amcs

APPROXIMATION OF PHENOL CONCENTRATION USING NOVEL HYBRID COMPUTATIONAL INTELLIGENCE METHODS

Paweł Pławiak [*,**], Ryszard Tadeusiewicz [**]

[*] Institute of Telecomputing
Cracow University of Technology, ul. Warszawska 24, F-5, 31-155 Cracow, Poland
e-mail: `plawiak@pk.edu.pl`

[**]Department of Automatics and Biomedical Engineering
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Cracow, Poland
e-mail: `plawiak@agh.edu.pl`

This paper presents two innovative evolutionary-neural systems based on feed-forward and recurrent neural networks used for quantitative analysis. These systems have been applied for approximation of phenol concentration. Their performance was compared against the conventional methods of artificial intelligence (artificial neural networks, fuzzy logic and genetic algorithms). The proposed systems are a combination of data preprocessing methods, genetic algorithms and the Levenberg–Marquardt (LM) algorithm used for learning feed forward and recurrent neural networks. The initial weights and biases of neural networks chosen by the use of a genetic algorithm are then tuned with an LM algorithm. The evaluation is made on the basis of accuracy and complexity criteria. The main advantage of proposed systems is the elimination of random selection of the network weights and biases, resulting in increased efficiency of the systems.

Keywords: soft computing, neural networks, genetic algorithms, fuzzy systems, evolutionary-neural systems, pattern recognition, chemometrics.

1. Introduction

The use of hybrid systems can eliminate weak points of single methods of artificial intelligence. The classical learning algorithm of neural networks can sometimes fall into local minima and therefore the problem is not solved properly. With the use of hybrid methods this could be minimized by global search and relatively easier finding of a lower error value. Also, in problems where it is difficult to obtain information about the surface error gradient, the global search space of weights and biases can be searched. Evolutionary methods are used to teach a neural network (determining the values of network weights and biases), searching for the optimal network structure (number of layers and neurons) and even for concurrent determination of the network structure and its learning (Rutkowski, 2008; Patan and Patan, 2011). Another advantage of hybrid systems is to use a genetic algorithm to optimize the input signal in order to minimize its volume while preserving the necessary amount of transmitted information to the proper operation of the system.

In many cases one can achieve much higher performance while the system complexity is only slightly increased (Tallon-Ballesteros and Hervas-Martinez, 2011). The most popular hybrid systems include evolutionary-neural (Font *et al.*, 2010; Chandra *et al.*, 2011; Su *et al.*, 2011; Tong and Schierz, 2011; Yang and Chen, 2012; Zhang *et al.*, 2011), evolutionary-fuzzy (Cheng *et al.*, 2010; Lin and Chen, 2011; Antonelli *et al.*, 2009; Cheshmehgaz *et al.*, 2012; Aydogan *et al.*, 2012) and neuro-fuzzy systems (Shahlaei *et al.*, 2012; Czogała and Łęski, 2000; Tadeusiewicz, 2010b; Tadeusiewicz and Morajda, 2012).

Although these systems are well known and have been used for many years, the subject is still an interesting topic and is in particular used for the analysis of non-linear multidimensional data, i.e., gas sensor signals from electronic noses and in image analysis (Tadeusiewicz 2011a; Tabor, 2010; 2009). Hybrid systems are used to identify gas mixtures using sensor arrays (Benrekia *et al.*, 2009; Snopok and Kruglenko, 2002), eliminate

the influence of humidity (Maziarz *et al.*, 2003), lower the sensor baseline drift, improve its long term-stability (Nomura *et al.*, 1998), enhance selectivity and sensitivity (Ihokura and Watson, 1994; Romain *et al.*, 2000), identify mixtures of gases with a single sensor with modulated temperature (Nakata *et al.*, 2001; Maziarz and Pisarkiewicz, 2008), and solve various classification problems in fragrance recognition systems (Yu *et al.*, 2009; Cevoli *et al.*, 2011; Ghasemi-Varnamkhasti *et al.*, 2011).

The authors previously analysed various systems based on artificial neural networks (Tadeusiewicz, 2010a), fuzzy logic and genetic algorithms for approximation of phenol concentration. The results indicated that some methods are inappropriate to solve a defined problem, i.e., the training of the systems with the classic LM algorithm for Feed Forward (FF) and Recurrent Neural Networks (RNNs) as well as neural networks trained by genetic algorithms exhibited unacceptable effectivity—their approximation of phenol concentration with the average error was at the level of 8.34%–9.15%.

In the case of tuning the parameters of neural networks only with an LM algorithm, the authors sometimes observed a local minimum or maximum of the function, not a global one construed as a lower error value. This usually results in increased errors, lower reliability and lower effectiveness of the system. In order to overcome these drawbacks, a modified algorithm was applied and the Genetic Algorithm (GA) was used for the preselection of the initial values of the network weights and biases used in LM calculations. Another issue is system complexity growing with the size of the training data set. For the training set containing a few tens of elements, radial neural networks exhibited the best scores in the performed analyses, but for larger input data sets they will obtain worse results, due to their complexity. The presented systems exhibit much lower complexity than those based on the RBF, which have achieved the highest score.

The use of novel hybrid systems proposed by the authors greatly increased the effectiveness of the systems based on RNNs. Such a solution improves the pattern recognition process and approximation of phenol concentration. RNNs are applied despite the absence of their use in similar research problems. The authors wanted to test their effectiveness. Results of some analysis (Table 6) showed that the error of phenol concentration for an RNN system was lower than for a system with a FF neural network (FF+GA+LM). The RNN system showed greater stability training, resulting in greater efficiency of finding a lower error value.

2. Measurements and data manipulation

The process of approximation of phenol concentrations consists of three stages: measurements, data manipulation, and data analysis. The measurements were performed with a commercially available e-nose from the Alpha MOS company, model FOX 4000 (M.O.S., 2002). It consists of an array of 18 semiconductor gas sensors.

A phenol specimen means the vial with appropriate solution of phenol. All the phenol specimens were heated before measurement for 5 min at 60°C. The volume of the gas mixture injected into the measuring chamber of the gas mixture was 0.5 ml. The measurement of each specimen of phenol lasted 121 seconds. Sampling was performed with a step of 1 second, and a single sample contains information about the resistance value of 18 sensors. A set of 121 samples was analysed. In effect, for a single specimen of phenol we obtain a matrix consisting of 121 rows corresponding to subsequent samples, and 18 columns corresponding to the responses of the sensors.

The object of this study was 47 specimens of 8 levels of molar concentration of phenol[1] before selection (0.00005M, 0.0001M, 0.0005M, 0.001M, 0.005M, 0.01M, 0.05M, 0.1M) and 24 specimens of 5 levels of molar concentration of phenol after the final selection (0.00005M, 0.0001M, 0.0005M, 0.01M, 0.1M). As a result of the selection, some of the specimens were rejected and the number of concentration levels was reduced. Due to the small number of specimens, inaccuracies and errors caused by, e.g., improper dosing of the gas mixture (some concentration levels of phenol are overlapping or are weakly separable), the incorrect measurements were rejected and eventually only five concentration levels were taken into consideration.

All of the preprocessing stages performed on a single specimen of phenol are described in Table 1 and shown in Fig. 2. As a result, 47 vectors built of 36 elements were obtained. Some part of them, after data selection, were used as input data for all the developed and evaluated AI systems.

Table 2 shows the matrices in which the data were collected. For the reference matrices, the letter m should be understood as the correct level of normalized concentrations of phenol[2].

---

[1]Molar concentration is the number of moles of phenol contained in 1 dm^3 of the measured gas mixture.

[2]Normalized concentration is the molar concentration converted into bales, $M = \log_{10}\left(M_r/10^{-5}\right)$, where M is the matrix of normalized values of phenol concentration, M_r is the matrix with real (molar) phenol concentration values of phenol, 10^{-5} is the reference value, equal to one-fifth of the smallest value of the expected response.
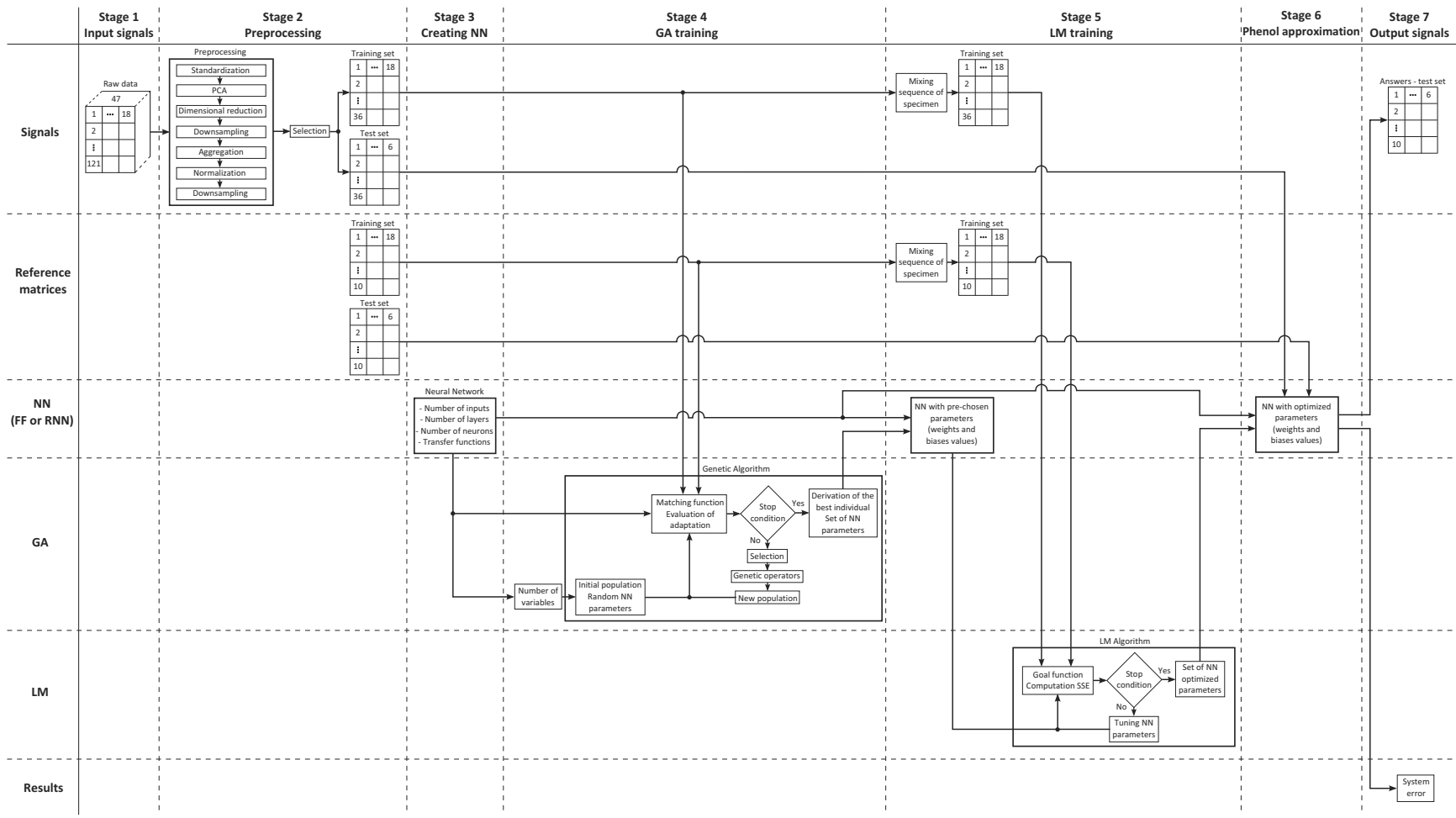
Fig. 1. Scheme of novel data analysis systems.

3. Data analysis systems

The goal of the analysis was approximation of the the correct level of phenol concentration. The authors previously tested different systems and algorithms and used them for quantitative analysis of phenol concentration. The data obtained from the e-nose were preprocessed, and then the assignment of the input data to appropriate classes was performed. The tools used for the approximation were different data analysis systems based on selected methods of artificial intelligence. According to the assumed criteria, it was observed that some complex systems based on fuzzy logic and radial neural networks are better in terms of accuracy while others need fewer parameters, but the phenol concentration was approximated with some errors.

The authors propose two novel data analysis systems that benefit from a combination of single methods of artificial intelligence such as Artificial Neural Networks (ANNs) and GAs (Tadeusiewicz, 2011b).

The systems belong to a group of evolutionary-neural systems in which the weights and biases of an ANN are determined as a result of training by a GA. Among different developed and tested systems, the best scores in the evaluation obtained the systems using the following RNNs and FF neural networks:

- feed-forward neural networks trained by the genetic algorithm and optimized by the Levenberg–Marquardt algorithm (FF+GA+LM),
- recurrent neural networks trained by the genetic algorithm and optimized by the Levenberg–Marquardt algorithm (RNN+GA+LM).

To simplify the naming and uniquely recognize both systems, in our further discussion we will use shorter names: FF+GA+LM and RNN+GA+LM. Both the systems are described in the following sections. A scheme of novel hybrid systems is presented in Fig. 1.

3.1. FF+GA+LM system. The first developed system, which obtained the fourth score in the evaluation (in Table 6), is FF+GA+LM. Its core is a feed-forward neural network, trained by a genetic algorithm and optimized by a Levenberg–Marquardt algorithm. Its structure is presented in Fig. 3. The network contains 4 layers. The preprocessing signal is fed to 36 inputs ($x_1, \ldots, x_{36}$) corresponding to 36 samples included in one specimen of phenol. The input layer consists of 36 neurons transmitting the signals (not trained). The first hidden layer consists of 10 neurons, each with 36 weights and 1 bias, with a log-sigmoid transfer function. The second hidden layer consists of 5 neurons, each with 10 weights and 1 bias, also with a log-sigmoid transfer function. The output layer consists of 1 neuron, with 5 weights and 1 bias, with a linear transfer function. The output signal

contains information from one network output (y_1). The output y_1 corresponds to an approximate value of phenol concentration.

Evolutionary algorithms mimic nature in the process of natural selection and evolution. They are used for random searches, global optimization and learning. Genetic algorithms are usually included in evolutionary algorithms (Rutkowski, 2008), and in our case they were used for random selection of weights and biases and for the training of neural networks. The course of a genetic algorithm for successive generations is shown in Fig. 4 for the FF+GA+LM system.

The example of matching the FF+GA+LM system's response to the model response obtained for a test set for normalized values is shown in Fig. 5 and for actual values in Fig. 6. The slope of the fitting line and the shift of the response points are related to the accuracy of the system.

The most important parameters of the FF+GA+LM system are summarized in Table 3. As we can see, the developed system approximates phenol concentration with the average error of 6.67% for normalized values and 13.00% for real values. They have also low computational complexity and consist of a relatively low and acceptable number of parameters. The neural network parameters were calculated in a training process consisting of 30 generations of the genetic algorithm. The best individual characterized by the lowest error consisted of a set of parameters. It was used for primary configuration of weights and biases of the networks. Such parameters were then optimized by the LM algorithm.

3.2. RNN+GA+LM system. The second developed system, which obtained the third score in the evaluation (in Table 6) is RNN+GA+LM. Its core is a recurrent neural network. It was also trained with a GA and optimized by an LM algorithm. Its structure is presented in Fig. 7. The network contains 4 layers. The preprocessing signal is fed to 36 inputs ($x_1, \ldots, x_{36}$) corresponding to 36 samples included in one specimen of phenol. The input layer consists of 36 neurons transmitting the signals (not trained). The first hidden layer with feedback consists of 10 neurons, each with 46 weights and 1 bias, with log-sigmoid transfer function. The second hidden layer, also with feedback, consists of 5 neurons, each with 15 weights and 1 bias, also with a log-sigmoid transfer function. The output layer consists of 1 neuron, with 5 weights and 1 bias, with a linear transfer function. The output signal contains information from 1 network output (y_1). The output y_1 corresponds to an approximate value of phenol concentration.

The course of a genetic algorithm for successive generations is shown in Fig. 8 for the RNN+GA+LM system. An example of matching the RNN+GA+LM system's response to the model response obtained for a test set for normalized values is shown in Fig. 9 and for

Table 1. Successive preprocessing stages of measured data, for a single specimen of phenol. PC is the principal component.

Stage	Name	Description	Number of dimensions	Range	Number of data	Figure
I	Raw data	121 samples 18 sensors responses	18	$[\,4.6{\cdot}10^3\,,\ 3.1{\cdot}10^5\,]$	$121\cdot18$ $=2178$	**2A**
II	Standardization	standard deviation $=1$ average value $=0$	18	$[\,\text{-}4.4\,,\ 3.2\,]$	$121\cdot18$ $=2178$	**2B**
III	Principal Component Analysis	maximizing the variance of the first coordinates	18	$[\,\text{-}10.4\,,\ 3.0\,]$	$121\cdot18$ $=2178$	**2C**
IV	Reduction	dimensional reduction leaving first 3 PC	3	$[\,\text{-}10.4\,,\ 3.0\,]$	$121\cdot3$ $=363$	**2D**
V	Reduction	downsampling: every six sample left	3	$[\,\text{-}10.4\,,\ 3.0\,]$	$22\cdot3$ $=66$	**2E**
VI	Aggregation	aggregation of first 3 PC	1	$[\,\text{-}10.4\,,\ 3.0\,]$	$66\cdot1$ $=66$	**2F**
VII	Normalization	to interval: $[\,\text{-}1\,,\ 1\,]$	1	$[\,\text{-}1\,,\ 1\,]$	$66\cdot1$ $=66$	**2G**
VIII	Reduction	downsampling: one of two samples left	1	$[\,\text{-}1\,,\ 1\,]$	$36\cdot1$ $=36$	**2H**
IX	Preprocessed data	all 47 specimens of phenol preprocessed	1	$[\,\text{-}1\,,\ 1\,]$	$36\cdot1$ $=36$	**2I**

Table 2. Data matrices.

Raw data

	Sensor 1	Sensor 2	Sensor 3	...	Sensor 18
Sample 1	$r_{1\,1}$	$r_{1\,2}$	$r_{1\,3}$	...	$r_{1\,18}$
Sample 2	$r_{2\,1}$	$r_{2\,2}$	$r_{2\,3}$	...	$r_{2\,18}$
Sample 3	$r_{3\,1}$	$r_{3\,2}$	$r_{3\,3}$	...	$r_{3\,18}$
⋮	⋮	⋮	⋮	⋱	⋮
Sample 121	$r_{121\,1}$	$r_{121\,2}$	$r_{121\,3}$	...	$r_{121\,18}$

$r \in [5.0\cdot10^3\,,\ 3.2\cdot10^5]$
47 matrices, each matrix corresponds to one specimen of phenol

Preprocessed data

	Specimen 1	Specimen 2	Specimen 3	...	Specimen 18/6
Sample 1	$p_{1\,1}$	$p_{1\,2}$	$p_{1\,3}$	...	$p_{1\,18/6}$
Sample 2	$p_{2\,1}$	$p_{2\,2}$	$p_{2\,3}$	...	$p_{2\,18/6}$
Sample 3	$p_{3\,1}$	$p_{3\,2}$	$p_{3\,3}$	...	$p_{3\,18/6}$
⋮	⋮	⋮	⋮	⋱	⋮
Sample 36	$p_{36\,1}$	$p_{36\,2}$	$p_{36\,3}$	...	$p_{36\,18/6}$

$p \in [-1\,,\ 1]$
18 columns for training set, 6 columns for test set
each column of the matrix corresponds to one specimen of phenol

Reference matrix

	Specimen 1	Specimen 2	Specimen 3	...	Specimen 18/6
Concentration of phenol	$m_{1\,1}$	$m_{1\,2}$	$m_{1\,3}$	...	$m_{1\,18/6}$

$m \in \{0.699\,,\ 1\,,\ 1.699\,,\ 3\,,\ 4\}$
Appropriate floating-point number indicating the correct level of phenol concentration
18 columns for training set, 6 columns for test set

real values in Fig. 10. The slope of the fitting line and the shift of the response points are related to the accuracy of the system.

The most important parameters of the RNN+GA+LM system are summarized in Table 4. As we can see, the developed system approximates the phenol concentration with the average error for normalized values of 5.88% and for real values of 11.26%. They also have low computational complexity and consist of a relatively low and acceptable number of parameters. The neural network parameters were calculated in a training process consisting of 30 generations of the
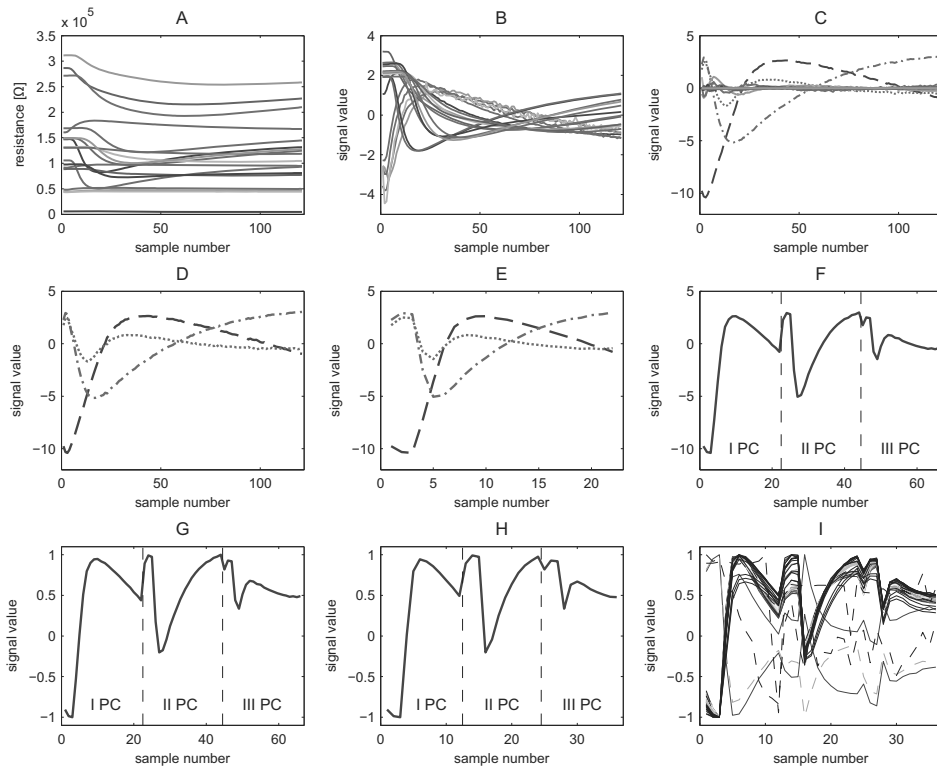
Fig. 2. Visualization of preprocessing stages for a single specimen of phenol: measured sensor responses, each color representing a separate response (A), signals after standardization (B), after PCA each PC is presented with a different color (C), the first three PCs (D), the first three PCs (every six sample left) (E), after aggregation of PC1 to PC3 (F), signal after normalization (G), signal after reduction of samples (every second sample left) (H), data for all specimens of phenol after preprocessing—the specimens of phenol corresponding to the same concentration of phenol have the same color in the graph (8 colors, 8 concentration of phenol) (I). All stages are described in Table 4.

genetic algorithm. The best individual characterized by the lowest error consisted of a set of parameters. It was used for the primary configuration of weights and biases of the networks. Such parameters were then optimized by the LM algorithm.

4. Comparison of system parameters

The most important parameters of the proposed hybrid and conventional systems are presented in Table 5. For artificial neural networks they include topology, the transfer function of neurons in successive layers and the training algorithm. For the fuzzy system these are the type with the number of inputs and outputs in brackets, and the number of inference rules. For the genetic algorithm this concerns the number of individuals, the probability of crossover and mutation, the number of generations and the method of scaling and selection. For the general parameters there is the number of elements of the training and test sets, the number of variables (i.e., parameters determined during training, such as weights and biases or parameters of membership functions in the premises and

conclusions), and the average error of approximation of concentrations of phenol for normalized values.

5. Evaluation criteria

A comparison of data analysis systems was performed on the basis of accuracy (AC) and complexity criterions (CC). AC covers 60% of the total evaluation and CC covers 40%.

The results were collected from 30 completed analyses of each of the systems, except for those trained with GAs, which require long calculation time, or an ANFIS, for which the subsequent results were identical. All results were normalized in such a way that the worst score in a group of systems is assigned 0% of the grade, and the best one 100%. Assessment values were normalized by the formulas (1) and (2):

$$y_+ = \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}} \right) \cdot 100\%, \qquad (1)$$

$$y_- = \left(\frac{x_{\max} - x}{x_{\max} - x_{\min}} \right) \cdot 100\%, \qquad (2)$$
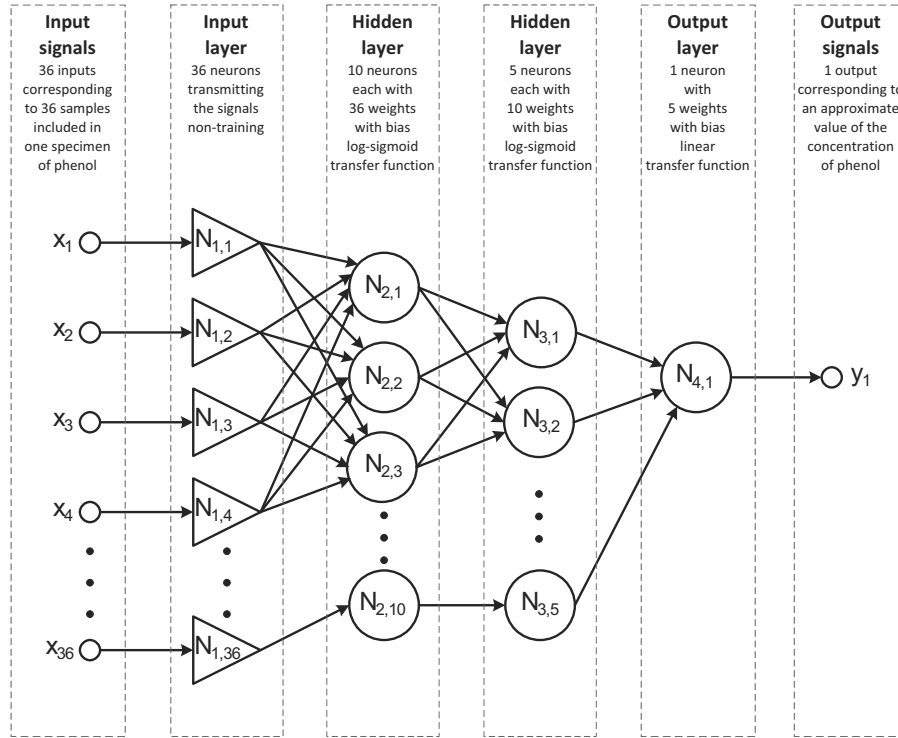
Input signals	Input layer	Hidden layer	Hidden layer	Output layer	Output signals
36 inputs corresponding to 36 samples included in one specimen of phenol	36 neurons transmitting the signals non-training	10 neurons each with 36 weights with bias log-sigmoid transfer function	5 neurons each with 10 weights with bias log-sigmoid transfer function	1 neuron with 5 weights with bias linear transfer function	1 output corresponding to an approximate value of the concentration of phenol

Fig. 3. Feed-forward neural network structure.

where y_+ is the normalized rating of the system for a given parameter x (used if the rating increases with the parameter x), y_- is the normalized rating of the system for a given parameter x (used if the rating decreases with the parameter x), x is the value of a parameter obtained by the current system evaluated, $x_{\max}$ is the maximum value of a parameter, obtained from all the analyses, $x_{\min}$ is the a minimum value of a parameter, obtained from all the analyses.

During the process of designing the systems, the emphasis was put on the criterion of accuracy, so the main goal was to achieve the lowest error. When the minimal error was reached, the overall number of system parameters was decreased.

The analysis covered 11 systems evaluated on the basis of five parameters for the accuracy criterion and three parameters for the complexity criterion. The evaluation was made on the basis of the average and maximum error for the normalized values, which were determined in Matlab, using the formula

$$\delta_{\text{av }n} = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{|M_n(i) - Y_n(i)|}{\text{range}_n} \right) \cdot 100\%, \quad (3)$$

$$\delta_{\max n} = \max \left(\frac{|M_n(i) - Y_n(i)|}{\text{range}_n} \right) \cdot 100\%, \quad (4)$$

where $\delta_{\text{av }n}$ is the average error for the normalized values, $\delta_{\max n}$ is the maximum error for the normalized values,

N is the number of elements in a test set, $M_n(i)$ is the i-th model (expected) response for the normalized values, $Y_n(i)$ is the i-th response generated by the system for the the normalized values, range_n is for normalized values equal to $4 - 0.699 = 3.301$, and the average error for real values calculated in Matlab, using the following formula:

$$\delta_{\text{av }r} = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{|M_r(i) - Y_r(i)|}{\text{range}_r} \right) \cdot 100\%, \quad (5)$$

where $\delta_{\text{av }r}$ is the average error for the real values, N is the number of elements in a test set, $M_n(i)$ is the i-th model (expected) response for the real values, $Y_n(i)$ is the i-th response generated by the system for the the real values, range_r is for real values equal to $0.1 - 5 \cdot 10^{-5} = 0.09995$.

The individual components of evaluational criteria, together with their corresponding weights, are described below. Computations were made in most cases for the thirty analyses.

- **Accuracy criterion (60%): evaluated system error**

  - AC1: Arithmetic mean of the average error ($\delta_{\text{av }n}$) generated by the evaluated system, for the whole test set for normalized values, with the weight equal to 35% of the assessment.

  - AC2: Arithmetic mean of maximum error ($\delta_{\max n}$) generated by the evaluated system,
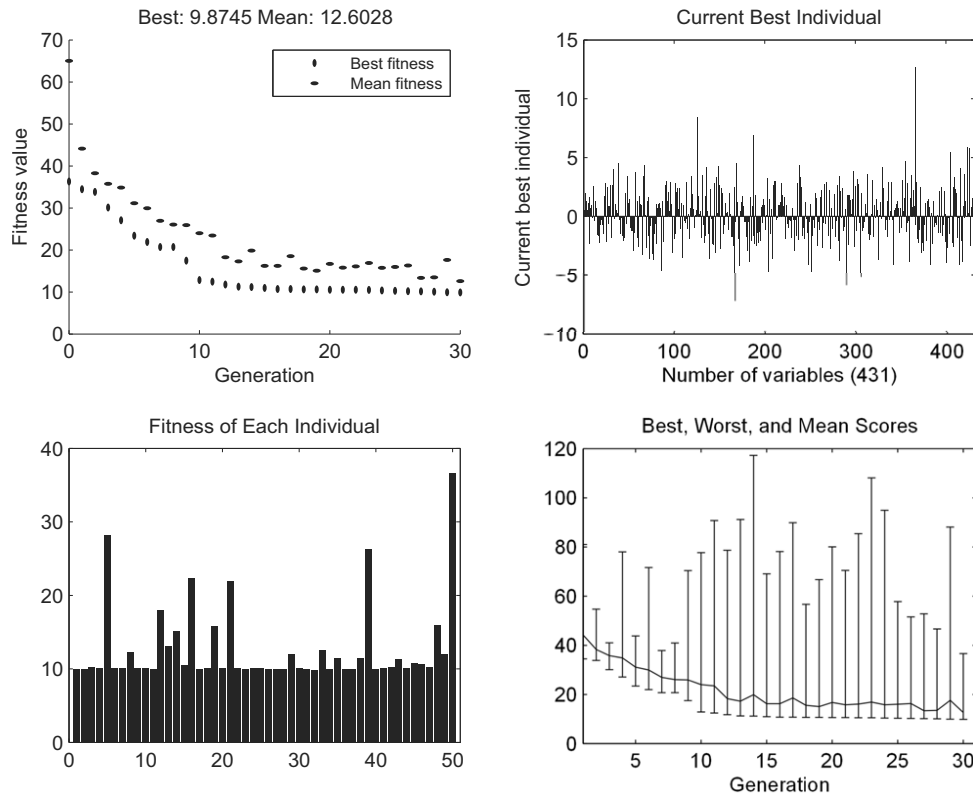
Fig. 4. Example training for an FF+GA+LM system using a genetic algorithm: properly decreasing (with successive generations) value of the fitness function for the best individual (vertical sign) and the average individual in the population (horizontal sign) (top left), set of modified values of all variables for the best current individual (with the smallest value of the fitness function—error); this simulation is also illustrated by all the weights and biases of each neuron (or the parameters of membership functions in the premises and conclusions of the fuzzy system) (top right), generation changes in the range of accommodation spread for all individuals in the population (bottom right), current value of the adaptation of all (in this case 50) individuals included in the population (bottom left).
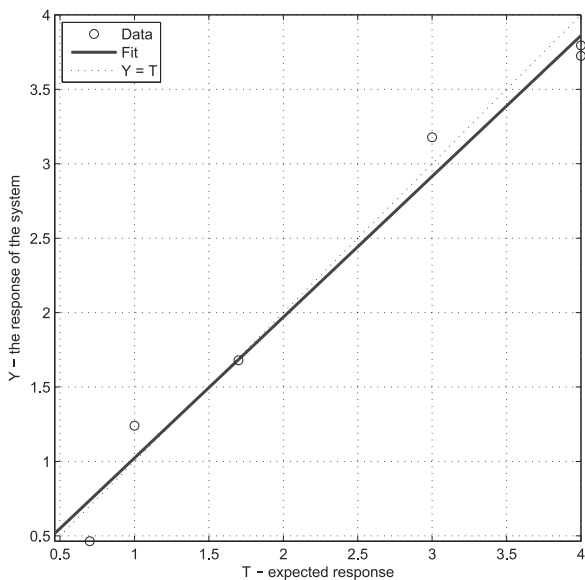

Fig. 5. FF+GA+LM system: example of the matching of the system's response to the model response for a test set for normalized values.
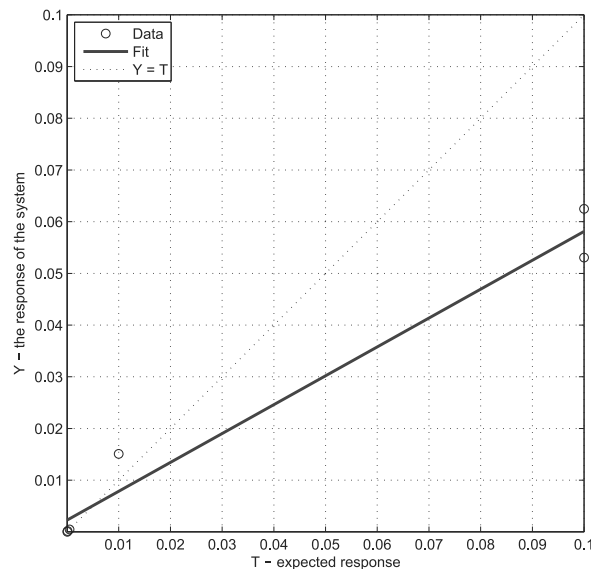
Fig. 6. FF+GA+LM system: example of the matching of the system's response to the model response for a test set for real values.

Table 3. Parameters of the hybrid system based on a feed forward neural network (FF+GA+LM).

FF + GA + LM			
Feed-forward neural network		**Genetic algorithm**	
Structure		Basic parameters	
Data flow:	unidirectional	Number of individuals in the population:	50
Topology:	multilayer	Type representation of genes:	floating-point vectors
Number of input – output:	36 – 1	Initial population:	random, uniform
Range of input signals:	[-1, 1]	Range of the genes are drawn to the initial population:	[-1.5, 1.5]
The set of model answers:	{0.7, 1.0, 1.7, 3.0, 4.0}	Stop the algorithm	
Number of layers under learning / in that hidden:	3 / 2	Target value of fitness function:	10^{-1}
The number of neurons in subsequent layers:	36 – 10 – 5 – 1	The target value of the function tolerance:	10^{-12}
Type of transfer function in the following layers:	log-sigmoid log-sigmoid linear	The maximum number of generations:	30
Training		The maximum computation time in seconds:	–
Method:	supervised	Efficiency and properties of the algorithm	
Training algorithm:	GA i LM	Type of crossover:	heuristic
Modified method of weights (biases):	–	Probability of crossover:	0.7
Error function (adaptation):	MSE	Type of mutation:	uniform
The target value of the error:	10^{-6}	Probability of mutation:	0.01
Number of epochs:	1000	The number of the fittest individuals that pass unchanged to the next generation:	2
The number of elements in the training set:	18	The method of scaling the fitness function:	ranking
The number of elements in the test set:	6	Method of selection the parents:	stochastic
Variables		The direction of migration:	bilateral
Number of weights per single neuron, in the following layers:	36 – 10 – 5	The value of the coefficient of migration:	0.2
Number of biases per single neuron, in the following layers:	1 – 1 – 1	Range of the migration:	20
The total number of variables (weights + biases network):	360 + 50 + 5 + 10 + 5 + 1 = 431	The initial value of the penalty:	–
Effectiveness		Hybrid optimization	
Average error for the concentration of phenol for normalized values:	6.67%	Hybrid Method:	–
Average error for the concentration of phenol for real values:	13.00%		

for a single measurement from the test set for normalized values, with the weight equal to 10% of the assessment.

- AC3: Maximum value of average error ($\delta_{\mathrm{av}\,n}$) generated by the evaluated system, for the whole test set for normalized values, the weight equal to 5% of the assessment.

- AC4: Minimum value of average error ($\delta_{\mathrm{av}\,n}$) generated by the evaluated system, for the whole test set for normalized values, with the

weight equal to 5% of the assessment.

- AC5: Arithmetic mean of average error ($\delta_{\mathrm{av}\,r}$) generated by the evaluated system, for the whole test set for real values, with the weight equal to 5% of the assessment.

- **Complexity criterion (40%): number of system parameters (variables)**

  - CC1: Number of variables (parameters) in an analysed system, with the weight equal to 25% of the assessment.
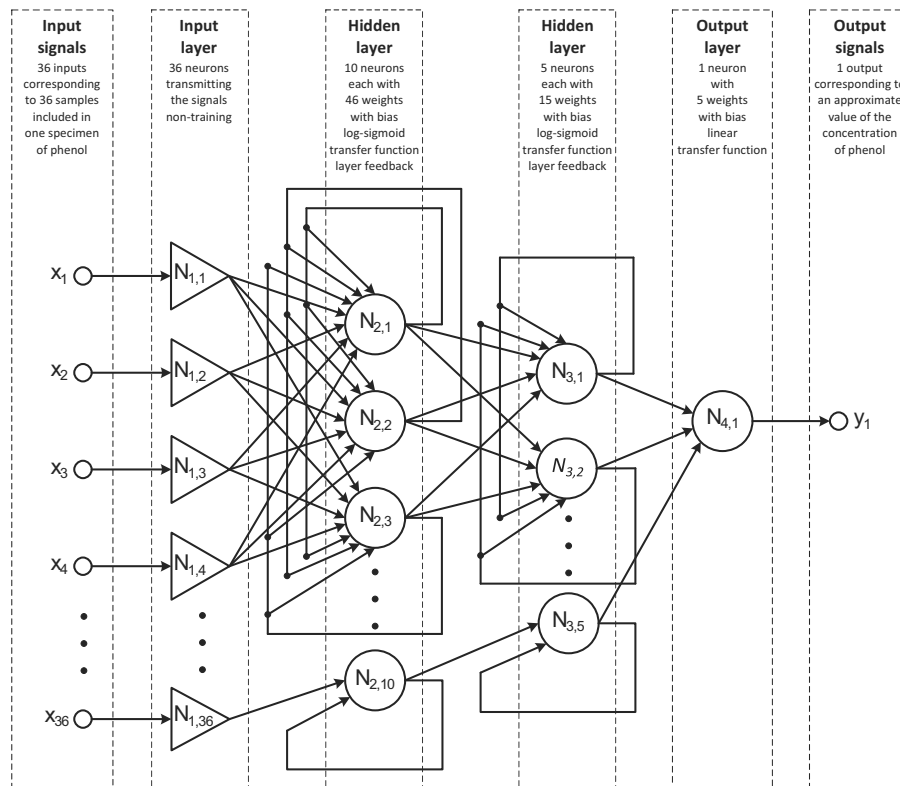
Fig. 7. Recurrent neural network structure.

– CC2: Average computation time (in seconds) needed to train the system, with the weight equal to 10% of the assessment.

– CC3: Average number of epochs needed to train the system, with the weight equal to 5% of the assessment.

6. Discussion

The comparison of results of quantitative analysis obtained for the developed hybrid and conventional systems is presented in Table 6 and Fig. 11.

The results of the performed analyses proved that hybrid combinations of AI methods are better in the evaluated criteria than conventional evolutionary-neural systems as well as feed-forward and recurrent neural networks. The new hybrid systems reached the results (WA3 parameter value—RNN+GA+LM: 77.50% and FF+GA+LM: 75.90%) close to the best, achieved by the radial neural networks (WA3—GRNN: 91.26% and RBF: 90.42%) in approximation of phenol concentration. According to Fig. 11, the four winning systems are highly efficient (over 75% score).

Noteworthy is also the fact that the FF+GA+LM system received (in one of the thirty analyses) the smallest average error in approximation of phenol concentration for normalised values (AC4: 2.64%). Also the result

obtained by the RNN+GA+LM system is very good (AC4: 3.05%).

However, it should be noted that the proposed hybrid systems are probably able to obtain even better results if the number of specimens is greater (due to their small number, increasing the number of specimens from 24 to several hundred). Such systems exhibit lower complexity (WA2) because the number of variables (CC1) in radial neural networks increases in proportion to the size of the training set. The accuracy of the novel systems (WA1—RNN+GA+LM: 73.27% and FF+GA+LM: 65.39%) should be also increased in comparison with radial neural networks (WA1—GRNN: 92.85% and RBF: 91.47%) as the developed hybrid systems were overtrained due to the relatively small number of specimens.

Regarding the complexity criterion, the new FF+GA+LM system exhibits worse results only in comparison with the FF neural network (WA2: 91.68% vs. 98.28%), but the latter obtained the worst score in the accuracy criterion (WA1: 47.67%). The RNN+GA+LM system also obtained a high score in the complexity criterion (WA2: 83.84%).

By using the GA, which pre-chooses the initial values of neural networks, their further optimization by the use of the LM algorithm is more efficient, which significantly affects the final efficiency of the systems
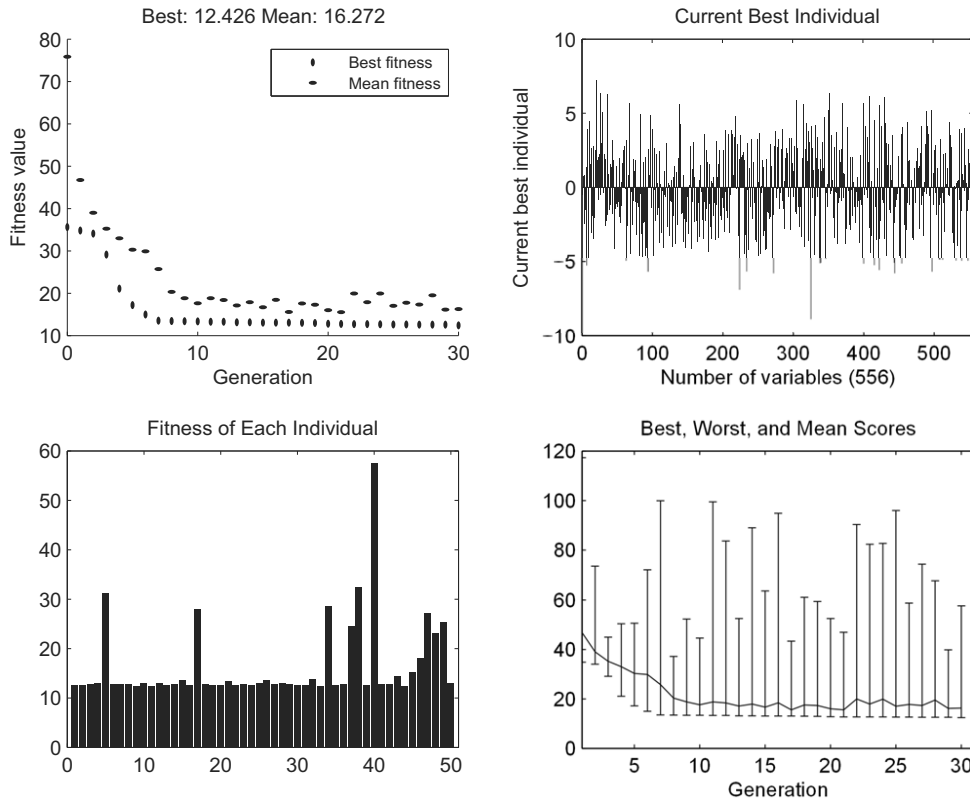
Fig. 8. Example training for an RNN+GA+LM system using a genetic algorithm: properly decreasing (with successive generations) values of the fitness function for the best individual and the average individual in the population (top left), set of modified values of all variables for the best current individual (with the smallest value of the fitness function—error); this is also illustrated by all the weights and biases of each neuron (or the parameters of membership functions in the premises and conclusions of the fuzzy system) (top right), generation changes in the range of accommodation spread for all individuals in the population (bottom right), current value of the adaptation of all (in this case 50) individuals included in the population (bottom left).
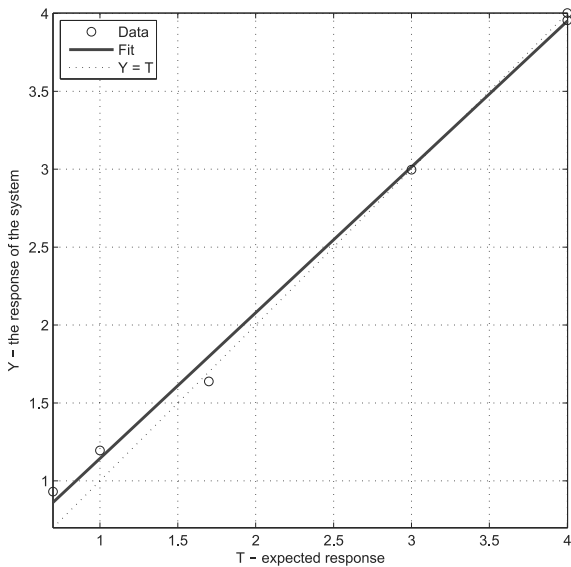

Fig. 9. RNN+GA+LM system: example of the matching of the system's response to the model response for a test set for normalized values.
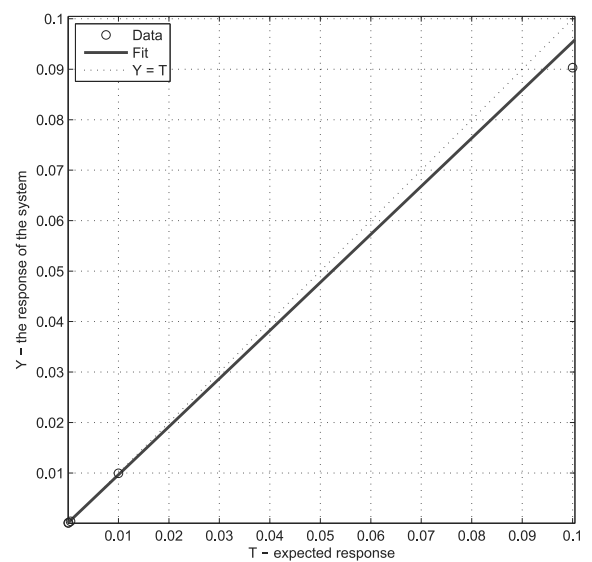

Fig. 10. RNN+GA+LM system: example of the matching of the system's response to the model response for a test set for real values.

Table 4. Parameters of the hybrid system based on a recurrent neural network (RNN+GA+LM).

RNN + GA + LM			
Recurrent neural network		**Genetic algorithm**	
Structure		Basic parameters	
Data flow:	bidirectional	Number of individuals in the population:	50
Topology:	multilayer	Type representation of genes:	floating-point vectors
Number of input – output:	36 – 1	Initial population:	random, uniform
Range of input signals:	[-1, 1]	Range of the genes are drawn to the initial population:	[-1.5, 1.5]
The set of model answers:	{0.7, 1.0, 1.7, 3.0, 4.0}	Stop the algorithm	
Number of layers under learning / in that hidden:	3 / 2	Target value of fitness function:	10^{-1}
The number of neurons in subsequent layers:	36 – 10 – 5 – 1	The target value of the function tolerance:	10^{-12}
Type of transfer function in the following layers:	log-sigmoid log-sigmoid linear	The maximum number of generations:	30
Training		The maximum computation time in seconds:	–
Method:	supervised	Efficiency and properties of the algorithm	
Training algorithm:	GA i LM	Type of crossover:	heuristic
Modified method of weights (biases):	–	Probability of crossover:	0.7
Error function (adaptation):	MSE	Type of mutation:	uniform
The target value of the error:	10^{-6}	Probability of mutation:	0.01
Number of epochs:	1000	The number of the fittest individuals that pass unchanged to the next generation:	2
The number of elements in the training set:	18	The method of scaling the fitness function:	ranking
The number of elements in the test set:	6	Method of selection the parents:	stochastic
Variables		The direction of migration:	bilateral
Number of weights per single neuron, in the following layers:	(36 + 10) – (10 + 5) – 5	The value of the coefficient of migration:	0.2
Number of biases per single neuron, in the following layers:	1 – 1 – 1	Range of the migration:	20
The total number of variables (weights + biases network):	460 + 75 + 5 + 10 + + 5 + 1 = 556	The initial value of the penalty:	–
Effectiveness		Hybrid optimization	
Average error for the concentration of phenol for normalized values:	5.88%	Hybrid Method:	–
Average error for the concentration of phenol for real values:	11.26%		

(WA3—RNN+GA+LM: 77.50% vs. RNN: 59.86% vs. RNN+GA: 52.28% and FF+GA+LM: 75.90% vs. FF: 67.92% vs. FF+GA: 54.04%).

The result of this approach is more effective approximation of the phenol concentration for novel hybrid systems (WA1—RNN+GA+LM: 73.27% vs. RNN: 40.45% vs. RNN+GA: 43.56% and FF+GA+LM: 65.39% vs. FF: 47.67% vs. FF+GA: 38.82%). One has to note that their complexity decreases (WA2—RNN+GA+LM: 83.84% vs. RNN+GA: 65.37% and FF+GA+LM: 91.68% vs. FF+GA: 76.86%) due to a

lower computation time (CC2) and a smaller number of epochs (CC3).

It should also be noted that the same genetic training or the same LM training as for the hybrid systems in the case of RNNs and FF neural networks is not sufficient because in the accuracy criterion they reached a score lower than 50% (WA1—RNN+GA+LM: 73.27% vs. RNN: 40.45% vs. RNN+GA: 43.56% and FF+GA+LM: 65.39% vs. FF: 47.67% vs. FF+GA: 38.82%).

The genetic training combined with training by the LM algorithm gives much better results in comparison

Table 5. Most important parameters of the tested data analysis systems. The hybrid systems described in the paper are shown in boldface.

System	Parameters						
	Neural network		Fuzzy system	Genetic algorithm		Overall	
	topology/ transfer functions	training algorithm	type (I-O)/ inference rules	individuals/ crossover/ mutation	generations/ scaling/ selection	training/ test set	variables/ error
Artificial neural networks (ANN)							
FF	36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear	LM	–	–	–	18 6	**431** **8.34**%
RNN	36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear	LM	–	–	–	18 6	**556** **9.15**%
RBF	36 − 18 − 1 radial − linear	mapping	–	–	–	18 6	**685** **3.65**%
GRNN	36 − 18 − 1 radial − linear	mapping	–	–	–	18 6	**684** **3.49**%
Fuzzy system (FUZZY)							
FUZZY	–	–	Sugeno (36 - 1) 17	–	–	18 6	**1853** **3.53**%
Evolutionary-neural systems (ANN + GA)							
FF GA	36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear	GA	–	50 heuristic (0.7) uniform (0.01)	100 ranking stochastic	18 6	**431** **9.12**%
RNN GA	36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear	GA	–	50 heuristic (0.7) uniform (0.01)	100 ranking stochastic	18 6	**556** **8.60**%
FF GA LM	**36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear**	**GA + LM**	**–**	**50 heuristic (0.7) uniform (0.01)**	**30 ranking stochastic**	**18 6**	**431** **6.67**%
RNN GA LM	**36 − 10 − 5 − 1 log-sigmoid log-sigmoid linear**	**GA + LM**	**–**	**50 heuristic (0.7) uniform (0.01)**	**30 ranking stochastic**	**18 6**	**556** **5.88**%
Neuro-fuzzy system (ANFIS)							
ANFIS	36 − 1260 − 1 radial − linear	BP	Sugeno (36 - 1) 17	–	–	18 6	**1853** **3.83**%
Evolutionary-fuzzy system (FUZZY + GA)							
FUZZY GA	–	–	Mamdani (36 - 1) 7	50 heuristic (0.7) uniform (0.01)	300 ranking stochastic	18 6	**518** **12.72**%

Table 6. Comparison of results obtained for the developed systems. Normalized values (y_+ and y_-) are given in parentheses. The parameters WA1, WA2 and WA3 are computed using normalized values of y_+ and y_-. The hybrid systems described in the paper are shown in boldface.

	Accuracy criterion			Complexity criterion		Weighted average	
AC1	arithmetic mean of average error for normalized values	AC4	minimum value of average error for normalized values	CC1	the number of variables (parameters)	WA1	accuracy criterion
AC2	arithmetic mean of maximum error for normalized values	AC5	arithmetic mean of average error for real values	CC2	average computation time	WA2	complexity criterion
AC3	maximum value of average error for normalized values			CC3	average number of epochs	WA3 −	criterion accuracy- -complexity

Rresults of quantitative analysis of systems

Pos.name		Accuracy criterion					Complexity criterion			Weighted average		
		AC1	AC2	AC3	AC4	AC5	CC1	CC2	CC3	WA1	WA2	WA3
	Unit	[%]	[%]	[%]	[%]	[%]	[ ]	[s]	[ ]	[%]	[%]	[%]
	Weight	35%	10%	5%	5%	5%	25%	10%	5%	60%	40%	100%
1	GRNN	3.49% (100%)	15.61% (68.99%)	3.49% (100%)	3.49% (76.37%)	0.10% (99.80%)	684 (82.21%)	1.00 (100%)	1.00 (100%)	92.85%	88.88%	**91.26**%
2	RBF	3.65% (98.25%)	8.72% (100%)	3.65% (99.25%)	3.65% (71.87%)	11.15% (38.79%)	685 (82.14%)	1.00 (100%)	1.00 (100%)	91.47%	88.84%	**90.42**%
3	**RNN GA LM**	**5.88% (74.09%)**	**12.22% (84.26%)**	**10.96% (65.29%)**	**3.05% (88.61%)**	**11.26% (38.15%)**	**556 (91.21%)**	**71.10 (66.34%)**	**54.90 (81.97%)**	**73.27**%	**83.84**%	**77.50**%
4	**FF GA LM**	**6.67% (65.52%)**	**15.70% (68.58%)**	**12.03% (60.31%)**	**2.64% (100%)**	**13.00% (28.54%)**	**431 (100%)**	**52.47 (75.29%)**	**52.20 (82.88%)**	**65.39**%	**91.68**%	**75.90**%
5	FUZZY	3.53% (99.57%)	21.17% (43.96%)	3.53% (99.82%)	3.53% (75.27%)	0.07% (100%)	1853 (0%)	1.00 (100%)	1.00 (100%)	88.34%	37.50%	**68.00**%
6	FF	8.34% (47.48%)	21.04% (44.58%)	19.15% (27.22%)	2.79% (95.84%)	13.19% (27.49%)	431 (100%)	2.10 (99.47%)	38.87 (87.34%)	47.67%	98.28%	**67.92**%
7	ANFIS	3.83% (96.29%)	21.29% (43.44%)	4.03% (97.49%)	3.69% (70.80%)	1.76% (90.64%)	1853 (0%)	1.00 (100%)	5.00 (98.66%)	84.99%	37.33%	**65.93**%
8	RNN	9.15% (38.74%)	21.15% (44.08%)	25.00% (0%)	3.11% (87.03%)	11.12% (38.95%)	556 (91.21%)	17.80 (91.93%)	85.17 (71.85%)	40.45%	88.97%	**59.86**%
9	FF GA	9.12% (39.03%)	19.99% (49.30%)	19.71% (24.59%)	4.40% (51.03%)	14.84% (18.41%)	431 (100%)	159.30 (23.99%)	100.00 (66.89%)	38.82%	76.86%	**54.04**%
10	RNN GA	8.60% (44.65%)	20.65% (46.31%)	14.26% (49.95%)	3.80% (67.58%)	18.17% (0%)	556 (91.21%)	209.25 (0%)	100.00 (66.89%)	43.56%	65.37%	**52.28**%
11	FUZZY GA	12.72% (0%)	30.95% (0%)	22.85% (10.01%)	6.23% (0%)	17.21% (5.28%)	518 (93.88%)	115.15 (45.19%)	300.00 (0%)	1.27%	69.97%	**28.75**%

with the sole LM or GA training, which is confirmed by the analysis results of the criterion AC1 (RNN+GA+LM: 5.88% vs. RNN: 9.15% vs. RNN+GA: 8.50% and FF+GA+LM: 6.67% vs. FF: 8.34% vs. FF+GA: 9.12%) and AC2 (RNN+GA+LM: 12.22% vs. RNN: 21.15% vs. RNN+GA: 20.65% and FF+GA+LM: 15.70% vs. FF: 21.04% vs. FF+GA: 19.99%) and AC3 (RNN+GA+LM: 10.96% vs. RNN: 25.00% vs. RNN+GA: 14.26% and FF+GA+LM: 12.03% vs. FF: 19.15% vs. FF+GA: 19.71%) and AC4 (RNN+GA+LM: 3.05% vs. RNN: 3.11% vs. RNN+GA: 3.80% and FF+GA+LM: 2.64% vs. FF: 2.79% vs. FF+GA: 4.40%) and AC5 (RNN+GA+LM: 11.26% vs. RNN: 11.12% vs. RNN+GA: 18.17% and FF+GA+LM: 13.00% vs. FF: 13.19% vs. FF+GA: 14.84%) in Table 6.

Comparing the two novel systems, slightly better was RNN+GA+LM due to higher accuracy (WA1: 73.27% vs. 65.39%), which one can observe as values of the parameters AC1, AC2, AC3 and AC5. Also the RNN+GA+LM system exhibits greater stability during
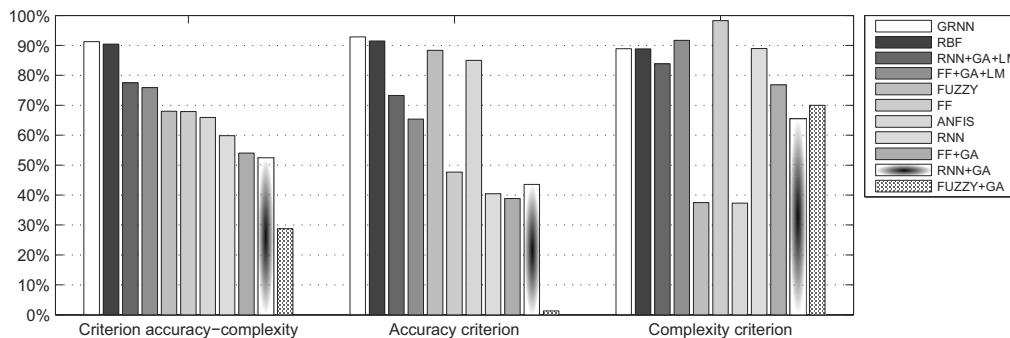
Fig. 11. Results of quantitative analysis of the systems from Table 6 for the parameters WA3, WA1 and WA2. The higher the value, the better.

learning than FF+GA+LM (AC1: 5.88% vs. 6.67%, AC2: 12.22% vs. 15.70%, AC3: 10.96% vs. 12.03% and AC5: 11.26% vs. 13.00%), at the expense of increased complexity due to an increased number of variables and longer computation time (CC1: 556 vs. 431 and CC2: 71.10 vs. 52.47 and CC3: 54.90 vs. 52.20).

7. Conclusions

The authors proposed two innovative hybrid data analysis systems. They were used for the approximation of five levels of phenol concentration. The core of such evolutionary-neural systems are feedforward and recurrent neural networks. RNNs or FF neural networks were initially trained by the genetic algorithm, and then optimized by the LM algorithm. The advantage of such systems is more effective training and relatively easier finding of lower error value. The effectiveness of neural networks trained only with a GA or only with an LM algorithm was increased by using new hybrid methods linking RNNs or FF neural networks, the GA and LM. The systems developed based on both RNNs and FFNNs yield high scores in the accuracy-complexity criteria, not much worse than the systems based on RNNs, but due to the complexity of the radial system they would exceed this system performance for larger data sets.

References

Antonelli, M., Ducange, P., Lazzerini, B. and Marcelloni, F. (2009). Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework, *International Journal of Approximate Reasoning* **50**(7): 1066–1080.

Aydogan, E., Karaoglan, I. and Pardalos, P. (2012). hGA: Hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems, *Applied Soft Computing* **12**(2): 800–806.

Benrekia, F., Attari, M. and Bermak, A. (2009). FPGA implementation of a neural network classifier for gas sensor array applications, *Proceedings of the 6th IEEE International Multi-Conference on Systems, Signals and Devices, Djerba, Tunisia.*

Cevoli, C., Cerretani, L., Gori, A., Caboni, M., Gallina, T., Toschi and Fabbri, A. (2011). Classification of Pecorino cheeses using electronic nose combined with artificial neural network and comparison with GC–MS analysis of volatile compounds, *Food Chemistry* **129**(3): 1315–1319.

Chandra, R., Frean, M., Zhang, M. and Omlin, C. (2011). Encoding subcomponents in cooperative co-evolutionary recurrent neural networks, *Neurocomputing* **74**(17): 3223–3234.

Cheng, M.-Y., Tsai, H.-C. and Sudjono, E. (2010). Evolutionary fuzzy hybrid neural network for project cash flow control, *Engineering Applications of Artificial Intelligence* **23**(4): 604–613.

Cheshmehgaz, H., Haron, H., Kazemipour, F. and Desa, M. (2012). Accumulated risk of body postures in assembly line balancing problem and modeling through a multi-criteria fuzzy-genetic algorithm, *Computers & Industrial Engineering* **63**(2): 503–512.

Czogała, E. and Łęski, J. (2000). *Fuzzy and Neuro-Fuzzy Intelligent Systems*, Physica-Verlag, Springer-Verlag Com., Heidelberg/New York, NY.

Font, J., Manrique, D. and Rios, J. (2010). Evolutionary construction and adaptation of intelligent systems, *Expert Systems with Applications* **37**(12): 7711–7720.

Ghasemi-Varnamkhasti, M., Mohtasebi, S., Siadat, M., Lozano, J., Ahmadi, H., Razavi, S. and Dicko, A. (2011). Aging fingerprint characterization of beer using electronic nose, *Sensors and Actuators B: Chemical* **159**(1): 51–59.

Ihokura, K. and Watson, J. (1994). *The Stannic Oxide Gas Sensor: Principles and Applications*, CRC Press, Boca Raton, FL.

Lin, C.-J. and Chen, C.-H. (2011). Nonlinear system control using self-evolving neural fuzzy inference networks with reinforcement evolutionary learning, *Applied Soft Computing* **11**(8): 5463–5476.

Maziarz, W. and Pisarkiewicz, T. (2008). Gas sensors in a dynamic operation mode, *Measurement Science and Technology* **19**(5): 055205.

Maziarz, W., Potempa, P., Sutor, A. and Pisarkiewicz, T. (2003). Dynamic response of a semiconductor gas sensor analysed with the help of fuzzy logic, *Thin Solid Films* **436**(1): 127–131.

M.O.S., A. (2002). *Technical note*, Toulouse, ND, www.alpha-mos.com.

Nakata, S., Neya, K. and Takemura, K. (2001). Non-linear dynamic responses of a semiconductor gas sensor: Competition effect on the sensor responses to gaseous mixtures, *Thin Solid Films* **391**(2): 293–298.

Nomura, T., Fujimori, Y., Kitora, M., Matsuura, Y. and Aso, I. (1998). Battery operated semiconductor CO sensor using pulse heating method, *Sensors and Actuators B* **52**(1): 90–95.

Patan, K. and Patan, M. (2011). Optimal training strategies for locally recurrent neural networks, *Journal of Artificial Intelligence and Soft Computing Research* **1**(22): 103–114.

Romain, A.-C., Nicolas, J., Wiertz, V., Maternova, J. and Andre, P. (2000). Use of a simple tin oxide sensor array to identify five malodours collected in the field, *Sensors and Actuators B: Chemical* **62**(1): 73–79.

Rutkowski, L. (2008). *Computational Intelligence: Methods and Techniques*, Springer, Berlin.

Shahlaei, M., Madadkar-Sobhani, A., Saghaie, L. and Fassihi, A. (2012). Application of an expert system based on Genetic Algorithm-Adaptive Neuro-Fuzzy Inference System (GA-ANFIS) in QSAR of cathepsin K inhibitors, *Expert Systems with Applications* **39**(6): 6182–6191.

Snopok, B. and Kruglenko, I. (2002). Multisensor systems for chemical analysis: State-of-the-art in electronic nose technology and new trends in machine olfaction, *Thin Solid Films* **418**(1): 21–41.

Su, C.-L., Yang, S. and Huang, W. (2011). A two-stage algorithm integrating genetic algorithm and modified Newton method for neural network training in engineering systems, *Expert Systems with Applications* **38**(10): 12189–12194.

Tabor, Z. (2009). Statistical estimation of the dynamics of watershed dams, *International Journal of Applied Mathematics and Computer Science* **19**(2): 349–360, DOI: 10.2478/v10006-009-0030-6.

Tabor, Z. (2010). Surrogate data: A novel approach to object detection, *International Journal of Applied Mathematics and Computer Science* **20**(3): 545–553, DOI: 10.2478/v10006-010-0040-4.

Tadeusiewicz, R. (2010a). New Trends in Neurocybernetics, *Computer Methods in Materials Science* **10**(1): 1–7.

Tadeusiewicz, R. (2010b). Place and role of intelligent systems in computer science, *Computer Methods in Materials Science* **10**(4): 193–206.

Tadeusiewicz, R. (2011a). How intelligent should be system for image analysis? *in* H. Kwasnicka and L.C. Jain (Eds.), *Innovations in Intelligent Image Analysis*, Studies in Computational Intelligence, Vol. 339, Springer-Verlag, Berlin/Heidelberg/New York, NY.

Tadeusiewicz, R. (2011b). Introduction to intelligent systems, *in* B.M. Wilamowski and J.D. Irvin (Eds.), *The Industrial Electronics Handbook—Intelligent Systems*, CRC Press, Boca Raton, FL.

Tadeusiewicz, R. and Morajda, J. (2012). Artificial intelligence methods, *in* P. Lula and G. Paliwoda-Pekosz (Eds.), *Analysis and Data Processing Computer Methods*, Cracow University of Economics Publishing House, Cracow.

Tallon-Ballesteros, A. and Hervas-Martinez, C. (2011). A two-stage algorithm in evolutionary product unit neural networks for classification, *Expert Systems with Applications* **38**(1): 743–754.

Tong, D. and Schierz, A. (2011). Hybrid genetic algorithm-neural network: Feature extraction for unpreprocessed microarray data, *Artificial Intelligence in Medicine* **53**(1): 47–56.

Yang, S.-H. and Chen, Y.-P. (2012). An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications, *Neurocomputing* **86**(1): 140–149.

Yu, H., Wang, J., Xiao, H. and Liu, M. (2009). Quality grade identification of green tea using the eigenvalues of PCA based on the E-nose signals, *Sensors and Actuators B: Chemical* **140**(2): 378–382.

Zhang, L., Tian, F., Kadri, C., Pei, G., Li, H. and Pan, L. (2011). Gases concentration estimation using heuristics and bio-inspired optimization models for experimental chemical electronic nose, *Sensors and Actuators B: Chemical* **160**(1): 760–770.

Pawel Pławiak received his diploma in electronics from the AGH University of Science and Technology in Cracow in 2012. At present, he works at the Institute of Telecomputing at the Cracow University of Technology and is a Ph.D. student at the Department of Automatics and Biomedical Engineering at the AGH University of Science and Technology. His research includes computational intelligence methods (e.g., artificial neural networks, genetic algorithms, fuzzy systems, support vectors machine and hybrid systems), pattern recognition, signal processing, sensor techniques, medicine and biocybernetics.

Ryszard Tadeusiewicz studied at the Electrical Engineering Department of the AGH University of Science and Technology in Cracow, from which he graduated (with honors) in 1971. Additionally, after receiving his degree in automatic control engineering, he studied at the Faculty of Medicine of the Medical Academy in Cracow, and undertook studies in the field of mathematical and computer methods in economics. Since 1971 he has been conducting research in the areas of bio-cybernetics, automatic control engineering, and computer science. In 1975 he was awarded the Ph.D. degree, and in 1981 the Doctor of

Science (habilitation) degree. In 1986 he became a professor and in 1991 a full professor at the AGH UST. His research interests cover neural networks, computer vision, biomedical engineering, and distance learning. He has written and published over 950 scientific papers, which have appeared in prestigious Polish and foreign scientific journals, as well as numerous conference presentations, both national and international. Professor Tadeusiewicz has also authored over 90 scientific monographs and books, including highly popular, often reprinted textbooks. Total number of Ph.D. graduate students supervised: 58, total number of graduate students advised as the principal adviser: 228. Elected three times as the rector of the AGH University of Science and Technology in Cracow, and several times as a member of the Governmental Central Committee for Scientific Degrees and Titles as well as the Ministry Council for Scientific Research. Awarded honorary doctorates by twelve Polish and foreign universities.