

Розроблено метод навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень. Метод проводить навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності; архітектури та параметрів окремого вузла мережі. В разі неможливості забезпечити задану якість функціонування штучних нейронних мереж за рахунок навчання параметрів штучної нейронної мережі відбувається навчання архітектури штучних нейронних мереж. Вибір архітектури, виду та параметрів функції належності відбувається з урахуванням обчислювальних ресурсів засобу та з врахуванням типу та кількості інформації, що надходить на вхід штучної нейронної мережі. Зазначений метод дозволяє проводити навчання окремого вузла мережі та здійснювати комбінування вузлів мережі. Розробка запропонованого методу обумовлена необхідністю проведення навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, з метою обробки більшої кількості інформації, при однаковості рішень, що приймаються. Зазначений метод навчання забезпечує в середньому на 10–18 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує помилок в ході навчання. Зазначений метод дозволить проводити навчання штучних нейронних мереж; визначити ефективні заходи для підвищення ефективності функціонування штучних нейронних мереж; підвищити ефективність функціонування штучних нейронних мереж за рахунок навчання параметрів та архітектури штучних нейронних мереж. Метод дозволить зменшити використання обчислювальних ресурсів систем підтримки та прийняття рішень; виробити заходи, що спрямовані на підвищення ефективності навчання штучних нейронних мереж; підвищити оперативність обробки інформації в штучних нейронних мережах

Ключові слова: штучні нейронні мережі, обробка інформації, інтелектуальні системи підтримки прийняття рішень

DEVELOPMENT OF A METHOD FOR TRAINING ARTIFICIAL NEURAL NETWORKS FOR INTELLIGENT DECISION SUPPORT SYSTEMS

V. Dudnyk

PnD

Department of Fire Training*

Yu. Sinenko

Associate Professor

Department of Fire Training*

M. Matsyk

Associate Professor

Department of Driving Combat Vehicles and Cars*

Ye. Demchenko

PhD, Head of Research Department

Research Department of Scientific and Methodological Support for the Development and Implementation of Programs for the Development of Weapons And Military Equipment and the State Defense Order**

R. Zhyvotovskyi

PhD, Senior Researcher, Head of Research Department

Research Department of the Development of Anti-Aircraft Missile Systems and Complexes**

Iu. Repilo

Doctor of Military Sciences, Professor

Department of Missile Troops and Artillery***

O. Zabolotnyi

PhD, Associate Professor, Leading Researcher

Center of Military Strategic Studies***

O. Symonenko

Senior Lecturer

Department of Automated Control Systems

Military Institute of Telecommunications

and Informatization named after Heroes of Kruty

Moskovska str., 45/1, Kyiv, Ukraine, 01011

P. Pozdniakov

PhD, Head of Department

Department of Tactic and General Military Sciences

Institute of Naval Forces National University «Odessa Maritime Academy»

Hradonachalnytska str., 20, Odessa, Ukraine, 65029

A. Shyshatskyi

PhD, Senior Researcher

Research Department of Electronic Warfare Development**

E-mail: ierikon12@gmail.com

*Hetman Petro Sahaidachnyi National Army Academy

Heroiv Maidanu str., 32, Lviv, Ukraine, 79026

**Central Scientifically-Research Institute of Arming

and Military Equipment of the Armed Forces of Ukraine

Povitroflotskyi ave., 28, Kyiv, Ukraine, 03168

***Ivan Chernyakhovsky National Defense University of Ukraine

Povitroflotskiy ave., 28, Kyiv, Ukraine, 03049

Received date 30.03.2020

Accepted date 18.05.2020

Published date 30.06.2020

Copyright © 2020, V. Dudnyk, Yu. Sinenko, M. Matsyk, Ye. Demchenko,

R. Zhyvotovskyi, Iu. Repilo, O. Zabolotnyi, O. Symonenko, P. Pozdniakov, A. Shyshatskyi

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>)

1. Introduction

Decision support systems (DSS) are actively used in all spheres of human life. They are especially common in the

processing of large data sets, providing information support for decision-making by decision-makers.

Currently, the basis of existing DSS are methods of artificial intelligence [1–10].

The creation of intelligent DSS has become a natural continuation of the widespread use of DSS of the classical type. Intelligent DSS provide information support for all production processes and services of enterprises (organizations, institutions). Intelligent DSS are used for design, manufacture and sale of products, financial and economic analysis, planning, personnel management, marketing, support for the creation (operation, repair) of products and long-term planning. Also, these intelligent DSS have been widely used to solve specific tasks of military purpose, namely [1, 2]:

- planning the deployment, operation of communication systems and data transmission;
- automation of troops and weapons control;
- collection, processing and generalization of intelligence information on the state of intelligence objects;
- forecasting the electronic situation in communication channels, etc.

The main tool for solving computational and other problems in modern intelligent DSS is evolving artificial neural networks (ANN).

The prospects for the use of evolving ANNs are due to the fact that the capabilities of ANNs, which do not have the possibility of evolution, do not meet the requirements for the efficiency of data processing and their learning capabilities.

Evolving ANNs have both universal approximating properties and fuzzy inference capabilities, which are the reasons why they are widely used to solve various problems of data mining, identification, emulation, forecasting, intelligent control, etc. ANNs provide stable operation in conditions of nonlinearity, uncertainty, stochasticity and chaos, various perturbations and disturbances.

Despite their successful use to solve a wide range of data mining problems, these systems have a number of disadvantages associated with their use.

The most significant shortcomings are the following:

- complexity of choosing the system architecture. As a rule, the model based on the principles of computational intelligence has a fixed architecture. In the context of ANN, this means that the neural network has a fixed number of neurons and connections. Therefore, adapting the system to new data coming in for processing that is different from previous data may be problematic;
- training in batch mode and training for several epochs requires significant time resources. Such systems are not adapted to work with a sufficiently high rate of new data for processing;
- a lot of the existing systems of computational intelligence can not determine the evolving rules by which the system develops and can also present the results of their work in terms of natural language.

Thus, the task of developing new ANN training methods, which will solve these difficulties, is urgent.

2. Literature review and problem statement

In [3], an analysis of the properties of ANNs, which were used in predicting the concentration of air pollutants is performed. It is emphasized that ANNs have a low convergence rate and a local minimum. The use of an extreme learning machine for ANN is proposed, which provides high efficiency of generalization at extremely high learning speed. The disadvantages of this approach include the accumulation of ANN errors during the calculations, the inability to choose the parameters and type of membership function.

The work [5] presents an operational approach to spatial analysis in the marine industry to quantify and reflect related ecosystem services. This approach covers the three-dimensionality of the marine environment, considering separately all marine areas (sea surface, water column and seabed). In fact, the method builds 3-dimensional sea models by estimating and mapping the associated marine domains through the adoption of representative indicators. The disadvantages of this method include the impossibility of flexible adjustment (adaptation) to evaluate models while adding (excluding) indicators and changing their parameters (compatibility and significance of indicators).

The work [6] presents a model of machine learning for automatic identification of requests and provision of information support services exchanged between members of the Internet community. This model is designed to process a large number of messages from social network users. The disadvantages of this model are the lack of mechanisms for assessing the adequacy of decisions and high computational complexity.

In [7], the use of ANN for the detection of heart rhythm abnormalities and other heart diseases is presented. The backpropagation algorithm is used as a method of ANN teaching. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

In the work [8] the use of ANN to detect the avalanche is presented. The backpropagation algorithm is used as a method of ANN training. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

In [9], the use of ANN for the detection of anomaly detection problems in home authorization systems is presented. The «winner gets everything» algorithm is used as a method of training Kohonen's ANN. The disadvantages of this approach are the accumulation of errors in the learning process, the limitation to learning only synaptic weights without learning the type and parameters of the membership function, as well as the need to store previously calculated data.

In [10], the use of ANN to identify problems in detecting abnormalities in the human encephalogram is presented. The method of fine-tuning of the ANN parameters is used as a method of ANN training. The disadvantages of this approach are the accumulation of errors in the learning process, the limitation to learning only synaptic weights without learning the type and parameters of the membership function.

In [12], the use of machine learning methods, namely ANN and genetic algorithms is presented. A genetic algorithm is used as a method of ANN training. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

In [13], the use of machine learning methods is presented, namely ANN and differential search method. During the research, a hybrid method of ANN training was developed, which is based on the use of the algorithm of error backpropagation and differential search. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

In [14], the development of methods for ANN learning using the combined approximation of the response surface, which provides the smallest errors of learning and forecasting is carried out. The disadvantage of this method is the accumulation of errors during training and the inability to change the architecture of the ANN during training.

The work [15] shows the use of ANN to assess the efficiency of the unit, using the previous time series of its performance. SBM (Stochastic Block Model) and DEA (Data Envelopment Analysis) models are used for ANN training. The disadvantages of this approach are the limited choice of network architecture and training only synaptic weights.

The work [16] shows the use of ANN for the evaluation of geomechanical properties. The backpropagation algorithm is used as a method of ANN training. Improving the characteristics of the backpropagation algorithm is achieved by increasing the training sample. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

The work [17] shows the use of ANN for estimating traffic intensity. The backpropagation algorithm is used as a method of ANN training. Improving the performance of the backpropagation algorithm is achieved by using bandwidth connections between each layer, so that each layer sets out only the residual function relative to the results of the previous layer. The disadvantage of this approach is its limitation to training only synaptic weights without training the type and parameters of the membership function.

These methods [1–17] are usually focused on learning synaptic weights or membership functions. However, these scientific works should not be used to analyze and predict the state of the electronic environment in special-purpose communication systems.

The use of the known [1–17] algorithms (methods, techniques) of training artificial neural networks for forecasting the electronic environment does not meet the existing and future requirements for them, namely:

- increasing the amount of input information that artificial neural networks can process;
- increasing the reliability of decision-making by intelligent decision support systems in the analysis of the electronic environment;
- increasing the speed of adaptation of the architecture and parameters of artificial neural networks in accordance with the tasks that arise;
- ensuring the predictability of the learning process of artificial neural networks in the analysis of the electronic environment;
- ensuring the calculation of large data sets for one era without saving previous calculations;
- impossibility to provide training for individual elements of the architecture of artificial neural networks;
- impossibility to combine the architecture of artificial neural networks (individual nodes).

3. The aim and objectives of the study

The aim of the study is to develop a method of training artificial neural networks for intelligent decision support systems that solve problems of analysis and forecasting of the electronic environment, which allows you to process more information with unambiguous decisions.

To achieve this aim, the following tasks were set:

- to determine the possibility of training the type and parameters of the membership function, as well as the architecture and parameters of a single network node in addition to training the synaptic weights of the artificial neural network;
- to conduct approbation of the proposed method.

4. Determination of the possibility of training the type and parameters of the membership function as well as the architecture and parameters of a particular network node

To solve the problems of analysis and forecasting of the electronic environment of special-purpose networks, the authors propose to use an evolving ANN cascade.

The architecture of the evolving ANN cascade [16–18] is presented in Fig. 1.

The zero layer of the system receives ($n \times 1$)-dimensional vector of input signals $x(k) = (x(k), x_1(k), x_2(k), \dots, x_n(k))^T$, which is then transmitted to the first hidden layer containing nodes-neurons, each of which has two inputs.

Output signals are formed at the outputs of $M^{[1]}$ nodes of the first hidden layer $\hat{y}_s^{[1]}$ $s = 1, 2, \dots, 0.5n(n-1) = c_n^2$. Then, these signals get to the selection unit SB , which performs the function of sorting the nodes of the first hidden layer according to the accepted criterion (usually the value of the mean square error $\sigma_{\hat{y}_s^{[1]}}^2$) so that $\sigma_{\hat{y}_1^{[1]}}^2 < \sigma_{\hat{y}_2^{[1]}}^2 < \dots < \sigma_{\hat{y}_s^{[1]}}^2$. The outputs of the $\hat{y}_1^{[1]*}$ and $\hat{y}_2^{[1]*}$ selection unit get to the input of a single node-neuron of the second layer, at the output of which the output signal $\hat{y}^{[2]*}$ is formed.

This output signal together with the output signal of the selection unit $\hat{y}_3^{[1]*}$ get to the input of the node-neuron of the next layer. The process of building cascades continues until the required accuracy of information processing is achieved.

As nodes of the considered evolving ANN cascade, two-input neuro-fuzzy systems that were considered earlier [1], and also two-input neuro-fuzzy nodes which architecture will be considered further can be used.

The problem of ANN learning is clearly complicated if the data coming to the input of the neural network are non-stationary and nonlinear, contain quasi-periodic, stochastic and chaotic components.

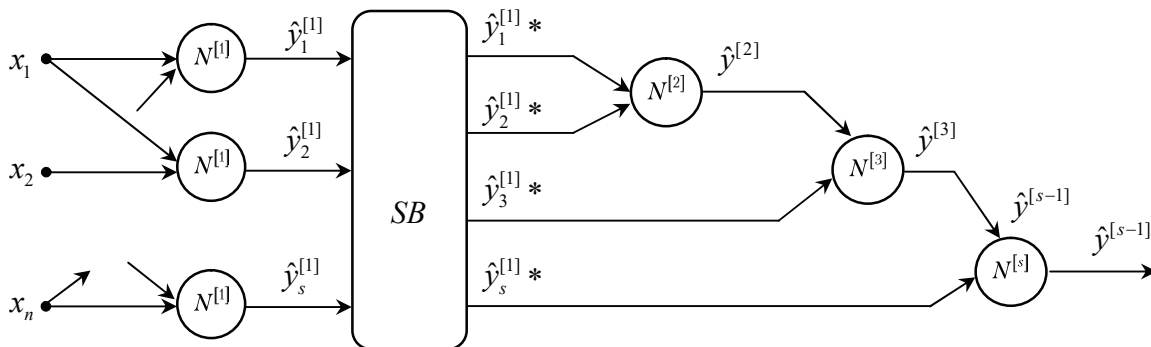


Fig. 1. Architecture of evolving ANN cascade [16–18]

In these conditions, nonlinear models based on the mathematical apparatus of computational intelligence [2–4] and, first of all, neuro-fuzzy systems proved to be the best. The advantages are the high approximating and extrapolating properties, learning ability, transparency and interpretability of the results. NARX-models should be also mentioned here and they look like:

$$\hat{y}(k) = f \left(\begin{matrix} y(k-1), y(k-2), \dots, \\ y(k-n_y), x(k-1), \dots, x(k-n_x) \end{matrix} \right) \quad (1)$$

where $\hat{y}(k)$ is the estimate (forecast) of the sequence at the moment of discrete time $k = 1, 2, \dots$; $f(\bullet)$ is some nonlinear transformation implemented by the neuro-fuzzy system; $x(k)$ is the exogenous factor that determines the behavior of $y(k)$.

It can be seen that the description (1) corresponds to the ANARX-models (Additive Nonlinear Autoregressive Exo-

genous – NARX) and WANARX-models (Additive Non-linear Autoregressive Exogenous Weighted – ANARX). Such models are well studied and there are a lot of architectures and algorithms for their training, but it is assumed that the order of the model is somehow predetermined.

In the case of structural nonstationarity of the studied series (in this case it is the analysis of the electronic situation), these orders are a priori unknown and must be adjusted in the learning process.

Given the above, the classic procedure for learning networks is the adjustment of synaptic weights, without taking into account other network learning opportunities, such as the type of architecture of individual network nodes and network composition (node combinations).

Fig. 2 shows the proposed method of training of the artificial neural network.

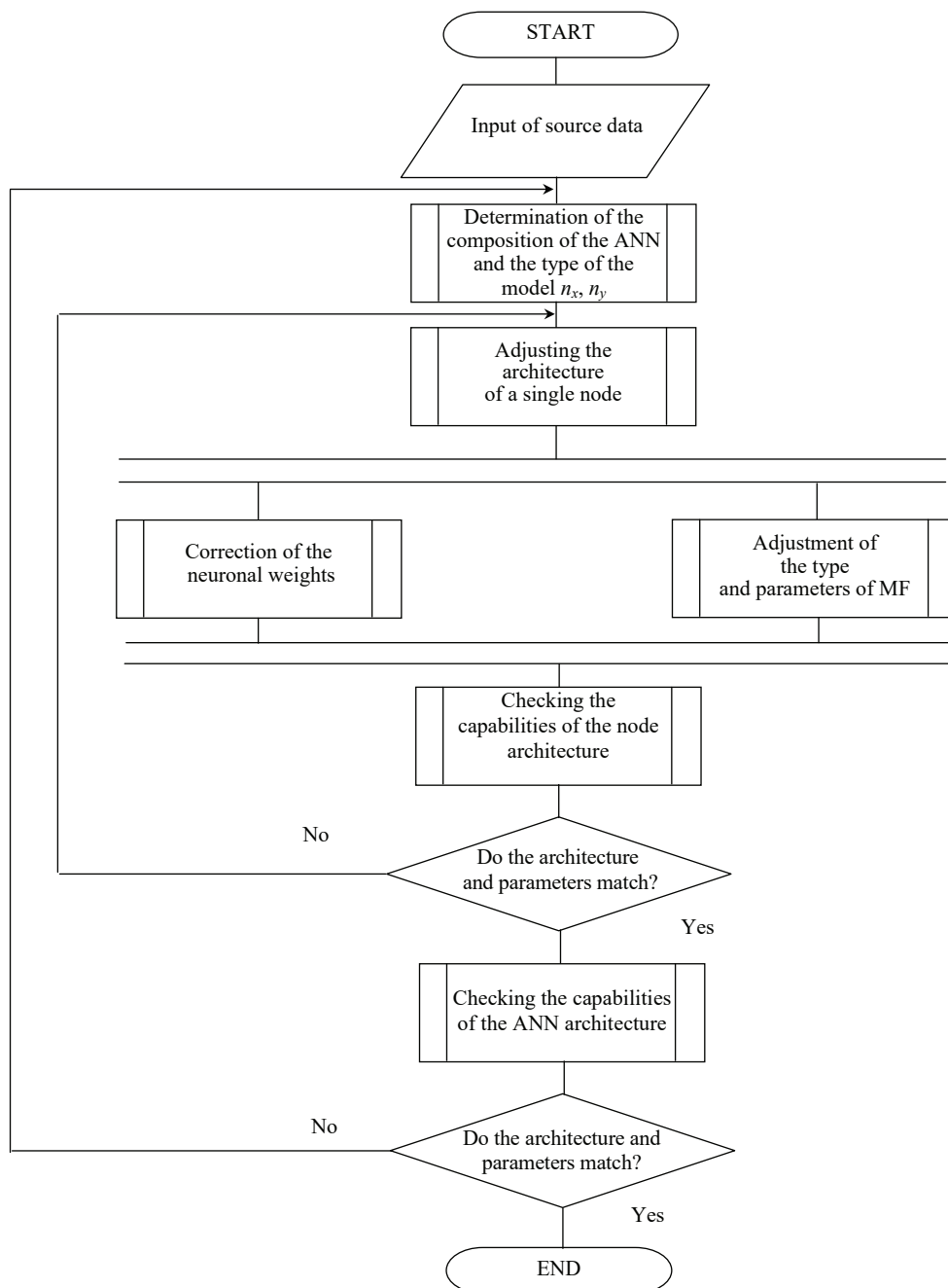


Fig. 2. Algorithm of functioning and training of the evolving ANN

Improvement of this learning algorithm consists in adding the following procedures: determining the composition of the ANN and the order of the model n_x, n_y ; adjusting the architecture of a single node; checking the capabilities of the architecture of a single node of the ANN and ANN as a whole to the known methods of training of artificial neural networks.

There is additional training of artificial neural networks, which was not taken into account in [1–17]:

- the architectures of artificial neural networks depending on the amount of source information (number of layers, number of hidden layers, number of connections between neurons in the layer and between layers);
- the architecture and parameters of a separate node of artificial neural networks;
- the possibility of combining nodes of an artificial neural network.

Step 1. The initial step consists in entering the initial data.

Step 2. Determination of the composition of the ANN (number of nodes) and the type of the model n_x, n_y .

Step 3. Correction of the architecture of a single ANN node.

Step 4. Correction of the weights of a single node neuron.

Step 5. Correction of the type and parameters of the membership function (MF). It should be noted that the implementation of steps 4 and 5 can be performed both sequentially and in parallel depending on the software implementation.

Steps 6–7. Check of the capabilities of the architecture of a single node.

Steps 8–9. Check of the architecture’s ability to process the amount of information that is coming to its input.

Let us consider the proposed method in detail.

Step 1. Entering the initial data.

At this stage, the initial parameters of the network are entered: the number of layers, the number of nodes, the number of connections between them, the initial values of the membership function.

Step 2. Determination of the composition of the ANN (number of nodes) and the type of the model n_x, n_y .

ANARX-model looks as follows [7, 17]:

$$\hat{y}(k) = f_1(y(k-1), x(k-1)) + f_2(y(k-2), x(k-2)) + \dots + f_n(y(k-n), x(k-n)) = \sum_{l=1}^n f_l(y(k-l), x(k-l)), \quad (2)$$

where $n = \max\{n_y, n_x\}$, the initial task of the synthesis of the prediction system is decomposed into many local problems of parametric identification of models-nodes with two inputs $y(k-l), x(k-l), l = 1, 2, \dots, n, \dots$

Fig. 3 shows the architecture of the ANARX-system, formed by two lines of pure delay elements z^{-1} ($z^{-1}y(k) = y(k-1)$) and n parallel connected nodes $N^{[l]}$.

Training of these nodes is carried out independently of each other, and the introduction of new nodes or the exclusion of redundant ones does not affect all other neurons, so the evolution of such a system is realized by elementary manipulation of the number of nodes.

As the node of the ANARX-system under consideration, the two-input neuro-fuzzy systems discussed earlier, as well as the two-input neo-fuzzy nodes are used.

Step 3. Correction of the architecture of a single ANN node.

When the speed of data processing and simplicity of numerical implementation of the computational intelligence system come to the fore, instead of neuro-fuzzy nodes of the ANARX model, it is advisable to use neuro-fuzzy neurons, which belong to the nonlinear learning systems.

The architecture of the neuro-fuzzy-neuron as a node of the ANARX-system is shown in Fig. 4. The advantages of the neuro-fuzzy neuron include high learning speed, computational simplicity, good approximating properties, and the ability to find a global minimum of the learning criterion.

The constituent elements of the neuro-fuzzy neuron are nonlinear synapses NS_y, NS_x , in which the rules of fuzzy zero-order Takagi-Sugeno inference are implemented, however, as it is easy to notice, the neuro-fuzzy-neuron is much simpler structurally than the neuro-fuzzy node shown in Fig. 3.

When the input of such a node receives signals $y(k-l), x(k-l)$, the output value is formed:

$$\hat{y}(k) = \varphi'_y(k) + \varphi'_x(k) = \sum_{i=1}^h w'_{iy} \mu_{iy}(y(k-l)) + \sum_{i=1}^h w'_{ix} \mu_{ix}(x(k-l)), \quad (3)$$

w'_{ix} is the synaptic weights of the l -function neuron, μ_{ix} is the membership function.

And at the output of the ANARX-model as a whole:

$$\hat{y}(k) = \sum_{l=1}^n \left(\sum_{i=1}^h w'_{iy} \mu_{iy}(y(k-l)) + \sum_{i=1}^h w'_{ix} \mu_{ix}(x(k-l)) \right), \quad (4)$$

since the neuro-fuzzy neuron is also an additive model, the ANARX model on neuro-fuzzy neurons is twice additive.

Step 4. Correction of the weights of a single node neuron.

Correction of neuronal weights in the ANN node is based on the known scientific approaches, described for example in [18, 19].

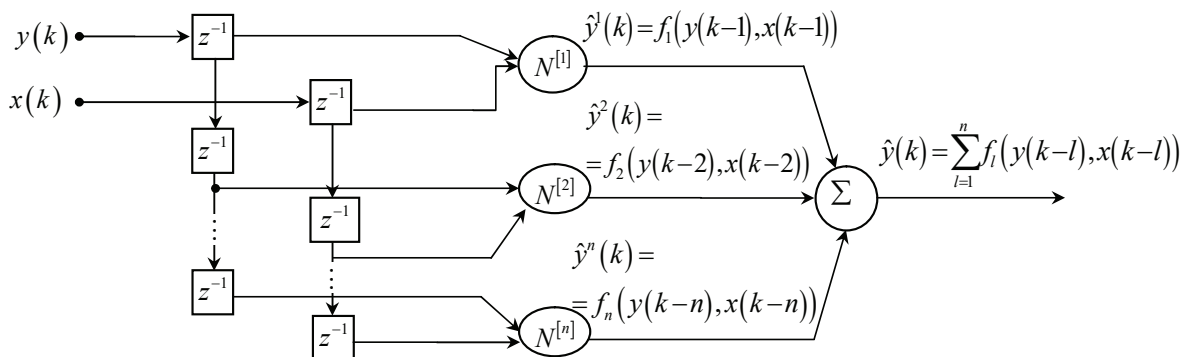


Fig. 3. Architecture of the ANARX-system [16–18]

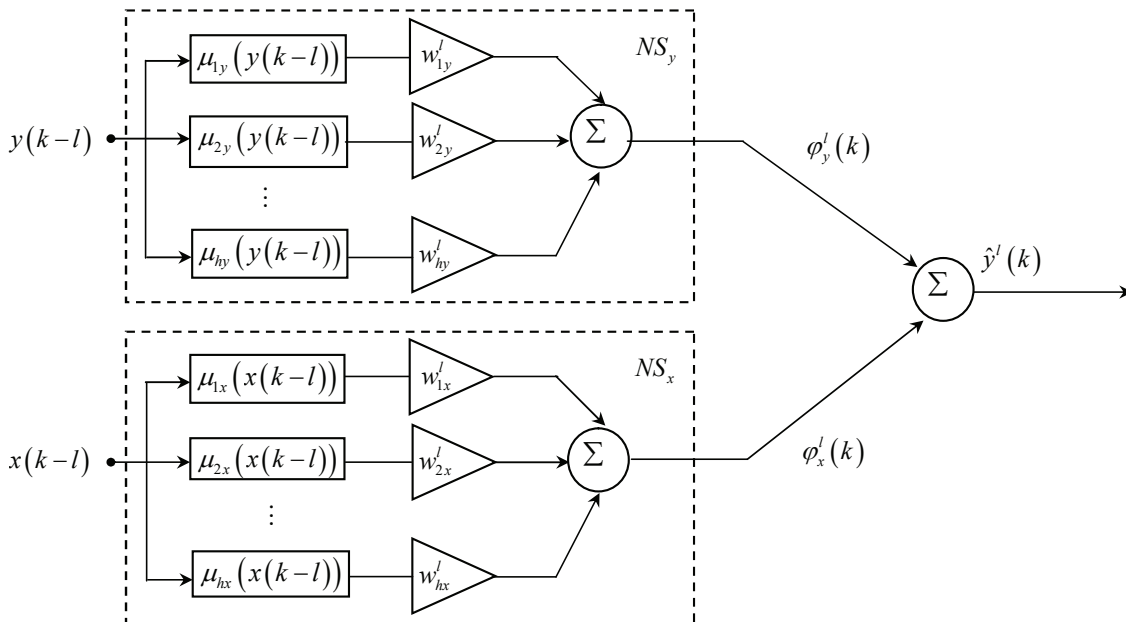


Fig. 4. Neo-fuzzy node of the ANARX system [16–18]

Step 5. Correction of the type and parameters of the membership function (FN).

As membership functions in the neuro-fuzzy-neuron, triangular constructions corresponding to the conditions of a single partition are usually used.

$$\sum_{i=1}^h \mu_{iy}(y(k-l)) = 1; \sum_{i=1}^h \mu_{ix}(x(k-l)) = 1, \tag{5}$$

which simplifies the design of the node, eliminating the layer of normalization. As membership functions of the neuro-fuzzy-neuron, it was proposed to use *B*-splines, which provide a higher quality of approximation and also meet the conditions of a single breakdown. In this case, for the *B*-spline of the *q*-th order, the following can be written:

$$\mu_{iy}^q(y(k-l)) = 1, \begin{cases} 1, \text{ if } c_{iy} \leq y(k-l) < c_{i+1,y} \\ 0 \text{ otherwise} \end{cases} \text{ for } q = 1, \begin{cases} \frac{y(k-l) - c_{iy}}{c_{i+q,y} - c_{iy}} \mu_{iy}^{q-1}(y(k-l)) + \frac{c_{i+q,y} - y(k-l)}{c_{i+q,y} - c_{i+1,y}} \times \\ \times \mu_{i+1,y}^{q-1}(y(k-l)), \text{ for } q > 1, \\ i = 1, \dots, h - q, \end{cases} \tag{6}$$

where $c_i = (c_{i1}, c_{i2}, \dots, c_{in})^T$.

$$\mu_{ix}^q(x(k-l)) = 1, \begin{cases} 1, \text{ if } c_{ix} \leq x(k-l) < c_{i+1,x} \\ 0 \text{ otherwise} \end{cases} \text{ for } q = 1, \begin{cases} \frac{x(k-l) - c_{ix}}{c_{i+q,x} - c_{ix}} \mu_{ix}^{q-1}(x(k-l)) + \frac{c_{i+q,x} - x(k-l)}{c_{i+q,x} - c_{i+1,x}} \times \\ \times \mu_{i+1,x}^{q-1}(x(k-l)), \text{ for } q > 1, \\ i = 1, \dots, h - q. \end{cases} \tag{7}$$

It should be mentioned that *B*-splines are a kind of generalized membership functions: for example, at *q*=2 we obtain traditional triangular membership functions, at *q*=4 we obtain cubic splines, etc.

Entering further vector variables:

$$\begin{aligned} w^l &= (w'_{1,y}, \dots, w'_{h,y}, w'_{1,x}, \dots, w'_{h,x})^T, \\ \varphi^l(k) &= (\mu_{1y}(y(k-l)), \dots, \mu_{hy}(y(k-l)), \\ &\mu_{1x}(x(k-l)), \dots, \mu_{hx}(x(k-l)))^T \end{aligned}$$

(4) can be rewritten:

$$\hat{y}^l(k) = w^{lT} \varphi^l(k) \tag{8}$$

and using the optimal gradient one-step Kachmazh-Widrow-Hoff algorithm:

$$\begin{aligned} w^h(k) &= w^h(k-1) + \\ &+ \frac{y(k) - w^{hT}(k-1)\varphi^h(x(k))}{\|\varphi^h(x(k))\|^2} \varphi^h(x(k)) \end{aligned} \tag{9}$$

we obtain taking into account (9):

$$\begin{aligned} w^l(k) &= w^l(k-1) + r_l^{-1}(k) \times \\ &\times (y(k) - w^{lT}(k-1)\varphi^l(k))\varphi^l(k), \\ r_l(k) &= \alpha r_l(k-1) + \varphi^{lT}(k)\varphi^l(k), \quad 0 \leq \alpha \leq 1, \end{aligned} \tag{10}$$

which has both filtering and tracking properties. It can also be seen that for $\alpha=1$ (9) it completely coincides with the optimal Kachmazh-Widrow-Hoff algorithm (10).

Steps 6–7. Check of the capabilities of a single node architecture.

At this stage, the ability of the architecture of the ANN node with certain parameters to perform a computational

task is tested. The ability to perform a computational task is determined by comparing the computational capabilities of the architecture and parameters of the ANN node and the required computing resources of the ANN node.

In the case of mismatch of computing resources of the ANN node, there is a change of parameters of the ANN node, and in case of impossibility to increase computing resources of the node – there is a change of architecture and parameters of the ANN node.

Steps 8–9. Check of the architecture’s ability to process the amount of information coming to its input.

Since each node of the ANARX $N^{[l]}$ model is configured independently of the others and is essentially a separate neuro-fuzzy system, to improve the quality of the obtained predictions, you can use the idea of combining an ensemble of neural networks. This approach naturally leads to the architecture of the weighted ANARX neuro-fuzzy-WANARX system shown in Fig. 5.

The output signal of this system can be written as:

$$\hat{y}(k) = \sum_{l=1}^n c_l \hat{y}^l(k) = c^T \hat{\bar{y}}(k), \quad (11)$$

where $\hat{\bar{y}}(k) = (\hat{y}^1(k), \hat{y}^2(k), \dots, \hat{y}^n(k))^T$, is the vector of the weight coefficients that adjust and determine the proximity of the signals $\hat{y}^l(k)$ to the process that is predicted (processed) by $y(k)$ and meets the conditions of non-bias:

$$\sum_{l=1}^n c_l = c^T I_n = 1, \quad (12)$$

where $I_n - (n \times 1)$ is the vector formed by units.

To find the vector c in batch mode, you can use the method of undetermined Lagrange multipliers. To do this, a sequence of errors is used:

$$\begin{aligned} v(k) &= y(k) - \hat{y}(k) = y(k) - c^T \hat{\bar{y}}(k) = \\ &= c^T I_n y(k) - c^T \hat{\bar{y}}(k) = c^T (I_n y(k) - \hat{\bar{y}}(k)) = c^T V(k). \end{aligned} \quad (13)$$

Lagrange function:

$$\begin{aligned} L(c, \lambda) &= \sum_k c^T V(k) c + \lambda (c^T I_n - 1) = \\ &= c^T R c + \lambda (c^T I_n - 1), \end{aligned} \quad (14)$$

where λ is the undetermined Lagrange multiplier, $R = \sum_k V(k) V^T(k)$ is the correlation matrix of errors.

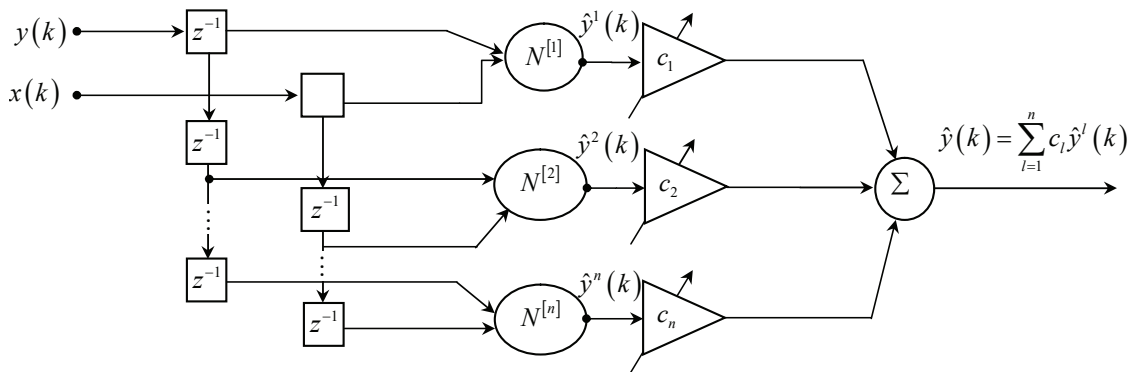


Fig. 5. Architecture of the WANARX-system [16–18]

The system of Karush-Kun-Tucker equations is also used:

$$\begin{cases} \nabla_c L(c, \lambda) = 2Rc + \lambda I_n = \vec{0}, \\ \frac{\partial L}{\partial \lambda} = c^T I_n - 1 = 0. \end{cases} \quad (15)$$

The solution of the system (15) will lead to:

$$\begin{cases} c = R^{-1} I_n (I_n^T R^{-1} I_n)^{-1}, \\ \lambda = -2 I_n^T R^{-1} I_n, \end{cases} \quad (16)$$

while the Lagrangian (17) at the saddle point takes the value:

$$L^*(c, \lambda) = (I_n^T R^{-1} I_n)^{-1}. \quad (17)$$

The implementation of the algorithm (16) may encounter significant difficulties in processing information with a high degree of correlation of signals $\hat{y}^l(k)$. This leads to poor conditionality of the matrix R , which must be rotated on each clock of real time k .

We write the Lagrange function (14) as:

$$L(c, \lambda) = \sum_k y(k) - c^T \hat{\bar{y}}(k) + \lambda (c^T I_n - 1) \quad (18)$$

and a gradient algorithm for finding its saddle point based on the Arrow-Hurwitz procedure [17, 18]:

$$\begin{cases} c(k) = c(k-1) - \eta_c(k) \nabla_c L(c, \lambda), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k) \frac{\partial L(c, \lambda)}{\partial \lambda}, \end{cases} \quad (19)$$

or

$$\begin{cases} c(k) = c(k-1) + \eta_c(k) \left(\begin{matrix} 2(y(k) - c^T(k-1)\hat{\bar{y}}(k)) \times \\ \times \hat{\bar{y}}(k) - \lambda(k-1) I_n \end{matrix} \right) = \\ = c(k-1) + \eta_c(k) (2v(k)\hat{\bar{y}}(k) - \lambda(k-1) I_n), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k) (c^T(k) I_n - 1), \end{cases} \quad (20)$$

where $\eta_c(k), \eta_\lambda(k)$ are the learning step parameters.

The Arrow-Hurwitz procedure is reduced to a saddle point under the fairly general assumptions of relative values $\eta_c(k), \eta_\lambda(k)$, but these parameters can be optimized to speed up the learning process.

To do this, we multiply the first relation (20) on the left by $\hat{y}(k)$:

$$\hat{y}(k)c(k) = \hat{y}^T(k)c(k-1) + \eta_c(k) \times \left(2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right) \tag{21}$$

and use an additional function that characterizes the criterion convergence:

$$\begin{aligned} \left(y(k) - \hat{y}^T(k)c(k) \right)^2 &= v^2(k) - 2\eta_c(k)v(k) \times \\ &\times \left(2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right) + \\ &+ \eta_c^2(k) \left(2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right). \end{aligned} \tag{22}$$

Solution of the differential equation:

$$\begin{aligned} \frac{\partial \left(y(k) - \hat{y}^T(k)c(k) \right)^2}{\partial \eta_c(k)} &= -2v(k) \left(2v(k)\|\hat{y}(k)\|^2 - \right. \\ &\left. - \lambda(k-1)\hat{y}^T I_n \right) + \\ &+ 2\eta_c(k) \left(2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right) = 0 \end{aligned} \tag{23}$$

allows us to get the optimal values of the learning step $\eta_c(k)$ in the form:

$$\eta_c(k) = \frac{v(k)}{2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n}, \tag{24}$$

substituting it in (20), it is finally possible to write down:

$$\begin{cases} c(k) = c(k-1) + \frac{v(k)\left(2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right)}{2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n}, \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\left(c^T(k)I_n - 1 \right). \end{cases} \tag{25}$$

It is easy to notice that the procedure (25) coincides with the Kachmazh-Widrow-Hoff algorithm (9).

Based on equations (11)–(25), the ability of the network architecture and parameters to process information with a given degree of reliability of the obtained results is checked. On the basis of the specified comparison, at the first stage the decision concerning the adjustment of ANN parameters, and in case of adjustment impossibility of the ANN architecture is made.

5. Approbation of the method of training artificial neural networks for intelligent decision support systems

Simulation of the proposed method was performed in the Matlab 2019 software environment.

To demonstrate the effectiveness of the proposed weighted ANARX system, a forecast of the electronic environment of special radio systems was made.

To evaluate the effectiveness of the proposed method, modeling was performed using the following components:

- personal computer with special software and Matlab 2019;
- Agilent OmniBER 718 digital flow analyzer with software and a set of connecting cables that measures parameters;

- TRC 274 H/V/UHF Jammer (20-3000), which simulated the operation of the electronic warfare system (transmitter power is 20 W, the frequency band that can be suppressed is 10 MHz, the type of interference is the noise interference with frequency manipulation; strategy of the REB complex is dynamic);

- MikroTik NetMetal 5 broadband radio access stations with the following parameters (128 positional quadrature amplitude manipulation; radiation bandwidth is 40 MHz, radiation power is 1 W; radiation frequency is 2.4 GHz).

To predict the state of the electronic environment, the training of the artificial neural network lasted for three sessions of 17 hours each (full measurement cycle of Agilent OmniBER 718 for broadband facilities).

After that, the forecast of the state of the electronic situation was carried out. Approbation was performed under equal conditions for each of the systems.

The square root of the RMSE standard error was used as a prediction quality criterion.

A multilayer perceptron (MLP), a radial basis function network (RBFN), and an adaptive neuro-fuzzy inference system (ANFIS) were used to compare prediction quality.

The results of forecasting the state of the electronic environment for different systems are presented in Table 1.

Table 1

Forecasting results for different systems

Name of the system	Number of adjustable parameters	RMSE (training)	RMSE (test)	Time, sec
MLP	53	0.1158	0.1507	0.1181
RBFN	23	0.1166	0.2255	0.1181
ANFIS	81	0.0659	0.1965	0.1181
Evolving cascade system with neo-fuzzy nodes	21	0.0584	0.1181	0.1181

Since the multilayer perceptron cannot operate in near-real-time mode, 2 variants of this system were selected for comparison.

The first multilayer perceptron was trained during one epoch. The second multilayer perceptron was trained for five epochs, and the number of adjustable parameters in this case was approximately equal to the number of adjustable parameters in the proposed systems. In both cases, the multilayer perceptrons contained 4 inputs and 7 nodes in the hidden layer. The number of adjustable parameters was 43.

For the second multilayer perceptron, the operating time was almost 2 times longer, but at the same time, the prediction quality was also almost 2 times better. Two radial basis function networks were also selected.

The number of parameters of the first radial basis function network was almost equal to the number of parameters of the proposed systems. The architecture of the second radial basis function network was chosen taking into account the quality of its operation. In the first case, the radial basis function network had 3 inputs and 7 nuclear functions. In the second case, the radial basis function network also had 3 inputs, but 12 nuclear functions.

The ANFIS system showed one of the best prediction results in this experiment. It had 4 inputs, 55 nodes and was studied for five epochs. For the training, the parameter α was

set equal to 0.62. The system contained 37 adjustable parameters. *B*-splines with $q = 2$ (triangular membership functions) were used as membership functions. The forecast quality of this system was quite high, and the training time was the shortest.

Table 2

Comparison of the forecasting results

Systems	Number of adjustable parameters	RMSE (training)	RMSE (test)	Time, sec
MLP (version 1)	44	0.0610	0.0710	0.4263
MLP (version 2)	44	0.0246	0.0391	0.9319
RBFN (version 1)	37	0.0661	0.0842	0.6562
RBFN (version 2)	62	0.0473	0.0604	1.1250
ANFIS	82	0.0247	0.0396	0.7131
ANARX with neuro-fuzzy nodes	39	0.0923	0.0952	0.4310
Weighted ANARX with neuro-fuzzy nodes	38	0.0437	0.0553	0.3760

The research of the developed method showed that this training method provides on average 10–18 % higher learning efficiency of artificial neural networks and does not accumulate errors during training (Table 1, 2). This can be seen by the efficiency of data processing in the last columns of Table 1, 2.

6. Discussion of the results of development of training method of artificial neural networks for intelligent decision support systems

The main advantages of the proposed evaluation method are:

- does not accumulate learning errors during the training of artificial neural networks by adjusting the parameters and architecture of the artificial neural network. This is explained by expressions (1)–(3), (9);
- unambiguity of the obtained results – expressions (3), (4);
- wide scope of use (decision support systems);
- simplicity of mathematical calculations;
- possibility of adapting the system during operation and its individual elements – expression (9);
- possibility of synthesizing the optimal structure of the decision-support system – expressions (11)–(25).

As for the limitations of this method, this method is adapted for analysis and forecasting of the electronic environment in terms of its uncertainty and high dynamics. However, the proposed method is able to successfully solve the problem of data analysis and forecasting with appropriate adaptation to a particular type of decision support systems.

The disadvantages of the proposed method include:

- loss of informativeness in the assessment (forecasting) due to the construction of the membership function. This loss of information can be reduced by choosing the type of membership function and its parameters in the practical implementation of the proposed methodology in decision support systems. The choice of the type of membership function depends on the computing resources of a particular electronic computing device.
- lower accuracy of assessment by a single parameter of state estimation;

- loss of accuracy of results during the reconstruction of the architecture of the artificial neural network.

This method will allow:

- to train artificial neural networks;
- to identify effective measures to improve the efficiency of artificial neural networks;
- to increase the efficiency of artificial neural networks by training the parameters and network architecture;
- to reduce the use of computing resources of decision support systems;
- to develop measures aimed at improving the learning efficiency of artificial neural networks;
- to increase the efficiency of information processing in artificial neural networks.

This research is a further development of research conducted by the authors. It is aimed at the development of theoretical foundations for improving the efficiency of artificial intelligence systems published earlier [1, 2, 29–32].

Areas of further research should be aimed at reducing the computational costs in the processing of various types of data in special-purpose systems.

7. Conclusions

1. A method of training artificial neural networks for intelligent decision support systems has been developed.

Improvement of the efficiency of information processing, reduction of the estimation and forecasting error are achieved by:

- learning not only the synaptic weights of the artificial neural network, but also the type and parameters of the membership function;
- learning the architecture of artificial neural networks;
- the possibility of combining elements of an artificial neural network, which will allow the adaptation of the architecture and parameters of the network to solve a specific type of problems;
- learning opportunities for individual elements of the artificial neural network;
- calculation of data for one epoch without the need to store previous calculations and this reduces the time for processing information by not having to access the database;
- absence of accumulation of learning errors of artificial neural networks as a result of processing information arriving at the input of artificial neural networks.

2. Approbation of the proposed method by the example of forecasting the state of the electronic environment is carried out. This example showed an increase in the efficiency of artificial neural networks at the level of 10–18 % of the efficiency of information processing through the use of additional training procedures for artificial neural networks.

Acknowledgments

The author’s team is grateful for the help to:

- doctor of technical sciences, professor Oleksiy Viktorovich Kuvshinov – deputy head of the educational and scientific institute of the National defense university of Ukraine named after Ivan Chernyakhovskiy;
- doctor of technical sciences, senior researcher Yuriy Vladimirovich Zhuravskiy – leading researcher of the research center of the Zhytomyr military institute named after S. P. Korolev;

– doctor of technical sciences, senior researcher Oleg Yaroslavovich Sova – head of the department of «Automated control systems» of the military institute of telecommunications and informatization named after the Heroes of Krut;

– candidate of technical sciences, associate professor Oleksandr Mykolayovych Bashkirov – leading researcher at the Central research institute of armament and military equipment of the Armed Forces of Ukraine.

References

1. Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskyi, A., Yarosh, S., Gatsenko, S. et. al. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. *Eastern-European Journal of Enterprise Technologies*, 5 (9 (95)), 60–76. doi: <https://doi.org/10.15587/1729-4061.2018.144085>
2. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The method of improving the efficiency of routes selection in networks of connection with the possibility of self-organization. *International Journal of Advanced Trends in Computer Science and Engineering*, 8 (1.2), 1–6, doi: <https://doi.org/10.30534/ijatcse/2019/0181.22019>
3. Zhang, J., Ding, W. (2017). Prediction of Air Pollutants Concentration Based on an Extreme Learning Machine: The Case of Hong Kong. *International Journal of Environmental Research and Public Health*, 14 (2), 114. doi: <https://doi.org/10.3390/ijerph14020114>
4. Katranzhy, L., Podskrebko, O., Krasko, V. (2018). Modelling the dynamics of the adequacy of bank's regulatory capital. *Baltic Journal of Economic Studies*, 4 (1), 188–194. doi: <https://doi.org/10.30525/2256-0742/2018-4-1-188-194>
5. Manea, E., Di Carlo, D., Depellegrin, D., Agardy, T., Gissi, E. (2019). Multidimensional assessment of supporting ecosystem services for marine spatial planning of the Adriatic Sea. *Ecological Indicators*, 101, 821–837. doi: <https://doi.org/10.1016/j.ecolind.2018.12.017>
6. Çavdar, A. B., Ferhatosmanoğlu, N. (2018). Airline customer lifetime value estimation using data analytics supported by social network information. *Journal of Air Transport Management*, 67, 19–33. doi: <https://doi.org/10.1016/j.jairtraman.2017.10.007>
7. Kachayeva, G. I., Mustafayev, A. G. (2018). The use of neural networks for the automatic analysis of electrocardiograms in diagnosis of cardiovascular diseases. *Herald of Dagestan State Technical University. Technical Sciences*, 45 (2), 114–124. doi: <https://doi.org/10.21822/2073-6185-2018-45-2-114-124>
8. Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. *Ice and Snow*, 56 (4), 502–510. doi: <https://doi.org/10.15356/2076-6734-2016-4-502-510>
9. Kanev, A., Nasteka, A., Bessonova, C., Nevmerzhitsky, D., Silaev, A., Efremov, A., Nikiforova, K. (2017). Anomaly detection in wireless sensor network of the «smart home» system. 2017 20th Conference of Open Innovations Association (FRUCT), 118–124. doi: <https://doi.org/10.23919/fruct.2017.8071301>
10. Sreeshakthy, M., Preethi, J. (2016). Classification of human emotion from deap EEG signal using hybrid improved neural networks with Cuckoo search. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 6 (3-4), 60–73. Available at: <https://www.slideshare.net/bpatrut/classification-of-human-emotion-from-deap-eeeg-signal-using-hybrid-improved-neural-networks-with-cuckoo-search>
11. Chica, J., Zaputt, S., Encalada, J., Salamea, C., Montalvo, M. (2019). Objective assessment of skin repigmentation using a multilayer perceptron. *Journal of Medical Signals & Sensors*, 9 (2), 88. doi: https://doi.org/10.4103/jmss.jmss_52_18
12. Massel, L. V., Gerget, O. M., Massel, A. G., Mamedov, T. G. (2019). The Use of Machine Learning in Situational Management in Relation to the Tasks of the Power Industry. *EPJ Web of Conferences*, 217, 01010. doi: <https://doi.org/10.1051/epjconf/201921701010>
13. Abaci, K., Yamacli, V. (2019). Hybrid Artificial Neural Network by Using Differential Search Algorithm for Solving Power Flow Problem. *Advances in Electrical and Computer Engineering*, 19 (4), 57–64. doi: <https://doi.org/10.4316/aecce.2019.04007>
14. Mishchuk, O. S., Vitynskyi, P. B. (2018). Neural Network with Combined Approximation of the Surface of the Response. *Research Bulletin of the National Technical University of Ukraine «Kyiv Politechnic Institute»*, 2, 18–24. doi: <https://doi.org/10.20535/1810-0546.2018.2.129022>
15. Kazemi, M., Faezirad, M. (2018). Efficiency estimation using nonlinear influences of time lags in DEA Using Artificial Neural Networks. *Industrial Management Journal*, 10 (1), 17–34. doi: <https://doi.org/10.22059/imj.2018.129192.1006898>
16. Parapuram, G., Mokhtari, M., Ben Hmida, J. (2018). An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. *Energies*, 11 (3), 680. doi: <https://doi.org/10.3390/en11030680>
17. Prokoptsev, N. G., Alekseenko, A. E., Kholodov, Y. A. (2018). Traffic flow speed prediction on transportation graph with convolutional neural networks. *Computer Research and Modeling*, 10 (3), 359–367. doi: <https://doi.org/10.20537/2076-7633-2018-10-3-359-367>
18. Bodyanskiy, Y., Pliss, I., Vynokurova, O. (2013). Flexible Neo-fuzzy Neuron and Neuro-fuzzy Network for Monitoring Time Series Properties. *Information Technology and Management Science*, 16 (1), 47–52. doi: <https://doi.org/10.2478/itms-2013-0007>
19. Bodyanskiy, Ye., Pliss, I., Vynokurova, O. (2013). Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks. *Oil and Gas Power Engineering*, 2 (20), 158–162. Available at: http://nbuv.gov.ua/UJRN/Nge_2013_2_18

20. Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 842.
21. Nelles, O. (2001). *Nonlinear System Identification*. Springer. doi: <https://doi.org/10.1007/978-3-662-04323-3>
22. Wang, L.-X., Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3 (5), 807–814. doi: <https://doi.org/10.1109/72.159070>
23. Kohonen, T. (1995). *Self-Organizing Maps*. Springer. doi: <https://doi.org/10.1007/978-3-642-97610-0>
24. Kasabov, N. (2003). *Evolving Connectionist Systems*. Springer. doi: <https://doi.org/10.1007/978-1-4471-3740-5>
25. Sugeno, M., Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28 (1), 15–33. doi: [https://doi.org/10.1016/0165-0114\(88\)90113-3](https://doi.org/10.1016/0165-0114(88)90113-3)
26. Ljung, L. (1999). *System Identification. Theory for the User*. PTR Prentice Hall, Upper Saddle River, 609. Available at: <https://www.twirpx.com/file/277211/>
27. Otto, P., Bodyanskiy, Y., Kolodyazhniy, V. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. *Integrated Computer-Aided Engineering*, 10 (4), 399–409. doi: <https://doi.org/10.3233/ica-2003-10409>
28. Narendra, K. S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1 (1), 4–27. doi: <https://doi.org/10.1109/72.80202>
29. Petruk, S., Zhyvotovskiy, R., Shyshatskiy, A. (2018). Mathematical Model of MIMO. 2018 International Scientific-Practical Conference Problems of Infocommunications. *Science and Technology (PIC S&T)*, 7–11. doi: <https://doi.org/10.1109/infocommst.2018.8632163>
30. Zhyvotovskiy, R., Shyshatskiy, A., Petruk, S. (2017). Structural-semantic model of communication channel. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. *Science and Technology (PIC S&T)*, 524–529. doi: <https://doi.org/10.1109/infocommst.2017.8246454>
31. Alieinykov, I., Thamer, K. A., Zhuravskiy, Y., Sova, O., Smirnova, N., Zhyvotovskiy, R. et. al. (2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. *Eastern-European Journal of Enterprise Technologies*, 6 (2 (102)), 16–27. doi: <https://doi.org/10.15587/1729-4061.2019.184394>
32. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskiy, R., Prokopenko, Y., Hurskiy, T. et. al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. *Eastern-European Journal of Enterprise Technologies*, 5 (9 (101)), 35–45. doi: <https://doi.org/10.15587/1729-4061.2019.180197>