In: Pattern Recognition in Biology
Editor: Marsha S. Corrigan, pp. 191-206
ISBN: 978-1-60021-716-6

Chapter 6

MODELING OF AMBULATORY HEART RATE USING LINEAR AND NEURAL NETWORK APPROACHES

Vitaliy Kolodyazhniy[a], Monique C. Pfaltz[b] and Frank H. Wilhelm[c]
Clinical Psychophysiology Laboratory,
Department of Clinical Psychology and Psychotherapy,
Institute for Psychology, University of Basel, CH-4055 Basel, Switzerland

Abstract

This chapter presents new results on modeling 24 hour (circadian) human heart rate data collected with the LifeShirt system using a variety of linear regression and neural network models. Such modeling is important in biopsychology, chronobiology, and chronomedicine where signals collected continuously from human subjects for one or several days need to be interpreted. Ambulatory heart rate is influenced by a variety of factors, including physical activity, posture, and respiration, and our models try to predict heart rate based on these factors. The analyses described in the chapter indicate that neural and especially neuro-fuzzy techniques provide better results in the modelling of human heart rate at the circadian scale than conventional linear regression. The advantages of the neuro-fuzzy approaches consist in their computational efficiency, better interpretability, and the possibility of incorporation of prior knowledge for easier model construction.

1. Introduction

Biological systems are very complex and their behavior is typically influenced by a large number of factors. Consider the human heart rate (HR): at a shorter time scale (seconds to minutes) it is influenced by respiration, speech, physical activity, and body posture. At a longer scale (minutes to hours), HR is influenced by emotions, food intake, and thermoregulation (environmental and body temperature). At even longer scale, HR is

[a] E-mail address: v.kolodyazhniy@unibas.ch
[b] E-mail address: monique.pfaltz@unibas.ch
[c] E-mail address: frank.wilhelm@unibas.ch

influenced by the circadian (near 24 hour long) patterns of everyday life: during sleep our hearts beat slower than when we are awake.

The character of these dependencies is often assumed to be linear for the ease of modeling. Linear techniques have been successfully applied to modeling the human HR at shorter time scales, e.g. to discriminate between physical and emotional activation causing an increase in HR [23].

However, a general linear model of the human HR is hardly feasible because of the complexity of nonlinear static and dynamic behavior of the regulation systems of our organisms, including the internal clock, the thermoregulation system, the cardiovascular system, and the nonlinear interactions of these systems.

Circadian variations of the human HR are of interest for biopsychology, chronobiology, and chronomedicine. In order to detect such variations, and especially to reliably estimate the underlying parameters of the circadian clock like its phase, amplitude, and free-running period (which is usually slightly different from 24 hours), nonlinear modeling approaches are required. This is a new and challenging field, and our goal here is to present a comparison of different modeling approaches that can be used for the investigation of the human circadian clock based on ambulatory measurements, when test subject keep their everyday life schedule while their physiological parameters are continuously recorded with a wearable device throughout several days or weeks.

Artificial neural networks (ANNs) [8] have been widely used in recent years to solve a wide range of problems such as data mining and processing of signals of different nature under uncertainty as to the structure and parameters of the underlying model. The ANNs possess universal approximation properties and learning capabilities, so they can be trained to identify unknown and very complex nonlinear input-output relationships. However, the ANNs represent the black-box approach to systems modelling since they are nontransparent models, and the interpretation of the knowledge stored in an ANN can be difficult. We will consider three popular types of neural networks for nonlinear regression: the multilayer perceptron (MLP), the radial basis function network, and the generalized regression neural network (GRNN).

Hybrid neuro-fuzzy approaches [5] emerged as a synergism of the neural nets and fuzzy systems [11], the two major directions in computational intelligence [10]. The neuro-fuzzy systems possess the learning capabilities similar to those of neural networks, and provide the interpretability and “transparency” of results inherent to the fuzzy approach. We will consider here two neuro-fuzzy models, the neo-fuzzy neuron (NFN) and the novel neuro-fuzzy Kolmogorov’s network (NFKN) that were designed to overcome such disadvantages of most neuro-fuzzy systems such as the slow convergence of the gradient-based learning procedures or high computational load of the genetic algorithm-based optimization of their parameters.

2. Regression Models

In this section, we consider a number of regression models that will be further used for modeling the ambulatory HR. The most general form of such models is

$$\hat{y} = \hat{f}(x), \; x = [x_1, \ldots, x_d]^T, \tag{1}$$

where $\hat{y}$ is the model output, x is the vector of inputs, $\hat{f}$ is the estimated input-output functional relation. We assume that the true function $y = f(x)$ is not known.

2.1. Linear Regression

The simplest model that we consider is given by the linear multiple regression equation:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d \,, \tag{2}$$

where $w_0, w_1, \ldots, w_d$ are the regression parameters.

2.2. Multilayer Perceptron

A multilayer perceptron (MLP) [8] is shown in Fig. 1. This is a feed-forward network of artificial neurons with sigmoid activation functions. The number of layers in an MLP is usually two or three (layers of neurons, input layer is not counted).

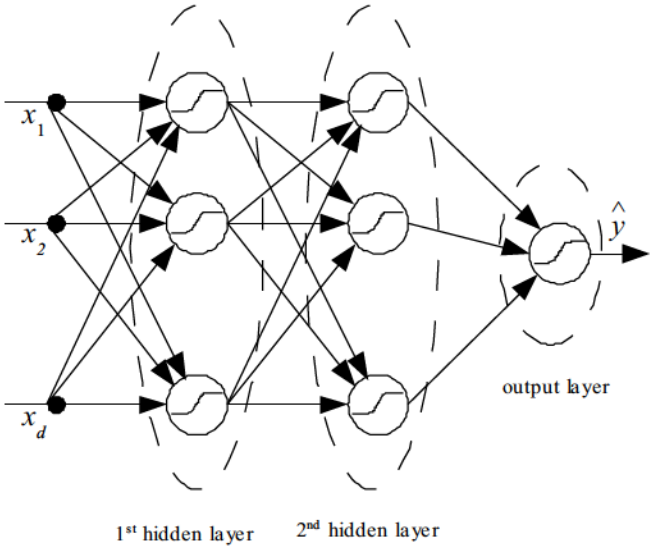

Figure 1. Multi-layer perceptron with 2 hidden layers.

The MLP is described by the following equation [18]:

$$\hat{y} = \Gamma^{[n_L]}(W^{[n_L]}\Gamma^{[n_L-1]}(W^{[n_L-1]}\ldots\Gamma^{[1]}(W^{[1]}x + w_0^{[1]}) + \ldots + w_0^{[n_L-1]}) + w_0^{[n_L]}) \,, \tag{3}$$

where $W^{[l]}$ is the matrix $(n^{[l]} \times n^{[l-1]})$ of the synaptic weights of the l-th layer, $n^{[l]}$ is the number of neurons in the l-th layer, $l = 1, \ldots, n_L$, $n^{[0]} = d$, $n^{[n_L]} = n_y$, n_L is the number of layers, x is the input vector $(d \times 1)$, $w_0^{[l]}$ is the vector $(n^{[l]} \times 1)$ of bias parameters of the l-th

layer, $\Gamma^{[i]}$ is a nonlinear operator implemented by a set of sigmoid activation functions of the l-th layer.

2.3. Radial Basis Function Network

A radial basis function network (RBFN) [1, 6] always contains two layers: the hidden layer of basis functions and the output layer (see Fig. 2).

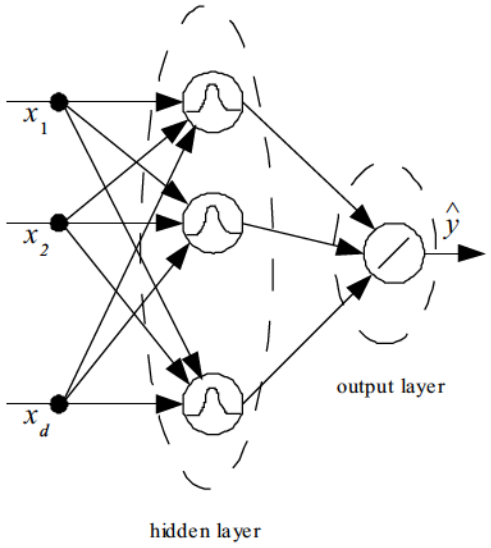

Figure 2. Radial basis function network.

The RBFN model is described by the equation

$$\hat{y} = \sum_{i=1}^{n_h} w_i \phi(\|x - v_i\|) + w_0, \tag{4}$$

where n_h is the number of neurons (basis functions) in the hidden layer, w_i is the weight of the i-th basis function, ϕ is the basis function (usually Gaussian), v_i is the vector $(n_d \times 1)$ of the center of the i-th basis function, w_0 is the bias parameter.

The most important advantage of the RBF networks consists in shorter training time as compared to the conventional multilayer networks trained by means of the error backpropagation technique. At the same time, the construction and training of an RBF network usually requires the use of at least two procedures. The first one is for the setting of the basis function parameters and the second one is for the training of the output layer weights.

It was shown in [9] that the RBFNs and neuro-fuzzy systems are functionally equivalent under minor restrictions, so similar training procedures can be used for both. The RBFNs are also considered to be more transparent than the MLP models because the radial basis

functions have a clear interpretation in terms of clustering of the input space in contrast to the neurons of the MLPs, especially with more than one hidden layer.

2.4. Generalized Regression Neural Network

A generalized regression neural network (GRNN) [19] has a similar architecture to that of the RBFN but it has as many hidden layer units (basis functions) as the number of samples in the training data set. So each sample from the training data set is memorized in the GRNN as a prototype.

2.5. Neo-fuzzy Neuron

The neo-fuzzy neuron (NFN) was proposed as a simple neuro-fuzzy architecture with very fast learning and guarantee for the convergence to the global minimum of the error surface [25]. It is also very well suited for hardware implementations, including purely analog circuits [17]. This neuron has been successfully applied to the problems of time series prediction, signal filtering, and restoration [20].

The NFN model is described by the expressions

$$\hat{y} = \sum_{i=1}^{d} f_i(x_i), \quad f_i(x_i) = \sum_{h=1}^{m_i} \mu_{i,h}(x_i) w_{i,h}, \tag{5}$$

where $f_i(x_i)$ is the nonlinear synapse of the i-th input, m_i is the number of membership functions per input i, $\mu_{i,h}(x_i)$ is the h-th membership function (MF) at the i-th input, $w_{i,h}$ is the tunable synaptic weight of the respective MF. These weights are tuned automatically by an optimization procedure that is used to fit the model to the data. The architecture of an NFN is shown in Fig. 3.

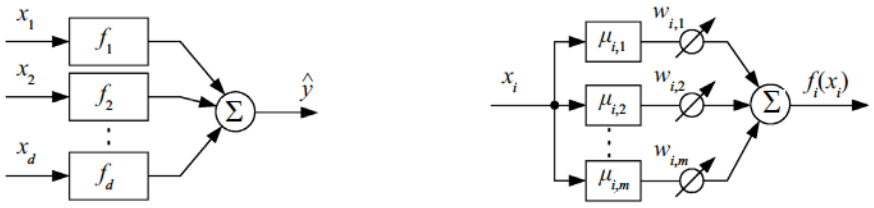

Figure 3. Neo-fuzzy neuron (*left*) and its nonlinear synapse (*right*).

The MFs in a neo-fuzzy neuron are fixed and equidistantly spaced. They are chosen such that at maximum two adjacent MFs in each synapse fire at a time and their sum is always unity:

$$\sum_{h=1}^{m_i} \mu_{i,h}(x_i) = 1, \quad i = 1,\ldots,d. \tag{6}$$

The functions of nonlinear synapses $f_i(x_i)$ are piecewise-linear since they are superpositions of triangular membership functions. An example of an approximation of a univariate nonlinear function by a nonlinear synapse is shown in Fig. 4 where the unknown approximated function is plotted as dotted line.

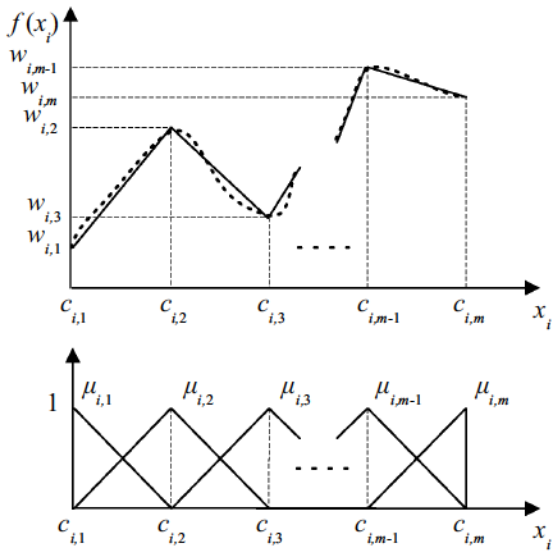

Figure 4. Approximation of a univariate nonlinear function by a nonlinear synapse.

Each nonlinear synapse is a single-input single-output fuzzy inference system. It contains fuzzy rules in the form:

$$\text{IF } x_i \text{ IS } X_{i,h} \text{ THEN } f_i = w_{i,h}, \quad i = 1,\ldots,d, \quad h = 1,\ldots,m_i, \tag{7}$$

where $X_{i,h}$ is the h-th linguistic label (fuzzy set) on the i-th input. The fuzzy sets are defined by the respective triangular MFs $\mu_{i,h}(x_i)$, and represent such linguistic values as “SMALL”, “MEDIUM”, “LARGE”, etc, depending on a particular fuzzy partition which can be chosen based on the domain knowledge and experience of the person that designs the model. In such a way, prior knowledge can be incorporated into the model and the neo-fuzzy model itself possesses inherent interpretability along with nonlinear approximation capabilities.

A disadvantage of the NFN model consists in the fact that it assumes that the nonlinearities of the inputs are separable. So the NFN model is *not* a universal approximator, in contrast to the neural nets [7] considered above or to the conventional fuzzy systems [16, 21]. But it possesses better approximation properties than the linear models, as it has more degrees of freedom allowing good piecewise-linear approximation for many processes and imposing very low requirements on the computational resources. The most important advantage of the NFN model is its extremely fast learning which can be performed in the online mode with very simple weight update rules [25, 2], or in just a single operation with the linear least-squares formula [3].

2.6. Neuro-fuzzy Kolmogorov's Network

The neuro-fuzzy Kolmogorov's network (NFKN) is a universal approximator based on the NFNs [14]. The NFKN architecture [4, 15] is comprised of two layers of neo-fuzzy neurons (NFNs) [25] and is described as follows:

$$\hat{f}(x_1,\ldots,x_d)=\sum_{l=1}^{n} f_l^{[2]}(o^{[1,l]}), \quad o^{[1,l]}=\sum_{i=1}^{d} f_i^{[1,l]}(x_i), \quad l=1,\ldots,n, \tag{8}$$

where n is the number of hidden layer neurons, $f_l^{[2]}(o^{[1,l]})$ is the l-th nonlinear synapse in the output layer, $o^{[1,l]}$ is the output of the l-th NFN in the hidden layer, $f_i^{[1,l]}(x_i)$ is the i-th nonlinear synapse of the l-th NFN in the hidden layer.

The NFKN architecture was named after the famous Kolmogorov's theorem on the representation of functions of multiple variables as a two-level superposition of univariate functions [13]. Indeed, the NFKN makes approximate representations of multivariate functions by a two-level superposition (8) of the nonlinear synapses (univariate approximators) of its hidden and output layers (see Fig. 5).

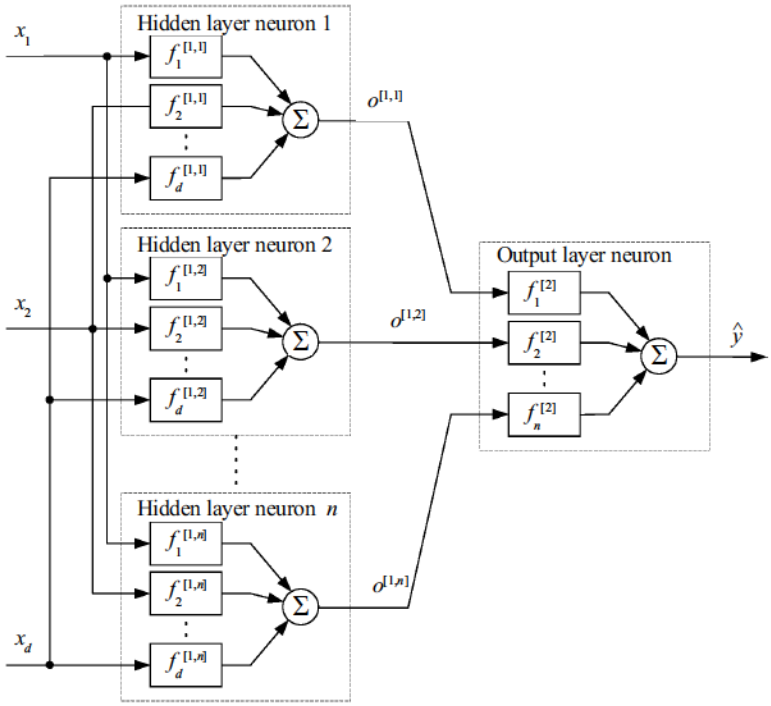

Figure 5. NFKN with d inputs and n hidden layer neurons.

The equations for the hidden and output layer synapses are

$$f_i^{[1,I]}(x_i) = \sum_{h=1}^{m_{1,i}} \mu_{i,h}^{[1]}(x_i) w_{i,h}^{[1,I]}, \quad f_l^{[2]}(o^{[1,I]}) = \sum_{j=1}^{m_{2,I}} \mu_{l,j}^{[2]}(o^{[1,I]}) w_{l,j}^{[2]},$$
$$l = 1,\ldots,n, \quad i = 1,\ldots,d, \tag{9}$$

where $m_{1,i}$ and $m_{2,I}$ are the number of membership functions (MFs) per input i and l in the hidden and output layers respectively, $\mu_{i,h}^{[1]}(x_i)$ and $\mu_{l,j}^{[2]}(o^{[1,I]})$ are the MFs, $w_{i,h}^{[1,I]}$ and $w_{l,j}^{[2]}$ are the tunable weights. The MFs in the NFKN at each input in the hidden and output layers are shared between all neurons of the respective layer.

The outputs of the NFKN are computed via the following two-stage fuzzy inference procedure:

$$\hat{y} = \sum_{l=1}^{n} \sum_{j=1}^{m_{2,I}} \mu_{l,j}^{[2]} \left[\sum_{i=1}^{d} \sum_{h=1}^{m_{1,i}} \mu_{i,h}^{[1]}(x_i) w_{i,h}^{[1,I]} \right] w_{l,j}^{[2]}. \tag{10}$$

Since the nonlinear synapses represent local piecewise-liner approximations, the overall NFKN model is also piecewise-linear and does not involve any operations other than addition and multiplication in computing its input-output mapping. So a trained NFKN model can be easily implemented even on embedded processors with very limited computational resources.

The description (10) corresponds to the following two-level fuzzy rule base:

$$\text{IF } x_i \text{ IS } X_{i,h} \text{ THEN } o^{[1,1]} = w_{i,h}^{[1,1]} d \text{ AND} \ldots \text{AND } o^{[1,n]} = w_{i,h}^{[1,n]} d,$$
$$i = 1,\ldots,d, \quad h = 1,\ldots,m_{1,i}, \tag{11}$$

$$\text{IF } o^{[1,I]} \text{ IS } O_{l,j} \text{ THEN } \hat{y} = w_{l,j}^{[2]} n, \quad l = 1,\ldots,n, \quad j = 1,\ldots,m_{2,I}, \tag{12}$$

where $X_{i,h}$ and $O_{l,j}$ are the antecedent fuzzy sets in the first and second level rules, respectively. Thus, the NFKN is an interpretable model, and the domain knowledge can be used in designing the model in a similar way as for the NFNs.

3. Experiments

In this section, we describe the application of the models introduced in section 2 to the prediction of heart rate in healthy test subjects. The prediction is based on the measurements of physical activity, posture, and respiration. A comparison of the models w.r.t. their prediction quality is presented.

3.1. Participants

Participants were 18 healthy adults (6 men and 12 women), mean age: 39.27 years, standard deviation (SD): 11.67; mean body mass index: 22.78, SD: 3.22. These test subjects were

recruited by means of postings and local newspaper advertisements. Participants on medication with direct effects on the autonomic nervous system were excluded. The study was approved by the local ethics committee for medical research and participants received a reimbursement of 200 CHF (approximately 160 USD).

3.2. Data Acquisition

Physiological recordings were performed by means of the LifeShirt System (VivoMetrics, Inc., Ventura, CA, USA). The LifeShirt is a non-invasive ambulatory monitoring system consisting of a data recorder, a garment with embedded sensors for continuous monitoring of ECG, respiration, 3-D accelerometry and other functions and a software package (VivoLogic®) for offline signal analysis. For detailed description of the LifeShirt system, see [22, 24].

The data recorder can be carried in a waist pack and the data is written to a flash memory card. Data were recorded at 10 Hz (accelerometry), 200 Hz (ECG), and 50 Hz (respiratory pattern), respectively. The following measures were automatically derived by means of the VivoLogic Software for the present analysis: heart rate ('HR', beats/min), motion ('AccM', summed absolute values of acceleration along 2 axes), posture ('AccP'), minute ventilation ('Vent', liters inhaled/min), tidal volume ('Vt', ml), breaths per minute ('Br/M').

3.3. Procedure

On arrival in the laboratory (between 8 and 9am), participants gave written informed consent. Subsequently, the sensors and the LifeShirt were attached, the data recording was started and the respiration signal was calibrated against a known fixed volume. Before leaving the laboratory, participants were instructed to pursue their regular everyday activities and to stop the recording after awakening the next morning. To avoid artifacts, they were advised not to pursue any sports.

3.4. Modeling

For modelling purposes, the data were downsampled to 30 sec sampling period (1/30 Hz). The posture channel was divided into 3 separate channels for supine, lateral, and upright body positions. With the other channels, it resulted in 7 input variables (one for 'AccM', three for 'AccP', and another three for the 'Vent', 'Vt', and 'Br/M' respectively). The dependent variable was the heart rate ('HR').

Mean length of a data set for one test subject was 2583.6 samples (21.53 hours), SD: 177.33 samples (1.48 hours). An example of a downsampled heart rate recording is shown in Fig. 6. A clear circadian pattern is visible: the heart rate is reduced and remains more stable during the night, while during the day it is on average higher due to the opposite phase of the circadian clock and has more variations because of physical activity, emotions, etc.

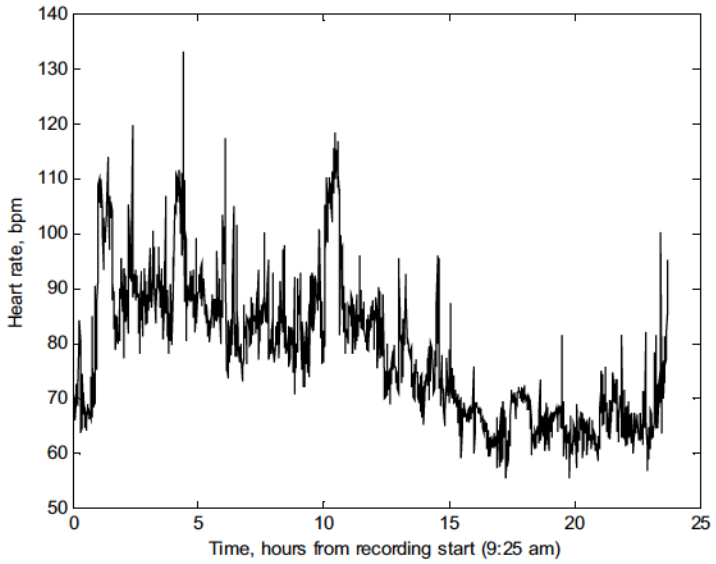

Figure 6. Heart rate recording for test subject N18.

The parameters of the compared models are listed in Table 1. The column 'Architecture' shows the number of inputs (first value), the number of outputs (last value), and the number of neurons in the respective intermediate layers (if applicable). The fewer parameters a model has, the better it is from the point of view of computational complexity.

Table 1. Models used for heart rate prediction (best values in bold, second best in italics)

Model	Activation function	Architecture	Number of parameters
Linear	Linear	**7-1**	**8**
MLP	Sigmoid	7-10-10-1	201
RBFN	Gaussian	7-25-1	201
GRNN	Gaussian	7-1292*-1	10336*
NFN	Triangular	**7-1**	*38*
NFKN	Triangular	*7-4-1*	104

*average value

For the linear model the parameters were the respective regression coefficients from (2). For the MLP, NFN, and NFKN models the only adjustable parameters were their synaptic weights, while for the RBFN and GRNN models these included also the respective prototype vectors v_i sized (7×1) for each basis function ϕ in (4). The radii of the basis functions were fixed and equal to 2.7 (RBFN) and 0.75 (GRNN) for all the basis functions. The NFN model had 8 MFs per 'AccM', 'Vent', 'Vt', and 'Br/M' inputs, and 2 MFs per each of the 3 'AccP' inputs. The NFKN model had 3 MFs per 'AccM', 'Vent', 'Vt', and 'Br/M' inputs, 2 MFs per each of the 3 'AccP' inputs in the hidden layer, and 8 MFs per each of the 4 synapses in its output layer. All model structures were chosen based on best performance.

The data sets for each subject were divided into two parts: one with samples with odd numbers and another one with samples with even numbers. For each of the compared models we performed two runs of testing: in the first run the samples with odd numbers were used for training, the samples with even numbers for checking, and in the second run vice versa. In each of the two runs the training and checking procedures were repeated 18 times with the respective training and checking data sets for all the 18 test subjects. For each subject, the data were standardized as follows: the means of all the variables in the training set were subtracted from both the training and checking sets, and then both the training and checking sets were divided by the respective standard deviations of the training set.

3.5. Learning Algorithms

The parameters of all the models during the training phase were estimated by minimizing the sum of squared errors on the training data set

$$E = \sum_{k=1}^{N} [y(k) - \hat{y}(k)]^2 = (Y - \hat{Y})^T (Y - \hat{Y}), \tag{13}$$

where N is the number of samples in the training set, $y(k)$ is the real output (target value) for the input vector $x(k) = [x_1(k), \ldots, x_d(k)]^T$ corresponding to the k-th sample from the training set, $\hat{y}(k)$ is the model output for the k-th sample, Y is the vector of target values, and $\hat{Y}$ is the vector of model outputs:

$$Y = [y(1), y(2), \ldots, y(N)]^T, \quad \hat{Y} = [\hat{y}(1), \hat{y}(2), \ldots, \hat{y}(N)]^T. \tag{14}$$

The models and their training procedures were implemented in Matlab 7.0. The parameters of the linear model and the NFN were found with the conventional linear least squares method.

The MLP, RBFN, and GRNN models were programmed with the Matlab Neural Network Toolbox. For training the MLP, the resilient propagation algorithm was used (function *trainrp*), which is considered to be one of the best training procedures for the MLPs, combining high speed and high precision. The RBFN and GRNN models were constructed with the *newrb* and *newgrnn* functions, respectively.

For the NFKN model, a separate Matlab toolbox was developed. The training of the NFKN model is based on the hybrid algorithm which involves linear least-squares optimization for the output layer and gradient-based learning of the hidden layer weights. The overall training algorithm is very fast and does not involve any nonlinear operations [15].

3.6. Results

The coefficient of determination R^2, indicating the proportion of "explained variance", was used to estimate the prediction quality. The R^2 index was computed as

$$R^2 = 1 - \frac{\sum_{k=1}^{N} e^2(k)}{\sum_{k=1}^{N} (y(k) - \bar{y})^2}, \quad \bar{y} = \frac{1}{N} \sum_{k=1}^{N} y(k), \tag{15}$$

where k is the number of a measurement in the sample Y, $e(k) = y(k) - \hat{y}(k)$ is the modelling error (residual) for the measurement k, and $\bar{y}$ is the mean value of the sample Y. The performance of the models in the first and second runs is summarized in Tables 2 and 3, respectively. R^2_{trn} stands for training, and R^2_{chk} for checking.

The fourth and the last columns in Tables 2 and 3 show the ratio of the R^2 indices w.r.t. their respective standard deviations $SD(R^2)$. Good performance of a model is indicated by high values of R^2 and low values of $SD(R^2)$ resulting in high values of the ratio $R^2 / SD(R^2)$.

The values of R^2_{chk} for individual test subjects for the runs 1 and 2 are shown in Fig. 7 and 8, respectively. The R^2_{chk} curves for the linear models in Figs. 7 and 8 are below all the respective curves for the nonlinear models, so the R^2_{chk} for the nonlinear models was always higher than that for the linear ones.

To test the significance of these results we used the nonparametric Wilcoxon matched pairs test on R^2_{chk} for all 18 test subjects. In both runs, the p-values for R^2_{chk} of each nonlinear model compared to the linear one are smaller than 0.0002, i.e. the results presented here are highly significant in the statistical sense.

Note that the NFN and the NFKN models had fewer parameters than the other nonlinear models. Nevertheless, the simplest NFN model with only 38 parameters outperformed the most complex GRNN models with more than 10000 parameters on average, and the NFKN model with 104 parameters performed like the RBFN and MLP models with 201 parameters each.

Table 2. Performance of the models in the 1st run (best values in bold, second best in italics)

Model	Mean R^2_{trn}	$SD(R^2_{trn})$	Mean $\frac{R^2_{trn}}{SD(R^2_{trn})}$	Mean R^2_{chk}	$SD(R^2_{chk})$	Mean $\frac{R^2_{chk}}{SD(R^2_{chk})}$
Linear	0.7622	0.0552	13.7986	0.7632	0.0566	13.4851
MLP	*0.8269*	**0.0411**	**20.1334**	**0.8145**	**0.0453**	**17.9920**
RBFN	0.8215	0.0440	18.6637	*0.8092*	0.0483	16.7421
GRNN	0.8131	0.0478	17.0092	0.7969	0.0527	15.1196
NFN	0.8020	0.0449	17.8543	0.7915	0.0496	15.9639
NFKN	**0.8320**	*0.0427*	*19.4870*	0.8042	*0.0461*	*17.4329*

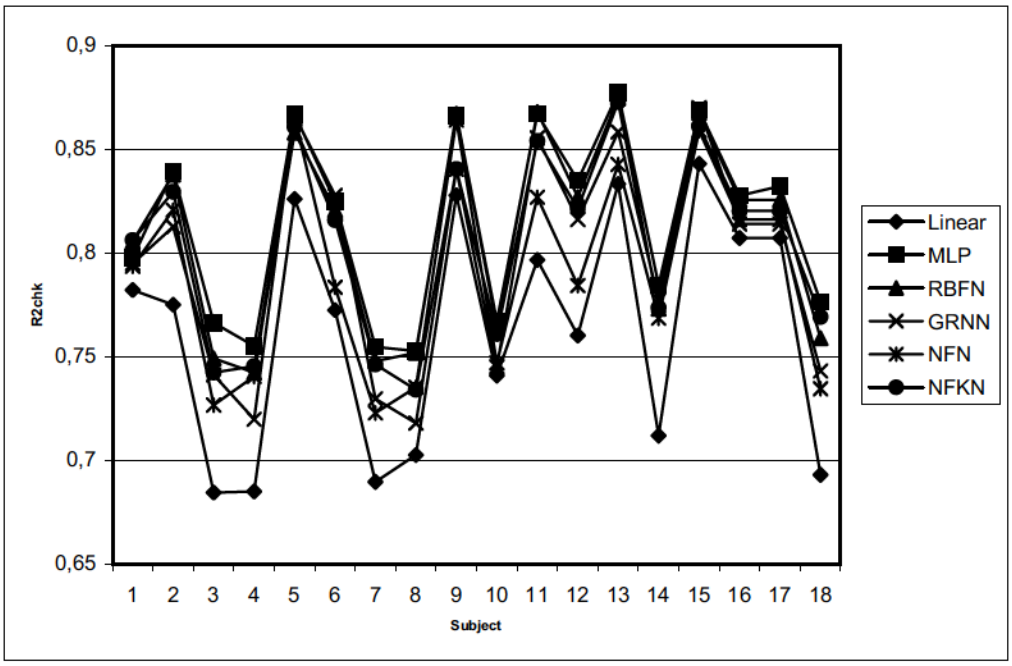

Figure 7. Values of R^2_{chk} for the 1st run for the compared models.

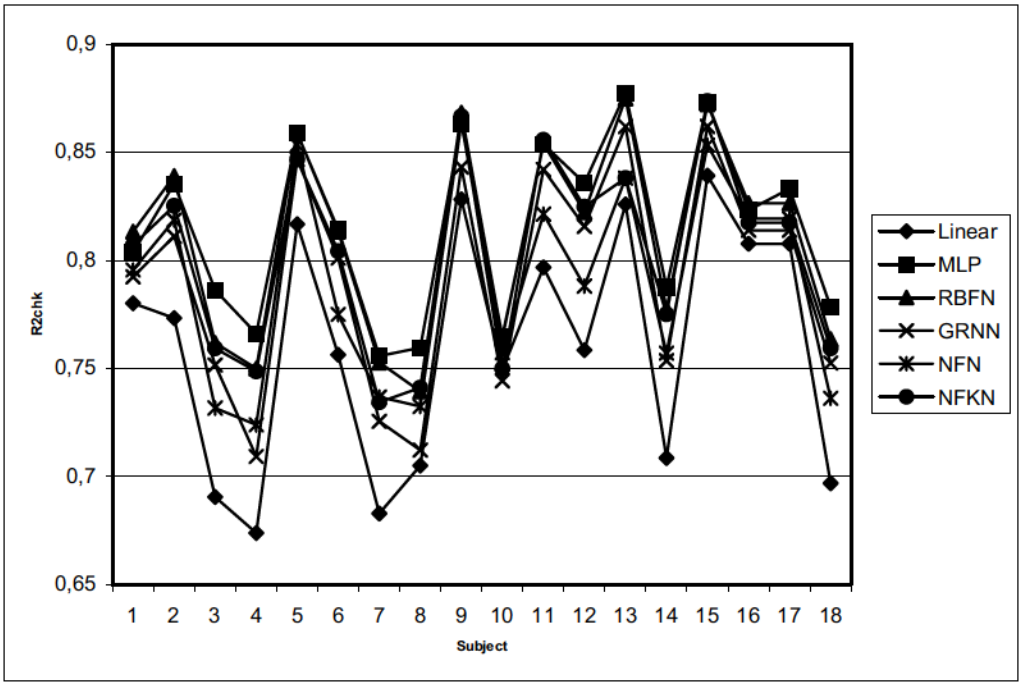

Figure 8. Values of R^2_{chk} for the 2nd run for the compared models.

Table 3. Performance of the models in the 2nd run (best values in bold, second best in italics)

Model	Mean R^2_{trn}	$SD(R^2_{trn})$	Mean $\frac{R^2_{trn}}{SD(R^2_{trn})}$	Mean R^2_{chk}	$SD(R^2_{chk})$	Mean $\frac{R^2_{chk}}{SD(R^2_{chk})}$
Linear	0.7645	0.0561	13.6238	0.7610	0.0558	13.6458
MLP	*0.8318*	**0.0424**	**19.6337**	**0.8150**	**0.0411**	**19.8062**
RBFN	0.8230	0.0470	17.5276	*0.8099*	0.0465	17.3986
GRNN	0.8161	0.0503	16.2402	0.7918	0.0511	15.4938
NFN	0.8030	0.0485	16.5533	0.7900	0.0476	16.6039
NFKN	**0.8332**	*0.0452*	*18.4324*	0.8024	*0.0455*	*17.6224*

4. Conclusion

The results presented in this chapter indicate that nonlinear models provide better accuracy in predicting human heart rate at the circadian scale than conventional linear models. These results are significant from the statistical point of view as confirmed by the nonparametric Wilcoxon test.

Neuro-fuzzy models, such as the NFN and the NFKN, are especially attractive because they provide the same level of accuracy as neural networks but have low computational complexity of both the input-output mapping and the training procedures, and are more transparent (interpretable). Another important advantage of the considered neuro-fuzzy models (the NFN and the NFKN) is their capability of working in high dimensions of input space because their computational complexity is increased *linearly* with the increase of the input dimension [4]. This is important for the development of complex models with many input variables.

Based on the presented results, it is expected that neural and neuro-fuzzy techniques described here will find application in analyzing human circadian rhythms. Linear techniques do not appear to provide reliable results for ambulatory monitoring where test subjects live their normal everyday life rather than stay in the controlled laboratory environment [12].

For the reliable detection of circadian parameters such as the phase of the internal clock based on indirect measurements of physiological parameters from, e.g., cardiovascular, respiratory, and temperature recordings, it will be necessary to solve a number of problems related to data preparation for the nonlinear models and estimating variance from between-individual differences in responsivity to behavioral variations such as physical activity. These problems include reliable calibration of the recording devices, signal filtering, and adequate preprocessing.

Acknowledgements

This work was supported by the 6th Framework Project EUCLOCK (No. 018741) and the Swiss National Science Foundation (grant 105311-105850).

References

[1] Billings, S.A. & Hong, X. (1998). Dual-orthogonal radial basis function networks for nonlinear time series prediction. *Neural Networks,* **11**, 479-493.

[2] Bodyanskiy, Ye., Kokshenev, I., & Kolodyazhniy, V. (2003). An adaptive learning algorithm for a neo fuzzy neuron. *Proc.* **3***rd Int. Conf. of European Union Society for Fuzzy Logic and Technology (EUSFLAT'2003), Zittau, Germany*, 375-379.

[3] Bodyanskiy, Ye., Kokshenev, I., & Kolodyazhniy, V. (2004). *A learning fuzzy nonlinear filter. Proc.* **9***th Biennial Baltic Electronics Conference (BEC'2004). Tallinn Univ. of Technology, Tallinn, Estonia,* 141-144.

[4] Bodyanskiy, Ye., Kolodyazhniy, V., & Otto, P. Neuro-Fuzzy Kolmogorov's Network for Time Series Prediction and Pattern Classification. In: Furbach U. editor. *Lecture Notes in Computer Science Series: Lecture Notes in Artificial Intelligence*, Vol. 3698, Berlin Heidelberg: Springer-Verlag; 2005; 191-202.

[5] Borgelt, C., Klawonn, F., Kruse, R., & Nauck, D. Neuro-Fuzzy-Systeme. Von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen. Reihe Computational Intelligence. Braunschweig: Vieweg; 2003.

[6] Chang, E.S., Chen, S., & Mulgrew, B. (1996). Gradient radial basis neural networks for nonlinear time series prediction. *IEEE Trans. on Neural Networks,* **7**, 190-194.

[7] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. of Control, Signals, and Systems,* **2**, 303-314.

[8] Haykin, S.: Neural Networks: A Comprehensive Foundation. Upper Saddle River: Prentice Hall; 1999.

[9] Jang, J.-S. R., & Sun, C.-T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Fuzzy Systems,* **4**,156-159.

[10] Jang, J.-S. R., Sun, C.-T., & Mizutani, E. Neuro-Fuzzy and Soft Computing – A Computational Approach to Learning and Machine Intelligence. Upper Saddle River, NJ: Prentice Hall; 1997.

[11] Jin, Y. Advanced Fuzzy Systems Design and Applications. Heidelberg: Physica-Verlag; 2003.

[12] Klerman, E.B., Lee, Y., Czeisler, C.A., & Kronauer, R.E. (1999). Linear demasking techniques are unreliable for estimating the circadian phase of ambulatory temperature data. *J. of Biological Rhythms* **14**, 260-274.

[13] Kolmogorov, A.N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR,* **114**, 953-956.

[14] Kolodyazhniy, V., Bodyanskiy, Ye., & Otto, P. Universal Approximator Employing Neo-Fuzzy Neurons. In: Reusch B. editor. *Computational Intelligence, Theory and Applications. Advances in Soft Computing,* Vol. 33, Berlin Heidelberg: Springer-Verlag; 2005; 631-640.

[15] Kolodyazhniy, V., & Otto, P.: Neuro-Fuzzy Modelling Based on Kolmogorov's Superposition: a New Tool for Prediction and Classification. *Lecture Notes in Informatics*, Vol. P-72, Bonn: Gesellschaft für Informatik; 2005; 273-280.

[16] Kosko, B. (1992). Fuzzy systems as universal approximators. *Proc.* **1***st IEEE International Conference on Fuzzy Systems*, *San Diego, CA,* 1153-1162.

[17] Miki, T., & Yamakawa, T. Analog implementation of neo-fuzzy neuron and its on-board learning. In: Mastorakis N.E. (Ed.), *Computational Intelligence and Applications*, Piraeus: WSES Press; 1999; 144-149.

[18] Narendra, K.S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks,* **1**, 4-27.

[19] Specht, D.F. (1991). A Generalized Regression Neural Network. *IEEE Trans. on Neural Networks* **2**, 568-576.

[20] Uchino, E., & Yamakawa, T. Soft computing based signal prediction, restoration, and filtering. In: Da Ruan editor, Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms. Boston: Kluwer Academic Publishers; 1997; 331-349.

[21] Wang, L.-X. (1992) Fuzzy systems are universal approximators. *Proc.* **1***st IEEE International Conference on Fuzzy Systems*, *San Diego, CA,* 1163-1170.

[22] Wilhelm, F.H., Pfaltz, M.C., & Grossman, P. (2006). Continuous electronic data capture of physiology, behavior and experience in real life: towards ecological momentary assessment of emotion. *Interacting with Computers* **18**, 171-186.

[23] Wilhelm, F.H., & Roth, W.T. (1998). Using minute ventilation for ambulatory estimation of additional heart rate. *Biological Psychology,* **49**, 137-150.

[24] Wilhelm, F.H., Roth, W.T., & Sackner, M. (2003). The LifeShirt: an advanced system for ambulatory measurement of respiratory and cardiac function. *Behavior Modification,* **27**, 671-691.

[25] Yamakawa, T., Uchino, E., Miki, T., & Kusanagi, H. (1992). A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. *Proc.* **2***nd International Conference on Fuzzy Logic and Neural Networks "IIZUKA-***92***", Iizuka, Japan,* 477-483.