network is associated with a *weight*. The input links of H_j have weights u_1^J, u_2^j, ... ,u_m^j. A node computes its output, the *activation value* by summing its weighted inputs, subtracting a threshold, and passing the result to a non-linear function *f*, the *activation function*. Outputs from neurones in one layer are fed as inputs to neurones in the next layer. So, when an input vector is applied to the input, an output vector is obtained at the output layer.

One way to implement pattern recognition tasks using artificial neural networks is by a supervised learning process. This pattern recognition process is defined by a training and a test phase. In the training phase a set of pre-determined classes is set and the existence of some mechanism that could correctly label or classify each example is assumed. The *features* of a new, previously unclassified example are then fed into the pattern recognition machine (neural network), known as the *classifier* [21]. Features are the correlation of activity among different input nodes.

For example, suppose that the input for an artificial neural network is a two-dimensional array of binary inputs. Features in this array could mean a pattern of horizontal or vertical node *activations*, or a corner junction between two line segments.

These relationships between the activations of different input nodes or neurones provide a basis for a higher-level, more abstract representation of the input information in the next layers [17]. The training phase aims to find the best set of weights for the network which allow the network to classify input vectors with a satisfactory level of accuracy. An initial set of weights is chosen randomly in a predefined interval. The weights are normally updated via the standard back-propagation algorithm (Section 2.1). For a well-trained network which represents the classification function, if an input vector $\mathbf{X} = (x_1, x_2, \dots, x_m)$ is applied to the input layer, the output vector $\mathbf{C} = (c_1, c_2, \dots, c_n)$ should be obtained, where c_i has the value 1 if the input vector $\mathbf{X}$ belongs to class c_i, and the value 0 otherwise. The training phase is terminated when difference between the computed activation of the output units and the desired activation of the output units falls below a pre-specified threshold.

Because units in the artificial neural network have learnt to respond to features as the network was trained, the networks has developed the ability to generalise, i.e. to classify unknown examples. Thus, the test phase would report the following: 'this example or case is from class c_i' or 'this example or case is from none of these classes' or 'this example or case is too difficult for me' [21].

2.1. Back-propagation learning algorithm

The general process of learning in an artificial neural network—using the standard back-propagation procedure—is given by the five-step algorithm in Table 1 [22].

Table 1
Back-propagation learning algorithm

1. Set activation for input units
2. Propagate forward the activation along the directed connections (links); possibly through hidden units
3. Compute error as a function of the difference between the actual and desired activation on output units
4. Propagate error back through the same links used for carrying activations
5. Change connection weights to minimise difference between actual and desired activations on output units

Some of the equations that underlie this algorithm are illustrated below. These equations define how activations are computed, errors are propagated, and weights are changed according to the back-propagation process.

$$\text{Net Input}_i = \sum_j w_{ij} a_j + \theta_i \tag{1.1}$$

$$a_i = \frac{1}{1 + \mathrm{e}^{-\text{Net Input}_i}} \tag{1.2}$$

$$E\text{output}_i = [d_i - a_i][(1 - a_i)a_i] \tag{1.3}$$

$$\Delta w_{ij}(t) = \alpha e_j a_i + \mu \Delta w_{ij}(t-1) \tag{1.4}$$

where

Symbol	Meaning
i	index of a unit
Net Input$_i$	net incoming activation to unit i
j	index of every unit connected to unit i
w_{ij}	weight of a connection from unit j to unit i
θ_i	threshold term associated with unit i
a_I	activation of unit i
n	number of output unit
d_i	desired activation for unit i
Eoutput$_i$	error of unit i, when the unit is an Output unit
α	the learning rate
μ	the momentum term
t	a time index

Eq. (1.2) defines the activation function for units in a back-propagation network. As illustrated in Fig. 2, this standard *sigmoid function* squashes net incoming activation to a unit (Eq. (1.1)), which ranges over $[-10,10]$, into the range $[0,1]$. The effect of θ_i, the *bias*, term, in Eq. (1.1) is to increase (or decrease) the net input to a unit. It acts as a threshold for the activation function.

Eqs. (1.3) and (1.4) define the learning mechanism of connectionist networks. Intuitively, this method works by determining where the network went wrong, and making changes to address the mistakes. The total error of the network is defined as the sum of squared differences between the actual and desired activations of output units.

Eq. (1.3) sets the error at an output unit. This function, the partial derivative of the error with respect to the net input, can be computed by applying the chain rule to Eq. (1.2) and to the total error of the network. The error at each hidden unit does not appear in this table, but it is a function of the activation of that unit and the errors of all units to which that unit sends an activation.

Finally, Eq. (1.4) shows that the adjustment of a link's weight is dependent upon the error at the receiving end of the link and the activation at the sending end of the link. In addition, this Equation contains a *momentum* term that is dependent on the previous change in the connection weight. The momentum term is intended to avoid oscillations in the weight adjustments.

3. Materials and methods

3.1. Subjects and data acquisition

An electrocardiogram measures the electric activity of the heart. This activity is described by five waves named *P*, *Q*, *R*, *S* and *T*, respectively. The *P* wave (that represents the *atrial depolarisation* of the heart) is followed by the *QRS* complex (*ventricular depolarisation*) and the *T* wave (*ventricular repolarisation*) [28] (Fig. 3a). A *RR* interval reflects the length of the time period between the *R*-spikes of two subsequent heartbeats (Fig. 3b).

A total of 127 middle-aged asymptomatic male subjects underwent a supine resting electrocardiogram at fixed respiratory frequencies. Four test were performed: Two at 6 and 12 breaths/min with no control of breath volume and two at 6 and 12 breaths/min under controlled breathing volume (30% of tidal volume).

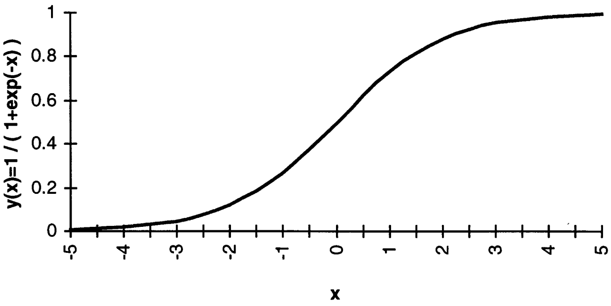

Fig. 2. Sigmoid function.

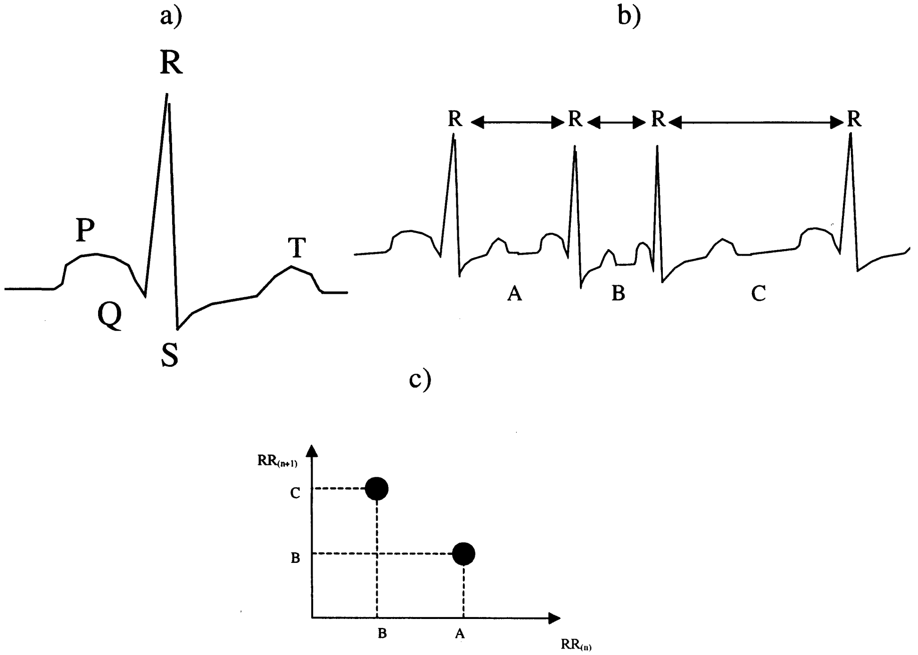

Fig. 3. Construction of a Poincaré Plot. (a) The normal electrocardiogram complex *PQRST*; (b) An electrocardiogram with three *RR* intervals; and (c) Poincaré plot with the first three *RR* interval.

Lead II electrocardiogram signals were digitised at 1 kHz with 12-bit resolution. Fig. 3a illustrates the normal electrocardiogram complex. The output from those four tests were four series of *RR* intervals; two series with 256 *RR* intervals, and two series with 100 *RR* intervals. From those four series, a single combined *RR* interval series containing 712 beat-to-beat time periods was generated.

Standard screening tests were performed to identify selected modifiable and non-modifiable coronary heart disease risk factors. Furthermore, coronary heart disease risk was calculated using a standard scoring system (Anderson) [2], which derives a coronary heart disease risk score from a combination of recognised coronary heart disease risk factors. The Anderson risk scores are calculated taking into consideration the following factors: age, total cholesterol, high density lipoprotein cholesterol (HDL), systolic blood pressure (SBP), smoking, diabetes and left ventricular hypertrophy assessed by electrocardiography (LVH). Patients were subsequently categorised in *low*, *medium*, *high*-risk groups, as well as in *very low*, *low*, *medium*, *high* and *very high* risk groups. In the case of three categories, the *low* risk category corresponds to the Anderson risk score range [0,10], *medium* to [11,20], and *high* to [21,31]. In the case of five categories, the *very low* risk category correspond to the Anderson risk score range [0,6], the *low* to [7,11], *medium* to [12,17], *high* to [18,23], and the *very high* risk category corresponds to the range [24,31].

3.2. Poincaré plots

The Poincaré plot is a *scatterplot* of a *RR* interval of the electrocardiogram (measured in milliseconds) against the *RR* interval immediately succeeding it. Thus, a Poincaré plot provides information related to the instantaneous beat-to-beat behaviour of the heart. Fig. 3c illustrates how a Poincaré plot is constructed from an electrocardiogram signal (in this simple case only three *RR* intervals are considered).

With this quantitative method, the degree of heart rate variability is graphically displayed as a pattern of points. When the basic beat-to-beat rhythm of the heart is very regular with little *interbeat* variability, points representing the *RR* interval are spread closely along the diagonal line at an angle of 45° to both axes. Points below this indicate a shorter *RR* interval relative to the preceding *RR* interval. Similarly, any points above the diagonal line indicate a *RR* interval longer than the preceding *RR* interval [26]. The Poincaré plot is a visual tool that can be applied to the analysis of *RR* interval allowing the capture of a heart rate variability pattern for each subject.

Poincaré plots have been used to determine the differences between healthy individuals and those with heart failure [29]. Fig. 4 shows the Poincaré plots of two subjects with a different coronary heart disease risk score. The pattern represented

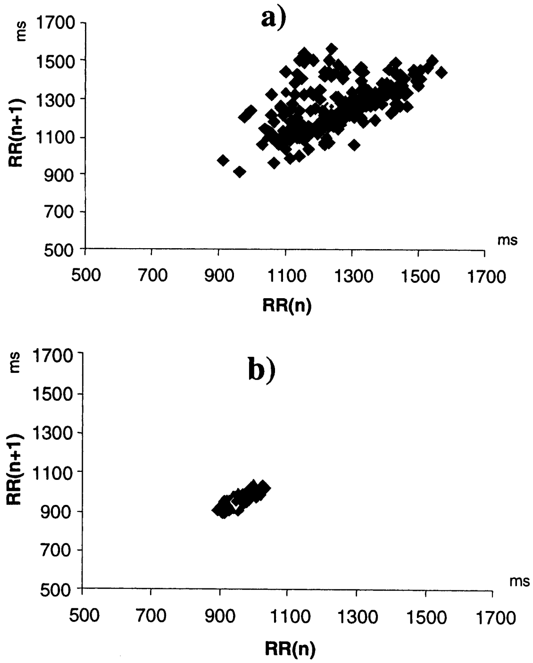

Fig. 4. Poincaré plot patterns. (a) Heart rate variability pattern corresponding to one patient with *low* risk; and (b) Pattern corresponding to patient with *high* risk.

in Fig. 4a corresponds to a patient with a low risk score, while the pattern in Fig. 4b corresponds to a patient with a high risk score. The low-risk subject in Fig. 4a shows a high heart rate variability (dispersion of points) and a low mean heart rate, while the high-risk subject in Fig. 4b is characterised by low variability and high mean heart rate.

3.3. Pre-processing and feature extraction procedures

The goal of pre-processing and feature extraction is to transform the information represented into a set of raw data for performing a pattern recognition algorithm. In neural network applications it aims to obtain a new set of data to be fed into a neural network architecture. This new set of data is obtained taking into consideration the nature of the data and the input format requirements of the network.

Typically, for performing pattern recognition tasks in signal processing applications and time series analysis the Fourier coefficients have been used as the set of features. As the need for more proficient classification have emerged, other feature extraction techniques as the Wigner-Ville transform [19] and *wavelets* [10] are being explored. Kwok-Kei and Wu [15] have proposed a feature extraction procedure, called Cellular Feature Extraction Method for extracting features of line drawings and large size images. The main idea of Cellular Feature Extraction Method is to ignore most of the non-target pixels that bear no useful information to the pattern recognition task.

In this study the main objectives of pre-processing are to adapt the two-dimensional format of the information represented in a Poincaré plot to an artificial neural network input format, and compress the effective size of the data represented in each pattern to a limited size. This data reduction is necessary to obtain a relatively short learning time and reduce the complexity of the network. Data reduction should also preserve sufficient information to discriminate between the three classes.

Also to establish future differences between coronary heart disease assessment methods based on artificial neural network is crucial to know how the electrocardiogram data is prepared for presentation to the artificial neural network. Moreover the method of pre-processing has an important influence on the obtained features.

Based on one Poincaré plot for each patient, an algorithm was implemented which quantifies the visual patterns presented for each patient. One way of interpreting in each pattern is to recognise the distribution of points in the plot.

The encoding was realised by dividing the plot into a number of squares, each square with a width equal to Δt ms and taking into account RR intervals between 500 ms and 1700 ms. By using a Δt equal to 100 ms, it is possible to divide the Poincaré plot space into 144 squares. The number of input nodes of the artificial neural network will then be the number of these squares.

In this experiment, two types of input values are used: binary and analogue inputs. The selection of these representations was done taking into consideration the following factors:

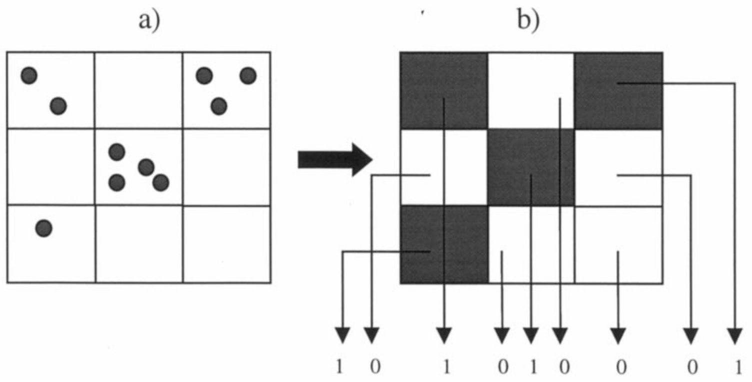

Fig. 5. Generation of binary inputs from Poincaré plots. (a) Poincaré plot with ten samples and divided into nine squares; and (b) generation of the input vector.

- The nature of the source: the input patterns are discrete values; and
- The range of the inputs: the entire range of possible values is considered and it can be reduced effectively between zero and one.

3.3.1. Binary inputs

With this binary-input pre-processing procedure the network recognises the presence or absence of sample points in the Poincaré plot. Thus the value of each square on the Poincaré plot is represented either by 0 (absence) or 1 (presence). For those squares that have samples the value is 1, otherwise, if there is no value, the value 0 is assigned.

Fig. 5 illustrates the process of binary inputs generation for a Poincaré plot with ten samples and nine squares (Fig. 5a). Fig. 5b shows how the input vector is defined and presented to the network according to a predefined order in sequence of analysed sectors. Using this technique the network can recognise the presence or absence of samples inside the different sectors of the plot (Fig. 5b).

This pattern representation method has the advantage of providing a simple and direct way for representing input patterns. Thus, the features are associated with a bit position in a pattern vector using binary digits 0 and 1 to indicate the presence and absence of the features in each pattern. The digit 1 has been used as the representation of an active pattern element, and 0 to indicate the absence of a feature. However, the simplicity of this method is based on the introduction of a certain degree of information loss as it treats all squares with one or more sample points equally. This is somewhat similar to Zadeh's *incompatibility principle* [14] which, informally, can be stated as follows: 'reduction of complexity through introduction of uncertainty'. In the following addresses this issue.

3.3.2. Analogue inputs

In the analogue-input pre-processing method the total number of samples present in each section is taken into account. It is necessary to scale or normalise all the values to indicate the degree of influence of each feature in a determined pattern. Two different analogue-input generation procedures were implemented and tested in this study, these are referred to as method/procedure I and II, respectively.

3.3.2.1. Analogue inputs, procedure I. In this pre-processing procedure the number of samples per square in the Poincaré plot is mapped into the unit interval [0,1]. The value (total number of samples per squared section) is divided by the 'total number' of samples on the plot. This means that in the extreme case, where all the samples of a pattern fall within a single square, the corresponding input value will be 1. For the other extreme case, where there is no sample point within a single square, the input value will be set to 0. And for those squares with a total number of samples less than the total number of samples on the plot, the input value will be between 0 and 1. With this representation it is possible to measure the degree to which a feature exists in a pattern.

Fig. 6 illustrates the first procedure of analogue-input generation for a Poincaré plot with ten samples that is divided into nine squared sections (Fig. 6a). Fig. 6b shows how the input vector is defined and presented to the network according to a predefined order in sequence of the analysed sectors. Using this procedure the network has a better 'vision' of each pattern (Fig. 6b) because of the differentiation of sections (different states according the number of samples). However with this representation it is expected that the network will still have difficulties in recognising the distribution of points, because the obtained values are not completely scaled between 0 and 1.

3.3.2.2. Analogue inputs, procedure II. Similarly to procedure I, in this input generation procedure the number of samples per square in the Poincaré plot is also mapped into the unit interval [0,1]. The value (total number of samples by squared section on the plot) is divided by the *maximum number* of samples present in a particular square on the actual processed plot. A plot is divided into a number of

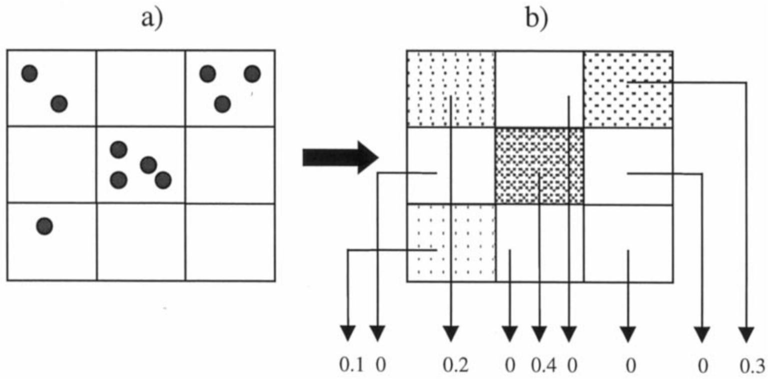

Fig. 6. Generation of inputs from Poincaré plots; analogue procedure I. (a) Poincaré plot with ten samples and divided into nine squares; and (b) generation of the input vector.

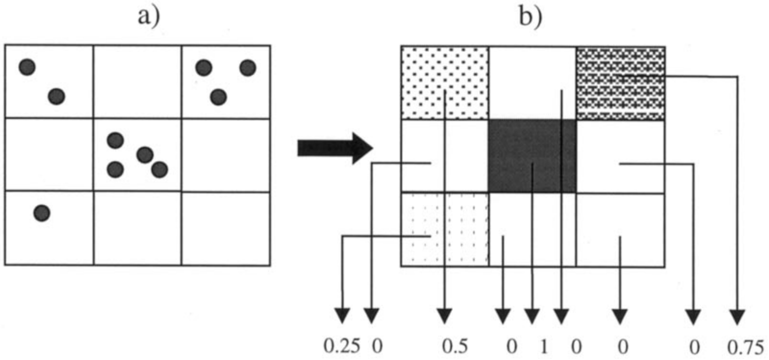

Fig. 7. Generation of inputs from Poincaré plots; analogue procedure II. (a) Poincaré plot with ten samples and divided into nine squares; and (b) generation of the input vector.

squares and the maximum number of samples N in a square is determined. The input value will be 1 for the square with N samples, the value is set to 0 for those squares without samples at all, and for those squares with a number of samples between 0 and N the value will lie within the closed interval (0,1).

Fig. 7 illustrates the second procedure II of analogue input generation for a Poincaré plot with 10 samples which is divided into nine squared sections (Fig. 7a). Fig. 7b shows how the input vector is defined and presented to the network according to a predefined order in sequence of analysed sectors. Thus scaled values between 0 and 1 are generated (Fig. 7b). Because there is a better differentiation between the values of the obtained intermediate states, a more accurate representation of the original patterns could be provided.

3.4. Defining output categories

Now that we have defined the format of the input patterns, we will consider the format for the output that we will expect the network to produce. The network should provide us with a classification according to the risk level of a subject. Therefore we have to decide a definition of the indicators that the network must produce that we can interpret as a risk category classification.

In this application we want an *exclusive* classification, i.e. we are interested in outputs such as: 'This subject is a *high* risk subject', or 'this subject is a *low* risk subject', and so on. Thus, an output binary format is suitable. If the vector $\mathbf{x} = (x_1, x_2, \dots, x_M)$ is applied to the input layer of the network, the output vector $\mathbf{c} = (c_1, c_2, \dots, c_N)$ should be obtained. Where the value c_i has a value 1 if the input vector belongs to class c_i and 0 otherwise. In this case one active output is interpreted as an indication of the coronary heart disease risk category or class to which the input pattern belongs. Initially, three coronary heart disease risk categories were used, namely, *high*, *medium* and *low*. An application with five risk categories was also implemented; in this experiment the categories were labelled *very high*, *high*, *medium*, *low* and *very low*. Tables 2 and 3 describe the defined neural outputs to classify three and five coronary heart disease risk categories, respectively.

3.5. Selecting the network and the training/test groups

For pattern recognition and similar tasks *multilayered* feed-forward networks have been successfully applied [9,20,8]. One of the limitation of this kind of architecture is the long training time for large networks and a propensity to not learn at all due to local minima in the error surface. Nevertheless, a solution for those limitations could be found with a proper pre-processing procedure of the inputs and by using the more recently developed back-propagation algorithms.

Tables 4 and 5 describe the groups of training and test cases that were used for this experiment. The set of training cases consists of 100 subjects and the test set of 27 subjects. The task is to classify three and five risk categories, respectively.

In the case of three categories, the *low* risk category corresponds to the Anderson risk score range [0,10], *medium* to [11,20], and *high* to [21,31]. Using these risk

Table 2
Description of network outputs for classifying three coronary heart disease risk categories

Coronary heart disease risk category	Defined neural network outputs		
High	1	0	0
Medium	0	1	0
Low	0	0	0

Table 3
Description of network outputs for classifying five coronary heart disease risk categories

Coronary heart disease risk category	Defined neural network outputs				
Very high	1	0	0	0	0
High	0	1	0	0	0
Medium	0	0	1	0	0
Low	0	0	0	1	0
Very low	0	0	0	0	1

Table 4
Number of training and test cases used to classify *three* risk categories

Coronary heart disease risk category	Number of training cases	Number of test cases
Low	23	9
Medium	57	9
High	20	9
Total	100	27

Table 5
Number of training and test cases used to classify *five* risk categories

Coronary heart disease risk category	Number of training cases	Number of test cases
Very low	11	4
Low	16	5
Medium	36	9
High	25	4
Very high	12	5
Total	100	27

ranges results in a dominance of the medium-risk subjects in both training sets (Tables 4 and 5)

In the case of five categories, the *very low* risk category correspond to the Anderson risk score range [0,6], the *low* to [7,11], *medium* to [12,17], *high* to [18,23], and the *very high* risk category corresponds to the range [24,31] (Table 5).

3.6. Architectures

Table 6 shows the various network architectures that were implemented in this study. In total seven experiments, comprising different types of pre-processing techniques (Section 3.3) and risk classes, were conducted (training and testing). To classify three classes, three binary input architectures were trained and tested using 144, 576 and 1024 inputs, respectively. The various experiments show the effect of increasing the number of inputs in the pattern-recognition process. Two architectures based on the analogue-input representation procedures described in Section 3.3 were trained and tested with 100 inputs. The performances of these experiments were compared with those obtained in the binary input encoding.

Table 6
Architectures that were used for the various experiments[a]

Experiment	Number of input units	Input representation	Units HL 1	Units HL 2	Output units
1	144	Binary	70	–	3
2	144	Analogue, procedure I	70	30	3
3	144	Analogue, procedure II	70	30	3
4	144	Binary	70	30	5
5	576	Binary	200	100	3
6	576	Binary	200	100	5
7	1024	Binary	500	200	3

[a] HL, hidden layer.

For the classification involving five classes (risk categories) only the binary-input architectures were implemented (for 144 and 576 inputs, respectively). The units of these architectures were fully interconnected and the *sigmoid* transfer function was used for both hidden and output layers. The criterion used to select the number of hidden layers and the number of units in the hidden layers was based on the best performance obtained. All the experiments presented in this paper were implemented as 2-hidden-layers configurations with the exception of experiment 1 (Table 6) because in this case a similar performance was obtained adding another layer. In order to determine the number of units in the hidden layer, multiple networks with different number of units were trained and tested for each experiment, always starting with a number of units that is small compared to the number of units in the input layer.

3.7. Training the network

A neural network design tool (simulator) served as a basis to realise the various networks and run the simulations. The input files for these simulations were generated by a module developed in C, and the back-propagation learning algorithm provided the key method for training the networks. Each of the 100 data files in the training set was presented to the network in a random fashion, and the error criterion was defined at 0.05 RMS (root mean square). To deal with the problem of over-training, the *SaveBest* option of the simulator was chosen. This option saves the network weights at various stages of learning (usually after 1000 presentations) to select the ones that produced the best performance. A non-constant *learning coefficient* and *momentum scheme*, according to the number of learning cycles reached, was performed as the best way of dealing with the dichotomy problem: no divergent behaviour and faster learning. Tables 7 and 8 show the details of this scheme for the output and hidden units, respectively. The output and hidden units start with a predefined learning coefficient and momentum, which values are reduced proportionally to a predefined rate when the network arrives at certain number of training cycles. The values of these two parameters are changed when the network has performed 10 000, 30 000, 70 000 and 100 000 training iterations.

Table 7
Values of the learning coefficient and momentum in the output units according to the number of learning iterations carried out by the network

Number of learning iterations	Learning coefficient value	Momentum value
0–10 000	0.15000	0.40000
10 001–30 000	0.07500	0.20000
30 001–70 000	0.01850	0.05000
70 001–150 000	0.00117	0.00312

Table 8
Values of the learning coefficient and momentum in the hidden units according to the number of learning iterations carried out by the network

Number of learning iterations	Learning coefficient value	Momentum value
0–10 000	0.30000	0.40000
10 001–30 000	0.15000	0.20000
30 001–70 000	0.03750	0.05000
70 001–150 000	0.00234	0.00312

In contrast to binary-input encoding, the networks with analogue-inputs never reached the error criterion (0.05 RMS) within a pre-set maximum learning period of 24 h. Table 9 shows the number of training cycles that were performed for each architecture using different input encoding procedures. For both cases of analogue-inputs only a network with 144 input units was implemented.

The classification performance over the 'training' group was 100% for each category with the binary input coding procedures. Whereas for the analogue-input encoding procedures, the number of correctly classified cases was below the total number of subjects in each category (average classification performance between 70 and 80%). Thus, the learning ability in those networks with binary inputs was higher than in those networks with analogue inputs. Given the fact that the binary encoding introduces a certain degree of information loss, for the sake of simplicity, this result is rather interesting.

4. Results

After training was performed, the 27 cases of the test group were presented to each of the various networks, and the output nodes of the networks were examined for each test case.

Table 10 presents the results of the experiments with binary input encoding for classifying three coronary heart disease risk classes, namely, *high*, *medium* and *low*. The percentages of test cases correctly classified for each risk class against the number of input units are depicted in Table 10. Three networks were trained and

Table 9
Number of training cycles carried out by each architecture with relation to the various input encoding procedures for classifying *three* risk categories

Encoding procedure	144 Inputs	576 Inputs	1024 Inputs
Binary	20 000	68 000	68 000
Analogue, procedure I	131 000	–	–
Analogue, procedure II	80 000	–	–
Binary (five risk categories)	50 000	50 000	–

Table 10
Classification of *three* risk categories using binary inputs

Number of inputs	*High* risk cases correctly classified	*Medium* risk cases correctly classified	*Low* risk cases correctly classified
144	4/9	6/9	4/9
576	5/9	5/9	3/9
1024	6/9	3/9	1/9

tested with different number of inputs, that is, 144, 576, and 1024 inputs, respectively (according to the binary encoding procedure explained in Section 3.3). By incrementing the number of neural inputs the percentage of correctly classified *high*-risk class improved, while the performance in the other classes decreased. In the case of 1024 inputs, the net correctly classified six out nine *high*-risk test cases, this was the best performance obtained for this class. The same performance could be achieved for *medium* risk class but with 144 inputs, while the best *low* risk performance was four out nine with 144 inputs.

From the medical point of view a high confidence level is required for the classification of high risk subjects. In order to determine the effectiveness of the various networks an evaluation criterion must not only take into account the number of high risk cases that are correctly classified, but also how good the network is in classifying counter-examples, i.e. cases that do not belong to the high risk category. The discrimination quality of the model learned by the networks was evaluated through the statistical measures of *sensitivity* and *specificity* for the classification between *high* risk subjects and those classified in other risk classes [27]. These measures, and the *performance* measure itself, are now defined with reference to the coronary heart disease risk assessment context.

Performance relates the full group of *true positives TP* (*high* risk cases correctly classified) and *true negatives TN* (*high* risk incorrectly classified) to the total number *N* of tested cases. *Performance* measures the ability of the model to produce correct answers and is defined in Eq. (4.1).

$$\mathrm{Performance} = (\mathrm{TP} + \mathrm{TN})/\mathrm{N} \tag{4.1}$$

Specificity calculates a ratio based on the *true negatives TP* (cases correctly classified into *low* and *medium* risk classes) and the *frequency of positives FP* (*medium* and *low* risk cases classified as *high* risk). *Specificity* measures the ability of the model to separate the target class cases (*high*-risk subjects) from the others (*medium*, *low*); it is defined in Eq. (4.2).

$$\mathrm{Specificity} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) \tag{4.2}$$

Sensitivity relates the observed frequency of true positives *TP* to the frequency of negatives *FN* (*high* risk cases classified as *medium* or *low* risk). *Sensitivity* measures the ability of the model to correctly identify the occurrence of a target case (*high* risk subject); it is defined as follows:

Table 11
Discriminatory quality of *high*-risk classification using binary inputs

Number of inputs	Performance (%)	Specificity (%)	Sensitivity (%)
144	48	90	44
576	48	61	55
1024	37	44	54

$$\text{Sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \tag{4.3}$$

Table 11 shows the results of the evaluation of *performance*, *sensitivity* and *specificity* against the number of neural inputs.

The best total classification performance (48%) was obtained using 144 inputs also with the best specificity of the experiments (90%), while the best sensitivity (55%) occurred using 576 inputs.

This evaluation shows that even when the percentage of correctly classified high risk cases is improved by adding units to the input layer, the total classification performance of the net is considerably reduced. Also, by adding input nodes, a certain degradation in the capacity of the network to separate high risk subjects from the others (specificity) can be observed. This means that the smaller the number of inputs the smaller is the confidence with which a non-high risk subject can be distinguished from a high risk individual. On the other hand, however, the ability of the network to identify the occurrence of a high risk case (sensitivity) is increased or nearly maintained for a large number of inputs.

Table 11 illustrates that although the performance of correctly classified *high*-risk cases using 1024 inputs is the best in these experiments, the level of confidence is poor (low sensitivity and low specificity). This means that many *high*-risk cases are erroneously identified as non-*high*-risk cases (*medium* or *low* ones), and that many non-*high* risk cases (*medium* or *low* ones) are incorrectly classified as *high*-risk cases.

Tables 12 and 13 summarise the results of classifying three risk categories (*high*, *medium* and *low* risk) using the two analogue-inputs generation procedures explained in Eq. (4.3). For every procedure a network with 144 neural inputs was trained and tested.

Using the analogue-input encoding procedures (I and II) there was no improvement in the correct detection of *high* risk cases (Table 14). However, the general classification performance (over all risk categories) was improved through encoding

Table 12
Correctly classified subjects based on analogue encoding procedure I, 144 inputs units, and *three* risk classes

Correctly classified *high* risk cases	Correctly classified *medium* risk cases	Correctly classified *low* risk cases
4/9	7/9	3/9

Table 13
Correctly classified subjects based on analogue encoding procedure II, 144 inputs units, and *three* risk classes

Correctly classified *high* risk cases	Correctly classified *medium* risk cases	Correctly classified *low* risk cases
4/9	5/9	5/9

Table 14
Correctly classified *high*-risk subjects using three different input encoding procedures and 144 inputs

Binary encoding procedure	Analogue, encoding procedure I	Analogue, encoding procedure II
4/9	4/9	4/9

Table 15
Discriminatory quality of *high*-risk classification with three different input encoding procedures and 144 input units

Input encoding procedure	Performance (%)	Specificity (%)	Sensitivity (%)
Binary	48	90	44
Analogue, procedure I	52	83	44
Analogue, procedure II	52	90	44

procedures schemes I and II (Table 15). The performances that were achieved with both analogue-input schemes were identical (52%) and better than using binary inputs. However, the best specificity was found in the binary and type II pre-processing procedures (90%). On the other hand, the sensitivity scores were low and there was no improvement related to any type of pre-processing procedure.

Two experiments were performed to classify five risk classes with 144 and 576 neural input units and using the binary input encoding procedure; Fig. 8 and Table 16 present the obtained performances. The classification performances for *very high*, *high* and *medium* risks were improved adding inputs to the network, but for the others categories it is found an inverse relation.

Table 16
Classification of *five* risk categories using 144 and 576 input units

Number of inputs	*Very high* correctly classified	*High* risk correctly classified	*Medium* correctly classified	*Low* correctly classified	*Very low* correctly classified
144	0/5	1/4	5/9	1/5	1/4
576	1/5	2/4	7/9	0/5	0/4

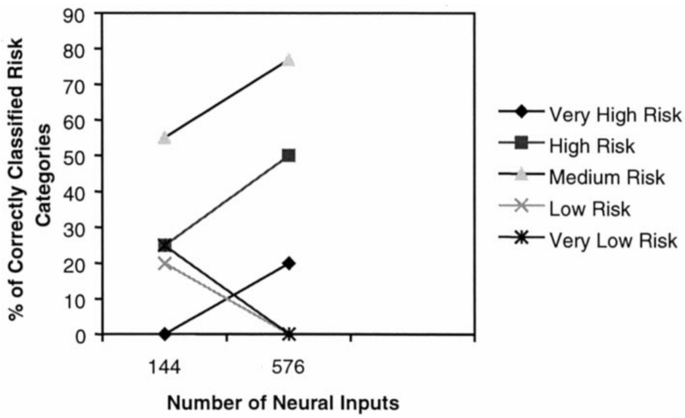

Fig. 8. Classification of five risk categories using 144 and 576 inputs, respectively.

5. Discussion

The work presented in this paper presents a variety of pre-processing experiments relating to short-term electrocardiogram data. The pre-processing procedures focus on the role of binary and analogue encoding procedures and their resolutions, and what effect the various representations have on the classification of coronary heart disease categories as determined by an artificial neural network.

The binary-input experiments suggest that through an increase of the number of neural input units the recognition capability of the neural network with regard to *high* risk subjects can be improved (Section 4). Further, it was shown that using an analogue-input scheme does not necessarily improve the classification accuracy of the network. A more detailed analysis of the binary-input approach shows that increasing the resolution of the input patterns leads to a decrease of the total performance (Eq. (4.1)) and specificity (Eq. (4.2)) with respect to detecting patients at *high* risk (Section 4).

As far as the analogue-input pre-processing encoding is concerned, it was shown that this approach can lead to a higher classification performance when compared with the binary-input scheme, and specifically in case of the analogue II procedure, it can lead to a higher specificity. Thus leading to the expectation the analogue-input approach, together with the addition of neural inputs, will increase the performance of the overall network. However, the experiments have also shown that this does not necessarily lead to an increased capability of distinguishing *high*-risk subjects from the other risk categories (*medium*, *low*). Furthermore, the inability of the network to learn within reasonable time constraints has proven to be another drawback of the analogue-input approach (Section 3). This is not the case with the binary-input procedure where a relatively short training time and better learning capability was observed (Section 3). The poor performance with the analogue-input data may be attributed to the greater variation found in the data. This may have considerable implications for learning and generalisation [12].

With regard to the experiments involving five risk categories (Section 3), it was observed that the addition of neural input nodes could improve the number of correctly classified subjects for the classes *very high*, *high*, and *medium* risk. However, this result was achieved at the cost of a poor performance for the classes *low* and *very low* risk (Section 4).

The number of units in the input layer appears to be the critical factor for influencing the classification quality of the network with respect to coronary heart disease risk classes on either extreme of the risk scale.

One of the limitations of these experiments is the sample size. Furthermore, the distribution of the risk categories within the data was biased towards the *medium*-risk category. Both factors are likely to have had an adverse effect on the classification accuracy of the overall system for both the binary and the analogue approach.

Because of the information loss due to the pre-processing procedures, it is expected that the accuracy of the results can be improved by different pre-processing techniques and by resolving inconsistencies (uneven distribution) in the underlying training data.

Back-propagation appears to be a reasonably robust and reliable approach if the input array is reasonably small, and if the patterns to be learned do not vary greatly in their size or position within the overall input pattern frame that is presented to the neural network. The structure of the data underlying the experiments presented in this paper was such that the input data array consisted of a few hundred data points, within a 1500×1500 frame, with input patterns that varied greatly with respect to their position in that frame. This resulted in a considerable variation of samples that fell inside the squared sections used to partition the frame. However, the fact that an increased number of input points still lead to a high number of correctly classified *high*-risk subjects may be useful to know in other approaches with similarly structured data.

6. Conclusions and future work

Coronary heart disease risk assessment is a time- and resource-consuming task. Estimating heart disease risk from short-term electrocardiogram signals is appealing as it is non-invasive and inexpensive both in terms of time and laboratory resources. This work has showed the usefulness of artificial neural networks in detecting individuals at varying risk of coronary heart disease based on short-term heart rate variability measurements under controlled breathing. The main thrust of this work lies in its detailed study of the influence a variety of pre-processing techniques on the classification behaviour and the architecture of artificial neural networks with regard to Poincaré-plot-encoded electrocardiogram data. Results obtained from these experiments indicate that, based on a relatively small data set, a simple binary-encoding of Poincaré plot data can produce useful results.

Further investigation will look at other *artificial neural network* paradigms (e.g. *Fuzzy Artmap* [9,5]) which may offer solutions to limitations inherent in the back-propagation approach. Some current work in the artificial neural network field aims at finding out more informative patterns for performing clustering functions [25]. It is envisaged that a future knowledge discovery framework could emerge from combining classification and clustering techniques of numerically-structured data (e.g. electrocardiogram) using neural networks and the knowledge discovery methods that work on mainly symbolic representations (e.g. blood pressure, cholesterol, etc.) [3,4].

References

[1] Airaksinen K, Ikaheimo MJ, Linnaluoto MK, Neimela M, Takkunen JT. Control impaired vagal heart rate control in coronary heart disease. Br Heart J 1988;58:592–7.

[2] Anderson KM, Wilson PW, Odell PM, Kanell WB. An updated coronary risk profile: a statement for health professionals. Circulation 1991;86:356–61.

[3] Azuaje F, Dubitzky W, Lopes P, Black N, Adamson K, Wu X, White J. Discovery of incomplete knowledge in electrocardiographic data, Proceedings of the Third International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, Singapore: World Scientific, 1998:286–94.

[4] Azuaje F, Dubitzky W, Lopes P, Black N, Adamson K, Wu X, White J. Knowledge discovery in electrocardiographic data based on neural clustering algorithms, Proceedings of the Vlll Mediterranean Conference on Medical and Biological Engineering and Computing (MEDICON'98), ISBN 9963-607-13-6.

[5] Carpenter GA, Grossberg S, Markuzon N, Reynolds JH, Rosen DB. Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans Neural Netw 1992;3:698–712.

[6] Cohen M, Hudson D, Deedwania P, Combination of chaotic and neural network modeling for diagnosis of heart failure, Proceedings of the International Conference on Computers and their applications 1997;1:254–7.

[7] Coronary heart disease: an epidemiological overview (HMSO, London, 1994) The Health of the nation.

[8] Dickhaus H, Heinrich H. Classifying biosignals with wavelet networks, IEEE Engineering in Medicine and Biology, Sept–Oct, 1996;15:103–11.

[9] Downs J, Harrinson R, Lee R, Cross S. Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks. Artif Intell Med 1996;8:403–28.

[10] Figliola A, Serrano E. Analysis of physiological time series using waveless transform, IEEE Engineering in Medicine and Biology, May–Jun, 1997;16:74–9.

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

[12] Jones D, Franklin P. Choosing a network: matching the architecture to the application. In: Maren AJ, Harston C, Pap R, editors. Handbook of Neural Computing Applications. San Diego: Academic Press, 1990:219–32.

[13] Kamen PW, Tonkin AM. Applications of the Poincare plots to heart rate variability: a new measure of functional status in heart failure. Aust NZ J Med 1995;25:18–26.

[14] Klir GJ, Folger TA. Fuzzy Sets, Uncertainty and Information. Englewood Cliffs, NJ: Prentice Hall, 1988.

[15] Kwok-Kei L, Wu P. Fast cellular feature extraction method. In: Information Intelligence and Systems, 1996 IEEE International Conference on Systems, Man and Cybernetics, IEEE, Piscataway, 1996, vol. 1, pp. 261–6.

[16] Lopes PL, Mitchel RH, White JA. The relationships between respiratory sinus arrhythmia and coronary heart disease risk factors in middle aged males. Automedica 1994;16:71–6.

[17] Maren AJ. Multilayer feedforward neural networks I: delta rule learning. In: Maren AJ, Harston C, Pap R, editors. Handbook of Neural Computing Applications. San Diego: Academic Press, 1990:85–105.

[18] Myers GA, Martin GJ, Magid NM, Barnett PS, Schaad JW, Weiss JS, Lesch M, Singer DH. Power spectral analysis of heart rate variability in sudden death: comparison to other methods. IEEE Trans Biomed Eng 1986;33:1149–56.

[19] Nuttall A. Wigner distribution function: relation to short-term spectral estimation, smoothing, and performance in noise, Naval Underwater Systems Center, Technical Report 8225, 1988.

[20] Reategui E, Campbell J, Leao B. Combining a neural network with case-based reasoning in a diagnostic system. Artif Intell Med 1997;9:5–27.

[21] Ripley BD. Pauern Recognition and Neural Networks. Cambridge: Cambridge University Press, 1996.

[22] Rumelhart DE, Hinton GE, Willians RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. Parallel Distributed Processing: Explorations in Microfeature of Cognition, vol. 1. Cambridge, MA: MIT Press, 1986:318–63.

[23] Shaper AG, Pocock SJ, Phillips AN, Walker M. A scoring system to identify men at high risk of heart attack. Health Trends 1987;19:37–9.

[24] Shen Z, Clarke M, Jones R, Alberti T. A neural network approach to the detection of coronary artery disease. In: Computers in Cardiology 1993 (IEEE Computer Society Press, Los Alamitos, 1993), pp. 221–4.

[25] Skapura D. Building Neural Networks. New York: ACM Press, 1995.

[26] Sosnowski M, Petelenz T, Leski J. Return maps: a non-linear method for the evaluation of respiratory sinus Arrhythmia, in: Computers in Cardiology 1994 (IEEE Computer Society Press, Los Alamitos, 1991), pp. 129–32.

[27] Swets J. Measuring the accuracy of diagnostic systems. Science 1988;240:1285–93.

[28] Timmis A, Nathan A. Essentials of Cardiology. Oxford: Blackwell, 1993.

[29] Woo MA, Stevenson W, Moser D, Trelease R, Harper R. Patterns of beat-to-beat heart rate variability in advanced heart failure. Am Heart J 1992;123:704–11.