

## Research Article

# ECG Prediction Based on Classification via Neural Networks and Linguistic Fuzzy Logic Forecaster

**Eva Volna, Martin Kotyrba, and Hashim Habiballa**

*University of Ostrava, 30 Dubna 22, 70103 Ostrava, Czech Republic*

Correspondence should be addressed to Martin Kotyrba; [martin.kotyrba@osu.cz](mailto:martin.kotyrba@osu.cz)

Received 16 July 2014; Accepted 20 November 2014

Academic Editor: Mohammed Chadli

Copyright © 2015 Eva Volna et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper deals with ECG prediction based on neural networks classification of different types of time courses of ECG signals. The main objective is to recognise normal cycles and arrhythmias and perform further diagnosis. We proposed two detection systems that have been created with usage of neural networks. The experimental part makes it possible to load ECG signals, preprocess them, and classify them into given classes. Outputs from the classifiers carry a predictive character. All experimental results from both of the proposed classifiers are mutually compared in the conclusion. We also experimented with the new method of time series transparent prediction based on fuzzy transform with linguistic IF-THEN rules. Preliminary results show interesting results based on the unique capability of this approach bringing natural language interpretation of particular prediction, that is, the properties of time series.

## 1. Background

Biometrical data is typically represented as an image or a quantification of measured physiological or behavioural characteristics. As this data should refer to very complex human behaviour or describe very precisely physiological characteristic (typically iris scan, fingerprint, palm vein image, hand scan, voice, walk pattern, etc.), this data can easily become very large and hard to process. For this reason, modern ways of data processing and classification are applied for biometrical data. The leading method is the usage of neural networks [1].

For more than four decades, computers have been used in the classification of the electrocardiogram (ECG) resulting in a huge variety of techniques [2] all designed to enhance the classification accuracy to levels comparable to that of a “gold standard” of expert cardiology opinion. Included in these techniques are multivariate statistics, decision trees, fuzzy logic, expert systems, and hybrid approaches [3]. The recent interest in neural networks coupled with their high levels of performance has resulted in many instances of their application in this field [4].

The electrocardiogram is a technique of recording bioelectric currents generated by the heart. Clinicians can evaluate the conditions of a patient’s heart from the ECG and perform further diagnosis. ECG records are obtained by sampling the bioelectric currents sensed by several electrodes, known as leads. A typical one-cycle ECG tracing is shown in Figure 3.

*1.1. Backpropagation Neural Networks.* A neural network is a parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected together with unidirectional signal channels called connections. Each processing element has a single output connection which branches into as many collateral connections as desired (each carrying the same signal, the processing element output signal). The processing element output signal can be of any mathematical type desired. All of the processing that goes on within each processing element must be completely local: that is, it must depend only upon the current values of the input signals

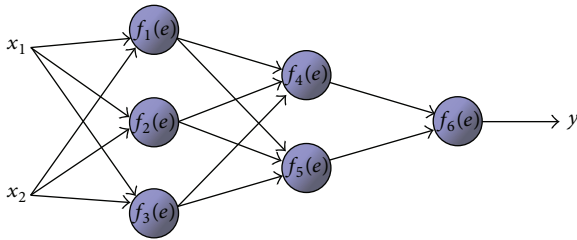


FIGURE 1: A backpropagation network architecture.

arriving at the processing element via impinging connections and upon values stored in the processing element's local memory [5].

The backpropagation neural network architecture is a hierarchical design consisting of fully interconnected layers or rows of processing units (with each unit itself comprised of several individual processing elements). Backpropagation belongs to the class of mapping neural network architectures and therefore the information processing function that it carries out is the approximation of a bounded mapping or function  $f : A \subset R^n \rightarrow R^m$ , from a compact subset  $A$  of  $n$ -dimensional Euclidean space to a bounded subset  $f[A]$  of  $m$ -dimensional Euclidean space, by means of training on examples  $(x_1, z_1), (x_2, z_2), \dots, (x_k, z_k), \dots$ . It will always be assumed that such examples of a mapping  $f$  are generated by selecting  $\mathbf{x}_k$  vectors randomly from  $A$  in accordance with a fixed probability density function  $p(\mathbf{x})$ . The operational use to which the network is to be put after training is also assumed to involve random selections of input vectors  $\mathbf{x}$  in accordance with  $p(\mathbf{x})$ . The backpropagation architecture described in this paper is the basic, classical version (Figure 1). The backpropagation learning algorithm is composed of two procedures: (a) forward propagation of signals and (b) backpropagation weight training [5].

*Feed-Forward.* Assume that each input factor in the input layer is denoted by  $x_i$ ; the  $y_j$  and  $z_k$  represent the output in the hidden layer and the output layer, respectively. And the  $y_j$  and  $z_k$  can be expressed as follows (1):

$$\begin{aligned} y_j &= f(X_j) = f\left(w_{oj} + \sum_{i=1}^I w_{ij}x_i\right), \\ z_k &= f(Y_k) = f\left(w_{ok} + \sum_{j=1}^J w_{jk}y_j\right), \end{aligned} \quad (1)$$

where the  $w_{oj}$  and  $w_{ok}$  are the bias weights for setting threshold values,  $f$  is the activation function used in both hidden and output layers, and  $X_j$  and  $Y_k$  are the temporarily computing results before applying activation function  $f$ . In this study, a sigmoid function is selected as the activation

function. Therefore, the actual outputs  $y_j$  and  $z_k$  in hidden and output layers, respectively, can be also written as

$$\begin{aligned} y_j &= f(X_j) = \frac{1}{1 + e^{-X_j}}, \\ z_k &= f(Y_k) = \frac{1}{1 + e^{-Y_k}}. \end{aligned} \quad (2)$$

The activation function  $f$  introduces the nonlinear effect to the network and maps the result of computation to a domain  $(0, 1)$ . This sigmoid function is differentiable. The derivative of the sigmoid function in (2) can be easily derived as  $f' = f(1 - f)$ .

*Backpropagation Weight Training.* The error function is defined as

$$E = \frac{1}{2} \sum_{k=1}^K e_k^2 = \sum_{k=1}^K (t_k - z_k)^2, \quad (3)$$

where  $t_k$  is a predefined network output (or desired output or target value) and  $e_k$  is the error in each output node. The goal is to minimize  $E$  so that the weight in each link is accordingly adjusted and the final output can match the desired output. To get the weight adjustment, the gradient descent strategy is employed. In the link between hidden and output layers, computing the partial derivative of  $E$  with respect to the weight  $w_{jk}$  produces

$$\frac{\partial E}{\partial w_{jk}} = -e_k f'(Y_k) y_j = -\delta_k y_j \quad \text{where } \delta_k = (t_k - z_k) f'(Y_k). \quad (4)$$

The weight adjustment in the link between hidden and output layers is computed by  $\Delta w_{jk} = \alpha \times y_j \times \delta_k$ , where  $\alpha$  is the learning rate, a positive constant between 0 and 1. The new weight herein can be updated by the following  $w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n)$ , where  $n$  is the number of iterations. Similarly, the error gradient in links between input and hidden layers can be obtained by taking the partial derivative with respect to  $w_{ij}$  as

$$\frac{\partial E}{\partial w_{ij}} = -\Delta_j x_j = f'(X_j) \sum_{k=1}^K \delta_k w_{jk}. \quad (5)$$

The new weight in the hidden-input links can be now corrected as  $\Delta w_{ij} = \alpha \times x_i \times \Delta_j$  and  $w_{ij}(n+1) = w_{ij}(n) + \Delta_j$ . Training the BP-networks with many samples is sometimes a time-consuming task. The learning speed can be improved by introducing the momentum term  $\eta$ . Usually,  $\eta$  falls in the range  $(0, 1)$ . For the iteration  $n$ , the weight change  $\Delta w$  can be expressed. The backpropagation learning algorithm used in artificial neural networks is shown in many text books [3–6].

*1.2. Fuzzy Logic.* Fuzzy logics form heterogeneous family of formalisms capable of successful modelling of uncertain and vague information processing [7]. The usage of fuzzy logic for analysis and prediction of time series can be perceived

as a complement method to neural network based methods. The symbolic background of fuzzy logic brings an advantage of human readable symbolic representation of prediction interpretation. It does not necessarily mean that fuzzy logic based time series analysis is more accurate and more efficient but its power lies in transparent and interpretable results that it gives [8–11].

Time series analysis and prediction are an important task that can be used in many areas of practice. The task of getting the best prediction to given series may bring interesting engineering applications in wide number of areas like economics, geography, or industry. Solution to the problem of obtaining best results in prediction of time series can be based on well-known and simple methods like Winters or Linear method. In this paper, we use a method based on two methods originally developed by members of Institute for Research and Applications of Fuzzy Modeling, which is a part of University of Ostrava. The aim of the paper is not to present the details of the methods already published but to present a tool implementing them. The first method is based on the notion of F-transform (fuzzy transform) devised by the group of Professor Perfilieva et al. [12]. The second approach uses the linguistic rules utilizing fuzzy logic and deduction that is a well-known formalism with very good results in variety of practical applications like industrial ones.

The idea of the fuzzy transform is to transform a given function defined in one space into another, usually simpler space, and then to transform it back. The simpler space consists of a finite vector of numbers. The reverse transform then leads to a function, which approximates the original one. More details can be found in [12].

The fuzzy transform is defined with respect to a fuzzy partition, which consists of basic functions. Let  $c_1 < \dots < c_n$  be fixed nodes within  $[a, b]$  such that  $c_1 = a$ ,  $c_n = b$ , and  $n \geq 2$ . We say that fuzzy sets  $A_1, \dots, A_n \in F([a, b])$  are basic functions forming a fuzzy partition of  $[a, b]$  if they fulfill the following conditions for  $i = 1, \dots, n$ :

- (1)  $A_i(c_i) = 1$ ;
- (2)  $A_i(x) = 0$  for  $x \in (c_{i-1}, c_{i+1})$ , where for uniformity of notation we put  $c_0 = c_1 = a$  and  $c_{n+1} = c_n = b$ ;
- (3)  $A_i$  is continuous;
- (4)  $A_i$  strictly increases on  $[c_{i-1}, c_i]$  and strictly decreases on  $[c_i, c_{i+1}]$ ;
- (5) for all  $x \in [a, b]$ ,

$$\sum_{i=1}^n A_i(x) = 1. \tag{6}$$

Let a fuzzy partition of  $[a, b]$  be given by basic functions  $A_1, \dots, A_n$ ,  $n \geq 2$  and let  $f: [a, b] \rightarrow R$  be a function that is known on a set  $\{x_1, \dots, x_T\}$  of points.

The  $n$ -tuple of real numbers  $[F_1, \dots, F_n]$  given by

$$F_i = \frac{\sum_{t=1}^T f(x_t) A_i(x_t)}{\sum_{t=1}^T A_i(x_t)}, \quad i = 1, \dots, n, \tag{7}$$

is a fuzzy transform of  $f$  with respect to the given fuzzy partition.

The numbers  $F_1, \dots, F_n$  are called the components of the fuzzy transform of  $f$ .

Let  $F_n[f]$  be the fuzzy transform of  $f$  with respect to  $A_1, \dots, A_n \in F([a, b])$ .

Then the function  $f_{F,n}$  given on  $[a, b]$  by

$$f_{F,n}(x) = \sum_{i=1}^n F_i A_i(x) \tag{8}$$

is called the inverse fuzzy transform of  $f$ .

Fuzzy IF-THEN rules can be understood as a specific conditional sentence of natural language of the form IF  $X_1$  is  $A_1$  AND  $\dots$  AND  $X_n$  is  $A_n$  THEN  $Y$  is  $B$ , where  $A_1, \dots, A_n$  and  $B$  are evaluative expressions (very small, roughly big, etc.). An example fuzzy IF-THEN rule is as follows.

*IF the number of cars sold in the current year is more or less small and the half-year sales increment is medium, THEN the upcoming half-year increment will be medium.*

The part of the rule before THEN is called the antecedent and the part after it is consequent. Fuzzy IF-THEN rules are usually gathered in a linguistic description:

$$\begin{aligned} R_1 &:= \text{IF } X_1 \text{ is } A_{11} \text{ AND } \dots \text{ AND } X_n \\ &\text{ is } A_{1n} \text{ THEN } Y \text{ is } B_1, \\ &\vdots \\ R_m &:= \text{IF } X_1 \text{ is } A_{m1} \text{ AND } \dots \text{ AND } X_n \\ &\text{ is } A_{mn} \text{ THEN } Y \text{ is } B_m. \end{aligned} \tag{9}$$

Time series prediction based on these two main approaches works as follows. Let time series  $x_t, t = 1, \dots, T$  be viewed as a discrete function  $x$  on a time axis  $t$ . Then  $F_n[x] = [X_1, \dots, X_n]$  is the fuzzy transform of the function  $x$  with respect to a given fuzzy partition. The inverse fuzzy transform then serves us as a model of the trend-cycle of a given time series. By subtracting the trend-cycle (inverse fuzzy transform) values from the time series lags, we get pure seasonal components. This is how the fuzzy transform helps us to model and decompose a given time series.

Logical dependencies between components  $X_1, \dots, X_n$  of the fuzzy transform may be described by the fuzzy rules. These rules are generated automatically from the given data and are used for forecasting the next components. Fuzzy transform components as well as their first and second order differences are used as antecedent variables. For forecasting future fuzzy transform components based on the generated fuzzy rules, a special inference method—perception-based logical deduction is used. The seasonal components are forecasted autoregressively. Finally, both forecasted components, trend-cycle and seasonal, are composed together to obtain the forecast of time series lags. These methods are integrated into an implementation, PC application called *linguistic fuzzy logic forecaster (LFLF)*, which enables as to produce linguistic descriptions that describe properties of data treated like a time series.

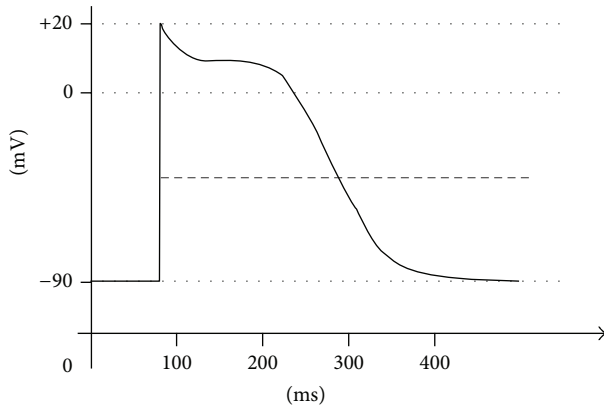


FIGURE 2: The cardiac action potentials.

### 2. Basic Principles of ECG Evaluation

ECG scanning has its own rules, which are in accordance with the laws of physics. The heart irritation spreads in all directions. In the case that the depolarisation spreads towards the electrode, which is placed on the body surface, a positive deflection is recorded on an ECG monitor. A negative deflection is recorded at the opposite end of the body. The ECG waveform is written with a chart speed of  $25 \text{ mm}\cdot\text{s}^{-1}$ . An algorithm describing the curve goes in the following steps. First, we evaluate the shape and rhythm of ventricular complexes or atrial, which can be either regular or irregular. Then we evaluate the frequency of ventricular complexes and atrial fibrillations. Contraction of each muscle of the human body (and thus the heart as well) is associated with electrical changes called depolarization, which can be detected by electrodes. The heart contains two basic types of cells: myocardial cells, which are responsible for generating the pressure necessary to pump blood throughout the body, and conduction cells, which are responsible for rapidly spreading electrical signals to the myocardial cells in order to coordinate pumping. A graph of an action potential of a muscle of cardiac cells is shown in Figure 2.

A normal electrocardiogram is illustrated in Figure 3. The figure also includes definitions for various segments and intervals in the ECG. The deflections in this signal are denoted in alphabetic order starting with the letter *P*, which represents atrial depolarization. The ventricular depolarization causes the *QRS* complex, and repolarization is responsible for the *T*-wave. Atrial repolarization occurs during the *QRS* complex and produces such a low signal amplitude that it cannot be seen apart from the normal ECG.

### 3. Signal Processing Using Neural Networks and Fuzzy Logic

In practice, a relatively reliable diagnostic program stored in ECG monitors has been used, which is a guideline for determining the final diagnosis of heart disorders. This program works according to the principle of IF-THEN rules. The values of the electrical signal are discretized and uploaded

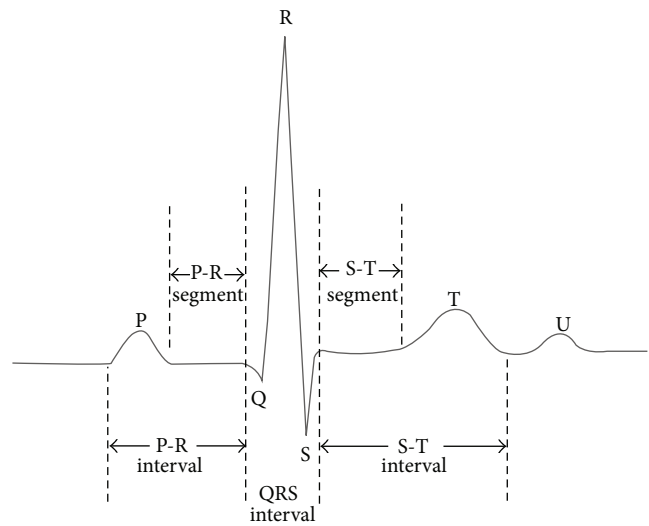


FIGURE 3: A typical one-cycle ECG tracing (adapted from [http://www.ni.com/white-paper/app/largeimage?lang=cs&imageurl=%2Fcms%2Fimages%2Fdevzone%2Ftut%2F2007-07-09\\_141618.jpg](http://www.ni.com/white-paper/app/largeimage?lang=cs&imageurl=%2Fcms%2Fimages%2Fdevzone%2Ftut%2F2007-07-09_141618.jpg)).

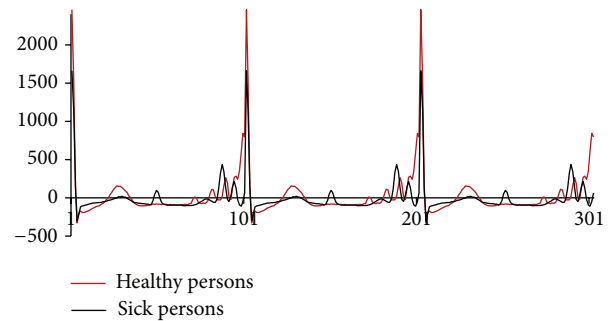


FIGURE 4: Comparison of mean values of ECG waveforms for healthy/sick persons.

into expert systems in the form of thousand rules. The aim of this paper is to use a different approach based on the principle of neural networks. The proposed methodology of solution could be summarized into the following steps:

- (1) a conversion of analog signal from the ECG monitor to a computer,
- (2) using multilayer networks that are fully connected,
- (3) obtaining ECG waveforms in collaboration with the University Hospital in Poruba, specifically at the Department Cardiac Surgery from sick patients and at the Department Traumatology from healthy patients (i.e., “healthy” with regard to heart diseases),
- (4) ECG waveforms built training/test sets,
- (5) neural network adaptation,
- (6) testing phases.

3.1. *Technical Equipment.* ECG measurements were performed using ADDA Junior with converter ADDA Junior,



FIGURE 5: Experimental results, test error for healthy/sick persons.

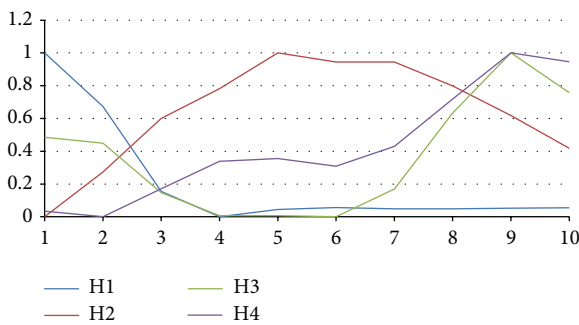


FIGURE 6: Patterns representing healthy persons.

which was connected to a computer via bidirectional parallel cable (CETRONICS). Technical parameters of the A/D converter (8-bit conversion) were the following:

- (i) 3 measuring ranges,
- (ii) measuring of a frequency of AC voltage at any channel,
- (iii) autoranging for measuring the frequency of 100 Hz, 1 kHz, and 10 kHz,
- (iv) input resistance of 300 kΩ,
- (v) measurement accuracy 1%.

Technical parameters of the D/A converter (a programmable voltage source ±10 V) were the following:

- (i) maximum current consumption of 15 mA (after optimizing 4A at the output),
- (ii) power of the converter ±15 V (stabilized).

### 4. Experimental Results

*4.1. Time Series Classification and Prediction via Neural Networks.* The training set consisted of modified ECG waveforms. We used a backpropagation neural network with topology 101-10-1. The output unit represents a diagnose 0/1, a healthy/sick person. A smaller number of inputs would

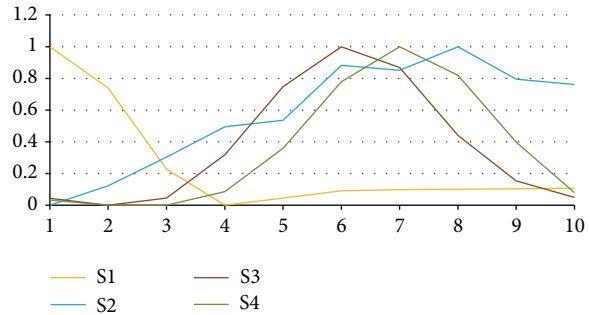


FIGURE 7: Patterns representing sick persons.

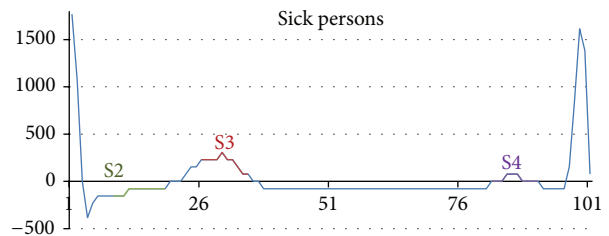


FIGURE 8: Some recognized patterns that occur in ECG time series.

not be appropriate due to the nature of the ECG waveform. We use 34 ECG time series associated with sick persons and 36 ECG time series associated with healthy persons. 25 time series of each group were used as a training set and the rest as a test set. Figure 4 shows a comparison of mean values of ECG waveforms for healthy/sick persons. We used the backpropagation method [5, 6] for the adaptation with the following parameters: the learning rate value is 0.1 and momentum is 0. The conducted experimental studies also showed that training patterns are mixed randomly in each cycle of adaptation. This ensures their greater diversity which acts as a measure of system stability. Uniform system in a crisis usually collapses entirely, while system with such diversity of trained patterns remains functional despite of crisis of its individual parts. The condition of end of the adaptation algorithm specified the limit value of the overall network error,  $E < 0.1$ .

The test set consisted of 20 samples (11 health and 9 sick persons) that were not included in the training set. The summary results for this type of experiment are shown in a graph in Figure 5. For clarity, the results of testing are given in percentage. The average test error was 0.194. A healthy population was detected with an average error of 0.263 and sick population with an average error of 0.109.

*4.1.1. Pattern Recognition Classifier Leading to Prediction.* For the purpose of adaptation of the pattern recognition classifier, it is necessary to remark that determination of training patterns is one of the key tasks. Improperly chosen patterns can lead to confusion of neural networks. During our experimental work, we made some study which included ECG pattern recognition. When creating appropriate patterns of the training set, we used characteristic curves shown as mean



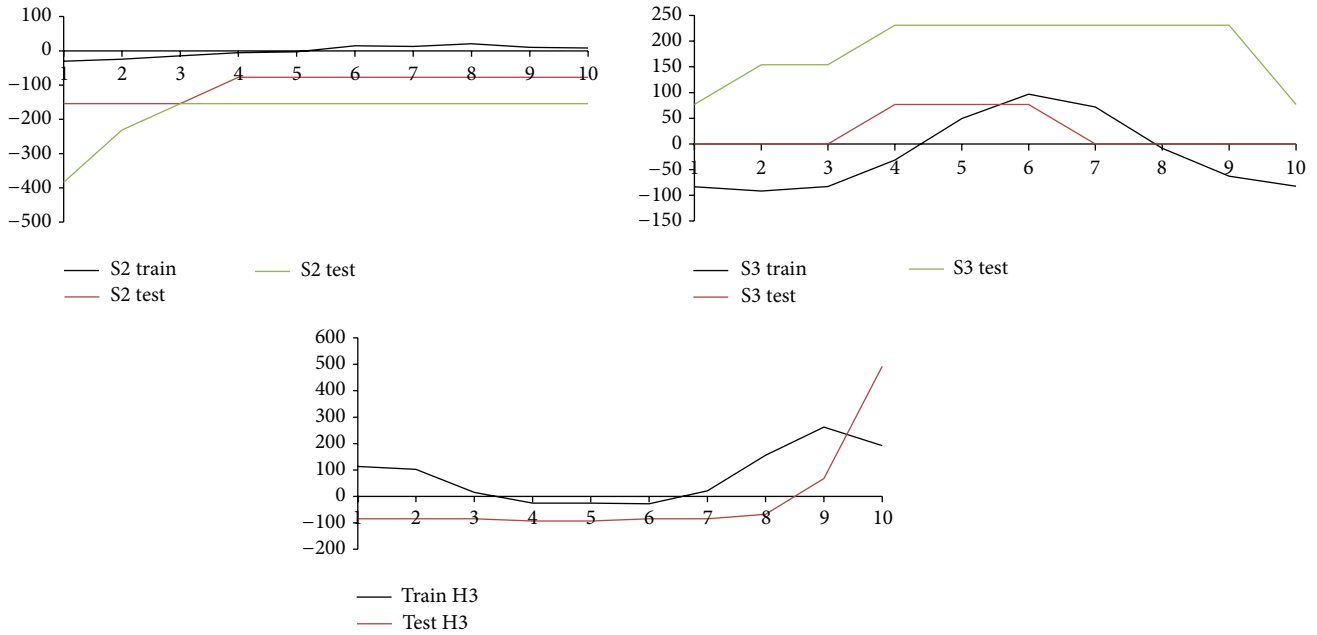


FIGURE 9: Training patterns, their representation in used test sets.

values from ECG waveforms for healthy and sick persons (Figure 4). We use two different groups of patterns. Patterns H1–H4 (Figure 6) represent healthy persons and patterns S1–S4 (Figure 7) represent sick persons. The whole training set is shown in Table 1.

Pattern recognition classifier is based on backpropagation neural network and is able to recognise wave structures in given time series [13, 14]. Artificial neural networks need training sets for their adaptation. In our experimental work, the training set consisted of 8 patterns representing the basic structure of the various waves in ECG graphs; see Figures 6 and 7. Input data is sequences always including  $n$  consecutive numbers, which are transformed into interval  $\langle 0, 1 \rangle$  by formula (10). Samples are adjusted for the needs of backpropagation networks with sigmoid activation function in this way [5, 6]:

$$x'_j = \frac{x_j - \min(x_i, \dots, x_{i+n-1})}{\max(x_i, \dots, x_{i+n-1}) - \min(x_i, \dots, x_{i+n-1})}, \quad (10)$$

$$(j = i, \dots, i + n - 1),$$

where  $x'_j$  is normalized output value of the  $j$ th neuron ( $j = i, \dots, i + n - 1$ ) and  $(x_i, \dots, x_{i+n-1})$  are  $n - 1$  consecutive output values that specify sequences (patterns) from the training set (e.g., training pairs of input and corresponding output vectors). Input vector contains 10 components. Output vector has got 8 components and each output unit represents one of 8 different types of ECG wave samples. A neural network architecture is 10-10-8 (e.g., 10 units in the input layer, 10 units in the hidden layer, and 8 units in the output layer). The net is fully connected. Adaptation of the neural network starts with randomly generated weight values.

We used the backpropagation method for the adaptation with the following parameters: the learning rate value is 0.1 and momentum is 0. We have utilized our experience from earlier times; that is, training patterns were mixed randomly in each cycle of adaptation. The condition of end of the adaptation algorithm specified the limit value of the overall network error,  $E < 0.1$ .

In order to test the efficiency of the method, we applied the same set of data that we used in the previous experimental part. Outputs from the classifier produce sets of values that are assigned to each recognized training pattern in the given test time series. It is important to appreciate what can be considered as an effective criterion related to consensus of similarity. The proposed threshold resulting from our experimental study was determined at least  $p = 70\%$ . Figure 9 shows a comparison of patterns, how were learned (S2, S3, H3 train) and how were recognized in test time series (S2, S3, H3 test). The neural network is able to discover some connections, which are almost imperceptible. Illustration of some recognized patterns that occur in ECG time series is shown in Figure 8. Outputs from the classifier carry a predictive character. The neural network determines if the time series belongs to a healthy or sick person on the basis of the recognised ECG patterns which appear in the time series history.

The methodology of testing is shown in Figure 10. This means that if the test pattern S1, S2, S3, or S4 appeared in ECG waveform with probability  $p_S \geq p$  ( $p = 70\%$ ), thus it was predicted to be “a sick person.” Then we work only with the remaining time series. If the test pattern H1, H2, H3, or H4 appeared in ECG waveform with probability  $p_H \geq p$  ( $p = 70\%$ ), thus it was predicted to be “a healthy person.” In all other cases, the ECG time series was unspecified. We examined a total of 20 data sets. Each of them

TABLE 1: The training set.

Patterns	Inputs										Outputs							
H1	1.000	0.672	0.155	0.000	0.045	0.057	0.049	0.049	0.053	0.055	1	0	0	0	0	0	0	0
H2	0.000	0.273	0.600	0.782	1.000	0.945	0.945	0.799	0.618	0.418	0	1	0	0	0	0	0	0
H3	0.485	0.449	0.147	0.007	0.007	0.000	0.169	0.632	1.000	0.757	0	0	1	0	0	0	0	0
H4	0.035	0.000	0.170	0.338	0.356	0.309	0.430	0.719	1.000	0.946	0	0	0	1	0	0	0	0
S1	1.000	0.740	0.228	0.000	0.045	0.091	0.098	0.101	0.104	0.107	0	0	0	0	1	0	0	0
S2	0.000	0.123	0.304	0.495	0.536	0.883	0.851	1.000	0.796	0.761	0	0	0	0	0	1	0	0
S3	0.044	0.000	0.045	0.319	0.748	1.000	0.868	0.440	0.154	0.050	0	0	0	0	0	0	1	0
S4	0.033	0.000	0.000	0.085	0.360	0.779	1.000	0.820	0.399	0.079	0	0	0	0	0	0	0	1

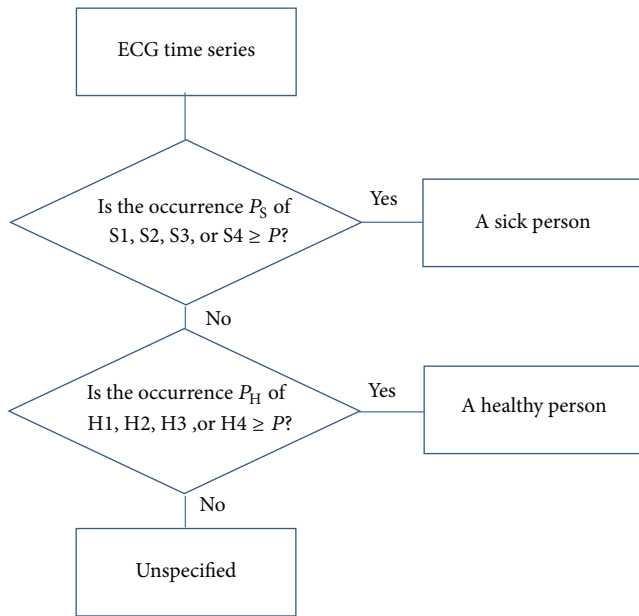


FIGURE 10: The methodology of testing.

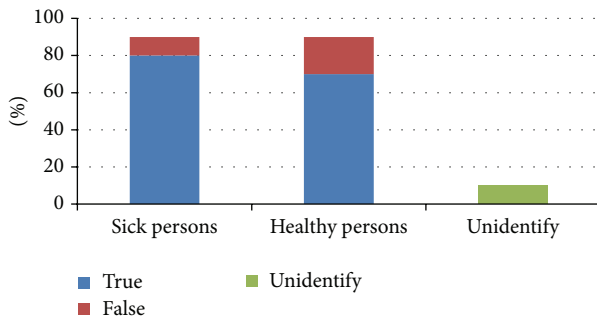


FIGURE 11: Experimental results, test error.

contains 101 values that assign 92 possible patterns. The whole number of examined patterns is 1840. The graph in Figure 11 demonstrates a summary of results, where “sick persons” represent patterns S1–S4 and “healthy persons” represent patterns H1–H4. The resulting prediction is based on the methodology; see Figure 10.

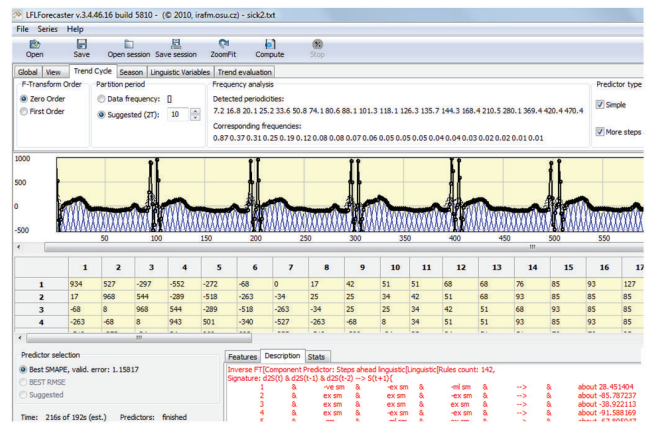


FIGURE 12: LFLF application.

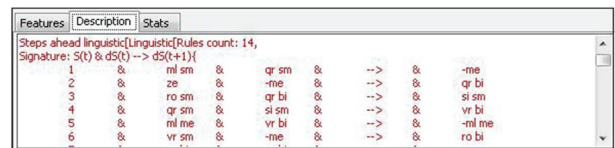


FIGURE 13: Winning predictor linguistic description (trend-cycle model).

#### 4.2. Time Series Classification and Prediction via Linguistic Fuzzy Logic Forecaster.

We tried also to utilize above presented method of time series analysis through linguistic fuzzy logic forecaster (LFLF) [15]; see Figure 12.

Basic usage of the application is to analyse given time series and find best predictor with respect to validation part of time series given. We evaluate efficiency of predictors by SMAPE (symmetric mean absolute percentage error). It enables us to make analysis of trend-cycle of a time series and also seasonal part. The main advantage lies in prediction based on transparent linguistic descriptions that provide the model of a time series behaviour. Linguistic variables are of the following types:

- (i) value: we directly mean the components of the fuzzy transform,
- (ii) difference, first order differences of fuzzy transform components that are given as follows: differences between components  $\Delta X_i = X_i - X_{i-1}$ ,

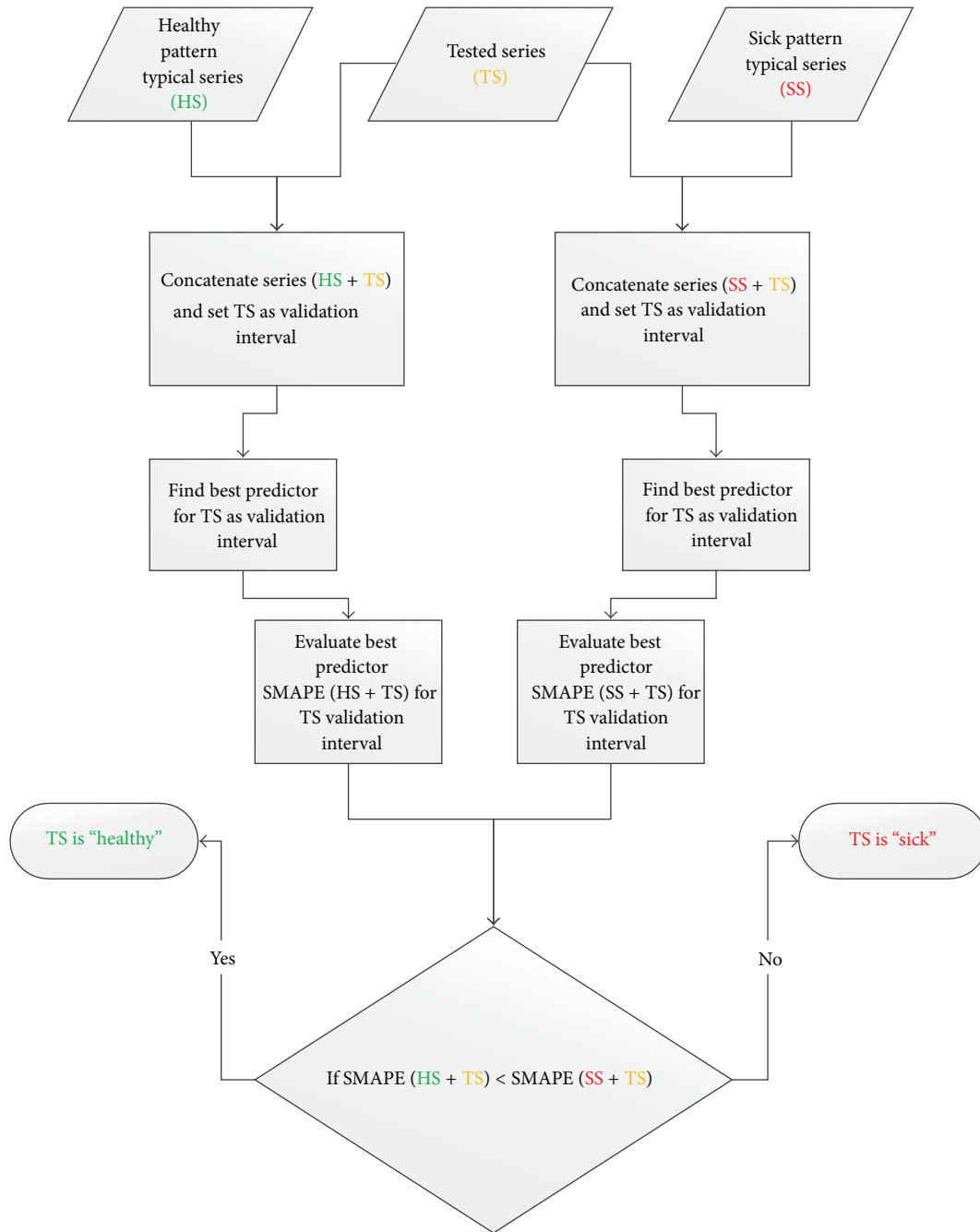


FIGURE 14: Recognition by linguistic fuzzy logic predictors for typical learning series.

(iii) second difference: these are values of second order differences of components of the fuzzy transform as follows:  $\Delta^2 X_i = \Delta X_i - \Delta X_{i-1}$ .

LFLF application enables us to define minimal and maximal number of these particular variables in a rule of linguistic description as well as the total number of antecedent variables.

A rule consisting of these variables has the following structure and can be described as a signature (fuzzy rules describing the trend-cycle model). Particularly,  $S$  denotes the

trend-cycle components,  $dS$  their differences, and  $d^2S$  their second order differences. The argument  $(t)$ ,  $(t - 1)$ , and so forth, denotes the time lag of the component.

For example, taking signature  $S(t) \& dS(t) \rightarrow d^2S(t + 1)$  denotes the fact that  $X_i$  and  $\Delta X_i$  are the antecedent variables and  $\Delta X_{i+1}$  is the consequent variable of the winning model and hence, we deal with rules of the form

$$\text{IF } X_i \text{ is } A_i \text{ AND } \Delta X_i \text{ is } A_{\Delta i} \text{ THEN } \Delta X_{i+1} \text{ is } A_{\text{Delta}i}. \tag{11}$$



TABLE 2: Example of algorithm evaluation on 10 “healthy” and 10 “sick” patients.

Patient	SMAPE (HS + TS)	SMAPE (SS + TS)	Result	Actual	Match
H11	3.38232	3.30989	Sick	Healthy	NO
H12	1.33555	3.40168	Healthy	Healthy	YES
H13	2.73377	4.58243	Healthy	Healthy	YES
H14	3.44581	2.37995	Sick	Healthy	NO
H15	2.04677	2.30998	Healthy	Healthy	YES
H16	3.73377	3.98572	Healthy	Healthy	YES
H17	1.03658	2.20151	Healthy	Healthy	YES
H18	3.22689	2.38111	Sick	Healthy	NO
H19	2.43544	3.42159	Healthy	Healthy	YES
H20	2.63355	3.79940	Healthy	Healthy	YES
S11	1.72265	1.31922	Sick	Sick	YES
S12	2.04804	0.38562	Sick	Sick	YES
S13	2.99464	0.51305	Sick	Sick	YES
S14	2.52248	0.67941	Sick	Sick	YES
S15	2.65972	1.75233	Sick	Sick	YES
S16	2.63674	2.39694	Sick	Sick	YES
S17	2.57638	1.65941	Sick	Sick	YES
S18	4.24496	2.85006	Sick	Sick	YES
S19	2.96533	1.01009	Sick	Sick	YES
S20	3.23630	1.10960	Sick	Sick	YES

Every single fuzzy rule can be taken as a sentence of natural language, for example, first rule from Figure 13.

IF  $X_i$  is ml sm AND  $\Delta X_i$  is qr sm, THEN  $\Delta X_{i+1}$  is -me may be read as follows.

If the number of cars sold in the current year is more or less small and the half-year sales increment is quite roughly small then the upcoming half-year increment will be negative medium.

4.2.1. Recognition of “Healthy” and “Sick” Patterns by LFLF. Our method to use linguistic fuzzy logic forecasting is based on simple idea that best predictor learning from both “healthy” and “sick” pattern samples, respectively, can be used for validation with tested pattern taken as validation part of the series. Then we can evaluate SMAPE for both these cases: compound series SMAPE (“healthy” + tested) and SMAPE (“sick” + tested).

If  $SMAPE (“healthy” + tested) < SMAPE (“sick” + tested)$  then the tested pattern is supposed to be “healthy”; otherwise, the tested pattern is supposed to be “sick.”

The idea is schematically shown in Figure 14.

For testing purposes, we created two necessary typical learning time series: “healthy” (HS) and “sick” (SS) according to the algorithm above. They both consist of 1010 samples made from 10 typical series of “healthy” and “sick” patients with 101 measured ECG values. Then we have created 10 concatenated series according to the scheme in Figure 14 with 10 randomly selected patients with “healthy” ECG measurement; that is, 20 files were produced (10x HS + TS and 10x SS + TS). The same concatenated series were also made from 10 “sick” patients measurements. This made us additional 20 files with concatenated series (10x HS + TS and

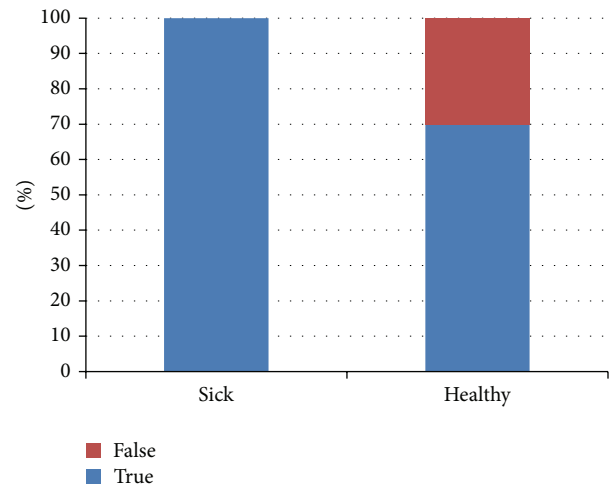


FIGURE 15: Experimental results, LFLF.

10x SS + TS). For 20 patients (Table 2) tested ECG we have 2 concatenated series giving SMAPE (HS + TS) and SMAPE (SS + TS).

Our method based on LFLF proved very good results for right identification of sick patient records. Nevertheless, it produces large amount of false positive identification of sick pattern for healthy patients (Figure 15). This result is consistent with our approach using neural networks. Of course, our preliminary research has a limited extent and should be perceived only as narrative result, which shows interesting properties especially in complementation of neural network results.

## 5. Conclusion

In this paper, a short introduction into the field of ECG waves recognition using backpropagation neural network has been given. Main objective was to recognise the normal cycles and arrhythmias and perform further diagnosis. We proposed two detection systems that have been created with usage of neural networks. One of them is adapted according to the training set. Here, each pattern represents the whole one ECG cycle. Then, an output unit represents a diagnose 0/1, a healthy/sick person. The second one approach uses neural network, in which training set contains two different groups of patterns for healthy/sick persons. According to the results of experimental studies, it can be stated that ECG waves patterns were successfully extracted in given time series and recognised using suggested method, as can be seen from figures in Experimental Result section. It might result in better mapping of the time series behaviour for better prediction.

Both approaches were able to predict with high probability if the ECG time series represents sick or healthy persons. It is interesting that a sick diagnose was recognised with higher accuracy in both experimental works.

The third approach based on LFLF is currently only in the stage of preliminary experiments, but it conforms to the former results based on neural networks. This approach is novel and could be good supplement to other soft-computing methods for this task.

## Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

The research described here has been financially supported by University of Ostrava Grants SGS16/PřF/2014 and SGS14/PřF/2014.

## References

- [1] P. Tangkraingki, C. Lursinsap, S. Sanguansintukul, and T. Desudchit, "Selecting relevant EEG signal locations for personal identification problem using ICA and neural network," in *Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*, pp. 616–621, June 2009.
- [2] C. D. Nugent, J. A. C. Webb, N. D. Black, and G. T. H. Wright, "Electrocardiogram 2: classification," *Automedica*, vol. 17, pp. 281–306, 1999.
- [3] S. Russell and P. Norvig, *Artificial Intelligence—A Modern Approach*, Prentice Hall, 2nd edition, 2003.
- [4] C. D. Nugent, J. A. Lopez, A. E. Smith, and N. D. Black, "Reverse engineering of neural network classifiers in medical applications," in *Proceedings of the 7th EFOMP Congress*, vol. 17, p. 184, Physica Medica, 2001.
- [5] L. Fausett, *Fundamentals of Neural Network*, Prentice Hall, 1994.
- [6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [7] V. Novák, *Fuzzy Sets and Their Applications*, Adam Hilger, Bristol, UK, 1989.
- [8] S. Aouaouda, M. Chadli, P. Shi, and H. R. Karimi, "Discrete-time  $H_-/H_\infty$  sensor fault detection observer design for nonlinear systems with parameter uncertainty," *International Journal of Robust and Nonlinear Control*, 2013.
- [9] M. Chadli, A. Abdo, and S. X. Ding, " $H_-/H_\infty$  fault detection filter design for discrete-time Takagi-Sugeno fuzzy system," *Automatica*, vol. 49, no. 7, pp. 1996–2005, 2013.
- [10] M. Chadli and H. R. Karimi, "Robust observer design for unknown inputs takagi-sugeno models," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 158–164, 2013.
- [11] S. Aouaouda, M. Chadli, and H. Karimi, "Robust static output-feedback controller design against sensor failure for vehicle dynamics," *IET Control Theory & Applications*, vol. 8, no. 9, pp. 728–737, 2014.
- [12] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, "Analysis and prediction of time series using fuzzy transform," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 3876–3880, Hong Kong, June 2008.
- [13] E. Volna, M. Kotyrba, and R. Jarusek, "Multi-classifier based on Elliott wave's recognition," *Computers & Mathematics with Applications*, vol. 66, no. 2, pp. 213–225, 2013.
- [14] E. Volná, M. Kotyrba, and R. Jarušek, "Prediction by means of Elliott waves recognition," in *Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems*, vol. 192 of *Advances in Intelligent Systems and Computing*, pp. 241–250, Springer, Berlin, Germany, 2013.
- [15] H. Habiballa, V. Pavliska, and A. Dvořák, "Software system for time series prediction based on F-transform and linguistic rules," in *Proceedings of the 8th International Conference on Applied Mathematics Aplimat*, pp. 381–386, Bratislava, Slovakia, 2009.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

