UDC 004.896

# NEURAL NETWORK DIAGNOSTICS OF AIRCRAFT PARTS BASED ON THE RESULTS OF OPERATIONAL PROCESSES

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

**Pukhalska H. V.** – PhD, Associate Professor, Associate Professor of the Department of Machinery Engineering Technology, 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.** – Dr. Sc., Professor, Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

**Gofman Ye. O.** – PhD, Senior Researcher of the Research Unit, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

## ABSTRACT

**Context.** The problem of synthesis of an optimal neural network model for diagnostics of aircraft parts after operational processes is considered. The object of the study is the process of synthesis of neural network diagnostic models for aircraft parts based on the results of operational processes

**Objective** is to synthesize neural network diagnostic models of aircraft parts after operational processes with a high level of accuracy.

**Method.** It is proposed to research the use of two approaches to the synthesis of neural network diagnostic models. So, using a system of indicators, the topology of the neural network is calculated, which will be trained using the method of Backpropagation method in the future. The second approach is based on the use of a neuroevolutionary approach, which allows for a complete synthesis of the neural network, dynamically modifying the topology of the solution in addition to the parameters. the final decisions are compared in the accuracy of work on the training and test data set. This approach will allow to determine the possibility and correctness of using neuroevolutionary methods for the synthesis of diagnostic models.

**Results.** Neuromodels for diagnostics of aircraft parts based on the results of operational processes have been obtained. The obtained results of comparing the methods used for synthesis made it possible to form recommendations for the implementation of neuroevolutionary methods in the synthesis of diagnostic neuromodels.

**Conclusions.** The results obtained during the experiments confirmed the operability of the mathematical software used and allowed us to form recommendations for further use of the considered methods in practice in order to synthesize diagnostic neuromodels. The prospects for further research may consist in expanding the input data sets in order to synthesize and study more complex topologies of neural network models.

**KEYWORDS:** diagnostics, aviation parts, synthesis, training, neuroevolution, data sampling, operational processes.

## ABBREVIATIONS

ANN is an artificial neural net;
CPU is central processing unit;
FDI is fault detection and identification method;
MGA is modification genetic algorithm;
SSD is solid-state drive;
UAE is United Arab Emirates.

## NOMENCLATURE

$\varepsilon$ is discrepancy, which is the difference between the real and calculated outputs;
$\delta$ is error of the neurons in the output layer;
$\theta$ is error of the neurons in the hidden layer;
$\mu$ is the learning rate;
$\sigma_{0,2}$ is yield strength;
$\sigma_{в}$ is tensile strength;
$comp$ is separate component of system;
$char$ is separate characteristic of component;
$DV_q$ is vector of desired outputs;
$f_a$ is activation function;
$FB$ is recurrent connections in ANN;
$k$ is number of components at system;
$l$ is number of neurons at the network input;
$m$ is number of dependent (categorical) features of sample instances;
$N_i$ is multiple neurons at the network input;
$N_{i_l}$ is neuron at the network input;
$N_o$ is multiple neurons at the network output;
$N_{o_p}$ is neuron at the network output;
$N_h$ is a multiple neurons of the hidden network layer;
$N_{h_r}$ is hidden network layer neuron;
$NN$ is a neural network;
$NN_{struct}$ is structure of neural network;
$p$ is number of neurons at the network output;
$r$ is number of neurons in the hidden network layer;
$status$ is main defect characteristic of component;
$Sample$ is data set;
$System$ is general information about aviation system;
$x_n$ is independent attribute of the sample instance;
$X$ is input vector of perceptron;
$w_{i_l h_r}$ is coefficient of the input and hidden layers;


DOI 10.15588/1607-3274-2022-2-7

$w_{h_r o_p}$ is coefficient of the hidden and output layers;

$W^{ih}$ is matrix of coefficients of the input and hidden layers;

$W^{ho}$ is matrix of coefficient of the hidden and output layers;

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

$Y$ is output vector of perceptron.

$Y_m$ is the vector of parameters calculated using the analytical model;

$Y_{real}$ is the vector of the output parameters; of the engine obtained by measuring using sensors.

## INTRODUCTION

At present, when the plane crashes have become a global problem, the problem of early detection of malfunctions of aircraft parts and systems has become particularly relevant [1, 2]. Traditionally, the process of diagnosing malfunctions of aviation systems is carried out using analytical models based on physical patterns, as well as by statistical processing of flight monitoring data. Specialists dealing with this problem install sensors that measure the parameters of aircraft engines during flights [1–3]. The flight monitoring data file usually contains parameters such as [1–3]:

– flight number;
– flight date;
– total engine operating time;
– temperature and air pressure at the engine inlet;
– temperature and gas pressure behind the turbine;
– temperature of the blades;
– oil level and temperature in the oil block;
– Mach number, etc.

The number of flight parameters can reach hundreds or more units.

After performing a certain number of flights, the engine, blades, transmission and other parts are removed from the aircraft and subjected to bench disassembly, during which a number of defects are identified and eliminated [1–4].

The task of the diagnostic engineer is to use flight monitoring data to identify system defects before they fail or before preventive disassembly. As already noted, traditionally this problem is solved by applying techniques based on physical laws: each defect causes certain deviations of certain parameters of work, physical characteristics, etc., therefore, analyzing their nature of change, it is possible to make assumptions about the appearance of defects that cause these changes. It is clear that due to the significant amounts of information and the complexity of the existing relationships between defects and measured parameters, the task of analyzing flight monitoring data and detecting defects is far from trivial and in many cases is not solved reliably and qualitatively enough [4].

The main directions determining the improvement of the quality of information technologies for diagnosing the technical condition of aviation should be considered the intellectualization of information processing processes involving data mining methods [2]. Such an approach is capable of improving the quality of recognition of the technical condition under the action of the above defined (measurable) and uncertain factors, as well as the integration of information processes (distributed local databases and knowledge into a global database and knowledge) [1, 3].

Data analyzing methods represent a new direction that complements and develops classical statistical research methods, often referred to in domestic and foreign literature as Data Mining and knowledge discovery. Data Mining uses modern intelligent technologies, including neural networks, fuzzy logic, expert systems. These technologies are used in this work to solve a wide range of problems of diagnostics of the technical condition of complex technical systems and their components [4].

In this article, it is proposed to solve this problem using a neural network basis, for example, using a multilayer perceptron with sigmoid activation functions. First of all, it should be noted that in the input vector $X$ of the perceptron, places should be provided for all monitoring parameters, the values of which are affected by the appearance of detected defects. Possible defects of the aircraft engine can be encoded in the output vector $Y$, for example, using zeros and ones. Vectors of desired outputs are compiled $DV_q$ based on the results of bench disassembly of engines [1–4].

**The object of study** is the process of using a model based on a neural network to diagnose aircraft parts after operational processes with high accuracy.

To test and investigate various approaches to the synthesis of neural network diagnostic models.

**The subject of the study** is a neural network model for the diagnosis of aircraft parts after operational processes, characterized by high accuracy.

Using information about operational processes and fixed traces of these processes to synthesize neural network diagnostic models.

**The purpose of the work** is to build and study a diagnostic neuromodel for aircraft parts after operational processes.

## 1 PROBLEM STATEMENT

The task of diagnosing aircraft parts based on the results of operational processes can be presented as a diagnostic task where it is necessary to determine whether the part is serviceable or not.

Thus, let's imagine an aviation system as a set of individual components $System = \{comp_1, comp_2, comp_3, ..., comp_k\}$, where $k$ is the number of components (parts) of the system. Each component has a number of characteristics of features that can be measured with bench measurements or in real time using specialized sensors $comp = \{char_1, char_2, char_3, ..., char_l; status\}$, and in addition to physical (or chemical) characteristics, the component also has a characteristic of its defect: $status$. Thus, to train a diagnostic neuromodel $NN$, a sample is ob-


DOI 10.15588/1607-3274-2022-2-7

tained: $Sample=\langle X,Y\rangle$, where $X$ the set of input features consists of the characteristics of the part $X=\{x_1=char_1, x_2=char_2, x_3=char_3,..., x_l=char_l\}$, and the set of output features consists of the characteristics of the defect of the part $Y=\{y=status\}$.

Then the diagnostic neuromodel of an aircraft part can be represented as an ANN: $NN$, consisting of structural elements and a set of parameters $NN=(struct, param)$. The structure of such a neuromodel is determined by sets of computational nodes – neurons and connections between them: $struct=\{N,c\}, N=\{N_i, N_h, N_o\}, c=\{c\}$. In turn, the aggregates of many neurons are divided into subsets by layers: $N_i=\{N_{i_1}, N_{i_2},..., N_{i_l}\}, l=1,2,...,|N_i|$ neurons of the input layer, $N_o=\{N_{o_1}, N_{o_2},..., N_{o_p}\}, p=1,2,...,|N_o|$ output and hidden $N_h=\{N_{h_1}, N_{h_2},..., N_{h_r}\}, r=1,2,...,|N_h|$. It should be noted that the neurons of the input layer take values from the set of input features $X$, so their number is equal. The subset of links consists of the links themselves and their weighting coefficients: $c=\{c_1, c_2,...c_k\}, k=1,2,...,|c|$, $w=\{w_k\}$.

Accordingly, the task can be presented as a synthesis of ANN with optimal structure and accuracy $NN=(struct, param)$, based on a sample of initial, experimental data about the object under study $Sample=\langle X,Y\rangle$. For further automation of the process of diagnostics of aircraft parts based on the results of operation, as a particular classification task.

## 2 REVIEW OF THE LITERATURE

The analysis of works in the field of automation of the process of diagnosing the condition of aircraft parts based on analytical models, including ANNs [1–4], demonstrates that today such work is being carried out extremely actively. However, it is worth noting that a number of such works are poorly covered due to a number of factors: secrecy, military or corporate secrecy, narrow specialization of the tasks being solved. A number of works do not cover engineering solutions or give only general theoretical and practical recommendations for solving such problems.

The use of the FDI method is recognized as a common approach in similar tasks [5–8]. This methodology for solving problems of automation of diagnostics of the technical condition of aircraft parts is based on the principle of comparing the measurement results of physical (or chemical) parameters of a real part (system) with the calculated parameters calculated on the basis of a mathematical model [5–8].

Fig. 1 shows the general scheme of using the FDI method to automate the task of automatic technical diagnostics. So in the diagram, where $X$ is the vector of control actions; $Y_m$ is the vector of parameters calculated using the analytical model of the part (system); $Y_{real}$ is the vector of the output parameters of the engine obtained by measuring using sensors; $\varepsilon = Y_{real} - Y_m$ the discrepancy, which is the difference between the vectors $Y_m$ and $Y_{real}$.

As a category of work, he suggests using ANNs as an analytical model of a technical part (or system) [9, 10]. The range of tasks solved using such a model within the framework of the FDI method is quite wide: from the tasks of monitoring and diagnosing the technical condition to debugging parameters [9, 10].

The main stages of the engineering methodology for building an INS model include [9, 10]:

1) preliminary data analysis at the stage of setting the task and choosing the neural network architecture;
2) data transformation (preprocessing) to build a more efficient network setup procedure;
3) the choice of neural network architecture;
4) selection of the neural network structure;
5) selection of the learning algorithm;
6) neural network training and testing;
7) analysis of the accuracy of the neural network solution;
8) making a decision based on the results obtained.

The analysis of the published works devoted to the use of ANNs for diagnosing the parameters of aircraft parts shows that in these works the main trends and characteristic features of solving the problems of diagnostics of parts based on ANN are highlighted. At the same time, they are devoted, as a rule, to solving particular problems, for example [11–14]:

– diagnosing the condition of the turbine blades of a gas turbine engine;
– formation of a space of diagnostic signs of the state of a gas turbine engine for the construction of a neural network classifier;
– indirect measurement of the temperature of gases behind the combustion chamber based on the ANN to diagnose the thermal condition of the engine.

They do not contain instructions on the choice of architecture, structure and methods of ANN training; there is no engineering methodology for designing such networks in relation to the tasks of diagnosing the technical condition of aircraft engines. Neural network methods for solving problems of diagnostics of aircraft parts are investigated below in order to identify the main patterns of their use and develop appropriate methods and techniques for the implementation of diagnostics of technical condition based on ANN [9, 10].

## 3 MATERIALS AND METHODS

In general, an ANN is a mathematical model, as well as its software implementation (or imitation), working on the principle of the human brain: it runs input data through a system of neurons: computing nodes interacting with each other, after which it outputs a certain result of calculations based on this interaction [15–19]. Also, in more complex architectures of such models, previous experience and mistakes of past launches play an important role in decision-making. This behavior of the model


DOI 10.15588/1607-3274-2022-2-7

leads to the thesis about a certain level of self-learning of the ANNs as an artificial intelligence system [15–19].

Today, ANNs solve a wide range of tasks: from digital image processing to forecasting financial processes. Accordingly, in some tasks, models based on ANNs can replace experts: in medicine the doctors, in technical tasks the operators, etc. [15–19].

The main advantage of ANNs is their ability to study patterns in training data and how best to associate it with the target variable that needs to be determined (or forecasted). From an analytical point of view, ANNs are capable of recreating any function and have proven themselves as a universal approximation device, that is, the emulation of some objects into other, more simplified ones [15–19].

The Multilayer Perceptron is one of the simplest ANN models that emulates a primitive model of the biological brain within the framework of machine learning and can be used to solve complex computational tasks such as classification or prediction. To put it simply, it can be noted that a perceptron is a model of a single neuron, which was the predecessor of larger and more complex ANNs capable of more accurately emulating brain function and using more natural approaches to model learning [15–19].

The basic computing nodes (perceptron blocks) are artificial neurons, simple computing blocks that have weighted input signals and generate an output signal using the activation function. The parameter of the weighting coefficient of such a neuron is similar to the coefficients used in the equation from the theory of linear regression [15–19]. Similar to linear regression, each neuron also has bias, which can be considered as an input weight, by default equal to one. For example, a neuron may have two input data sources, in which case three weights are required: one for each input source and one for the weights. Weights are often initialized with small random values, but more complex initialization schemes can be used for more complex ANNs topologies [15–19]. As in linear regression, large weights indicate an increased complexity of the model. It is desirable that the weights in the network are small, then regularization methods are applicable. The weighted input data is summed up and transmitted via an activation function, sometimes called a transfer function. This is a simple display of the summed weighted input and output of a neuron. The function determines the threshold at which the neuron is activated and the strength of the output signal. Nonlinear activation functions are traditionally used. This allows the network to combine input data in a more complex way and, in turn, expand the capabilities of the functions that they can model [15–19].

Neurons are organized into a network. A number of neurons are called a layer, and one network can consist of several layers. The architecture of neurons in a network is often referred to as network topology. The initial input layer, which accepts input data from a dataset, is called visible because it is an open part of the network [15–19]. The layers after the input are called hidden because they are not directly exposed. The simplest network structure is to have a single neuron in the hidden layer that directly outputs the value [15–19]. With the availability of computing power and efficient software libraries, it is possible to build neural networks of deep learning, which means a lot of hidden layers. The last hidden layer is called the output layer, and it is responsible for the output of values or their vector in the appropriate format. After setting up, the neural network needs to be trained on your dataset [15–19].

One of the most common methods of ANNs training is the Backpropagation method [20, 21]. Having a simple perceptron, as in Fig. 2, it is noted the input layer, where data is received by one hidden layer, and the output layer [20, 21]. The input layer contains the number of neurons corresponding to the input data of the neuron, one of which is called the displacement neuron. The displacement neuron always contains the same value, for example, one and is designed to supply a constant displacement to all subsequent neurons with which it is connected, it can be disabled by setting it to 0. Next comes a hidden layer consisting of a given number of neurons, again one of which is a displacement neuron. Note that it is connected only to the subsequent output layer, no connections are received from the input layer, since it does not change its state. The result of the network is calculated on the output layer, in which the number of neurons is determined beforehand. As a rule, this number depends on the number of target variables [20, 21].

Each subsequent layer is connected to the previous layer by links with certain weight coefficients. There may be several hidden layers in the network. The network is called a direct distribution network because the first layer is connected to the second, the second to the third, and so on, and there are no feedbacks, for example, from the output layer to the input. Networks with feedbacks are called recurrent networks and are more complex and resource-intensive in operation [20, 21].

For convenience, in all cases, the displacement neuron number is assumed to be zero. The coupling coefficients of the input and hidden layers can be denoted as $w_{i_l h_r}$, and the matrix of these coefficients is denoted by $W^{ih}$. Thus, $w_{i_0 h_1}$ it determines the connection of the input layer displacement neuron with the first neuron of the hidden layer, and $w_{i_3 h_2}$ sets the connection of the third neuron of the input layer with the second neuron of the hidden layer [20, 21].

The coupling coefficients of the hidden and output layers can be denoted as $w_{h_r o_p}$, and the matrix of these coefficients will be called $W^{ho}$ large.


DOI 10.15588/1607-3274-2022-2-7

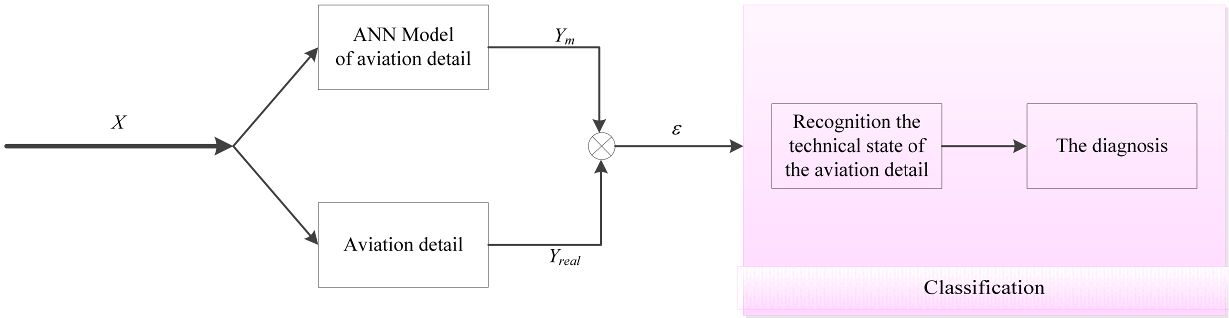


Figure 1 – Implementation of FDI method

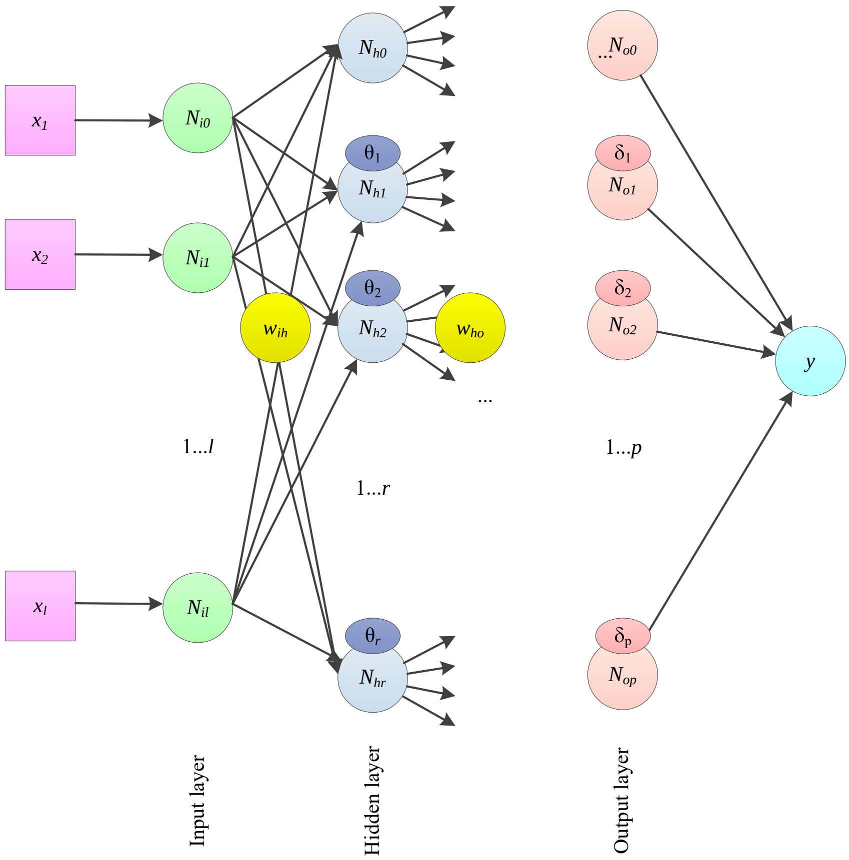


Figure 2 – General implementation of Backpropagation method

$$W^{ih}=\begin{pmatrix} w_{i_0h_1} & w_{i_0h_2} & w_{i_0h_3} & \dots & w_{i_0h} \\ w_{i_1h_1} & w_{i_1h_2} & w_{i_1h_3} & \dots & w_{i_1h} \\ w_{i_2h_1} & w_{i_2h_2} & w_{i_2h_3} & \dots & w_{i_2h} \\ \dots & \dots & \dots & \dots & \dots \\ w_{ih_1} & w_{ih_2} & w_{ih_3} & \dots & w_{ih} \end{pmatrix}, \qquad (1)$$

$$W^{ho}=\begin{pmatrix} w_{h_0o_1} & w_{h_0o_2} & \dots & w_{h_0o} \\ w_{h_1o_1} & w_{h_1o_2} & \dots & w_{h_1o} \\ \dots & \dots & \dots & \dots \\ w_{ho_1} & w_{ho_2} & \dots & w_{ho} \end{pmatrix}. \qquad (2)$$

The values of the input layer can be represented by a vector $X = N_i$. The values of the neurons of the hidden layer $N_{h_r}$, they make up a vector $N_h$, and the output values $N_{o_p}$ are a vector $N_o$. The information is processed sequentially, first the values of the hidden layer $N_h$ are calculated, then the values of the output layer $N_o$ [20, 21].

The formula for calculating the values of the hidden layer is indicated by a number:

$$N_{h_r} = f_a\left(\sum_{i=0...l} N_{i_l} \cdot w_{ih}\right). \qquad (3)$$


DOI 10.15588/1607-3274-2022-2-7

Each neuron calculates a combined input consisting of the sum of the products of the input value by the corresponding weight, and then the result is run through the activation function of this neuron $f_a$.

By analogy with the previous layer, the formula (4) is compiled to calculate the output values. The combined input is the sum of the products of the values of the intermediate layer by the values $N_{h_r}$ of the weights $w_{h_r o_p}$. The result is fed to the activation function [20, 21].

$$N_{o_p} = f_a\left(\sum_{h=0...r} N_{h_r} \cdot w_{h_r o_p}\right). \qquad (4)$$

The essence of the method is that when submitting a training set of examples, the result of the network is compared with the target value, errors in the output layer are determined as $\delta$ (Fig. 2), and then these errors are propagated in the opposite direction and the errors of the neurons of the hidden layers are calculated as $\theta$, and at the last step, the values of all weights are adjusted based on the values errors found [20, 21].

It is necessary to use the general (5), in which the difference between the target $N_{o_p}$ and real values $y_p$ is multiplied with the value of the derivative of the activation function:

$$\delta_p = \left(y_p - N_{o_p}\right) \cdot f_a^{'}\left(net_p\right). \qquad (5)$$

Next, we will find the errors of the neurons of the hidden layer: $\theta$. The error for the displacement neuron is not calculated:

$$\theta_r = f_a^{'}\left(net_r\right) \sum_{p=1...p} \delta_p \cdot w_{ho} \,. \qquad (6)$$

Thus, the error of a hidden layer neuron is a combination of the errors of all the neurons that it affects. The larger the connection $w_{h_r o_p}$, the more the error of the output layer $\delta$ affects the error of the neuron of the hidden layer. Thus, the error is propagated backwards from the network output to its hidden layers [20, 21].

The final stage is the adjustment of the weights of the arrays $W^{ih}$ and $W^{ho}$ [20, 21]:

$$w_{ih}^{'} = w_{ih} + \Delta w_{ih} = w_{ih} + \mu N_{i_l} \theta_r \,, \qquad (7)$$

$$w_{ho}^{'} = w_{ho} + \Delta w_{ho} = w_{ho} + \mu N_{h_r} \delta_p \,, \qquad (8)$$

where $\mu$ is the learning rate, which is set in the range $[0.1;0.4]$.

However, analyzing the above method, it can be concluded that in general, the training of the model based on the ANN is reduced to iterative iteration of trial and error, since the Backpropagation method does not involve the selection and fine-tuning of the architecture, but works with the already selected topology. Moreover, a number of papers note problems in the areas of local optima. Therefore, since the 2010s, more and more attention has been paid to neuroevolutionary methods of ANN synthesis [22, 23]. Such approaches existed before, but it was with the growth of computational capabilities that they began to show better results in comparison with gradient learning methods [22, 23].

The neuroevolutionary approach to the synthesis of INS uses evolutionary methods to create an ANN: the selection of its parameters, topology and rules. Neuroevolution is usually used as part of the reinforcement learning paradigm, and it can be contrasted with traditional deep learning methods that use gradient descent in a neural network with a fixed topology. Due to more flexible settings of synthesis parameters, the process allows fine-tuning and selecting the ANNs architecture for each task, avoiding the problem of retraining [22, 23].

Of course, this approach involves the use of large computing and time resources. So the synthesis process begins with the installation of metaparameters and the synthesis framework: the accuracy of the ANN, the number of epochs, the learning rate and topological complexity. The complexity can be set by limiting the number of hidden layers and neurons in them, the presence of feedbacks in the neurons of the hidden layer, etc. [22, 23].

As a neuroevolutionary method, consider a MGA. So, at the beginning, restrictions are set on the structural complexity of the final solution: the presence of feedbacks ($FB = 0 \,||\, FB = 1$), the number and depth of hidden layers ($|N_h|$) and stopping criteria. After that, a population is generated from simple ANN, and their genetic information is encoded based on interneuronal connections. Further, relying on the mechanisms of selective pressure and smart crossover, the main stages of GA are performed: crossing, mutation of a new generation and selection of individuals into the parent pool [23]. In general, the method can be represented schematically as in Fig. 3.

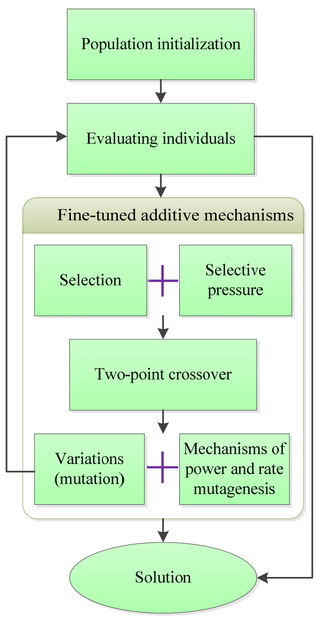


Figure 3 – General scheme of MGA method


DOI 10.15588/1607-3274-2022-2-7

## 4 EXPERIMENTS

The blades of the first stage of the compressor of the Klimov TV3-117 engine, having operational damage to the feather of the blades of the engines, were selected as the object of research [24]. In the studies, engines that were in operation in different countries were observed, respectively, the physical characteristics of the operational processes differed. From this it can be concluded that aircraft parts have different operating time and, accordingly, different degrees of damage to the blades. The engines were operated and observed in the following countries and enterprises: Yemen, India, UAE, Peru, Cyprus, Utair (Tyumen), Algeria, Spain [24].

Table 1 shows an example of sampling input data.

The table shows that $x_1$ is the average temperature in the region where the operational process took place; $x_2$ and $x_3$ are the values of the chord, in sections A2–A2 and A8–A8; $x_4$ is HB, the hardness of the initial blade, HRC; $x_5$ is $\sigma_{0,2}$, yield strength, MPa; $x_6$ is $\sigma_в$ tensile strength, MPa; $x_7$ is the frequency of natural vibrations of the blades, Hz.

$y_1$: $T_1$ total operating time; $y_2$: T2 operating time up to first repair.

For the experiments, a workstation with the following characteristics was used: Intel Core i5-8250U CPU (1.60–3.40 GHz (Intel Turbo Boost 2.0), 4 cores and 8 threads), 16 Gb RAM (dual-channel mode), SK hynix SC308 128 GB SSD (M.2), the Java programming language.

## 5 RESULTS

Table 2 shows the selected information features with their weight coefficients.

Table 3 shows a comparison of the results of the two methods. So the work of the methods was compared according to the following parameters:

– work time: time spent on the synthesis of ANN;

– accuracy of work on the training sample: accuracy of the model during training;

– accuracy of work on the test sample: accuracy of the model during testing.

Tables 4 and 5 shows the neural network models obtained.

Table 1 – Example of fragment from data set

| Blade number | Average temperature | Blade A2-A2 | Blade A8-A8 | HRC | $\sigma_{0.2}$ | $\sigma_в$ | Self-frequency of natural vibrations | $T_1$ | $T_2$ |
|---|---|---|---|---|---|---|---|---|---|
| Index | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $y_1$ | $y_2$ |
| India-1 | 24.6 | 26.7 | 28.2 | 38 | 970 | 1180 | 603.2 | 1652 | 724 |
| India-2 | 24.6 | 26.55 | 28.22 | 38 | 970 | 1180 | 617.1 | 1652 | 724 |
| India-3 | 24.6 | 26.73 | 28.1 | 38 | 970 | 1180 | 631.8 | 1652 | 724 |
| India-4 | 24.6 | 26.75 | 28.09 | 38 | 970 | 1180 | 623.9 | 1652 | 724 |
| India-5 | 24.6 | 26.59 | 28.12 | 38 | 970 | 1180 | 634.9 | 1652 | 724 |
| India-6 | 24.6 | 26.56 | 28.22 | 38 | 970 | 1180 | 624 | 1652 | 724 |
| India-7 | 24.6 | 26.6 | 28.13 | 38 | 970 | 1180 | 629.9 | 1652 | 724 |
| India-8 | 24.6 | 26.53 | 28 | 38 | 970 | 1180 | 637.2 | 1652 | 724 |
| India-9 | 24.6 | 26.83 | 28.2 | 38 | 970 | 1180 | 615 | 1652 | 724 |
| India-10 | 24.6 | 26.3 | 28.28 | 38 | 970 | 1180 | 625.4 | 1652 | 724 |
| … | … | … | … | … | … | … | … | … | … |
| Yemen-20 | 20.5 | 26.63 | 28 | 990 | 451 | 32 | 950 | 1100 | 627.4 |

Table 2 – Results of feature selection

| | $y_2$ | $y_3$ |
|---|---|---|
| $x_1$ | 0.2153 | –0.1901 |
| $x_2$ | –0.0323 | –0.0030 |
| $x_3$ | –0.3629 | –0.0191 |
| $x_6$ | 0.7211 | 0.4971 |
| $x_7$ | 0.0844 | 0.0336 |

Table 3 – Results of using different methods for neuromodels synthesis

| Target variable | Method | The synthesis time. s | Accuracy of work on the training sample | Accuracy of work on the test sample |
|---|---|---|---|---|
| $y_1$ | Backpropagation | 15.3726 | 0.0002 | 0.00025 |
| | MGA | 64.2397 | 0.00017 | 0.00024 |
| $y_2$ | Backpropagation | 15.2863 | 0.0001 | 0.0001 |
| | MGA | 52.6493 | 0.00014 | 0.0001 |


DOI 10.15588/1607-3274-2022-2-7

Table 4 – Coefficients matrices of resulting neuromodels for $y_1$

| | Number of layer | Number of neuron at layer | Number of input of neuron | | | |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 3 |
| Backpropagation $y_1$ | 1 | 1 | –40.2890 | 64.2278 | –1.2651 | 152.6108 |
| | | 2 | –8.7750 | –7.2652 | 0.0004 | –104.6105 |
| | | 3 | 10.5814 | 11.9088 | 0.0000 | 92.3337 |
| | | 4 | 22.2327 | –9.9015 | 0.2474 | 130.9445 |
| | | 5 | –37.1694 | –34.9135 | 0.0009 | 57.2398 |
| | 2 | 1 | 17.5119 | –10.0749 | 0.0060 | –42.5008 |
| MGA $y_1$ | Number of layer | Number of neuron at layer | Number of input of neuron | | | |
| | | | 0 | 1 | 2 | 3 |
| | 1 | 1 | –10.3678 | 3.9329 | 0.1771 | 44.4167 |
| | | 2 | –42.0171 | 0.0407 | 0.0058 | 0.3170 |
| | | 3 | –79.2515 | 0.2169 | 0.1018 | –85.1717 |
| | | 4 | 20.4838 | 0.5891 | –0.0395 | 10.0486 |
| | | 5 | 21.3511 | 0.1410 | 0.0512 | 9.8286 |
| | 2 | 1 | 37.8691 | –3.0681 | 0.8152 | 41.7886 |

Table 5 – Coefficients matrices of resulting neuromodels $y_2$

| | Number of layer | Number of neuron at layer | Number of input of neuron | | | |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 3 |
| Backpropagation $y_2$ | 1 | 1 | –2.3766 | –2.7395 | –2.7277 | 5.7025 |
| | | 2 | 1.8790 | 0.0000 | 1.7174 | 0.0000 |
| | | 3 | –15.9818 | 0.0000 | –15.2190 | 0.0000 |
| | | 4 | –7.6228 | 1.7200 | –7.7104 | 0.8612 |
| | | 5 | 37.7405 | 0.0000 | 36.0519 | 0.0000 |
| | 2 | 1 | –7.1881 | 1.8330 | –7.0854 | 4.2028 |
| MGA $y_2$ | Number of layer | Number of neuron at layer | Number of input of neuron | | | |
| | | | 0 | 1 | 2 | 3 |
| | 1 | 1 | 13.5323 | –9.2208 | 5.8346 | –15.7445 |
| | | 2 | –0.6086 | –0.6000 | 0.4055 | 0.4643 |
| | | 3 | –4.1636 | –4.1129 | 2.0460 | –106.7364 |
| | | 4 | 7.7830 | –2.6386 | 4.4783 | –13.5350 |
| | | 5 | 8.0768 | –2.3306 | 2.9106 | –13.5143 |
| | 2 | 1 | –2.3971 | –4.4495 | 6.9384 | –3.2805 |

## 6 DISCUSSION

For the operating time in both cases, the combination of informatively important features is the same. And in both cases, the frequency of natural vibrations of the blades is an important sign.

When initializing the synthesis process using MGA, restrictions were set on the absence of feedbacks and excessive growth of hidden layers. Based on the assessment of the complexity of the task, the optimal number of neurons in the hidden layer was chosen 4 [25]. During neuroevolutionary synthesis, this number of neurons was confirmed.

Comparing the operating time, it can be noted that the MGA method worked much slower, this is due to the fact that the method worked in single-threaded mode and completely synthesized a new network architecture, operating with a population of non-network models. From this we can conclude that in simple tasks, neuroevolutionary methods may need increased time resources. At the same time, the higher accuracy of the synthesized solution (which has been confirmed experimentally) may not fully justify such time expenditures.

However, in complex tasks, when the process of input data preprocessing is not possible or is largely difficult and the accuracy of the model is extremely important, neuroevolutionary methods can show great efficiency. This is due to the lower dependence of the operation of such methods on the noise of the input data, as well as the proportionally increasing time spent on training complex topologies using iterative methods.

## CONCLUSIONS

The urgent scientific and applied problem of synthesis of an optimal neural network model for diagnostics of aircraft parts after operational processes has been solved.

**The scientific novelty** lies in the fact that it is proposed to use different methods for the synthesis of neuromodels. Thus, the Backpropagation method was used to train a predefined ANN structure based on an assessment of the complexity of the simulated task. The MGA method was also used for neuroevolutionary synthesis of the model. As a result, both methods presented similar perceptron topologies with the same structures.

**The practical significance** lies in the fact that the rationality of approaches to the synthesis of neuromodels has been investigated. So for $y_1$ and $y_2$, MGA worked slower by 23.93% and 29.03%, respectively. At the same time, the accuracy of the resulting models differed by 0.3–0.1 percentage points. From this we can form a recommendation: for such tasks, the use of neuroevolutionary methods may not be justified precisely in the case of the time resources spent. However, for more complex tasks, where accuracy is more important, the neuroevolutionary approach will be preferable.

**Prospects for further research** are to expand the dataset of input characteristics of aircraft parts to use


DOI 10.15588/1607-3274-2022-2-7

complex ANN topologies and monitor the accuracy of their operation.

## 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" "Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects" (state registration number 0121U107499) and "Intelligent methods and tools for diagnosing and predicting the state of complex objects" (state registration number 0122U000972).

## REFERENCES

1. Smith D. J. Reliability, Maintainability and Risk: Practical Methods for Engineers, Oxford, Butterworth-Heinemann, 2021, 516 p.
2. Høyer C. B., Nielsen T. S., Nagel L. L., Uhrenholt L., Boel L.W.T. Investigation of a fatal airplane crash: autopsy, computed tomography, and injury pattern analysis used to determine who was steering the plane at the time of the accident. A case report, *Forensic Science, Medicine and Pathology*, 2012, Vol. 8(2), P. 179–188. DOI: 10.1007/s12024-011-9239-4
3. Maltry G.W. Airplane Crash Analysis [Electronic resource]. Access mode: https://www.edtengineers.com/blog-post/airplane-crash-analysis
4. Yunusov S., Labendik V., Guseynov S. Monitoring and Diagnostics of Aircraft Gas Turbine Engines: Improvement of Models and Methods for Diagnosis of Gas Path of Gas Turbine Engines, Chisinau: LAP LAMBERT Academic Publishing, 2014, 204 p.
5. Johri P., Anand A., Vain J., Singh J., Quasim M.T. System Assurances: Modeling and Management (Emerging Methodologies and Applications in Modelling, Identification and Control), Cambridge: Academic Press, 2022, 614 p.
6. Sun W., Paiva A.R.C., Xu P., Sundaram A., Braatz R.D. Fault Detection and Identification using Bayesian Recurrent Neural Networks, *Computers & Chemical Engineering*, 2019, Vol. 141, pp. 1–43. DOI: 10.1016/j.compchemeng.2020.106991
7. Nguyen H.V., Golinval J.-C. Fault detection based on Kernel Principal Component Analysis, *Engineering Structures*, 2010, Vol. 32(11), pp. 3683–3691. DOI: 10.1016/j.engstruct.2010.08.012
8. Aldrich C., Auret L. Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods, Berlin: Springer, 2013, 396 p.
9. Adouni A., Chariag D., Diallo D., Hamed M.B., Sbita L. FDI based on Artificial Neural Network for Low-Voltage-Ride-Through in DFIG-based Wind Turbine, *ISA Transactions*, 2016, Vol. 64, pp. 353–364. DOI: 10.1016/j.isatra.2016.05.009
10. Plikynas D., Akbar Y. H. Neural Network Approaches to Estimating FDI Flows: Evidence from Central and Eastern Europe, *Eastern European Economics*, Vol. 44, No. 3, 2006, pp. 29–59.
11. Babenko O., Pribora T. Analysis of the results of the study of the frequencies and forms of natural vibrations of the working blade of the 1st stage of the SLP, Bulletin of Engine Building, 2018, Vol. 2, pp. 91–98. [In Russian]
12. Dvirnyk Ya., Pavlenko D. The influence of dust erosion on the gas dynamic characteristics of the axial compressor of the GTE Vestnik dvigatelstroeniya, Bulletin of Engine Building, 2017, Vol. 1, pp. 56–66. [In Russian]
13. Yefanov V., Prokopenko O., Ovchinnikov O., Vnukov U. Erosion resistance of compressor blades of helicopter gas turbine engines with various types of coatings, Bulletin of Engine Building, 2017, Vol. 1, pp. 120–123. [In Russian]
14. Dvirnyk Ya., Pavlenko D. Patterns of wear of the compressor blades of helicopter engines operating in a dusty atmosphere, Bulletin of Engine Building, 2016, Vol. 1, pp. 42–51. [In Russian]
15. Huang H. Statistical Mechanics of Neural Networks, Berlin: Springer, 2022, 314 p.
16. Ekman M. Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow, Boston, Addison-Wesley Professional, 2021, 752 p.
17. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook, Berlin, Springer, 2018, 520 p.
18. Wu L., Cui P., Pei J., Zhao L. Graph Neural Networks: Foundations, Frontiers, and Applications, Berlin, Springer, 2022, 752 p.
19. Kneusel R.T. Math for Deep Learning: What You Need to Know to Understand Neural Networks, San Francisco, No Starch Press, 2021, 344 p.
20. Chapmann J. Neural Networks: Introduction to Artificial Neurons, Backpropagation Algorithms and Multilayer Feedforward Networks (Advanced Data Analytics), Scotts Valley, CreateSpace Independent Publishing Platform, 2017, 108 p.
21. Wadi H. Learn From Scratch Backpropagation Neural Networks using Python GUI & MariaDB, Chicago, Independently published, 2021, 590 p.
22. Milani A., Carpi A., Poggioni V. Evolutionary Algorithms in Intelligent Systems, Basel: Mdpi AG, 2020, 144 p.
23. Leoshchenko S., Oliinyk A., Subbotin S., Lytvyn V., Shkarupylo V. Modification and parallelization of genetic algorithm for synthesis of artificial neural networks, *Radio Electronics, Computer Science, Control*, 2019, No. 4, pp. 68–82. DOI: 10.15588/1607-3274-2019-4-7
24. Subbotin S., Pukhalska H., Leoshchenko S., Oliinyk A., Gofman Ye. Neuromodeling of operational processes, *Radio electronics, computer science, control*, 2022, No. 1, pp. 120–129.
25. Leoshchenko S., Subbotin S., Oliinyk A., Narivs'kiy O. Implementation of the indicator system in modeling complex technical systems, *Radio electronics, computer science, control*, 2021, No. 1, pp. 117–127. DOI: 10.15588/1607-3274-2021-1-12.

Received 15.03.2022.
Accepted 22.03.2022.


DOI 10.15588/1607-3274-2022-2-7

УДК 004.896

## НЕЙРОМЕРЕЖЕВЕ ДІАГНОСТУВАННЯ АВІАЦІЙНИХ ДЕТАЛЕЙ ЗА РЕЗУЛЬТАТАМИ ЕКСПЛУАТАЦІЙНИХ ПРОЦЕСІВ

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

**Пухальська Г.В.** – канд. техн. наук, доцент кафедри технологія машинобудування 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.** – д-р техн. наук, професор, професор кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

**Гофман Є. О.** – канд. техн. наук, старший науковий співробітник науково-дослідної частини Національного університету «Запорізька політехніка», Запоріжжя, Україна.

### АНОТАЦІЯ

**Актуальність.** Розглянуто завдання синтезу оптимальної нейромережевої моделі для діагностики авіаційних деталей після експлуатаційних процесів. Об'єктом дослідження є процес синтезу нейромережевих діагностичних моделей для авіаційних деталей за результатами експлуатаційних процесів.

**Мета роботи** полягає в синтезі нейромережевих діагностичних моделей авіаційних деталей після експлуатаційних процесів з високим рівнем точності.

**Метод.** Запропоновано дослідити використання двох підходів до синтезу нейромережевих діагностичних моделей. Так використовуючи систему індикаторів, обчислюється топологія нейронної мережі, яка в подальшому буде навчена з використанням методу зворотного поширення помилки. Другий же підхід грунтується на використанні нейроеволюційного підходу, який дозволяє зробити повний синтез нейронної мережі, динамічно модифікуючи крім параметрів і топологію рішення. підсумкові рішення порівнюються в точності роботи на навчальному і тестовому наборі даних. Такий підхід дозволить визначити можливість і коректність використання нейроеволюційних методів для синтезу діагностичних моделей.

**Результати.** Отримано нейромоделі для діагностики авіаційних деталей за результатами експлуатаційних процесів. Отримані результати порівняння використовуваних для синтезу методів дозволили сформувати рекомендації для імплементації нейроеволюційних методів в процеси синтезу діагностичних нейромоделей.

**Висновок.** Отримані в ході експериментів результати підтвердили працездатність використовуваного математичного забезпечення і дозволили сформувати рекомендації для подальшого використання розглянутих методів на практиці з метою синтезу діагностичних нейромоделей. Перспективи подальших досліджень можуть полягати в розширенні вхідних наборів даних з метою синтезу і дослідження більш складних топологій нейромережевих моделей.

**КЛЮЧОВІ СЛОВА**: діагностування, авіаційні деталі, синтез, навчання, нейроеволюція, вибірка даних, експлуатаційні процеси.

УДК 004.896

## НЕЙРОСЕТЕВОЕ ДИАГНОСТИРОВАНИЕ АВИАЦИОННЫХ ДЕТАЛЕЙ ПО РЕЗУЛЬТАТАМ ЭКСПЛУАТАЦИОННЫХ ПРОЦЕССОВ

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

**Пухальская Г.В.** – канд. техн. наук, доцент кафедры технология машиностроения 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.** – д-р техн. наук, профессор, профессор кафедры программных средств Национального университета «Запорожская политехника», Запорожье Украина.

**Гофман Е. А.** – старший научный сотрудник научно-исследовательской части Национального университета «Запорожская политехника», Запорожье Украина.

### АННОТАЦИЯ

**Актуальность.** Рассмотрена задача синтеза оптимальной нейросетевой модели для диагностики авиационных деталей после эксплуатационных процессов. Объектом исследования является процесс синтеза нейросетевых диагностических моделей для авиационных деталей по результатам эксплуатационных процессов.

**Цель работы** заключается в синтезе нейросетевых диагностических моделей авиационных деталей после эксплуатационных процессов с высоким уровнем точности.

**Метод.** Предложено исследовать использование двух подходов к синтезу нейросетевых диагностических моделей. Так используя систему индикаторов, вычисляется топология нейронной сети, которая в дальнейшем будет обучена с использованием метода обратного распространения ошибки. Второй же подход основывается на использовании нейроэволюционного подхода, который позволяет произвести полный синтез нейронной сети, динамично модифицируя помимо параметров и топологию решения. итоговые решения сравниваются в точности работы на обучающем и тестовом наборе данных. Такой


DOI 10.15588/1607-3274-2022-2-7

подход позволит определить возможность и корректность использования нейроэволюционных методов для синтеза диагностических моделей.

**Результаты.** Получены нейромодели для диагностики авиационных деталей по результатам эксплуатационных процессов. Полученные результаты сравнения используемых для синтеза методов позволили сформировать рекомендации для имплементации нейроэволюционных методов в процессы синтеза диагностических нейромоделей.

**Выводы.** Полученные в ходе экспериментов результаты подтвердили работоспособность используемого математического обеспечения и позволили сформировать рекомендации для дальнейшего использования рассматриваемых методов на практике с целью синтеза диагностических нейромоделей. Перспективы дальнейших исследований могут заключаться в расширении входных наборов данных с целью синтеза и исследования более сложных топологий нейросетевых моделей.

**КЛЮЧЕВЫЕ СЛОВА:** диагностирование, авиационные детали, синтез, обучение, нейроэволюция, выборка данных, эксплуатационные процессы.

## ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Smith D.J . Reliability, Maintainability and Risk : Practical Methods for Engineers / D. J. Smith. – Oxford : Butterworth-Heinemann, 2021. – 516 p.
2. Investigation of a fatal airplane crash: autopsy, computed tomography, and injury pattern analysis used to determine who was steering the plane at the time of the accident. A case report / [C. B. Høyer, T. S. Nielsen, L. L. Nagel et al.] // Forensic Science, Medicine and Pathology. – 2012. – Vol. 8(2). – P. 179–188. DOI: 10.1007/s12024-011-9239-4
3. Maltry G.W. Airplane Crash Analysis [Electronic resource]. Access mode: https://www.edtengineers.com/blog-post/airplane-crash-analysis
4. Yunusov S. Monitoring and Diagnostics of Aircraft Gas Turbine Engines: Improvement of Models and Methods for Diagnosis of Gas Path of Gas Turbine Engines / S. Yunusov, V. Labendik, S. Guseynov. – Chisinau : LAP LAMBERT Academic Publishing, 2014. – 204 p.
5. System Assurances: Modeling and Management (Emerging Methodologies and Applications in Modelling, Identification and Control) / [P. Johri, A. Anand, J. Vain et al.]. – Cambridge : Academic Press, 2022. – 614 p.
6. Fault Detection and Identification using Bayesian Recurrent Neural Networks / [W. Sun, A.R.C. Paiva, P. Xu et al.] // Computers & Chemical Engineering. – 2019. – Vol. 141. – P. 1–43. DOI: 10.1016/j.compchemeng.2020.106991
7. Nguyen H. V. Fault detection based on Kernel Principal Component Analysis / H. V. Nguyen, J.-C. Golinval // Engineering Structures. – 2010. – Vol. 32(11). – P. 3683–3691. DOI: 10.1016/j.engstruct.2010.08.012
8. Aldrich C. Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods / C. Aldrich, L. Auret. – Berlin : Springer, 2013. – 396 p.
9. FDI based on Artificial Neural Network for Low-Voltage-Ride-Through in DFIG-based Wind Turbine / [A. Adouni, D. Chariag, D. Diallo et al.] // ISA Transactions. – 2016. – Vol. 64. – P. 353–364. DOI: 10.1016/j.isatra.2016.05.009
10. Plikynas D. Neural Network Approaches to Estimating FDI Flows: Evidence from Central and Eastern Europe / D. Plikynas, Y. H. Akbar // Eastern European Economics. – 2006. – Vol. 44, No. 3. –P. 29–59.
11. Бабенко О. Н. Анализ результатов исследования частот и форм собственных колебаний рабочей лопатки 1 ступени КНД / О. Н. Бабенко, Т. И. Прибора // Вестник двигателестроения. – 2018. – № 2. – С. 91–98.
12. Двирник Я.В. Влияние пылевой эрозии на газодинамические характеристики осевого компрессора ГТД / Я. В. Двирник, Д. В. Павленко // Вестник двигателестроения. – 2017. – № 1. – С. 56–66.
13. Эрозионная стойкость лопаток компрессора вертолетных газотурбинных двигателей с различными типами покрытий / [В. С. Ефанов, А. Н. Прокопенко, А. В. Овчинников и др.] // Вестник двигателестроения. – 2017 – № 1. – С. 120–123.
14. Павленко Д. В. Закономерности изнашивания рабочих лопаток компрессора вертолетных двигателей, эксплуатирующихся в условиях запыленной атмосферы / Д. В. Павленко, Я. В. Двирник // Вестник двигателестроения. – 2016. – № 1. – С. 42–51.
15. Huang H. Statistical Mechanics of Neural Networks / H. Huang. – Berlin : Springer, 2022. – 314 p.
16. Ekman M. Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow / M. Ekman. – Boston : Addison-Wesley Professional, 2021. – 752 p.
17. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook / C. C. Aggarwal. – Berlin : Springer, 2018. – 520 p.
18. Graph Neural Networks: Foundations, Frontiers, and Applications / [L. Wu, P. Cui, J. Pei, L. Zhao]. – Berlin : Springer, 2022. – 752 p.
19. Kneusel R. T. Math for Deep Learning: What You Need to Know to Understand Neural Networks / R. T. Kneusel. – San Francisco : No Starch Press, 2021. – 344 p.
20. Chapmann J. Neural Networks: Introduction to Artificial Neurons, Backpropagation Algorithms and Multilayer Feedforward Networks (Advanced Data Analytics) / J. Chapmann. – Scotts Valley : CreateSpace Independent Publishing Platform, 2017. – 108 p.
21. Wadi H. Learn From Scratch Backpropagation Neural Networks using Python GUI & MariaDB / H. Wadi. – Chicago : Independently published, 2021. – 590 p.
22. Evolutionary Algorithms in Intelligent Systems / [A. Milani, A. Carpi, V. Poggioni]. – Basel : Mdpi AG, 2020. – 144 p.
23. Modification and parallelization of genetic algorithm for synthesis of artificial neural networks / [S. D. Leoshchenko, A. O. Oliinyk, S. A. Subbotin at al.] // Radio Electronics, Computer Science, Control. – 2019. – № 4. – P. 68–82. DOI: 10.15588/1607-3274-2019-4-7
24. Neuromodeling of operational processes / [S. A. Subbotin, H. V. Pukhalska, S. D. Leoshchenko et al.] // Radio electronics, computer science, control. – 2022. – № 1. – P. 120–129.
25. Implementation of the indicator system in modeling complex technical systems / [S. D. Leoshchenko, S. A. Subbotin, A. O. Oliinyk, O. E. Narivs'kiy] // Radio electronics, computer science, control. – 2021. – № 1. – P. 117–127. DOI: 10.15588/1607-3274-2021-1-12.


DOI 10.15588/1607-3274-2022-2-7