Adaptive neuro-fuzzy classifier for evaluating the technology effectiveness based on the modified Wang and Mendel fuzzy neural production MIMO-network

V I Syryamkin, S V Gorbachev, M V Shikhman

National Research Tomsk State University, Tomsk, Russia
E-mail: egs@sibmail.com

Abstract. The paper describes practical results gained during synthesis of the classification model for estimating technology effectiveness when solving multi-criteria problems of expert data analysis (Foresight) aimed to identify technological breakthroughs and strategic perspectives of scientific, technological and innovative development.

1. Introduction

Foresight research is becoming a popular tool for solving multi-criteria tasks of expert data analysis that allow identifying technological breakthroughs and strategic perspectives of scientific, technological, and innovative development [1]. As a system of methods, Foresight is constantly developing and improving; for the last 20 years, an extensive experience of its practical application has been accumulated [2]. The analysis of publications proves the effectiveness of the combination of qualitative and quantitative methods of expert data processing [3]. Moreover, with the constant complication of the system of methods, it is important to develop intellectual approaches that can provide scientifically valid transparent conclusions for Foresight research [4].

The purpose of this study is to develop a modified MIMO-structure of the adaptive fuzzy Wang and Mendel neural production network based on the fuzzy decision tree and adaptive algorithms of neuro-fuzzy inference when solving multi-criteria expert analysis problems (foresight).

2. Initialization of the modified Wang and Mendel fuzzy neural production MIMO-network

Let us have a data set D describing multidimensional feature objects (innovative technologies), where in each line there are m numerical values for the attributes $x_1,x_2,...,x_m$ and one class label C_i of the set $C = \{C_1,C_2\}$.

Let $x_{k\,j}$ be the value of the k-th input variable of the j-th observation; C_j is the value of output variable of the j-th observation; K is the number of observations; m is the number of input variables.

Rule i: IF x_1 is A_{1i} AND x_2 is A_{2i} AND ... AND x_m is A_{mi} , THEN $y_1 = r_i\,(1)$ AND $y_2 = r_i(2)$, where A_{ki} is the linguistic term by which the k-th input variable is estimated.

Figure 1 shows the structural-functional diagram of the constructed adaptive neuro-fuzzy classifier with a logic output that implements the MIMO- mapping. The module solves the problem of evaluating the effectiveness of innovative technologies as multidimensional feature objects using the fuzzy decision tree (stage 1) and fine-tuning of the parameters of membership functions and conclusion rules by hybrid learning algorithms (step 2). It consists of seven layers with the following basic functions of the nodes.

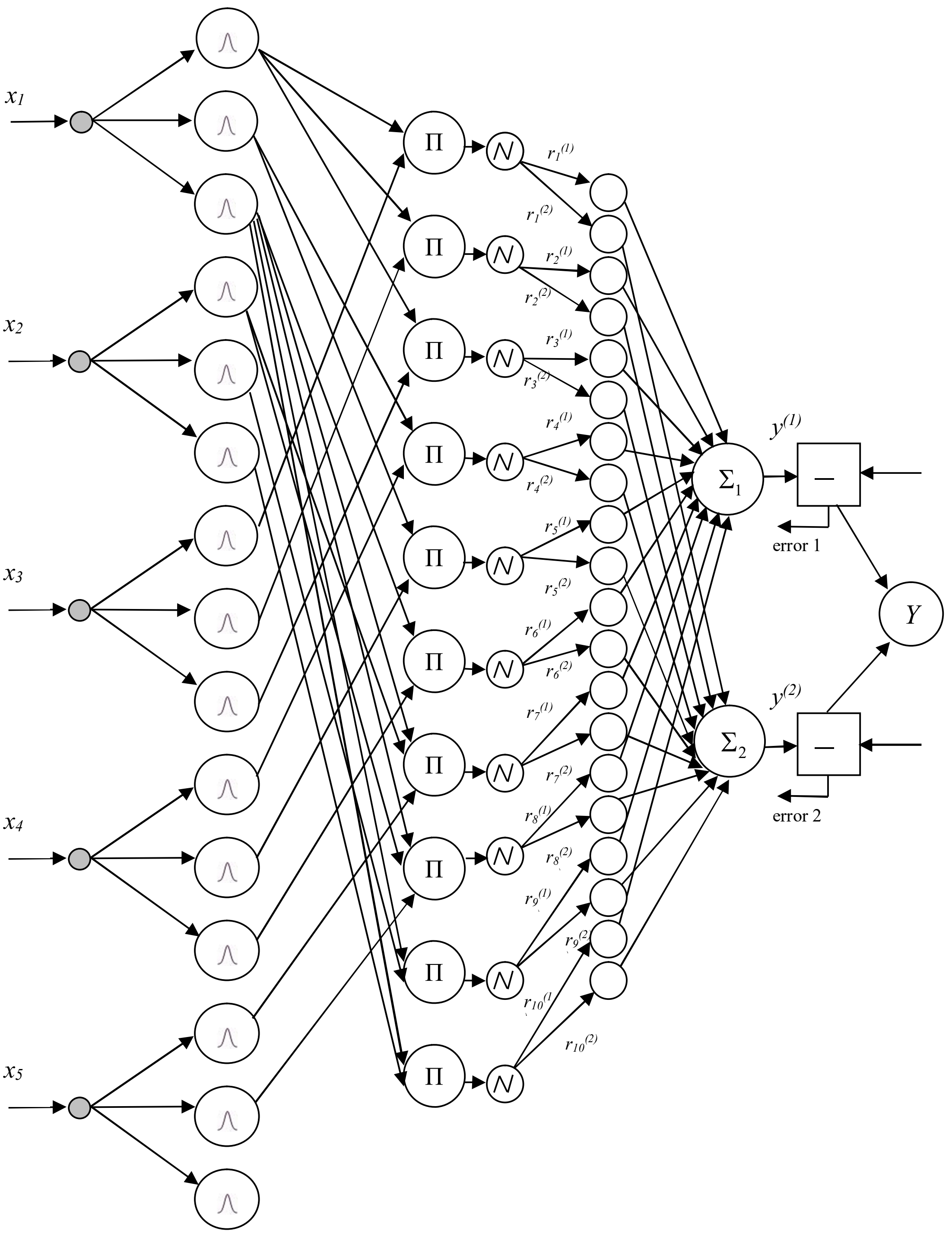

Figure 1. Structural-functional diagram of the adaptive neuro-fuzzy classifier based on the modified Wang and Mendel fuzzy neural production network with MIMO-structure.

Layer 1 (input).

Indicators evaluated by experts as innovative technologies, for example, the degree of the technology influence on global problems arising within the framework of the most pressing global crises (environmental, food, geopolitics, and energy) are considered as inputs.

The nodes in the *1*-st layer accurately convey the input attribute values into the *2*-nd layer.

Layer 2 (fuzzification) is characterized by the conversion of numerical attribute values into linguistic terms in order to reduce the information and present it in a form understandable for experts. Each *k*-th input variable is evaluated by linguistic term A_{ki}.

Layer 3 (aggregation) determines the degree of truth conditions for each of the fuzzy rules (branches of the tree); it is a non-adaptive layer of *AND*-neurons that simulate logical conjunction *AND* by multiplication of membership functions:

$$w_{ij} = \prod_k \mu_{ki}\left(x_{kj}\right) = \prod_k \frac{1}{1+\left|\frac{x_{kj}-c_{ki}}{\sigma_{ki}}\right|^{2b_{ki}}}, i=1,\dots,n,$$

where *i* is the rule number, the corresponding branch of the decision tree; in this case *i=1,...,10*;
n is the number of fuzzy rules.

The multiplication is calculated over all variables x_k presented at the *i*-th tree branch and included in the *i*-th rule.

w_{ij} can be interpreted as the weight or the degree of truth of the *i*-th rule (tree branch) condition in the *j*-th example.

The nodes of this layer are non-adaptive, they expect the relative degree (weight) for the implementation of fuzzy rules among all the rules in the *j*-th example according to the formula:

$$\overline{w_{ij}} = \frac{w_{ij}}{\sum_{i=1}^{n} w_{ij}}.$$

Layer 5 (activation) has 2 leaves at each branching point of the tree (rules) in accordance with the number of classes. On the leaves of the tree, it determines the degree of truth of the rule conclusions (the contribution of each rule) for each class using the weighting parameters r_i^1 and r_i^2 that are initialized, for example, by the method of the nearest neighbor search.

Note that the initialization of the parameters of membership functions (centers and widths) of the rule conditions and conclusions is provisional and they are to be fine-tuned on the next phase.
For each class, adaptive nodes in layer 5 calculate the degree of truth of rule conclusions with the propagation of the *j*-th signal according to the formula:

$$y_{ij}^{(1)} = \overline{w_{ij}} r_i^{(1)},\ y_{ij}^{(2)} = \overline{w_{ij}} r_i^{(2)},\ i=1,\dots,n.$$

Layer 6 (defuzzification) combines the degree of truth of the conclusions of each rule to obtain the membership functions of the output variable. For each class, non-adaptive nodes in this layer generate the output variable values $y_j^{(1)}$ and $y_j^{(2)}$, which represent the degree of truth allocating the *j*-th signal to the class as the sum of the truth degrees of all rules*:*

$$y_j^{(1)} = \sum_{i=1}^{n} y_{ij}^{(1)},\ y_j^{(2)} = \sum_{i=1}^{n} y_{ij}^{(2)}.$$

Thus, for the *j*-th example, neurons of the *6*-th layer implement the function approximation, which can be written as follows:

$$y_j = \frac{\sum_{i=1}^{n}\left(r_i \prod_k \mu_{ki}\left(x_{kj}\right)\right)}{\sum_{i=1}^{n}\prod_k \mu_{ki}\left(x_{kj}\right)} = \frac{\sum_{i=1}^{n} r_i \prod_k \frac{1}{1+\left|\frac{x_{kj}-c_{ki}}{\sigma_{ki}}\right|^{2b_{ki}}}}{\sum_{i=1}^{n}\prod_k \frac{1}{1+\left|\frac{x_{kj}-c_{ki}}{\sigma_{ki}}\right|^{2b_{ki}}}}. \qquad (1)$$

After fine-tuning parameters (step 2), *layer 7 (output)* calculates the total degree of truth Y_j of the predicted class l_j of the j-th signal according to the method of the right modal value (based on the maximum degree of truth):

$$\text{the network output } Y_j = \max\left(y_j^{(1)}, y_j^{(2)}\right),$$

$$\text{the predicted class } C_j = \arg\max_{l=1,2}\left(y_j^{(l)}\right).$$

The accuracy of fuzzy inference is achieved by collaboration of rules, which belong to the same class.

3. Adaptive tuning of parameters of the neuro-fuzzy classifier

The constructed network of fuzzy inference with forward signal propagation (Figure 2) has five input variables, each of which was fuzzified into three linguistic terms. Branches of the fuzzy decision tree that connect the layers 2 and 3 form the rule base. It is obvious that each node in layer 3 corresponds to exactly one rule. Thus, the rule base includes 10 rules.

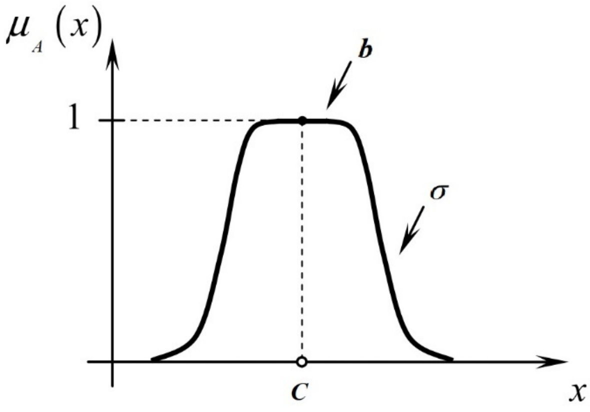

Figure 2. Generalized bell-shaped membership function.

We describe a hybrid-learning algorithm for center and width parameters of the bell-shaped membership functions and rule conclusions. In the hybrid algorithm, the parameters to be adapted are divided into two groups: linear parameters r_i of the *5*-th layer and the parameters c_{ki} and d_{ki} of nonlinear membership functions in the *2*-nd layer. Correction of the parameters is performed in two stages.

Stage 1.

When fixing specific values of the membership function parameters, we calculate linear parameters r_i by solving a system of linear equations (the value obtained in the initialization are in the first cycle). With the known values of membership functions, formula (1) in the j-th example is a linear combination of the parameters r_i with constant coefficients $\overline{w_{ij}}$:

$$y_j = \sum_{i=1}^{n} \overline{w_{ij}} r_i, \tag{2}$$

where $\overline{w_{ij}} = \dfrac{w_{ij}}{\sum_{i=1}^{n} w_{ij}} = \dfrac{\prod_k \mu_{ki}\left(x_{kj}\right)}{\sum_{i=1}^{n} \prod_k \mu_{ki}\left(x_{kj}\right)} = const$ is the normalized degree of truth of conditions at the i-th rule, $i=1,...,n$.

With K training examples ($X(j)$, d_j), $j=1, ...K$) and by substitution of the output signal with the expected value of d_j from (2), we obtain the system of K linear equations:

$$\begin{cases} \overline{w_{11}}r_1 + \ldots + \overline{w_{n1}}r_1 = d_1 \\ \ldots \\ \overline{w_{1K}}r_1 + \ldots + \overline{w_{nK}}r_1 = d_K \end{cases}$$

which can be conveniently written in the matrix form:

$$\begin{bmatrix} \overline{w_{11}} & \overline{w_{21}} & \ldots & \overline{w_{n1}} \\ \overline{w_{12}} & \overline{w_{22}} & \ldots & \overline{w_{n2}} \\ \ldots \\ \overline{w_{1K}} & \overline{w_{2K}} & \ldots & \overline{w_{nK}} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \ldots \\ r_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \ldots \\ d_K \end{bmatrix}.$$

In the abbreviated form: $w \cdot r = d$.

The dimension of matrix *W* is equal to *K·n*. With the help of pseudoinverse of *W* matrix, the solution can be obtained in one step:

$$r = W^{+}d,$$

where W^{+} is pseudoinverse of matrix *W*.

Pseudoinverse of matrix *W* is done using the singular value decomposition *(SVD)* with the following reduction of its dimension.

Recursive Greville and Fadeev algorithms belong to universal methods for finding the matrix pseudo-return. In this paper, we give the Greville algorithm for the pseudo-inversion of matrices.

4. The Greville's algorithm for the pseudo-inversion of matrices

Let us imagine that there is matrix $W \in R^{K\times n}$ and w_k is its *k*-th column, *k=1, ..., n.*

Let W_k be a matrix composed of the first *k* columns of the matrix *W*:

$$W_k = [w_1 \; w_2 \; \ldots \; w_k].$$

In case *k = 1:* $W_1 = w_1$.

In case *k = 2,..., n*: $W_k = [W_{k-1} \; w_1]$.

$W_n = W$.

Matrix $W^{+} \in R^{n\times K}$ can be computed using the recursive algorithm.

1. Initialization

$$W_1^{+} = \begin{cases} 0, \text{ if } w_1 \\ \dfrac{w_1^T}{\|w_1\|^2}, \text{ otherwise.} \end{cases}$$

2. In a cycle for *k = 2,...,n*

$$W_k^{+} = \begin{bmatrix} W_{k-1}^{+}(I - w_k f_k) \\ f_k \end{bmatrix}.$$

where *I* is the *1*-st matrix of order *K*,

$$f_k = \begin{cases} \dfrac{c_k^T}{\|c_k\|^2}, \; c_k = \left(i - W_{k-1}W_{k-1}^{+}\right)w_k, \; c_k \neq 0 \\ \dfrac{w_k^T\left(W_{k-1}^{+}\right)^T W_{k-1}^{+}}{1 + \|W_{k-1}^{+}w_k\|^2}, \; c_k = 0 \end{cases}$$

Obtained on the last step, the matrix $W_n^{+}$is the desired pseudo-inversion matrix.

Stage 2.

After fixing the values of the linear parameters r_i , the actual outputs y_j of the network for each j-th sample, $j=1,..., K$ are calculated. We use a linear relationship for this calculation:

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix} = Wr$$

Next, we calculate the error vector $\varepsilon = y - d$.

The training is based on minimizing the differentiable target function according to the method of least squares specified with the use of the Euclidean norm as a mean square deviation E [5]:

$$E = \frac{1}{2}\left(y_j - d_j\right)^2,$$

where y_j is the output value obtained as a result of fuzzy inference;
d_j is output (reference) value from the training sample.

The necessary condition for minimization of the error is that its derivatives in the parameters c_{ki}, σ_{ki}, and b_{ki} of generalized bell-shaped membership functions of the k-th input variable $\left(k \in [1,m]\right)$ on the i-th branch (the rule) are equal to zero.

The error signals are sent through the connected network in the direction to the input of the network (back propagation of error) up to the 2-nd layer, where components of the target function gradient may be calculated. After the formation of the gradient vector, the parameters are specified using one of the gradient methods of learning, for example, the method of steepest descent [6]:

$$c_{ki}(t+1) = c_{ki}(t) - \eta_c \frac{\partial E(t)}{\partial c_{ki}},$$

$$\sigma_{ki}(t+1) = \sigma_{ki}(t) - \eta_\sigma \frac{\partial E(t)}{\partial \sigma_{ki}},$$

$$b_{ki}(t+1) = b_{ki}(t) - \eta_b \frac{\partial E(t)}{\partial b_{ki}}.$$

After refinement of non-linear parameters, the process of adaptation of linear parameters (the 1-st stage) and non-linear parameters (the 2-nd stage) is restarted. This cycle is repeated until all process parameters are stabilized.

Formulas for calculating the target function gradient for one learning example *(x,d)* take the following form:

$$\frac{\partial E(t)}{\partial c_{ki}} = \left(y(x) - d\right)\sum_{i=1}^{n} r_i \frac{\partial \overline{w_i}}{\partial c_{ki}},$$

$$\frac{\partial E(t)}{\partial \sigma_{ki}} = \left(y(x) - d\right)\sum_{i=1}^{n} r_i \frac{\partial \overline{w_i}}{\partial \sigma_{ki}},$$

$$\frac{\partial E(t)}{\partial b_{ki}} = \left(y(x) - d\right)\sum_{i=1}^{n} r_i \frac{\partial \overline{w_i}}{\partial b_{ki}}.$$

where the partial derivatives to the parameters of the *bell-shaped membership function* are calculated by the following formula:

$$\frac{\partial \overline{w_h}}{\partial c_{ki}} = \frac{\delta_{hi} m(x_k) - l(x_k)}{\left[m(x_k)\right]^2} \prod_{s,s\neq k} \mu_{si}(x_s) \frac{\left[\frac{2b_{ki}}{\sigma_{ki}}\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}-1}\right]}{\left[1+\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}}\right]^2},$$

$$\frac{\partial \overline{w_h}}{\partial \sigma_{ki}} = \frac{\delta_{hi} m(x_k) - l(x_k)}{\left[m(x_k)\right]^2} \prod_{s,s\neq k} \mu_{si}(x_s) \frac{\left[\frac{2b_{ki}}{\sigma_{ki}}\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}}\right]}{\left[1+\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}}\right]^2},$$

$$\frac{\partial \overline{w_h}}{\partial b_{ki}} = \frac{\delta_{hi} m(x_k) - l(x_k)}{\left[m(x_k)\right]^2} \prod_{s,s\neq k} \mu_{si}(x_s) \frac{\left[-2\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}} \ln\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)\right]}{\left[1+\left(\frac{x_k - c_{ki}}{\sigma_{ki}}\right)^{2b_{ki}}\right]^2}$$

for *h=1,2,..,n*, (*n* is a number of tree branches (rules)),
δ_{ki} denotes the Kronecker delta:

$$\delta_{ki} = \begin{cases} 1, & k = i \\ 0, & k \neq i \end{cases}.$$

$l(x_k) = \prod_s \mu_{si}(x_s)$, $m(x_k) = \prod_s \mu_{si}(x_s)$ for all variables x_s occurring on the *i*-th branch of the decision tree and included into the *i*-th rule.

5. The calculation of the target function gradient relative to the parameters of the Gaussian membership function

Note that the final formulas for the calculation of the target function gradient relative to the parameters of the membership function depend on the used definition of the function error on the network output and on the type of the membership function. For example, when using the *Gaussian function*:

$$\mu(x) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right].$$

The appropriate formula of the target function gradient for a single example (*x, d*) from the training samples takes the following form:

$$\frac{\partial E}{\partial c_{ki}} = (y-d)(r_i - y)\overline{w_i}\frac{2(x_k - c_{ki})}{(\sigma_{ki})^2},$$

$$\frac{\partial E}{\partial \sigma_{ki}} = (y-d)(r_i - y)\overline{w_i}\frac{2(x_k - c_{ki})^2}{(\sigma_{ki})^3},$$

where $\overline{w_i} = \frac{\prod_k \mu_{ki}(x_{kj})}{\sum_{i=1}^{n}\left[\prod_k \mu_{ki}(x_{kj})\right]}$.

In the practical implementation of the hybrid learning method of the network, the dominant factor in adaptation is the first stage when the weights r_i are chosen using pseudo-inversion [7]. To balance its influence, the second stage (selection of non-linear parameters by gradient descent method) is repeated in each cycle.

Table 1 presents the average ratios of RMS errors in experiments on test data in 10-multiple cross-validation, the standard deviation of errors is presented in parentheses.

Table 1. The root means square error classification before and after the parametric settings of the neuro-fuzzy classifier.

No	**Test set**	**Fuzzy decision tree**		**Modified Wang and Mendel fuzzy neural production MIMO- network**	
		Average ratio of RMS errors	**Standard deviation**	**Average ratio of RMS errors**	**Standard deviation**
1	Breast	1.49	(0.00)	0.29	(0.0)
2	Credit	7.81	(0.00)	2.18	(0.48)
3	Cylinder	6.16	(4.51)	1.43	(3.18)
4	Diabetes	21.84	(1.27)	4.19	(1.15)
5	Gamma	21.13	(0.70)	2.53	(3.41)
6	Glass	39.13	(0.00)	5.85	(0.43)
7	Haberman	26.67	(0.00)	3.27	(1.47)
8	Heart	14.44	(5.60)	2.89	(0.87)
9	Ionosphere	3.99	(7.35)	0.53	(0.00)
10	Iris	8.00	(2.67)	3.48	(2.56)
11	Liver	36.76	(4.65)	4.13	(3.16)
12	Segmentation	12.38	(2.33)	0.14	(0.07)
13	Spam	28.98	(0.00)	2.51	(0.18)
14	Steel	20.78	(0.93)	4.17	(0.24)
15	Vehicle	25.37	(1.80)	2.95	(1.64)
16	Wine	5.00	(6.43)	0.18	(3.82)

6. Conclusion

In conclusion, we note that the practical importance of the modified Wang and Mendel fuzzy neural production MIMO-network is determined by the great potential opportunities of its application in intellectual systems for analysis and control of multi-criteria decision-making [8] in the case of processing multidimensional semi-expert data describing promising technologies and scientific and technical (innovation) objects.

The results of the study are listed below.

1. Instead of classical fuzzy production network, we can use neuro-fuzzy decision tree structure that automatically defines the base of fuzzy rules for solving problems of complex object classification in a multidimensional semi-space of features using expert data as a neuro-fuzzy system. We can also apply the inverse cycles of iteration in hybrid method of back propagation of error for adjusting the parameters (centers and widths of membership functions and the weight parameters of the rule conclusions in the leaves of the tree).

2. Synthesized knowledge base may be interpreted as a decomposition of the space of influencing factors on the field with blurred boundaries, within which the response function takes a fuzzy value. The number of such fuzzy regions is equal to the number of rules.

3. The described neural network method of adapting parameters of a decision tree (a fuzzy hierarchical system) based on the error backpropagation from leaves to the root node improves

classification accuracy of the fuzzy decision tree saving the interpretation without changing the tree structure.

4. In contrast to the known heuristic methods for the synthesis of neuro-fuzzy networks, which lead to the construction of a parametrically complex and cumbersome model with a large number of neurons, the method proposed increases the speed of the classical algorithm because of the optimized network and reduced the number of calculations.

5. From the point of view of the expert methods, we have received a new intellectual logically transparent foresight tool for conducting foresight studies

Acknowledgments
The paper was supported by “The Tomsk State University competitiveness improvement programme” under grant (No 8.2.24.2018) and by the Russian Foundation for Basic Research (grant No 16-29-04388).

The authors are grateful to Tatiana B. Rumyantseva from Tomsk State University for English language editing.

References

[1] Sokolov A. V. 2007 *Foresight* **1(1)** 8–15
[2] Georghiou I., Cassingera Harper J., Keenan M., Miles I. 2018 *The Handbook of Technology Foresight* (Edward Elgar Publishing: England)
[3] Morozova N. V. 2014 *Bulletin of the Chuvash University* **3** 178–182
[4] Gorbachev S. V., Syryamkin V. I., Syryamkin M. V., Vaganova E. V. 2017 *Innovation 2017* 66–70
[5] Kruglov V. V., Long M. I., Golunov R Y 2001 *Fuzzy logic and artificial neural networks* (FIZMATLIT: Moscow)
[6] Ezhov A. A., Shumsky S. A. *1998 Neurocomputing and its application in economics and business: teaching aid* (MEPhI: Moscow)
[7] Gorbachev S. V. 2015 *International Conference on Cognitive Computing and Information Processing* 341–346
[8] Abramova T. V., Vaganova E. V., Gorbachev S. V., Gribovsky M. V., Syryamkin V I, Syryamkin M. V., Yakubovskaya T. V. 2012 *The cognitive system of monitoring and forecast of scientific-technological development of the state* (Publishing house of Tomsk state University: Tomsk)