

# Prediction of Data Transmission Route Congestion in Telecommunication Systems Based on a Modified Elman Neural Network

Eduard Bovda <sup>1</sup> and Yuriy Samokhvalov <sup>2</sup>

<sup>1</sup> *Military Institute of Telecommunications and Informatization named after Heroes of Kruty, Knyaziv Ostrozkyh Street 45/1, Kyiv, 01011, Ukraine*

<sup>2</sup> *Taras Shevchenko National University, Volodymyrska Street 64/13, Kyiv, 01601, Ukraine*

## Abstract

The article analyzes existing approaches and methods of forecasting abnormal situations in telecommunication systems. The importance of the problem of forecasting congestion of data transmission routes is shown, and it is proposed to use the Elman neural network for its solution. A modification of this network and a method of predicting congestion of data transmission routes in the telecommunications network, which is based on a modified Elman neural network, are given. This method allows to increase the accuracy and speed of forecasting the congestion of routes in the network by increasing the bandwidth of the network and reducing the complexity of calculations.

## Keywords <sup>1</sup>

Data transmission routes, forecasting, telecommunication network, neural network, Elman network, stochastic time efficiency.

## 1. Introduction

The basis of modern distributed systems are telecommunication networks, which are complex technical systems and usually operate in dynamic environments [1, 2]. At the same time, the management of such networks should ensure the solution of tasks that ensure data transmission with a given quality [2]. Given this, telecommunication network management systems should include a subsystem for predicting abnormal situations (overload of data transmission routes, errors, etc.), which will allow the network administrator to take timely preventive measures. Therefore, forecasting the state of telecommunications networks is an important task of network administration.

A lot of research has been devoted to predicting the states of complex technical systems. Among them, the following methods and approaches are most often used. Thus, in [3], a method for predicting computer network states based on biometric algorithms is considered. In [4, 5], the method of temporal extrapolation, in [6, 7, 8], the method of spatial extrapolation, in [9], the method of causal relationship and expert methods, and in [10], a method is proposed in which data on the behavior of an object whose features are related to time are presented as the results of observations at uniform time intervals and are represented by a time series. You can also use the method of paired comparisons, which is considered in the work [11]. In addition, recently, neural network-based approaches have been widely used to predict the states of telecommunication networks and have shown their effectiveness. Such approaches are discussed in [12-15]. Papers [12, 13, 14] consider neural networks that allow obtaining the desired results without human intervention with low computational costs, and [15, 16] consider hybrid neural networks that allow assessing and predicting the state of computer networks with high accuracy of classification of the current and predicted state of the computer network, and [17] considers the use of a probabilistic neural network to solve the problems of classifying and predicting the state of the network transport environment.

Based on the fact that the forecasting problem is a special case of the regression problem, the following types of neural networks can be used to solve it: multilayer perceptron, radial basis networks,

---

*Information Technology and Implementation (IT&I-2023), November 20-21, 2023, Kyiv, Ukraine*

EMAIL: edepig8305@ukr.net (A. 1); yu1953@ukr.net (A. 2)

ORCID: 0000-0002-8267-2120 (A. 1); 0000-0001-5123-1288 (A. 2)



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

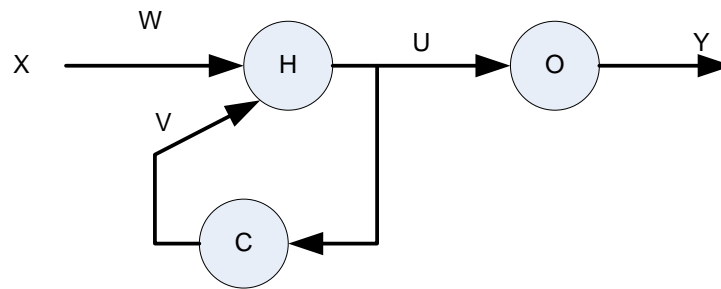
generalized regression networks, Volterra networks, and Elman networks. The analysis [18] of the use of such networks in solving forecasting problems indicates the expediency of using time series computation, which will be based on the Elman neural network.

At the same time, the direct use of this network increases the load on the telecommunications network as a whole, as well as the complexity of computing. This makes it impossible to predict its state in real time. Therefore, the question arises of creating a model that would solve this problem.

The article proposes a modification of the Elman network and a method for predicting the overload of telecommunication network data transmission routes based on it, which allows for effective management of a telecommunication network in conditions of high dynamics and complexity of connections between nodes.

## 2. Modified Elman neural network

An Elman neural network is a type of recurrent network. An Elman network consists of a multilayer perceptron with feedback. This function allows to take into account previous actions and accumulate information to support management decision-making based on time series forecasting. In other words, time series forecasting is reduced to the task of interpolation (determining intermediate values of a value) of a function of many variables and solving the problem of approximation (reduction to a simplified form) of a multidimensional function, which inherently affects the quality of forecasting. Figure 1 shows a diagram of the Elman neural network, which consists of three layers: the input (distribution) layer, the hidden layer, and the output (processing) layer. In this case, the hidden layer has a feedback on itself [18, 19].



**Figure 1:** Schematic of the Elman neural network

In this diagram, X is the input of the neural network; Y is the output of the neural network; C is the context state for the input X; W is the weight matrix of the input layer; V is the weight matrix of the hidden layer feedback; U is the weight matrix connecting the output of the hidden layer with the input of the output layer; H is the hidden layer of neurons, where each input X is connected to each neuron of the hidden layer; O is the output layer of neurons.

In the Elman network, the forecasting process is simulated by the output signal of some nonlinear dynamic system that depends on a number of factors, including past states of the system. Elman proposed to introduce an additional feedback layer into the network, called the contextual or state layer. This layer receives signals from the output of the hidden layer and, through the delay elements C, feeds them to the previous one, the input layer, thus preserving the processed information from previous cycles within the network [18].

Unlike a conventional feed-forward network, the input image of a recurrent network is not a single vector, but a sequence of input image vectors fed to the input in a given order, with the new state of the hidden layer depending on its previous states. Then the Elman network can be described by the following relations in matrix form:

$$Y_t = F(U \times F(W \times X_t + V \times C_t)) \quad (1)$$

$$C_t = F(W \times X_{t-1} + V \times C_{t-1}) \quad (2)$$

where  $X_t$  is the input signal;

$Y_t$  is the output of the neural network;

$C_t$  is the context state at iteration t for input X;

W is the weight matrix of the input layer;

V is the weight matrix of the hidden layer feedback;

$U$  is the weight matrix connecting the output of the hidden layer with the input of the output layer;

$X_{t-1}$  is the signal at the previous iteration;

$C_{t-1}$  is the state of the context at the previous iteration;

$F$  is the vector of the activation function;

$H$  is hidden layer of neurons, where each input  $X$  is connected to each neuron of the hidden layer;

$O$  is the output layer of neurons.

A telecommunication network can be considered as a set of its elements: information directions, routes, nodes, channels, and service quality characteristics. Therefore, as the input of the neural network, we will have a set of parameters of the network directions in the form of an input signal  $X_t = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ , where  $x_1$  is a type of traffic (voice, video, data) being transmitted;  $x_2$  is volume of service traffic between nodes;  $x_3$  is an information throughput capacity;  $x_4$  is delay of packets in the information direction;  $x_5$  is the value of jitter in the information sector;  $x_6$  is quality of routes between nodes;  $x_7$  is number of packets with errors (*IPER*);  $x_8$  is number of packets lost (*IPLR*). The input signal is formed as a result of monitoring the elements of the telecommunications network.

The output of the neural network  $Y_t$  is an output neuron (adder) that allows you to calculate the deviation of the values detected by the neurons from the value of the operating state of the telecommunications network routes.

### 3. Prediction of telecommunication network route overload using a modified Elman network

The essence of the forecasting is to determine the value of congestion of telecommunication network routes, which is calculated by the parameters of the information direction, in order to meet the requirements of network optimization and quality of service for packets of various types. The architecture of the modified Elman recurrent neural network is shown in Fig. 2.

Figure 2 shows a network with multiple inputs of the Elman network, where the number of neurons in the input layer  $m$  and a hidden layer  $n$  and  $q$  and one output block. Let  $x_{it}$  ( $i = 1, 2, \dots, m$ ) denote the set of input neuronal vectors at a given time  $t$ ,  $y_{t+1}$  indicates the network output at a given time  $t + 1$ ,  $u_{jt}$  ( $j = 1, 2, \dots, n$ ) denote the output of hidden layer neurons in time  $t$  and  $c_{jt}$  ( $j = 1, 2, \dots, n$ ) and  $s_{jt}$  ( $j = 1, 2, \dots, n$ ) indicate the neurons of the recurrent layer;  $w_{ij}$  is the weight that connects the node  $i$  in the input layer of neurons to the node  $j$  in hidden layer.  $v_j$  and  $q_j$  is weights that connect the unit  $j$  in hidden layer neurons with a node in the recurrent layer.

The hidden layer looks like this - the inputs of all neurons in the hidden layer are given by the network:

$$NET_{ji}(k) = \sum_{i=1}^m w_{ij}x_{it}(k-1) + \sum_{j=1}^n v_{ij}c_{it}(k) + \sum_{g=1}^l q_{ij}s_{it}(k) \quad (3)$$

where  $c_{ji}(k) = u_{jt}(k-1)$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ ;  $q_{ji}(k) = c_{jt}(k-1)$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$

The outputs of hidden neurons are obtained from the expression:

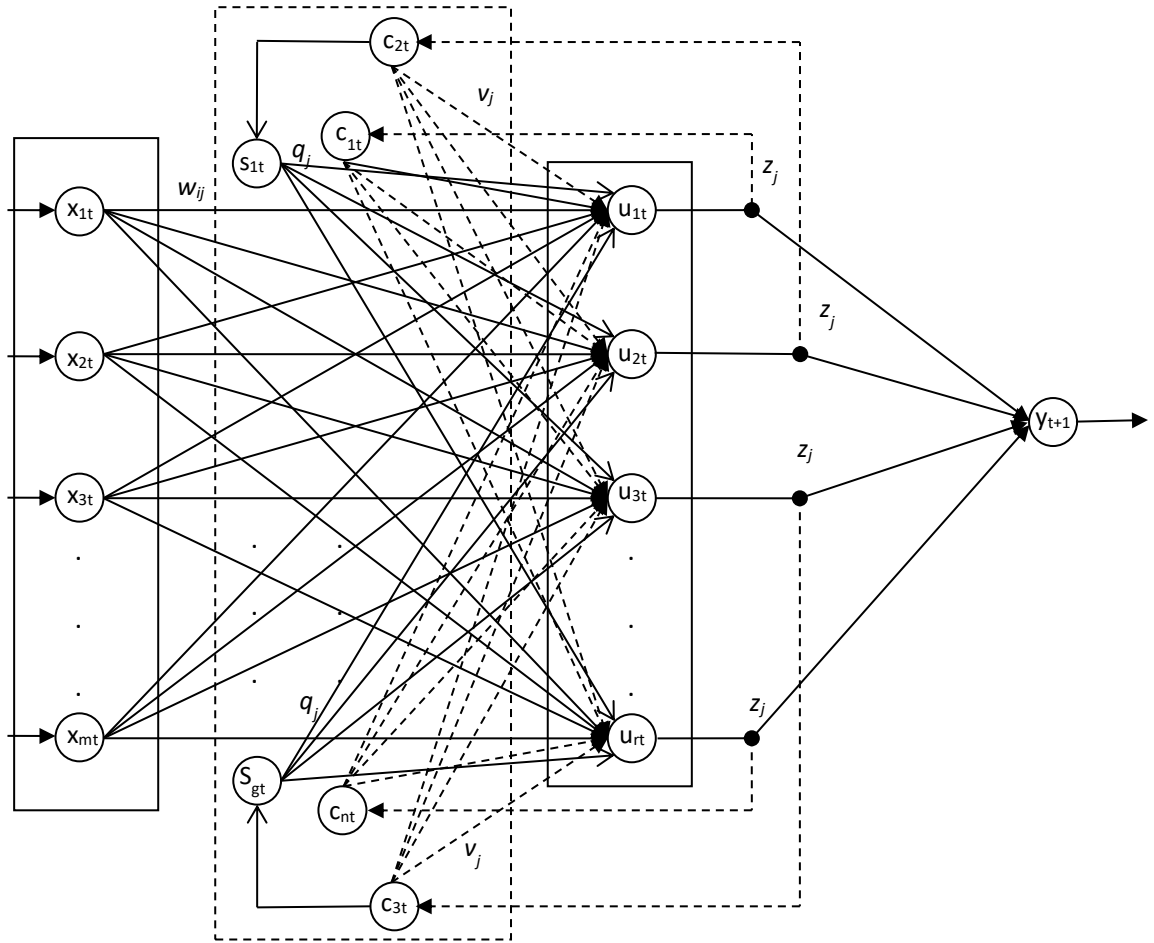
$$u_{ji}(k) = f_H \left( \sum_{i=1}^m w_{ij}x_{it}(k) + \sum_{j=1}^n v_{ij}c_{it}(k) + \sum_{g=1}^l q_{ij}s_{it}(k) \right) \quad (4)$$

where the sigmoidal function in the hidden layer is selected as the activation function:  $f_H(x) = 1/(1 + e^{-x})$ .

The output signal of the hidden layer is defined as follows:

$$y_{t+1}(k) = f_T \left( \sum_{j=1}^r z_j u_{jt}(k) \right) \quad (5)$$

where  $f_T(x)$  is mapping as a function of neuronal activation.



**Figure 2:** Elman network for predicting route congestion

#### 4. An algorithm for training an Elman network with stochastic time efficiency

Stochastic time efficiency is the minimization of time when training a neural network (in our case, Elman) based on error correction (learning with a teacher). The backpropagation algorithm is a supervised learning algorithm that minimizes the global error using the gradient descent method [20]. For the model of stochastic time efficiency of the Elman network, we assume that the resulting output error  $\varepsilon_{e_n} = d_{t_n} - y_{t_n}$  and sampling error  $n$  is defined as:

$$E(t_n) = 0,5\varphi(t_n)(d_{t_n} - y_{t_n})^2 \quad (6)$$

where  $t_n$  is response time (sampling)  $n$  ( $n = 1, 2, \dots, N$ ),  $d_{t_n}$  is an actual value,  $y_{t_n}$  is an output at a given time  $t_n$ , a  $\varphi(t_n)$  is the effective function of stochastic time. Let us define  $\varphi(t_n)$  in this way:

$$\varphi(t_n) = \frac{1}{\beta} \exp \left\{ \int_{t_0}^{t_n} \mu(t) dt + \int_{t_0}^{t_n} \sigma(t) dB(t) \right\} \quad (7)$$

the effective time function of the data is considered as a function of the time variable. The corresponding error of all the data in each network is then retrained and determined as:

$$E = \frac{1}{N} \sum_{n=1}^N E(t_n) = \frac{1}{2N} \sum_{n=1}^N \varphi(t_n) \cdot (d_{t_n} - y_{t_n})^2 \quad (8)$$

The main point of the learning algorithm is to minimize the value of the network route congestion function until it reaches the specified minimum value  $\xi$  by repeated training. At each repetition, the value of the network route congestion function is calculated and the global error is obtained. The gradient of the network route congestion function is defined as  $\Delta E = \partial E / \partial W$ . For nodes in the input layer, the weight gradient  $w_{ij}$  is given by the formula:

$$\Delta w_{ij} = -\eta \frac{\partial E(t_n)}{\partial w_{ij}} = \eta \varepsilon_{t_n} z_j \varphi(t_n) f'_H(NE T_{j t_n}) x_{i t_n} \quad (9)$$

for nodes in the recurrent layer, the weight gradient is set by the formulas:

$$\Delta v_j = -\eta \frac{\partial E(t_n)}{\partial v_{ij}} = \eta \varepsilon_{t_n} z_j \varphi(t_n) f'_H(NE T_{j t_n}) c_{i t_n}, \quad (10)$$

$$\Delta q_j = -\eta \frac{\partial E(t_n)}{\partial q_{ij}} = \eta \varepsilon_{t_n} q_j \varphi(t_n) f'_H(NE T_{j t_n}) q_{i t_n},$$

for weight nodes in the hidden layer - a weight gradient  $v_j$  is given by the formula:

$$\Delta z_j = -\eta \frac{\partial E(t_n)}{\partial z_j} = \eta \varepsilon_{t_n} z_j \varphi(t_n) f'_H(NE T_{j t_n}), \quad (11)$$

where  $\eta$  is learning speed  $f'_H(NE T_{j t_n})$ , is a derivative of the activation function.

Based on this update rule for scales  $w_{ij}$ ,  $v_j$ ,  $q_j$  and  $z_j$  are given by the formulas:

$$w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij}^k \quad (12)$$

$$v_j^{k+1} = v_j^k + \Delta v_j^k \quad (13)$$

$$q_j^{k+1} = q_j^k + \Delta q_j^k \quad (14)$$

$$z_j^{k+1} = z_j^k + \Delta z_j^k \quad (15)$$

The Elman neural network should change the weights to minimize the error between the network's prediction and the prediction target. Such a procedure can be effectively implemented using the methods of mathematical logic, in particular the method [21].

The Elman network training algorithm includes the following steps [20, 22, 23]:

*Step 1.* Normalize the input data. In the Elman neural network, we select 8 parameters as input values in the input layer. Then we define the network parameters, such as the learning rate  $\eta$ , which is between 0 and 1, the maximum number of iterations and the initial weights.

*Step 2.* First, the scales  $w_{ij}$ ,  $v_j$ ,  $q_j$  and  $z_j$  follow a uniform distribution on the interval (-1, 1).

*Step 3.* The stochastic efficiency over time is introduced by the function  $\varphi(t)$  to the error function  $E$ . Select the drift function  $\mu(t)$  and the function of a static indicator that characterizes the trend of changes in the network state  $\partial(t)$ .

*Step 4.* Set the minimum error  $\xi$  route congestion. Based on the network training goal, the value of the route congestion function is calculated:

$$E = (1/N) \sum_{n=1}^N E(t_n). \quad (16)$$

If the value  $E$  less than the specified minimum error, we go to step 5, if it is greater, we go to step 6.

*Step 5.* Change the connecting weights: calculate the gradient of the connecting weights  $w_{ij}, \Delta w_{ij}, v_j, \Delta v_j^k, q_j, \Delta q_j^k, z_j, \Delta z_j^k$ . Then the weights are changed from the current level to the previous layer  $w_{ij}^{k+1}, v_j^{k+1}, q_j^{k+1}, \Delta z_j^{k+1}$ .

When predicting the congestion of data transmission routes in a network, the problem of so-called "dead neurons" may arise. One of the limitations of any competing layer is that some neurons may not be involved. That is, the neurons with initial weight vectors are far removed from the input vectors and never win the competition, regardless of the training period. As a result, such vectors are not used in training and the corresponding neurons never win (are dead). Therefore, in order to enable such neurons to win, the learning algorithm provides for the possibility of a "winning neuron" losing its activity. For this purpose, neuronal activity is recorded based on the calculation of the potential of each neuron in the process of predicting the performance of data transmission routes and neuronal training.

First, the layer neurons are assigned a potential  $p_i(0) = \frac{1}{c}$ , where  $c$  is the number of neurons (clusters). Then:

- if the value of the potential  $p_i$  becomes less than the level  $p_{\min}$ , then the neuron is excluded from consideration;
- if  $p_{\min} = 0$ , then the neuron is considered;
- if  $p_{\min} = 1$ , the neurons win in turn, since in each cycle of searching for a "winning neuron" only one of them is ready to be considered for the possibility of defeating the others.

On the  $k$ th training cycle, the potential is calculated according to the rule::

$$p_i(k) = \begin{cases} p_i(k-1) + \frac{1}{c}, & i \neq j \\ p_i(k-1) - p_{\min}, & i = j \end{cases}, \quad (17)$$

where  $j$  is the number of the "winning neuron".

After providing equal opportunities for neurons to win and calculating the error, the neuron with the number  $k$  will be determined by the formula:

$$d_k = \min_j d_j. \quad (18)$$

The neurons of this layer are sets that are used according to the above rules (see formula 17). The output value of the layer will be the total potential of all "winning neurons" according to the network direction parameters based on the input values in the input layer.

*Step 6.* The value of the route congestion function is calculated:

$$y_{t+1} = f_T \left( \sum_{j=1}^m v_j f_H \left( \sum_{i=1}^m w_{ij} x_{it} + \sum_{j=1}^n c_j z_{jt} + \sum_{j=1}^g q_j s_{jt} \right) \right). \quad (19)$$

The learning process ends when this value is equal to the specified minimum value.

## 5. Conclusion

A method for predicting the overload of data transmission routes in a telecommunications network based on a modified Elman neural network is presented. The peculiarity of this method is to take into account the characteristics of the network by calculating the potential of the network neurons.

This makes it possible to increase the accuracy and speed of predicting route congestion in the network by increasing the network capacity and reducing the computational complexity of the neural network. The work of the Elman network algorithm with stochastic time efficiency is considered.

The proposed method for predicting the congestion of data transmission routes in a telecommunications network can also be used to predict other computer network states such as data throughput and delay.

## 6. References

- [1] The Law of Ukraine on Electronic Communications No. 1089-IX, 16.12.2020.
- [2] Bovda E.M. Conceptual bases of synthesis of an automated communication control system for military purposes / E.M. Bovda, V.A. Romaniuk, Y.A. Pluhovyi // Collection of scientific works of VITI. - 2016. - №1. - P. 6 - 18.

- [3] Stallings W. Network Security Essentials (2nd Edition). Prentice Hall, 2002. 432 p. Economic cybernetics: a textbook / [O. Chubukova, V. Ruban, L. Antoshkina, et al: YugoVostok, 2014. 454 p.
- [4] Kaufman C., Perlman R., Speciner M. Network Security: Private Communications in a Public World. Pearson Education, Limited, 2021. 752 p.
- [5] Aggarwal C. C. Neural Networks and Deep Learning. Cham : Springer International Publishing, 2023. URL: <https://doi.org/10.1007/978-3-031-29642-0> (date of access: 12.02.2024).
- [6] Anand, Adarsh and Ram, Mangey. Systems Performance Modeling, Berlin, Boston: De Gruyter, 2021. 181 p.
- [7] Additive Manufacturing, Design, Functionally Graded Additive Manufacturing. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959 : ASTM International, 2021. URL: <https://doi.org/10.1520/iso/astmtr52912-eb>.
- [8] Divitsky A.S., Borovyk L.V., Salnyk S.V., Gol V.D. Analysis of methods for predicting changes in data transmission routes in wireless self-organized networks. Collection of scientific papers of Kharkiv National Air Force University, 2020, 1(63)
- [9] Divitskyi A. Salnyk S. Hol V. & Storck A. Method of identification of data routes in wireless self-organized networks | Collection "Information Technology and Security". Information Technology and Security, 2021. URL: <https://doi.org/10.20535/2411-1031.2021.9.1.249839>.
- [10] Chen C. H. Handbook of Pattern Recognition and Computer Vision. 5th ed. University of Massachusetts Dartmouth, USA : WORLD SCIENTIFIC, 2020. 584 p.
- [11] Awan Z. K., Khan A., Iftikhar A. Hybrid Neural Networks: From Application Point of View. LAP Lambert Academic Publishing, 2012.
- [12] Chen Y., Kak S., Wang L. Hybrid neural network architecture for on-line learning // Intelligent Information Management. 2010. Vol. 2. P. 253-261.
- [13] Wan L., Zhu L., Fergus R. A Hybrid neural network-latent topic model // Proc. of the 15th Intern. Conf. on Artificial Intelligence and Statistics (AISTATS). La Palma, Canary Islands, 2012. Vol. 22. P. 1287-1294.
- [14] Zhong X., Enke D. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financial Innovation. 2019, vol. 5(1), pp. 1–20..
- [15] Y. Karpenko, Formation of the Enterprise Strategy based on the Industry Life Cycle / Independent Journal of Management & Production. 2021. Vol. 12, no. 3. P. s262–s280. URL: <https://doi.org/10.14807/ijmp.v12i3.1537>
- [16] Borah S., Panigrahi R. Applied Soft Computing: Techniques and Applications. Florida, United States : Apple Academic Press, 2022. 286 p.
- [17] Elman J.L. Finding Structure in Time // Cognitive science 14. -1990. - P. 179-211.
- [18] Lorentz C. SUPERVISED LEARNING TECHNIQUES. TIME SERIES FORECASTING. EXAMPLES with NEURAL NETWORKS and MATLAB. Independently Published, 2020. 277 p.
- [19] C.Strong, C.Barrett, C.Liu, T.Arnon, C.Lazarus, Algorithms for Verifying Deep Neural Networks / et al. Stanford University, USA: Foundations and Trends in Optimization, 2021. 404 p.
- [20] Samokhvalov, Y.Y. Problem-oriented theorem-proving method in fuzzy logic (po-method). Cybern Syst Anal 31, 682–690 (1995).
- [21] Rotstein A.P. Intelligent identification technologies. Rotstein A.P. - Vinnytsia, Universum-Vinnytsia, 1999. - 320 p.
- [22] M. Ghiasi, T. Niknam, Z. Wang, M. Dehghani, N. Ghadimi, M. Mehrandezh, ELECTRIC POWER SYSTEMS RESEARCH / University of Lisbon Higher Technical Institute, Lisboa, Portugal, 2023. URL: <https://doi.org/10.1016/j.eprs.2022.108975>