

## IMPLEMENTATION OF THE INDICATOR SYSTEM IN MODELING OF COMPLEX TECHNICAL SYSTEMS

**Leoshchenko S. D.** – Post-graduate student of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Subbotin S. A.** – Dr. Sc., Professor, Head of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Oliinyk A. O.** – PhD, Associate Professor, Associate Professor of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Narivs'kiy O. E.** – Dr. Sc., Professor, Technical Director, Ukrspecmash ltd., Berdyansk, Ukraine.

### ABSTRACT

**Context.** The problem of determining the optimal topology of a neuromodel, which is characterized by a high level of logical transparency in modeling complex technical systems, is considered. The object of research is the process of applying an indicator system to simplify and select the topology of neuromodels.

**Objective of the work** is to develop and use a system of indicators to determine the level of complexity of the modeling problem and gradually select the optimal logically transparent topology of the neuromodel.

**Method.** A method is proposed for selecting an optimal, logically transparent neural network topology for modeling complex technical systems using a system of corresponding indicators. At the beginning, the method determines the overall level of complexity of the modeling task and, using the obtained estimate, determines the method for further optimization of the neuromodel. Then, using Task data and input data characteristics, the method allows to obtain the most optimal structure of the neural model for further modeling of the system. The method reduces training time and increases the level of logical transparency of neuromodels, which significantly expands the practical use of such models, without using neuroevolution methods, which may not be justified by resource-intensive tasks.

**Results.** The developed method is implemented and investigated in solving the problem of modeling the dynamics of pitting processes of steel alloys. Using the developed method made it possible to reduce the training time of the model by 22%, depending on the computing resources used. The method also increased the level of logical transparency of the model by reducing the number of computing nodes by 50%, which also indicates faster and more efficient use of resources.

**Conclusions.** The conducted experiments confirmed the operability of the proposed mathematical support and allow us to recommend it for use in practice in the design of topologies of neuromodels for further solving modeling, diagnosis and evaluation problems. Prospects for further research may consist in the development of methods for structural optimization of previously synthesized models and the development of new methods for feature selection.

**KEYWORDS:** complexity assessment, indicator system, modeling, neuromodel, sampling, training, error, gradient.

### ABBREVIATIONS

A3C is asynchronous advantage actor-critic  
ANN is an artificial neural net;  
CAwP is complexity is associated with people;  
OC is organized complexity;  
OS is organized simplicity;  
RC is random complexity;  
RNN is recurrent neural network.

### NOMENCLATURE

$K_{input}$  is a number of element types in the neural network;

$K_{corrY}$  is a number of independent variables that strongly correlate with the original features;

$K_{imp}$  is a number of the most significant independent variables among factors;

$K_{ntcorrX}$  is a number of independent variables that are weakly dependent on others or do not correlate with each other;

$n$  is a number of input features that characterize sample instances;

$N_i$  is a multiple neurons at the network input;

$N_i$  is a neuron at the network input;

$N_o$  is a multiple neurons at the network output;

$N_{o_p}$  is a neuron at the network output;

$N_h$  is a multiple neurons of the hidden network layer;

$N_{h_r}$  is a hidden network layer neuron;

$Num_{elementype}$  is a number of element types in the neural network;

$NN$  is a neural network;

$NN_{struct}$  is a structure of neural network;

$l$  is a number of neurons at the network input;

$Lev_{accmeas}$  is a measurement accuracy level;

$Lev_{fctr}$  is a level of significant and less significant and/or non-significant factors;

$Lev_{manag}$  is a level of possible control and management;

$Lev_{task}$  is a conditional difficulty level of the task;

$Lev_{smpifctm}$  is a level of possible simplification of the structure;

$m$  is a number of dependent (categorical) features of sample instances;

$p$  is a number of neurons at the network output;

$q$  is a number of connections between neurons in the network;

$r$  is number of neurons in the hidden network layer;

$Sample$  is a data set;

$w$  is a multiple of connections between neurons;

$w_q$  is a connection between neurons in the network;

$x_n$  is a independent attribute of the sample instance;

$X$  is a set of independent attribute (variables);

$y_m$  is a value of the dependent variable (attribute) of the sample instance;

$Y$  is a set of values of dependent variables.

## INTRODUCTION

Usually, it can be concluded that ANN can developed hidden knowledge from data: forecasting, classification, pattern recognition skills, etc. are formed, but its logical structure usually remains hidden from the user [1–3]. The problem of manifestation (contrast) of this hidden logical structure is solved by reducing neural networks to a special logically transparent sparse form.

When designing an ANN the main questions usually arise [1]:

1) determine the set of neurons ( $N = \{N_i, N_o, N_h\}$ ), where  $N_i = \{N_{i_1}, N_{i_2}, \dots, N_{i_l}\}$

$N_o = \{N_{o_1}, N_{o_2}, \dots, N_{o_p}\}$   $N_h = \{N_{h_1}, N_{h_2}, \dots, N_{h_r}\}$

$l = 1, 2, \dots, |N_i|$ ,  $p = 1, 2, \dots, |N_o|$ ,  $r = 1, 2, \dots, |N_h|$ ) required to solve the problem;

2) determine the structure  $NN_{struct}$  (topology, number of layers, etc.) of the ANN.

Usually, after this or even at the testing stage, a third task/question arises: how to make the work of the ANN understandable to the user (logically transparent) and what benefits such an understanding can bring [4].

When solving the first question, the researcher is faced with two opposite points of view [5, 6]. One of them claims that the more neurons you use, the more reliable the network will be. Proponents of this position cite the example of the human brain. Indeed, the more neurons there are, the greater the number of connections between them, and the more complex tasks a neural network can solve. In addition, if you use a deliberately larger number of neurons than is necessary to solve the problem, then the neural network will definitely train. If you start with a small number of neurons, the network may not be able to train how to solve the problem, and the whole process will have to be repeated first with a large number of neurons. This point of view is popular among neural network software developers and is confirmed by a number of criteria [4–6].

The second point of view is based on an empirical rule: the more configurable parameters, the worse the approximation of the function in those areas where its values were pre-

viously unknown [4–6]. From a mathematical point of view, the tasks of training neural networks are reduced to continuing the function of a given finite number of points for the entire domain of definition. The second approach defines the required number of neurons as the minimum required. The main disadvantage is that this minimum required number is unknown in advance, and the procedure for determining it by gradually increasing the number of neurons is very time-consuming [5]. Based on the experience of various groups in the field of medical diagnostics, space navigation and psychology, it can be noted that all these tasks never required more than a few dozen neurons.

In the end, a network with a minimum number of neurons should better approximate the function, but determining this minimum number of neurons requires a lot of intellectual costs and experiments on network training. If the number of neurons is excessive, then you can get the result on the first attempt, but there is a risk of building a poor approximation. And as a specific methodology, you can use the approach: double the number of neurons in the network after each failed learning attempt.

An alternative approach is the use of neuroevolution methods [7–9]. When the ANN population is formed at the beginning. And then, in the course of using evolutionary operators (mutation, selection, crossover), the best solutions are obtained. In the form of big data with a large number of hidden relationships or incomplete data (or with the approach of teaching without a teacher), such methods and approaches show good results (high level of accuracy of work, choice of the optimal structure of the ANN). However, for not very complex tasks, such methods may not become justifiably resource-intensive, because they will work with populations of ANNs and constantly evaluate them [8]. However, there is a more reliable way to estimate the minimum number of neurons: using the contrast procedure [10]. In addition, the contrast procedure allow to answer the second question: what should be the network structure.

Therefore, using this example, we can conclude that at the beginning it is still necessary to more accurately assess the complexity of the problem in order to determine the strategy for finding the optimal structure of the neuromodel in the next one.

In this case, the term optimal structure of the ANN will be understood as the highest possible level of logical transparency of the ANN [1]. So, one of the main disadvantages of neural networks, from the point of view of many users, is that the ANN solves the problem, but cannot provide an explanation of exactly how the answer was received. In other words, you can't extract an algorithm for solving the problem from a trained network. In the case of using simple feed-forward ANN with a fixed and small number of neurons in the hidden layer, this can still be studied. However, for more complex topologies, we recommend using special methodologies, such as the contrast procedure.

That is why the scientific and applied task of developing new methods for assessing the complexity of the problem and then selecting the optimal structure of the ANN as a neuromodel in solving applied problems, such as modeling, classification and evaluation, is relevant.

**The object of study** is the process of using an indicator system to select the topology of a neuromodel.

Existing methodologies are general recommendations and do not have a solid mathematical basis for defining clear rules.

**The subject of study** is a system of criteria for determining the optimal structure of a neuromodel.

To date there are methods of neuroevolutionary synthesis of ANN, but they are quite resource-intensive, which may not be justified for simpler practical tasks. Therefore, the paper proposes a new approach based on presenting information about the task and evaluating input data.

**The purpose of the work** is to develop and use a system of indicators to determine the level of complexity of the modeling problem and gradually select the optimal logically transparent topology of the neuromodel.

## 1 PROBLEM STATEMENT

Let it be a problem which can be characterized by a conditional level of complexity ( $Lev_{task}$ ). The difficulty level is a specific integral estimate consisting of several characteristics:

– the sample  $Sample = \langle X, Y \rangle$  of input data, where  $X = \{x_1, x_2, \dots, x_n\}$  is the set of independent variables-features, and  $Y = \{y_1, y_2, \dots, y_m\}$  is the set of values of dependent variables, and-the number of input features that characterize sample instances;

–  $Lev_{smp\lfctn}$  is the level of possible simplification of the structure (for example, the amount of input data);

–  $Lev_{fctr}$  is a total number of significant and less significant and/or non-significant factors;

–  $Lev_{accmeas}$  is a level of measurement accuracy;

–  $Lev_{manag}$  is the level of possible control and management.

Based on this the problem can be present as follows: for the synthesis of ANN ( $NN$ ), it is necessary to define a set of neurons  $N = \{N_i, N_o, N_h\}$  consisting of subsets of input  $N_i = \{N_{i_1}, N_{i_2}, \dots, N_{i_l}\}, l = 1, 2, \dots, |N_i|$ , output  $N_o = \{N_{o_1}, N_{o_2}, \dots, N_{o_p}\}, p = 1, 2, \dots, |N_o|$ , hidden neurons  $N_h = \{N_{h_1}, N_{h_2}, \dots, N_{h_r}\}, r = 1, 2, \dots, |N_h|$  and a set of weights of connections between neurons  $w = \{w_q\}$ . Having determined the values of the elements of sets, we can consider the synthesis of ANN is complete.

To do this, it will be used an estimate of the complexity of the problem, after receiving which we will solve how we will design the ANN topology  $NN_{struct}$  in the future. If the problem can be classified as OS by its level of complexity, then using the feature selection, we will get the most informative ones and using the study of input data, it will be indicated the required number of neurons in the hidden layer of the network

$N_h = \{N_{h_1}, N_{h_2}, \dots, N_{h_r}\}, r = 1, 2, \dots, |N_h|$ . After that, it will be possible to perform neuromodel training.

## 2 REVIEW OF THE LITERATURE

Complexity is a characteristic that reflects the extent to which the project or implementation of a system or element is difficult to understand and verify (ISO/IEC/IEEE 24765).

A complex system is a system consisting of many interacting components (subsystems), as a result of which a complex system acquires new properties that are absent at the subsystem level and cannot be reduced to the properties of the subsystem level.

The complexity of a system is determined by the number of its constituent elements and possible connections between them. The degree of complexity is measured by the variety of the system. Diversity characterizes the number of possible states of the system.

According to the work of Peter M. Senge [11], system complexity exists in two main forms.

Complexity of granularity (structural complexity) occurs as a result of a large number of systems, system elements, and established relationships in any of the two main topologies (hierarchy or network). This complexity is related to the systems as they are; namely, their static existence.

Dynamic complexity (behavioral complexity) is related to the relationships that arise between ready-made, functioning systems in the course of their operation, that is, between expected and even unexpected behavior that actually occurs.

Weaver [12] formulated the initial point of view, identifying the following categories of complexity: organized simplicity, organized complexity, random complexity. These categories and later reflections, in particular Flood and Carson [13] and the author of the book, give grounds for using the following classification of complexity.

OS occurs when there are a small number of essential factors and a large number of less significant and/or non-essential factors. At first, the situation may seem complicated, but after studying it, less significant and insignificant factors can be excluded from the picture and hidden simplicity can be discovered [12, 13].

OC prevails in such physical and abstract systems, the structure of which is organized in such a way as to be understandable, and therefore pliable to scientists in describing complex behavior and structuring the process of creating complex systems and managing their life cycles. This is a wealth that should not be oversimplified [12, 13].

RC occurs when there are many variables that exhibit random, random behavior to a high degree. It can also be the result of a lack of necessary control over the structure of complex heterogeneous systems due to inadequate architecture management during the system lifecycle (creeping complexity) [12, 13].

CAwP occurs where the perception of any system causes a sense of complexity. In this context, people become systems of observation. This category also can be linked to systems in which people are elements and can

contribute significantly to organized simplicity, organized complexity, and disorderly complexity. Reasonable or unreasonable behavior of individuals in specific situations is naturally a significant factor in relation to complexity [12, 13].

### 3 MATERIALS AND METHODS

As already reported, it will be used the difficulty assessment of the task at the beginning for future work. So, based on the estimate, it will be determined which of the 4 types of Weaver [12, 13] classification the task belongs to. The main distribution is shown in Fig. 1. This is how the input data of the problem is studied at the beginning: the larger the sample size of the input data, the higher the score will be, because such tasks require either more complex operations for preprocessing input data (data reduction, feature selection, etc.), or more complex topologies of the ANN to encode all the features and track their relationship.

Next, it will be defined the level of possible simplification of the structure ( $Lev_{smp\l fctn}$ ). Usually it will be talked about the general possibility of deleting small information input data. So,  $Lev_{smp\l fctn} = -1$  if the number of input features is not large (by default, it is suggested up to 200, but it is possible to increase/decrease due to the technical resources used) and simplification is not possible,  $Lev_{smp\l fctn} = 0$  if the number is large (from 200 to 1000), but it is possible to reduce them, and  $Lev_{smp\l fctn} = 1$  if the number is large and the reduction is not possible (when working online) or too resource-intensive.

The study of the total number of significant and less significant and/or non-significant factors requires preliminary factor analysis [14–18]. In view of the simplification of this stage, it is proposed to perform the usual factor analysis using the principal component method. So,  $Lev_{fctr} = -1$  if the factor analysis did not show a large number of significant factors;  $Lev_{fctr} = 0$  if the number of significant factors is not large, but there are a certain number of less significant factors, or during the factor analysis there were correlation matrices that were poorly determined (their determinant was 0); and  $Lev_{fctr} = 1$  if the factor analysis showed a large number of significant factors and/or a large number of less significant ones.

During determining the level of measurement accuracy it will be guided by two main cases:

–  $Lev_{accmeas} = 1$  if the measurements are related to the human factor, then people (or human sensory systems) become observation systems;

–  $Lev_{accmeas} = 0$  if measurements are made using special sensors (sensor systems).

The level of possible control and management can also show only two states, so:

–  $Lev_{manag} = 1$  if the system exhibits highly random, disorderly behavior, or this behavior is the result of a lack of necessary control over the structure of complex heterogeneous systems;

–  $Lev_{manag} = 0$  if the system was under control (human or automated) and even if certain signs (e.g. deformities) were under control.

It should be noted that this paper will provide mechanisms exclusively for the first category: OS [12, 13]. After all, it should be noted that if a system, for example, after evaluating a problem, its complexity is determined as an OC, then at the beginning it is recommended to synthesize a neuromodel (for example, using neural network methods), and only then carry out structural optimization, if necessary. This is explained by the fact that in this case it is not possible to select information features according to the problem conditions, but it is possible to simplify and exclude links in the model. If the score is defined as CAwP, then it is recommended to check the truth of the input data before selecting information features. In the form of RC, it is recommended to use a special class of neural network methods for the synthesis of neuromodels, namely swarm intelligence methods, or use the RNN topology with the A3C method to train it [19, 20]. This approach will allow for much better data processing, because swarm intelligence methods do not have the disadvantages of most neuroevolutionary methods and emulate the behavior of independent agents [7–9], and RNN allows you to encode most hidden relationships in its structure.

Therefore, if the task was evaluated by the OS, we will proceed to the second stage. First, you need to determine the number of element types in the neural network. In our case  $Num_{elemtype} = 3$ , by default, because it is assumed that only input, hidden and output neurons are used. Next, you need to determine the number of independent variables that strongly correlate with the original features  $K_{corrY}$ . Additionally, the number of independent variables that are weakly dependent on others or do not correlate with each other is determined  $K_{ntcorrX}$ . Additionally, information about the number of the most significant independent variables among the factors is pulled up from the previous stage  $K_{imp}$ . After that, the number of neurons in the hidden layer can be determined by the formula:

$$|N_h| = K_{input} - K_{corrY} - K_{imp} - K_{ntcorrX} \quad (1)$$

### 4 EXPERIMENTS

To research the developed method, it will be used the problem of modeling the dynamics of pitting processes of steel alloys [16, 17]. Pitting (spot) corrosion is a type of corrosion destruction to which only passive metals and alloys are subjected. Pitting corrosion is observed in nickel, zirconium, chromium – nickel, chromium, aluminum alloys, etc. during pitting corrosion, only certain areas of the surface are destroyed, on which deep lesions are formed-pitting (point ulcers). AISI 304 steel is used in

many areas, for example, in the manufacture of prefabricated and welded metal structures, components of pipe fittings, as well as household equipment: balcony and staircase fences, kitchen equipment, automobile exhaust systems [21, 22]. Such structural materials have high corrosion resistance in many aggressive environments, but can be subjected to pitting corrosion in solutions containing chloride ( $C_{Cl}$ ). That is why it is an urgent task to build models of the dependence of pitting formation on the characteristics of the steel used and reversible solutions. A promising basis for such models is the ANN, due to their ability to learn from experimental data, generalize data, and extract knowledge from data [1–3]. ANNs can be used to build models of dependencies based on observations in the absence of uncertainty of dependencies in the data. Table 1 shows a fragment of the input data.

The data sample consisted of 50 point observations characterized by the values of input features:  $x_1$  is  $pH$  (medium characteristic, dimensionless);  $x_2$  is  $C_{Cl}$ , mg/l (chloride concentration in the medium);  $x_3$  is  $V_{ox}$ , Vol. % (volume of oxides in steel);  $x_4$  is the amount of oxides up to 1.98 microns in size per 100 visual obstacles of an optical microscope ( $\times 320$ ) in steel;  $x_5$  is the amount of oxides ranging in size from 1.98 microns to 3.95 per 100 visual interference of an optical microscope ( $\times 320$ ) in steel;  $x_6$  is  $L_{ox}$ , microns (average distance between oxides in steel);  $x_7$  is  $d_3$ , microns (average diameter of austenite grain in steel);  $x_8$  is  $P_{\alpha}$ , Vol. % (volume-ferrite in steel);  $x_9$  is  $C$  the mass percentages (carbon content in Steel);  $x_{10}$  is  $Mn$ , % (manganese content in steel);  $x_{11}$  is  $Si$ , % (silicon content in Steel);  $x_{12}$  is  $Cr$ , % (chromium

content in steel);  $x_{13}$  is  $Ni$ , % (nickel content in steel);  $x_{14}$  is  $N$ , % (nitrogen content in steel);  $x_{15}$  –  $Ti$ , % (titanium content in Steel);  $x_{16}$  is  $S$ , % (sulfur content in steel);  $x_{17}$  is  $P$ , % (phosphorus content in steel);  $x_{18}$  is  $X_o$ ,  $m^3/kg$  (specific magnetic susceptibility of austenite).

Dependent variables:  $y_1$  is  $\Delta Cr$ , mg ( $10^{-5}$ ) the chromium losses from steel after its exposure in solution;  $y_2$  is  $\Delta Fe$ , mg ( $10^{-5}$ ) the iron losses from steel after its exposure in solution;  $y_3$  is  $\Delta Ni$ , mg ( $10^{-5}$ ) the nickel losses from steel after its exposure in solution;  $y_4 = ZCr$ ;  $y_5$  if  $ZCr < 1$ , then steel in solution with such parameters (signs  $x_1, x_2$ ) is subjected to pitting corrosion with the formation of stable pitting, otherwise, steel in solution with such parameters (signs  $x_1, x_2$ ) is subjected to pitting corrosion with the formation of metastable pitting;  $y_6 = ZNi$ ;  $y_7$  if  $ZNi < 1$ , then pittings grow intensively, otherwise, pittings do not increase in size intensively.

## 5 RESULTS

The results of factor analysis are demonstrated at the table 2. This results will be used for evaluate the level of task and for calculations of number hidden neurons.

Pairwise correlation coefficients of input features are demonstrated at table 3 and will be used in future calculations of number hidden neurons.

Pairwise correlation coefficients of input and output features are demonstrated at table 4 and will be used in future calculations of number hidden neurons.

Table 5 demonstrate the comparison of neuromodel learning outcomes.

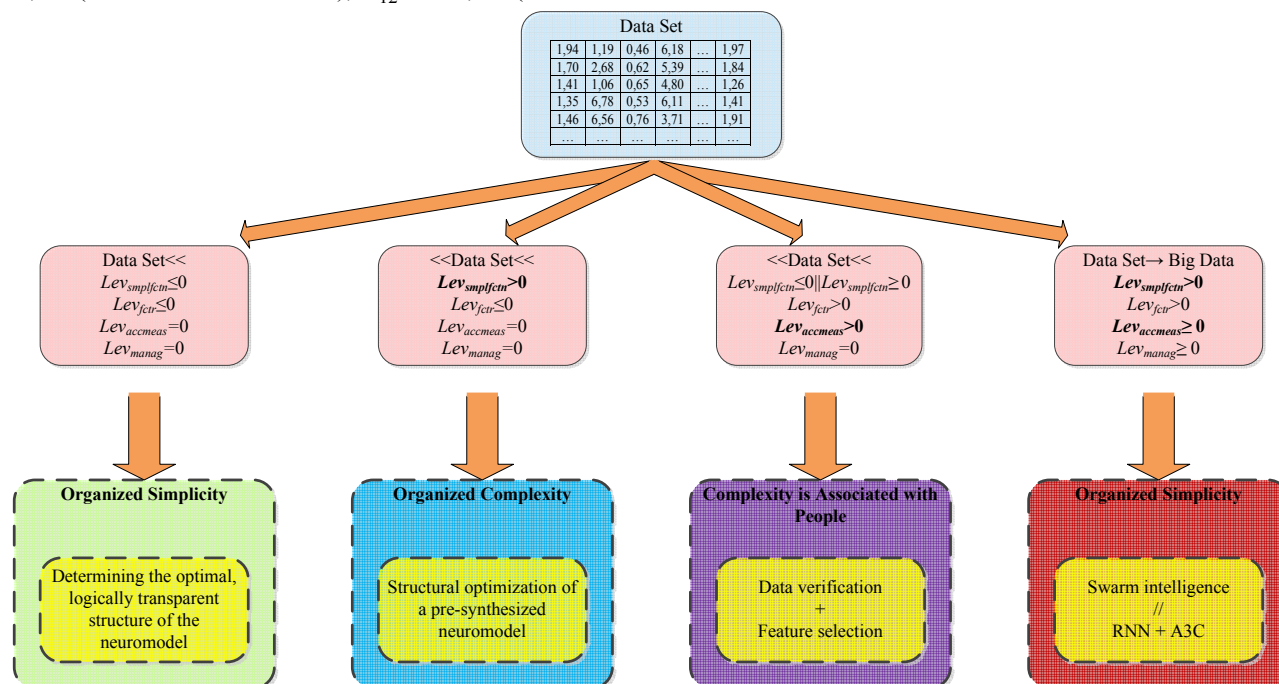


Figure 1 – General scheme for determining the level of difficulty

Table 1 – General information about data set

$x_1$	$x_2$	$x_3$	...	$x_{18}$	$y_1$	$y_2$	$y_3$	...	$y_7$
4	300	0,0199	...	2,27	0,01421	0,18776	0,01620	...	0
4	300	0,0216	...	2,27	0,01109	0,63335	0,01337	...	0
5	300	0,0199	...	2,27	0,00275	0,00296	0,01479	...	1
5	300	0,0216	...	2,27	0,00379	0,00275	0,01400	...	1
...	...	...	...	...	...	...	...	...	...
8	600	0,0216	...	2,27	0,00412	0,02114	0,03708	...	1

Table 2 – Results of factor analysis

$X$	Factor 1	Factor №2	Factor №3
$x_1$	-0.000000	-0.000000	-0.894181
$x_2$	0.000000	0.000000	-0.447706
$x_3$	-0.989883	0.031646	-0.000000
$x_4$	0.677879	-0.644594	0.000000
$x_5$	-0.980149	0.158833	0.000000
$x_6$	0.817799	0.493587	-0.000000
$x_7$	0.837964	-0.210144	-0.000000
$x_9$	0.533363	0.743210	0.000000

Table 3 – Pairwise correlation coefficients of input features

$\Gamma$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{16}$	$x_{17}$	$x_{18}$
$x_1$	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$x_2$	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$x_3$	0.00	0.00	1.00	-0.68	0.99	-0.83	-0.80	0.94	-0.46	0.15	-0.06	0.32	-0.13	-0.49	-0.16	0.25	0.89
$x_4$	0.00	0.00	-0.68	1.00	-0.78	0.18	0.59	-0.79	0.00	-0.39	-0.49	-0.86	0.58	-0.09	-0.40	0.31	-0.62
$x_5$	0.00	0.00	0.99	-0.78	1.00	-0.75	-0.80	0.96	-0.39	0.20	0.04	0.44	-0.23	-0.40	-0.05	0.15	0.89
$x_6$	0.00	0.00	-0.83	0.18	-0.75	1.00	0.54	-0.71	0.69	-0.02	0.32	0.17	-0.16	0.82	0.61	-0.65	-0.73
$x_7$	0.00	0.00	-0.80	0.59	-0.80	0.54	1.00	-0.61	0.28	-0.08	0.18	-0.31	0.13	0.13	-0.34	-0.03	-0.60
$x_8$	0.00	0.00	0.94	-0.79	0.96	-0.71	-0.61	1.00	-0.30	0.16	0.11	0.44	-0.20	-0.40	-0.19	0.10	0.93
$x_9$	0.00	0.00	-0.46	0.00	-0.39	0.69	0.28	-0.30	1.00	-0.61	-0.18	-0.03	0.35	0.92	0.48	-0.94	-0.11
$x_{10}$	0.00	0.00	0.15	-0.39	0.20	-0.02	-0.08	0.16	-0.61	1.00	0.87	0.70	-0.95	-0.38	0.12	0.42	-0.20
$x_{11}$	0.00	0.00	-0.06	-0.49	0.04	0.32	0.18	0.11	-0.18	0.87	1.00	0.83	-0.93	-0.01	0.26	0.01	-0.23
$x_{12}$	0.00	0.00	0.32	-0.86	0.44	0.17	-0.31	0.44	-0.03	0.70	0.83	1.00	-0.88	0.17	0.54	-0.26	0.17
$x_{13}$	0.00	0.00	-0.13	0.58	-0.23	-0.16	0.13	-0.20	0.35	-0.95	-0.93	-0.88	1.00	0.09	-0.38	-0.11	0.16
$x_{14}$	0.00	0.00	-0.49	-0.09	-0.40	0.82	0.13	-0.40	0.92	-0.38	-0.01	0.17	0.09	1.00	0.77	-0.94	-0.28
$x_{16}$	0.00	0.00	-0.16	-0.40	-0.05	0.61	-0.34	-0.19	0.48	0.12	0.26	0.54	-0.38	0.77	1.00	-0.68	-0.24
$x_{17}$	0.00	0.00	0.25	0.31	0.15	-0.65	-0.03	0.10	-0.94	0.42	0.01	-0.26	-0.11	-0.94	-0.68	1.00	-0.02
$x_{18}$	0.00	0.00	0.89	-0.62	0.89	-0.73	-0.60	0.93	-0.11	-0.20	-0.23	0.17	0.16	-0.28	-0.24	-0.02	1.00

Table 4 – Pairwise correlation coefficients of input and output features

$\Gamma$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$
$x_1$	-0.0240	-0.3045	0.0292	0.1242	0.1260	0.1317	0.1132
$x_2$	-0.3102	-0.2726	0.1651	-0.2519	-0.3563	-0.2464	-0.1601
$x_3$	0.1198	0.0172	0.2271	0.2153	0.0417	-0.0259	0.1020
$x_4$	0.0757	0.0193	-0.0486	-0.1540	0.0086	0.1167	-0.0194
$x_5$	0.0862	0.0104	0.2032	0.2145	0.0337	-0.0461	0.0909
$x_6$	-0.2262	-0.0467	-0.2617	-0.1657	-0.0708	-0.0532	-0.1225
$x_7$	-0.1016	-0.0253	-0.1831	-0.2068	0.0060	-0.0075	-0.0460
$x_8$	0.0404	-0.0113	0.1788	0.1881	0.0445	-0.0828	0.1013
$x_9$	-0.3200	-0.1464	-0.1721	-0.0998	-0.0484	-0.1651	-0.0282
$x_{10}$	0.1397	0.1280	-0.0134	0.0441	-0.0052	0.0852	-0.0676
$x_{11}$	-0.0254	0.0618	-0.1145	-0.0126	-0.0192	-0.0108	-0.0820
$x_{12}$	-0.0717	0.0230	-0.0507	0.0833	-0.0306	-0.0713	-0.0511
$x_{13}$	-0.0442	-0.0918	0.0562	-0.0462	0.0249	-0.0267	0.0806
$x_{14}$	-0.2920	-0.1089	-0.1929	-0.0861	-0.0731	-0.1287	-0.0718
$x_{16}$	-0.1545	-0.0229	-0.1222	0.0133	-0.0840	-0.0535	-0.0961
$x_{17}$	0.3191	0.1352	0.1549	0.0456	0.0576	0.1793	0.0345
$x_{18}$	0.0025	-0.0547	0.1910	0.1750	0.0472	-0.1046	0.1274

Table 5 – Comparison of neuromodel learning outcomes

Target variable	Number of hidden neurons	Training Time, s	Average error	Independent variables used	Number of hidden neurons	Training Time, s	Average error
$y_1$	8	4.2400	0.00336	$x_2, x_9$	4	3.2190	0.00165
$y_2$	8	5.6440	0.00686	$x_1, x_2, x_9, x_{18}$	4	4.3480	0.00639
$y_3$	8	4.3280	0.00641	$x_1, x_2, x_8, x_9, x_{18}$	4	3.3090	0.00181
$y_4$	8	5.5470	0.00344	$x_1, x_2, x_9, x_{18}$	4	4.9320	0.16357
$y_5$	8	4.4030	0.00316	$x_1, x_2, x_9, x_{18}$	4	3.3320	0.00191
$y_6$	8	7.8150	0.72003	$x_1, x_2, x_9, x_{18}$	4	4.9700	0.84592
$y_7$	8	4.4480	0.00501	$x_1, x_2, x_9, x_{18}$	4	4.3400	0.00357

## 6 DISCUSSION

Let's analyze the input data and sample it. So,  $Lev_{smp/fctn} = 0$  because the number of input attributes is not large and it is possible to reduce them.

In our case  $Lev_{fctr} = 0$ , due to the fact that during the factor analysis there were correlation matrices that were poorly determined (their determinant was 0). Factor loads are shown in table 2.

Under the conditions of the problem, measurements were made using special sensors, so  $Lev_{acmeas} = 0$ . And  $Lev_{manag} = 0$  because the experimental system was always under supervision and control.

In table 3 shows the coefficients of paired correlation of input features with each other. It should be noted that  $x_{15}$  does not correlate at all.

From the results of table 3, we can conclude that the signs  $x_1, x_2$  and  $x_{15}$ , ta do not depend on other signs, and all other signs are strongly related.

Table 4 shows the pairwise correlation coefficients between input and output features. It should be noted that  $x_{15}$  does not correlate at all.

From the results of table 4, it can be concluded that the most individually significant for the initial features are the following:  $x_1, x_2$  and  $x_9$ .

So we get the coefficients  $K_{corrY} = 3, K_{ntcorrX} = 3, K_{imp} = 8$ . The result is the most optimal number of hidden neurons  $|N_h| = 4$ .

It will be tested and compared the average error values for the complete data and for the data after feature selection and with the optimal number of hidden neurons in the hidden layer defined by (1). The comparison results are shown in Table 5. The Forward-Backward greedy algorithm was used to select features [23].

Based on the results of the initial analysis, the task was assigned to the OS group. After all, the input sample was not excessive, and the risks of human influence were excluded. The only significant complicating factor is the poorly conditioned correlation matrices (their determinant was 0), which was confirmed by factor analysis.

Therefore, at the beginning, neuromodel training was tested using all input data, and since the results of factor analysis revealed 8 most significant variables: for Factor 1

it is  $x_3 - x_7$ ; for Factor 2 it is  $x_4, x_9$ ; for Factor 3 it is  $x_1$ , then at the beginning  $|N_h| = 8$ .

Analyzing the initial results, it should be noted that there is a fairly large difference between the training time of the model: from 4.24 to 7.82. at the same time, the size of the average error was quite acceptable and ranged from 0.00316 to 0.72003.

The results obtained on the data after reduction showed that the accuracy in some cases (for  $y_1, y_2, y_3, y_5$  and  $y_7$ ) improved, and the time was significantly reduced, moreover, the time distribution between training iterations was distributed more evenly. However, for a number of cases ( $y_4, y_6$ ), the accuracy of the model has decreased. This may be due to a slight influence of the eliminated features, which may be of a group nature. Therefore, we can conclude that to feature selection in this case, we should use methods that can select not only individual features, but also groups of features [24, 25].

## CONCLUSIONS

The urgent scientific and applied problem of determining the optimal and logically transparent structure of a neuromodel is solved.

**The scientific novelty** lies in the fact that a method has been developed for selecting the optimal, logically transparent topology of a neural network for modeling complex technical systems using a system of corresponding indicators. Using the problem complexity assessment and a system of indicators, the method allows you to determine the most optimal and logically transparent structure of the neural network.

**The practical significance** lies in the fact that the developed method allows you to use clear mechanisms to increase the level of logical transparency of the model based on a comprehensive assessment of the problem. When testing the method, the learning time of neuromodels was accelerated by an average of 22%. Therefore, the results of experiments indicate a real simplification of the structure and more rational use of computing resources.

**Prospects for further research** are the development of methods for structural optimization of pre-synthesized neuromodels in the case of complexity of the OC type problem and the use of feature selection methods that would allow taking into account group information content.

## ACKNOWLEDGEMENTS

The work was carried out with the support of the state budget research projects of the state budget of the National University “Zaporozhzhia Polytechnic” “Intelligent methods and software for diagnostics and non-destructive quality control of military and civilian applications” (state registration number 0119U100360) and “Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects”.

## REFERENCES

1. Goldberg Y. Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies). California, Morgan & Claypool Publishers, 2017, 310 p.
2. Aggarwal C.C. Neural Networks and Deep Learning: A Textbook. Berlin, Springer, 2018, 520 p.
3. Valigi N. Zero to AI, A non-technical, hype-free guide to prospering in the AI. New York, Manning Publications, 2020, 264 p.
4. Artasanchez A. Artificial Intelligence with Python, Your complete guide to building intelligent apps using Python 3.x. Birmingham, Packt Publishing, 2020, 618 p.
5. Oliinyk A., Subbotin S., Leoshchenko S., Ilyashenko M., Myronova N., Mastinovskiy Y. Additional training of neuro-fuzzy diagnostic models, *Radio Electronics, Computer Science, Control*, 2018, No. 3, pp. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
6. Leoshchenko S., Oliinyk A., Subbotin S., Zaiko T. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems, *2018 International Scientific-Practical Conference Problems of Informatics. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018, proceedings*. Kharkiv, IEEE, 2018, pp. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
7. Iba H. Evolutionary Approach to Machine Learning and Deep Neural Networks, Neuro-Evolution and Gene Regulatory Networks. New York, Springer, 2018, 258 p.
8. Omelianenko I. Hands-On Neuroevolution with Python, Build high-performing artificial neural network architectures using neuroevolution-based algorithms. Birmingham, Packt Publishing, 2019, 368 p.
9. Blokdyk G. Neuroevolution of augmenting topologies, Second Edition. Ohio, SSTARCOoks, 2018, 128 p.
10. How is Contrast Encoded in Deep Neural Networks? [Electronic resource]. Access mode, <https://arxiv.org/abs/1809.01438>
11. Senge P. M. The Fifth Discipline, The Art & Practice of The Learning Organization. New York, Doubleday, 2006, 445 p.
12. Weaver W. Science and complexity, *American Scientist*, 1948, Vol. 36, No. 4, pp. 536–544.
13. Flood R.L. Dealing with Complexity. Berlin, Springer, 1993, 296 p.
14. Spiegelhalter D. The Art of Statistics, How to Learn from Data. New York, Basic Books, 2019, 448 p.
15. Bruce P. Practical Statistics for Data Scientists, 50 Essential Concepts, 1st Edition. California, O’Reilly Media, 2017, 318 p.
16. Finch W. H. Exploratory Factor Analysis [Text], 1st Edition. California, SAGE Publications, 2019, 144 p.
17. Rencher A. C. Methods of Multivariate Analysis [Text], 3rd Edition. New Jersey, John Wiley & Sons, 2012, 800 p.
18. Dean A. Design and Analysis of Experiments (Springer Texts in Statistics), 2nd Edition. Berlin, Springer, 2017, 865 p.
19. Sewak M. Deep Reinforcement Learning, Frontiers of Artificial Intelligence / M. Sewak. Berlin, Springer, 2020, 220 p.
20. Calix R. A Deep Learning Algorithms, Transformers, gans, encoders, cnns, rnns, and more. Traverse City, Independently published, 2020, 428 p.
21. Narivs’kyi O.E. Corrosion fracture of platelike heat exchangers, *Materials Science*, 2005, Vol. 41, No. 1, pp. 122–128.
22. Narivs’kyi O.E. Micromechanism of corrosion fracture of the plates of heat exchangers, *Materials Science*, 2007, Vol. 43, No. 1, pp. 124–132.
23. Kuhn M. Feature Engineering and Selection, A Practical Approach for Predictive Models (Chapman & Hall/CRC Data Science Series). London, Chapman and Hall (CRC Press), 2019, 310 p.
24. Oliinyk A., Subbotin S., Lovkin V., Ilyashenko M., Blagodariov O. Parallel method of big data reduction based on stochastic programming approach, *Radio Electronics, Computer Science, Control*, 2018, № 2, pp. 60–72.
25. Oliinyk A., Leoshchenko S., Lovkin V., Subbotin S., Zaiko T. Parallel data reduction method for complex technical objects and processes, *Dependable Systems, Services and Technologies, (DESSERT’2018)*, *The 9th IEEE International Conference, Kyiv, 24–27 May, 2018 : proceedings*. Los Alamitos, IEEE, 2018, pp. 528–531.

Received 12.12.2020.  
Accepted 18.02.2021.

УДК 004.896

## ЗАСТОСУВАННЯ ІНДИКАТОРНОЇ СИСТЕМИ ПРИ МОДЕЛЮВАННІ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

**Леощенко С. Д.** – аспірант кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя Україна.

**Субботін С. О.** – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

**Олійник А. О.** – канд. техн. наук, доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

**Нарівський О. Е.** – д-р техн. наук, професор, технічний директор, ТОВ «Укрспецмаш», Бердянськ, Україна.



#### АНОТАЦІЯ

**Актуальність.** Розглянуто задачу визначення оптимальної топології нейромоделі, що відрізняється високим рівнем логічної прозорості при моделюванні складних технічних систем. Об'єктом дослідження є процес застосування індикаторної системи для спрощення та вибору топології нейромоделі.

**Мета роботи** полягає у розробці та використанні системи індикаторів для визначення рівню складності задачі моделювання та поступовому підборі оптимальної логічно прозорої топології нейромоделі.

**Метод.** Запропоновано метод для підбору оптимальної, логічно прозорої топології нейронної мережі для моделювання складних технічних систем з використанням системи відповідних індикаторів. На початку метод визначає загальний рівень складності задачі моделювання та використовуючи отриману оцінку визначає спосіб подальшої оптимізації нейромоделі. Потім використовуючи дані про задачу та характеристики вхідних даних метод дозволяє отримати найбільш оптимальну структуру нейронної моделі для подальшого моделювання системи. Метод дозволяє скоротити час навчання та підвищити рівень логічної прозорості нейромоделі, що значно розширює практичне використання таких моделей, без використання нейроevolюційних методів, що можуть бути не виправдано ресурсоемними при ряді задач.

**Результати.** Розроблений метод реалізовано та досліджено при вирішенні задачі моделювання динаміки піттінгових процесів сталних сплавів. Використання розробленого методу дозволило скоротити час навчання моделі на 22%, в залежності від використовуваних обчислювальних ресурсів. Також метод дозволило підвищити рівень логічної прозорості моделі скоротивши кількість обчислювальних вузлів на 50%, що також свідчить про прискорення та більш раціональне використання ресурсів.

**Висновки.** Проведені експерименти підтвердили працездатність запропонованого математичного забезпечення і дозволяють рекомендувати його для використання на практиці при проектуванні топологій нейромоделей для подальшого вирішення задач моделювання, діагностування та оцінювання. Перспективи подальших досліджень можуть полягати в розробці методів структурної оптимізації попередньо синтезованих моделей та розробці нових методів відбору інформаційних ознак.

**КЛЮЧОВІ СЛОВА:** оцінка складності, система індикаторів, моделювання, нейромоделі, вибірка, навчання, помилка, градієнт.

УДК 004.896

#### СИНТЕЗ И ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ С ВЕРОЯТНОСТНЫМ КОДИРОВАНИЕМ СТРУКТУРЫ

**Леощенко С. Д.** – аспирант кафедры программных средств Национального университета «Запорожская политехника», Запорожье Украина.

**Субботин С. А.** – д-р. техн. наук, профессор, заведующий кафедрой программных средств Национального университета «Запорожская политехника», Запорожье Украина.

**Олейник А. А.** – канд. техн. наук, доцент, доцент кафедры программных средств Национального университета «Запорожская Политехника», Запорожье Украина.

**Наривский А. Э.** – д-р техн. наук, профессор, технический директор ООО «Укрспецмаш», Бердянсь, Украина.

#### АННОТАЦИЯ

**Актуальность.** Рассмотрена задача определения оптимальной топологии нейромоделей, которая отличается высоким уровнем логической прозрачности при моделировании сложных технических систем. Объектом исследования является процесс применения индикаторной системы для упрощения и выбора топологии нейромоделей.

**Цель работы** заключается в разработке и использовании системы индикаторов для определения уровня сложности задачи моделирования и постепенном подборе оптимальной логически прозрачной топологии нейромоделей.

**Метод.** Предложен метод для подбора оптимальной, логично прозрачной топологии нейронной сети для моделирования сложных технических систем с использованием системы соответствующих индикаторов. В начале метод определяет общий уровень сложности задачи моделирования и используя полученную оценку определяет способ дальнейшей оптимизации нейромоделей. Затем используя данные о задаче и характеристики входных данных метод позволяет получить наиболее оптимальную структуру нейронной модели для дальнейшего моделирования системы. Метод позволяет сократить время обучения и повысить уровень логической прозрачности нейромоделей, что значительно расширяет практическое использование таких моделей, без использования нейроevolюционных методов, которые могут быть не оправданно ресурсоемкими при ряде задач.

**Результаты.** Разработанный метод реализован и исследован при решении задачи моделирования динамики питтинговых процессов сталных сплавов. Использование разработанного метода позволило сократить время обучения модели на 22%, в зависимости от используемых вычислительных ресурсов. Также метод позволил повысить уровень логической прозрачности модели сократив количество вычислительных узлов на 50%, что также свидетельствует об ускорении и более рациональном использовании ресурсов.

**Выводы.** Проведенные эксперименты подтвердили работоспособность предложенного математического обеспечения и позволяют рекомендовать его для использования на практике при проектировании топологии нейромоделей для дальнейшего решения задач моделирования, диагностирования и оценивания. Перспективы дальнейших исследований могут заключаться в разработке методов структурной оптимизации предварительно синтезированных моделей и разработке новых методов отбора информативных признаков.

**КЛЮЧЕВЫЕ СЛОВА:** оценка сложности, система индикаторов, моделирование, нейромоделі, выборка, обучение, ошибка, градиент.

## ЛІТЕРАТУРА / LITERATURE

1. Goldberg Y. Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies) / Y. Goldberg, G. Hirst. – California : Morgan & Claypool Publishers, 2017. – 310 p.
2. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook / C. C. Aggarwal. – Berlin : Springer, 2018. – 520 p.
3. Valigi N. Zero to AI: A non-technical, hype-free guide to prospering in the AI era / N. Valigi, G. Mauro. – New York : Manning Publications, 2020. – 264 p.
4. Artasanchez A. Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3.x / A. Artasanchez, P. Joshi. – Birmingham : Packt Publishing, 2020. – 618 p.
5. Additional training of neuro-fuzzy diagnostic models / [A. Oliinyk, S. Subbotin, S. Leoshchenko et al.] // Radio Electronics, Computer Science, Control. – 2018. – № 3. – P. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
6. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems / [S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko] // 2018 International Scientific-Practical Conference Problems of Informatics. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018 : proceedings. – Kharkiv: IEEE, 2018. – P. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
7. Iba H. Evolutionary Approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks / H. Iba. – New York : Springer, 2018. – 258 p.
8. Omelianenko I. Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms / I. Omelianenko. – Birmingham : Packt Publishing, 2019. – 368 p.
9. Blokdyk G. Neuroevolution of augmenting topologies: Second Edition / G. Blokdyk. – Ohio : 5STARCOoks, 2018. – 128 p.
10. How is Contrast Encoded in Deep Neural Networks? / A. Akbarinia, K. R. Gegenfurtner [Electronic resource]. – Access mode: <https://arxiv.org/abs/1809.01438>
11. Senge P. M. The Fifth Discipline: The Art & Practice of The Learning Organization / P. M. Senge. – New York : Doubleday, 2006. – 445 p.
12. Weaver W. Science and complexity / W. Weaver // American Scientist. – 1948. – Vol. 36, No. 4. – P. 536–544.
13. Flood R.L. Dealing with Complexity / R. L. Flood, E. R. Carson. – Berlin : Springer, 1993. – 296 p.
14. Spiegelhalter D. The Art of Statistics: How to Learn from Data / D. Spiegelhalter. – New York : Basic Books, 2019. – 448 p.
15. Bruce P. Practical Statistics for Data Scientists: 50 Essential Concepts: 1st Edition / P. Bruce, A. Bruce. – California : O'Reilly Media, 2017. – 318 p.
16. Finch W. H. Exploratory Factor Analysis: 1st Edition / W. Holmes Finch. – California : SAGE Publications, 2019. – 144 p.
17. Rencher A. C. Methods of Multivariate Analysis: 3rd Edition / A. C. Rencher, W. F. Christensen. – New Jersey : John Wiley & Sons, 2012. – 800 p.
18. Dean A. Design and Analysis of Experiments (Springer Texts in Statistics): 2nd Edition / A. Dean, D. Voss, D. Draguljić. – Berlin : Springer, 2017. – 865 p.
19. Sewak M. Deep Reinforcement Learning: Frontiers of Artificial Intelligence / M. Sewak. – Berlin : Springer, 2020. – 220 p.
20. Calix R. A Deep Learning Algorithms: Transformers, gans, encoders, cnns, rnns, and more / R. A. Calix. – Traverse City : Independently published, 2020. – 428 p.
21. Narivs'kyi O. E. Corrosion fracture of platelike heat exchangers / O. E. Narivs'kyi // Materials Science. – 2005. – 41, № 1. – P. 122–128.
22. Narivs'kyi O.E. Micromechanism of corrosion fracture of the plates of heat exchangers / O. E. Narivs'kyi // Materials Science. – 2007. – 43, №1. – p. 124–132.
23. Kuhn M. Feature Engineering and Selection: A Practical Approach for Predictive Models (Chapman & Hall/CRC Data Science Series) / M. Kuhn, K. Johnson. – London: Chapman and Hall (CRC Press), 2019. – 310 p.
24. Parallel method of big data reduction based on stochastic programming approach / [A. Oliinyk, S. Subbotin, V. Lovkin, M. Ilyashenko, O. Blagodariov] // Radio Electronics, Computer Science, Control. – 2018. – № 2. – P. 60–72.
25. Parallel data reduction method for complex technical objects and processes / [A. Oliinyk, S. Leoshchenko, V. Lovkin et al.] // Dependable Systems, Services and Technologies, (DESSERT'2018) : The 9th IEEE International Conference, Kyiv, 24–27 May 2018 : proceedings. – Los Alamitos: IEEE, 2018. – P. 528–531.