I. J. Computer Network and Information Security, **2024, 4, 65-87**
Published Online on August 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.04.05

Universal On-board Neural Network System for Restoring Information in Case of Helicopter Turboshaft Engine Sensor Failure

Serhii Vladov
Kremenchuk Flight College of Kharkiv National University of Internal Affairs, Kremenchuk, 39605, Ukraine
E-mail: serhii.vladov@univd.edu.ua
ORCID iD: https://orcid.org/0000-0001-8009-5254

Ruslan Yakovliev
Kremenchuk Flight College of Kharkiv National University of Internal Affairs, Kremenchuk, 39605, Ukraine
E-mail: director.klk.hnuvs@gmail.com
ORCID iD: https://orcid.org/0000-0002-3788-2583

Victoria Vysotska
Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: Victoria.A.Vysotska@lpnu.ua
ORCID iD: https://orcid.org/0000-0001-6417-3689

Dmytro Uhryn
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: d.ugryn@chnu.edu.ua
ORCID iD: https://orcid.org/ 0000-0003-4858-4511

Yuriy Ushenko*
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine
E-mail: y.ushenko@chnu.edu.ua
ORCID iD: https://orcid.org/0000-0003-1767-1882
*Corresponding author

Received: 13 March 2023; Revised: 07 May 2023; Accepted: 16 June 2023; Published: 08 August 2024

Abstract: This work focuses on developing a universal onboard neural network system for restoring information when helicopter turboshaft engine sensors fail. A mathematical task was formulated to determine the occurrence and location of these sensor failures using a multi-class Bayesian classification model that incorporates prior knowledge and updates probabilities with new data. The Bayesian approach was employed for identifying and localizing sensor failures, utilizing a Bayesian neural network with a 4–6–3 structure as the core of the developed system. A training algorithm for the Bayesian neural network was created, which estimates the prior distribution of network parameters through variational approximation, maximizes the evidence lower bound of direct likelihood instead, and updates parameters by calculating gradients of the log-likelihood and evidence lower bound, while adding regularization terms for warnings, distributions, and uncertainty estimates to interpret results. This approach ensures balanced data handling, effective training (achieving nearly 100% accuracy on both training and validation sets), and improved model understanding (with training losses not exceeding 2.5%). An example is provided that demonstrates solving the information restoration task in the event of a gas-generator rotor r.p.m. sensor failure in the TV3-117 helicopter turboshaft engine. The developed onboard neural network system implementing feasibility on a helicopter using the neuro-processor Intel Neural Compute Stick 2 has been analytically proven.

Index Terms: Helicopter Turboshaft Engines, Sensor Failure, Automatic Control Systems, Bayesian Neural Networks, Training, Universal Neural Network Diagram, Restoring.

1. Introduction

In the helicopter technology advancement phase, the key task is to ensure their flight safety and effective functioning. This task's critical aspect is the sensors' parametric failure detection, as well as their characteristics degradation during operation. On board the helicopter, sensors large number are responsible for collecting and transmitting information about various flight parameters such as altitude, speed, frequency, temperature, etc. Important decisions related to safety and flight control are based on reliable data provided by these sensors [1–3].

Sensors used in helicopter turboshaft engines (TE) are exposed to various factors that can lead to their degradation. For example, the engine operation's extreme temperatures can wear down sensor materials, and constant vibrations can cause mechanical damage or changes in the sensitive sensor elements' characteristics. In addition, over time, the materials from which the sensors are made may lose their initial properties, which leads to a deterioration in the measurement accuracy [4, 5].

Built-in self-diagnosis systems have several significant disadvantages. First, the implementation and maintenance cost is a major concern. Such systems development, integration and support require significant financial investments, which increases the helicopter TE operating overall costs. Second, built-in self-diagnosis systems can face difficulties in identifying and localizing failures, especially in environments where many components are interconnected. Even with high automation, pinpointing the malfunction's exact location and cause can be challenging. In addition, there is a false positives risk, where the system may falsely identify failures, leading to unnecessary inspections and maintenance, increasing costs and downtime. Dependence on software is also a significant drawback. The built-in self-diagnosis system reliability depends on the software quality, which may contain errors or require regular updates. Any failures in the software can the self-diagnosis system operation negatively affect [6, 7].

The sensors' redundancy, although it increases the measurements' reliability, also has its disadvantages. The several sensors used to measure one parameter increase the system's total weight and its costs and maintenance installation. This is particularly crucial in helicopters, where the weight of every gram matters. Furthermore, redundancy adds to the system's overall complexity, potentially increasing failure points and complicating the diagnostic and repair process [8–10]. Data conflicts between sensors can also complicate the determining process of which sensor is malfunctioning, which can require additional analysis and increase decision-making time. The increased requirements for processing data from backup sensors require more complex algorithms for their processing and analysis, which can increase the load on the helicopter TE computing system [11, 12].

Real-time data analysis has specific challenges that also require improvement. The main disadvantage is the high requirements for computing resources. Processing data in large amounts in real time requires powerful computing power and fast information processing, which can increase the weight and energy consumption of the system. The analysis effectiveness depends on the algorithm settings accuracy, which may require special knowledge and considerable time to train the models. There are also false conclusions risks due to incorrect or poor-quality data, which can lead to the system's status misunderstanding and unnecessary maintenance actions. The real-time data analysis systems integration of high complexity with existing helicopter control systems also requires significant efforts from developers and technical personnel [13, 14].

Taking into account all traditional approaches listed shortcomings, the neural networks' use is a promising solution for improving parametric sensor failure detection systems. Neural networks can process data in large amounts in real time with high speed and accuracy, which allows for reducing the requirements for computing resources and providing efficient data analysis. They can be configured to detect complex relations and patterns, which greatly increases fault identification accuracy. Neural networks can also adapt to new operating conditions and improve their models based on accumulated data, which reduces the risk of false conclusions [15–17].

The neural network technologies integration with the existing helicopter TE automatic control systems (ACS) [18, 19] can simplify the analysis and diagnosis process, reducing the false alarms and increasing number the system reliability as a whole. The neural network technologies also allow you to automate the setting up and training algorithms process, reducing the need for specialized knowledge and shortening the new solution's time implementation [20–22].

The majority control method proposed in works [23, 24] for addressing the information restoring issue using the gas turbine engines (GTE) neural network model offers several key advantages. First, this method increases the data restoring reliability and accuracy because it uses multiple neural network models to estimate the parameters, and the results are determined based on the obtained values majority. This approach reduces the individual model errors influence and increases the system's overall stability. Second, the majority control method facilitates the anomalies and failures detection and correction, since discrepancies in the different models' results can signal potential problems with data or sensors. Finally, this method is flexible and adaptable, allowing new models or algorithms easy integration without significant impact on the already existing system, which makes it an effective tool for improving the GTE diagnostic and monitoring processes. The main provisions proposed in [23, 24] were further developed in [25] by adapting them to the peculiarities of helicopter TE operation. In [26], a decomposition neural network was created as a local neural network set for solving the GTE sensor failures event restoring lost information task. However, in [26] the neural network composition is not described, which creates difficulties in the proposed approach to reproduction and analysis. In addition, the method has limitations, as it is developed exclusively for turbojet engines, which narrows its application to GTE other types, in particular, helicopter TE. It is also important to note that [26] does not show a specific example of the restoring lost information task, which makes it difficult to assess the proposed approach's effectiveness and practical usefulness.

Considering the above, the work aims to develop and enhance a system for restoring lost information due to helicopter TE sensor failures by adapting the majority control method, which employs multiple neural network models to improve data restoration reliability and accuracy. Additionally, it addresses limitations noted in [26], such as restricted application to turbojet engines and insufficient detail on neural network composition and specific information recovery examples.

2. Material and Methods

As mentioned above, one of the cases of helicopter TE failures concerns sensors, which can have both short-term and long-term failures. According to [26], an intelligent controller is integrated into the helicopter TE ACS (Fig. 1) to restore lost information in the sensor parametric failure event. This controller assesses the sensor states and, in the failure event, utilizes the necessary parameter values calculated by the built-in neural network to generate control actions for the actuators.

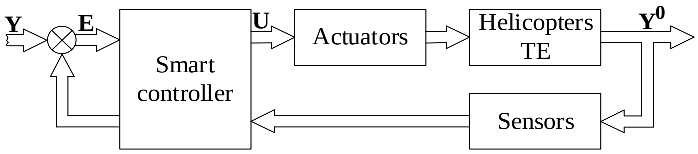


Fig.1. Generalized block diagram of helicopter turboshaft engines' automatic control systems (author's research, based on [23])

In practice, given the helicopter TE input and output parameter vectors' large size, the variety in operating modes, and the extensive time required for the neural network train, it is practical to decompose the neural network into a local neural network set. Such a system structure, used within an intelligent controller, is shown in Fig. 2. The basic neural network (*BNN*) includes several additional neural networks ($NN_1 ... NN_i$) and logical blocks (*LB*) that define the interaction algorithms among the modules. NN_1 calculates the gas-generator rotor r.p.m. (n_{TC}) at the current time based on the helicopter TE's other thermogas-dynamic parameters measurements, simulating the n_{TC} parameter. Similarly, NN_2 calculates the gas temperature in front of the compressor turbine (T_G^*), and NN_3 calculates the free turbine rotor speed (n_{FT}). The restoring lost information system (Fig. 2) can be modified by adding neural networks (NN_k) for other sensors recording various parameters on board the helicopter. LB_1 switches the measured parameters n_{TC}, T_G^* and n_{FT} to the neural network inputs, enabling calculations to counter sensor failures; LB_2 and LB_3 connect individual neural networks in the failed sensors' place if necessary to counter their failures; *BNN* identifies and localizes sensor failures; Y^R represents the reconstructed helicopter TE parameters vector.

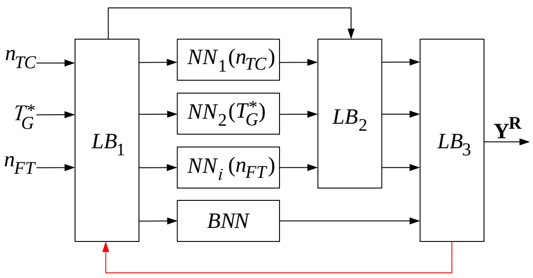


Fig.2. The restoring helicopter turboshaft engine parameters values universal neural network diagram (author's research, based on [26])

Given the nonlinear dependencies among the thermogas-dynamic parameters recorded on board, the helicopter TE, three-layer perceptrons are used as neural networks $NN_1 ... NN_i$, as shown in Fig. 3, according to [26]. In these networks, the input layer consists of the neural network input signals, represented by the input signal $x(t)$ and a constant signal equal to 1. The constant unity signal provides the bias for this task. The hidden layer consists of three neurons with sigmoid activation functions. This layer of each neuron, according to [27], is represented by the equations:

$$O_j = \frac{1}{1 + e^{S_j}}, \tag{1}$$

$$S_j = \sum_{i=1}^{m} W_{ij} O_i, \tag{2}$$

where O_i is the input layer *i*-th neuron output signal; W_{ij} is the hidden layer *j*-th neuron *i*-th input synaptic weight; *m* is the input layer neurons number.

The initial perceptron layer one neuron consists, which creates the signal $\hat{y}(t)$ is the hidden layer neurons output signals weighted sum, i.e.

$$y(t) = \sum_{j=1}^{n} V_j O_j, \tag{3}$$

where O_j is the hidden layer j-th neuron output signal; V_j is the output layer neuron j-th input synaptic weight; n is the hidden layer neurons number.

The perceptron V_j and W_{ij} synaptic weights adjustment is performed using the backpropagation method [27–29], according to which the optimality criterion $J(t) = \varepsilon^2(t) = \frac{\left(y(t) - y(t)\right)^2}{2}$ minimization is performed using a gradient procedure [27]:

$$V_j(t) = V_j(t-1) - \mu_V \frac{\partial J(t)}{\partial V_j} = V_j(t-1) - \mu_V \varepsilon(t) O_j, \tag{4}$$

$$W_{ij}(t) = W_{ij}(t-1) - \mu_W W_{ij} = W_{ij}(t-1) - \mu_W \varepsilon(t) V_j(t)\left(1 - O_j\right) O_i, \tag{5}$$

where μ_V and μ_W are numerical coefficients that indicate the gradient search duration and influence the beginning speed.

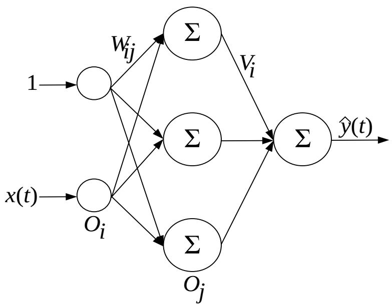


Fig.3. Structure of a three-layer perceptron for the implementation of $NN_1...NN_i$ [27]

In this context, the perceptron with one hidden layer use (as a neural network for $NN_1...NN_i$ type) makes sense, taking into account the helicopter TE parameters' large number processing complex task decomposition. The main aim is to simplify calculations, reduce training time and optimize network architecture. The single hidden layer perceptron is a neural network well-studied and well-known type that performs well in many classifications and function approximation tasks. It has sufficient computing power to solve tasks related to processing the helicopter TE-measured parameters (for example, calculating the n_{TC}, T_G^*, n_{FT} values or other thermogas-dynamic parameters). The neural networks' more complex modifications use, which may require more training data or more computing power, can lead to unnecessary system complexity, increased training time, and degraded performance in fending off sensor failures on board a helicopter task context [27, 30, 31]. Therefore, the perceptron with one hidden layer use is a practical solution to ensure the intelligent controller an efficient and economical implementation to restore lost information in the helicopter TE sensors in one failure parametric event.

The primary task in restoring lost information during sensor parametric failures is selecting the appropriate *BNN* architecture. This choice is crucial for the data restoration process, as it dictates the method for analysing and interpreting the received signals from the system that controls the helicopter TE parameters. The *BNN* is responsible for identifying the occurrence of a problem with the sensors and determining their location, which is important for further actions to restore the parameters of helicopter TE.

To select the *BNN* architecture for the helicopter TE restoring lost information system, a mathematical task is formulated to determine the sensor failure location occurrence and localization. For the helicopter TE sensors failure detecting task, the best approach is to use the Bayesian approach due to its ability to effectively estimate probabilities and a priori knowledge based on the data obtained [32, 33]. Bayesian statistics allows you to model uncertainty and take into account sensor failure and location prior probabilities before obtaining new information. This approach is particularly useful in the system diagnostics context, where there may be uncertainty and changes varying levels in sensor performance over time. The Bayesian approach improves the predictions provides accuracy a more flexible approach to solving diagnostic tasks, and also allows the new data integration into the decision-making process, making it ideal for complex systems involving multivariate aspects and dynamic changes in the environment.

To formulate the helicopter TE sensors failure location determining the occurrence and localization mathematical task using the Bayesian approach, this work uses a multi-class Bayesian classification model taking into account a priori knowledge and updating probabilities based on new information. It is assumed that x is the helicopter TE parameters input vector, containing data from all sensors, D_i is the i-th sensor (i = 1, 2, ..., n, n is the number of the sensors) failure event, $P\left(D_i = 1 \middle| x\right)$ is the i-th sensor failure probability at the input vector x, L_i is the i-th sensor failure location, which can be different places in the sensor system.

Each sensor D_i failure probabilities estimation based on the input vector x is carried out by the expression:

$$P\left(D_i = 1 \middle| x\right) = \frac{P\left(x \middle| D_i = 1\right) P\left(D_i = 1\right)}{P\left(x\right)}, \tag{6}$$

where $P\left(x \middle| D_i = 1\right)$ is the receiving the input vector x probability if the i-th sensor fails, $P\left(D_i = 1\right)$ is the i-th sensor failure priori probability, and $P(x)$ is the input vector x receiving total probability.

For each i-th sensor, the failure location L_i probability is assessed based on the input vector x and the sensor failure D_i fact is carried out by the expression:

$$P\left(L_i = l \middle| x, D_i = 1\right) = \frac{P\left(x \middle| L_i = l, D_i = 1\right) P\left(L_i = l \middle| D_i = 1\right)}{P\left(x \middle| D_i = 1\right)}, \tag{7}$$

where $P\left(x \middle| L_i = l, D_i = 1\right)$ is the input vector x receiving probability, given that the i-th sensor occurred at the l-th location, $P\left(L_i = l \middle| D_i = 1\right)$ represents the prior probability that the i-th sensor failure occurred at the l-th location.

The overall probability for the incoming vector x is calculated using the following expression:

$$P\left(x\right) = \sum_{i=1}^{n} P\left(x \middle| D_i = 1\right) P\left(D_i = 1\right) + P\left(x \middle| D_i = 0\right) P\left(D_i = 0\right), \tag{8}$$

where $P\left(x \middle| D_i = 0\right)$ is the input vector x in the i-th sensor failure absence receiving probability, and $P\left(D_i = 0\right)$ is the i-th sensor failure absence priori probability.

When new data x' is received, the sensor failure probabilities are updated according to Bayes' theorem. The i-th sensor failure updated probability, given the new data x' is calculated as:

$$P\left(D_i = 1 \middle| x'\right) = \frac{P\left(x' \middle| D_i = 1\right) P\left(D_i = 1\right)}{P\left(x'\right)}. \tag{9}$$

Similarly, according to new data x', the i-th sensor failure probability at the l-th location is determined as:

$$P\left(L_i = l \middle| x', D_i = 1\right) = \frac{P\left(x' \middle| L_i = l, D_i = 1\right) P\left(L_i = l \middle| D_i = 1\right)}{P\left(x' \middle| D_i = 1\right)}. \tag{10}$$

To obtain the entire sensor system's overall failure probability and failure location, all sensors are considered simultaneously. When the failure vector of all sensors is designated as $D = (D_1, D_2, \ldots, D_n)$, and the all sensors failure location vector is designated as $L = (L_1, L_2, \ldots, L_n)$, the failure occurrence total probability is calculated as:

$$P\left(D \middle| x\right) = \prod_{i=1}^{n} P\left(D_i \middle| x\right), \tag{11}$$

and the failure location overall probability is:

$$P\left(L \middle| x, D\right) = \prod_{i=1}^{n} P\left(L_i \middle| x, D_i\right). \tag{12}$$

It is also important to consider how the failure of one or more sensors will affect the system as a whole. To do this, the system failure S probability is determined as a function of failures of individual sensors by the expression:

$$P\left(S|x\right)=f\left(P\left(D_1|x\right),P\left(D_2|x\right),...,P\left(D_n|x\right)\right).\tag{13}$$

Taking into account the probabilities of failure and their locations, optimal actions to restore the system are determined. To do this, it is assumed that A is a set of possible actions (for example, checking sensors, replacing components, etc.). The expected utility for each action $a \in A$ is calculated as:

$$U\left(a|x\right)=\sum_{i=1}^{n}P\left(D_i|x\right)\cdot C\left(a,D_i\right),\tag{14}$$

where $C(a, D_i)$ is the losses when acting *as* in the *i*-th sensor failure case.

The optimal action a^* is calculated as:

$$a^*=\arg\max_{a\in A}U\left(a|x\right).\tag{15}$$

Expressions (6)–(15) are used in the *i*-th sensor instantaneous failure case. If helicopter TE sensor failures have a time dependence (i.e. short-term or long-term failures), expressions (6)–(15) are modified using the Markov process model. If t is time, x_t is the input vector at time t, then the sensor failure probability at time t is calculated as:

$$P\left(D_i=1|x_t,t\right)=\frac{P\left(x_t|D_i=1,t\right)P\left(D_i=1|t\right)}{P\left(x_t|t\right)},\tag{16}$$

and the *i*-th sensor failure probability, taking into account preliminary data, is as follows:

$$P\left(D_i=1|x_{1:t}\right)=\frac{P\left(x_{1:t}|D_i=1\right)P\left(D_i=1\right)}{P\left(x_{1:t}\right)},\tag{17}$$

where $x_{1:t}$ is the vector of observations from time 1 to t.

In the case of dependence on helicopter TE sensor failures, the correlation between them is taken into account. The total failure probability of two sensors is given by:

$$P\left(D_i=1,D_j=1|x\right)=P\left(D_i=1|x\right)P\left(D_j=1|D_i=1,x\right),\tag{18}$$

where $P\left(D_j=1|D_i=1,x\right)$ is the *j*-th sensor failure probability, provided that the *i*-th sensor fails at the input vector *x*.

The spatially dependent probability is calculated as:

$$P\left(L_i=l|x,D_i=1\right)=\frac{P\left(x|L_i=l,D_i=1\right)P\left(L_i=l|S\right)}{P\left(x|D_i=1\right)},\tag{19}$$

where $P\left(L_i=l|S\right)$ is the *l*-th failure locating possibility, taking into account the spatial dependencies *S*.

Taking into account the above, the complex probability is calculated as:

$$P\left(D_i=1,|x,t,S\right)=\sum_z\int P\left(D_i=1,|x,z,t,S\right)P\left(z|x,t,S\right)dz,\tag{20}$$

and the failure location complex probability is:

$$P\left(L_i=l|x,D_i=1,t,S\right)=\sum_z\int\frac{P\left(x|L_i=l,D_i=1,z,t,S\right)P\left(L_i=l|D_i=1,z,S\right)}{P\left(x|D_i=1,t,S\right)}dz,\tag{21}$$

where $P\left(D_i=1,|x,z,t,S\right)$ is the *i*-th sensor failure probability taking into account the input vector *x*, hidden variable *z*, time *t* and spatial dependencies *S*; $P\left(z|x,t,S\right)$ is the hidden variable *z* probability, taking into account the input vector *x*, time *t* and spatial dependencies *S*; $P\left(x|L_i=l,D_i=1,z,t,S\right)$ is the input vector *x* receiving probability, provided that the *i*-th sensor failure occurred in the *l*-th place, taking into account the hidden variable *z*, time *t* and spatial dependencies *S*; $P\left(L_i = l \middle| D_i = 1, z, S\right)$ is the probability that the *i*-th sensor failure occurred in the *l*-th location, taking into account the hidden variable *z* and spatial dependencies *S*.

It should be noted that hidden variables (or latent variables) can influence the sensor failure probability, but are not directly observed during the measurement process. These variables may include internal system states, external conditions, operating conditions, or any other factors that cannot be directly measured but affect the sensors' behaviour. Hidden variables allow you to model complex relations in a system that are difficult or impossible to account for using visible variables. They help make models more flexible and accurate in predicting failure probability and fault location.

To implement complex probabilities of failure and localize the failure location, as described above, it is advisable to use a Bayesian neural network (Fig. 4) [32, 33]. Bayesian neural networks (BNN) integrate the Bayesian inference principles with traditional neural networks to handle uncertainty and account for hidden variables. The BNN use is justified because they can learn from data in large amounts, providing flexibility in modelling complex systems, while simultaneously providing uncertainty estimates, which is important for accurately identifying sensor failures and locating them. Thanks to their Bayesian updating ability, these networks can dynamically adapt to new data and provide high accuracy and reliability in predicting and diagnosing complex technical systems, which include helicopter TE.

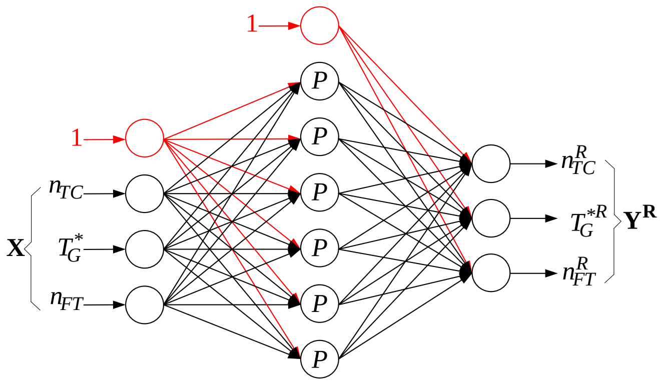


Fig.4. The Bayesian neural network general view as a basic neural network for the helicopter turboshaft engine sensors failure location occurrence and localization determining (author's research, based on [26])

At the Bayesian neural network training initial stage is the initialization stage, and its structure is determined: the layers *L* number, each layer n_l for $l = 1...L$ neurons number, activation functions f_l, and the prior distributions choice for weighting coefficients *W* and shifts *b* is calculated as:

$$W_{i,j} \sim N\left(0, \sigma_\omega^2\right), \;\; b_i \sim N\left(0, \sigma_b^2\right). \tag{22}$$

After this, the parameters for variational approximation are determined when applying the variational Bayes method:

$$q(W) = P\left(W \middle| D\right). \tag{23}$$

At the next stage, the prediction stage, the estimated network outputs for input data *x* are calculated using the current values of the weighting coefficients *W* and biases *b*:

$$h_0 = x, \tag{24}$$

$$h_l = f_l\left(W_l h_{l-1} + b_l\right) \text{ for } l = 1 \ldots L. \tag{25}$$

At the next stage, the probability estimation stage, the log-likelihood is calculated for the Bayesian neural network outputs relative to the real labels by the expression:

$$\log P\left(y \middle| x, W\right) = \log P\left(y \middle| h_L\right). \tag{26}$$

Using likelihood maximization or variational Bayesian inference, a probability estimate is made. This work applies variational Bayesian inference, according to which the evidence lower bound indicator (*ELBO*) is estimated as:

$$ELBO = \mathrm{E}_{q(W)} \log P\left(y \middle| x, W\right) - D_{KL}\left(q(W) \middle\| P(W)\right), \tag{27}$$

where $D_{KL}\left(q(W)\|P(W)\right)=\sum_{i=0}^{n}q(W_i)\log\left(\frac{q(W_i)}{P(W_i)}\right)$ is the Kullback-Leibler divergence (KL-divergence) [34, 35].

At the next stage, the weights updating stage, the log-likelihood gradients or *ELBO* are calculated relative to the model parameters (weights and biases) or:

- log-likelihood gradient:

$$L=-\log P\left(y|x,W\right), \tag{28}$$

$$\frac{\partial L}{\partial W_l}=\frac{\partial L}{\partial h_L}\cdot\frac{\partial h_L}{\partial W_l}, \tag{29}$$

$$\frac{\partial L}{\partial b_l}=\frac{\partial L}{\partial h_L}\cdot\frac{\partial h_L}{\partial b_l}, \tag{30}$$

- *ELBO* gradient [36]:

$$\frac{\partial ELBO}{\partial W_l}=\mathrm{E}_{q(W)}\cdot\left(\frac{\partial\log P\left(y|x,W\right)}{\partial W_l}\right)-\frac{\partial D_{KL}\left(q(W)\|P(W)\right)}{\partial W_l}, \tag{31}$$

$$\frac{\partial ELBO}{\partial b_l}=\mathrm{E}_{q(W)}\cdot\left(\frac{\partial\log P\left(y|x,W\right)}{\partial b_l}\right)-\frac{\partial D_{KL}\left(q(W)\|P(W)\right)}{\partial b_l}. \tag{32}$$

Using optimization methods such as stochastic gradient descent (SGD) or Adam's method, the network parameters are updated as follows:

$$W\leftarrow W-\eta\nabla_W\log P\left(y|x,W\right)\text{ or }W\leftarrow W-\eta\nabla_W ELBO, \tag{33}$$

where η is the training rate.

To prevent overfitting, regularization terms are added to the losses, which for a Bayesian neural network are represented as prior distributions of parameters:

$$L_{reg}=\sum_{l=1}^{L}\left(\frac{W_l^2}{2\sigma_\omega^2}+\frac{b_l^2}{2\sigma_b^2}\right). \tag{34}$$

Taking into account L2-regularization - binding of weight coefficients to the normal distribution, the expression for determining the total losses takes the form:

$$L_{total}=\log P\left(y|x,W\right)-L_{reg}. \tag{35}$$

The proposed Bayesian neural network training algorithm differs in several key ways from the traditional approach:

- The proposed algorithm employs variational approximation to estimate the BNN parameters before distribution. This approach approximates the complex posterior distribution with a simpler variational distribution $q(W)$, facilitating efficient probability and gradient calculations. In contrast, the traditional algorithm does not use variational approximation.
- Instead of direct likelihood maximization, the proposed algorithm uses evidence lower bound (*ELBO*) as a metric for optimization. *ELBO* includes both the data likelihood and a regularization term in the Kullback-Leibler divergence (KL-divergence) form between the variation distribution and the prior distribution (27). This ensures balanced data accounting and regularization.
- To ensure stable and efficient training, the proposed parameter updating algorithm includes both log-likelihood (29)–(30) and *ELBO* (31)–(32) gradients calculation, in particular terms related to variational approximation (33).
- To prevent overfitting, the proposed algorithm adds regularization terms to the losses (34), which is critical for

Bayesian models. This enhances the model's ability to generalize, considering data and parameter uncertainty.
- The proposed algorithm incorporates posterior distributions and uncertainty estimates for result interpretation. This enables a model deeper understanding and facilitates informed decision-making based on predictions and their associated uncertainty.

Thus, the proposed Bayesian neural network training algorithm emphasizes variational approximation, *ELBO* calculation, regularization, and result interpretation with consideration for uncertainty. This approach offers greater flexibility and accuracy compared to traditional methods, particularly when handling complex and uncertain data. Fig. 5 presents the proposed Bayesian neural network training algorithm pre-code segment.

```
# Initialize network structure and parameters
initialize_network()

# Choose prior distributions for weights and biases
initialize_priors()

# Training loop
for epoch in range(num_epochs):
    for batch in data_loader:
        x, y_true = batch

        # Forward pass
        y_pred = forward_pass(x)

        # Compute log-likelihood
        log_likelihood = compute_log_likelihood(y_pred, y_true)

        # Compute ELBO (if using variational inference)
        elbo = compute_elbo(log_likelihood, priors, variational_params)

        # Backward pass and parameter update
        gradients = compute_gradients(elbo, network_params)
        update_params(network_params, gradients, learning_rate)

        # Regularization
        apply_regularization(network_params, priors)

    # Validation step (optional)
    validate_network(validation_data)

# Final evaluation on test data
evaluate_network(test_data)
```

Fig.5. Pseudocode fragment of the proposed Bayesian neural network training algorithm (author's research)

With the proposed training algorithm, the prediction, probability estimation, parameter updating, and regularization stages are iterated through each training epoch until a stopping criterion is met either achieving stable accuracy on the validation set or reaching an epoch-specified number. To determine stable accuracy on the validation set, accuracy changes are monitored over several epochs. This criterion can be implemented by assessing the difference between accuracies or by evaluating the accuracy values average and variance.

The accuracy on the validation set for the current epoch t and the previous epoch $t - 1$ is calculated as:

$$\Delta Accuracy_t = Accuracy_t - Accuracy_{t-1}. \tag{36}$$

If $\left|\Delta Accuracy_t\right|$ less than the given threshold ϵ for a certain number of epochs N, then training stops at:

$$\text{If } \left|\Delta Accuracy_t\right| < \varepsilon \text{ for } t = E - N + 1, E - N + 2, ...E \text{ then "stop".} \tag{37}$$

The average accuracy and variance on the validation sample for the last N epochs are calculated as:

$$\mu_{Accuracy} = \frac{1}{N} \cdot \sum_{i=E-N+1}^{E} Accuracy_i, \tag{38}$$

$$\sigma^2_{Accuracy} = \frac{1}{N} \cdot \sum_{i=E-N+1}^{E} \left(Accuracy_i - \mu_{Accuracy}\right)^2. \tag{39}$$

If the variance is less than the given threshold δ, then training is stopped under the condition:

$$\text{If } \sigma^2_{Accuracy} < \delta \text{ then "stop".} \tag{40}$$

The criterion of reaching a certain number of epochs is based on setting the maximum number of epochs $E_{\max}$, after which training stops:

$$\text{If } E > E_{\max} \text{ then "stop".} \tag{41}$$

Combining the criteria for achieving stable accuracy on the validation sample and achieving a certain number of epochs, it is essential to determine the thresholds ϵ, δ and the epochs number maximum $E_{\max}$, monitor the accuracy on the validation sample, and calculate the stopping criteria (36), (38)–(40). When one of the criteria (36), (38)–(40) is met, the BNN training is stopped according to (37), (40), (41). The stop criteria pseudocode fragment is shown in Fig. 6.

```
# Initialization
epsilon = 0.01
delta = 0.001
E_max = 100
N = 5

# Training loop
for epoch in range(1, E_max + 1):
    train_epoch()
    accuracy_val = evaluate_validation_accuracy()

    accuracies.append(accuracy_val)

    # Check stopping criteria
    if epoch > N:
        delta_accuracy = abs(accuracies[-1] - accuracies[-2])
        if delta_accuracy < epsilon:
            count_stable_epochs += 1
        else:
            count_stable_epochs = 0

        if count_stable_epochs >= N:
            print(f"Stopping training at epoch {epoch} due to stable accuracy")
            break

        mean_accuracy = np.mean(accuracies[-N:])
        variance_accuracy = np.var(accuracies[-N:])
        if variance_accuracy < delta:
            print(f"Stopping training at epoch {epoch} due to low variance in accuracy")
            break

# Final epoch check
if epoch >= E_max:
    print("Reached maximum number of epochs")
```

Fig.6. The Bayesian neural network training stopping criteria pseudocode fragment (author's research)

Hence, the above stopping criteria help to ensure that the model will not be overtrained and the training process will be completed in a reasonable time while maintaining the given generalization ability.

3. Experiment

For the computational experiment, a personal computer equipped with an AMD Ryzen 5 5600 processor (6 cores and 12 threads running at 3.3 GHz) and DDR-4 RAM 32 GB was used. The experiment's first stage involved creating, analysing, and training sample pre-processing that included parameters recorded on board the helicopter (Table 1) [37–41].

The training data preprocessing detailed account is provided in [37–41]. This preprocessing involves evaluating data homogeneity, splitting it into control and test samples, and assessing representativeness through cluster analysis. Homogeneity is assessed using the Fisher-Pearson criterion [42], comparing observed frequencies with χ^2 critical values ($r - k - 1 = 13$, $\alpha = 0.05$). The 3.588 χ^2 value is below the critical 22.362, confirming sample consistency and normal distribution. The Fisher-Snedecor criterion [43], which assesses larger and smaller dispersion values, results in the 1.28

value, which is below the 3.44 critical value, further validating sample consistency. Cluster analysis identified VIII classes (Fig. 7), and after randomization, the training and test samples were divided in a 2:1 ratio (67 and 33 %, respectively). Analysis [37–41, 44] indicates that both samples are representative, with nearly identical cluster distances, as summarized in Table 2.

Table 1. Training sample fragment (author's research, published in [37–41])

Number	Gas generator rotor r.p.m., n_{TC}	Free turbine rotor speed, n_{FT}	Gas temperature in front of the compressor turbine, T_G
1	0.929	0.943	0.932
2	0.933	0.982	0.964
3	0.952	0.962	0.917
4	0.988	0.987	0.908
5	0.991	0.972	0.899
6	0.997	0.963	0.915
7	0.968	0.962	0.922
8	0.962	0.969	0.989
9	0.954	0.947	0.954
…	…	…	…
256	0.973	0.981	0.953

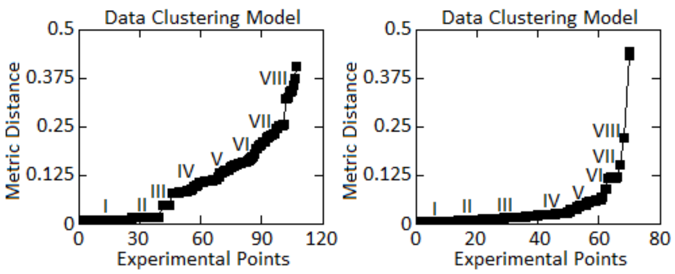


Fig.7. Diagrams of clustering results (I ... VIII classes) on initial experimental (a) and training (b) samples (author's research, published in [37–41])

Table 2. The training and test samples optimal amounts determining results (author's research)

Sample	Description	Amounts
Training	It is utilized in the model train for solving practical tasks based on the provided data.	256 (100 %)
Test	It is employed to evaluate the developed model and the control sample accuracy.	84 (the training sample 33 %)
Control	It is used to monitor neural network training and ensure its representativeness.	172 (the training sample 67 %)

In the computational experiment, 1000 training epochs were conducted to examine the BNN *Accuracy* (Fig. 8) and *Loss* (Fig. 9) on both training and test samples. The accuracy function assesses how effectively the BNN determines the i-th sensor failure complex probability as per (20) and the i-th sensor failure locating complex probability as per (21), reflecting the percentage of the correctly identified value. The *Loss* function measures the deviation between the model's predictions and actual values, guiding improvements in the training process by minimizing this loss. Fig. 8 and Fig. 9 analysis enables the training effectiveness evaluation and, if necessary, the implementation of the measures to enhance performance.

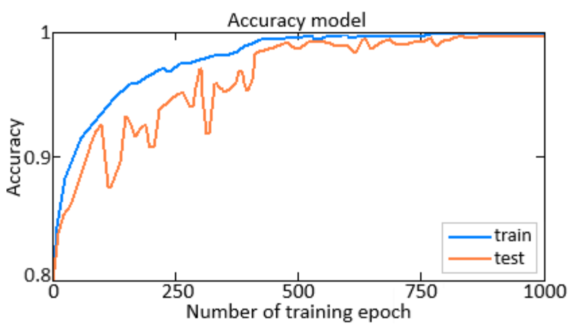


Fig.8. Accuracy diagram on training (blue curve) and test samples (orange curve) Bayesian neural network (author's research)

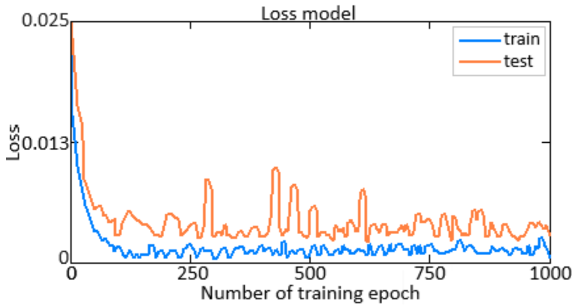


Fig.9. Loss diagram on training (blue curve) and test samples (orange curve) Bayesian neural network (author's research)

The obtained data highlight the BNN's high performance and stability during training. The model's accuracy, evaluated on both training and test samples, approaches unity, reflecting its capability to effectively generalize the helicopter TE parameters and determine the complex probability and location of sensor failure. Additionally, loss values on both datasets remain below 0.025 (2.5 %), indicating a low error rate in determining the sensor failure complex probabilities and its location. These results affirm the training process's success and the proposed algorithm's efficacy in reliably generating the desired complex probability values for sensor failure and its location.

Confidence is determined using an activation function that transforms the model's output into a probability range (typically 0 to 1). In this work, a sigmoidal activation function $\sigma(z)=\frac{1}{1+e^{-z}}$ is used, where z is the model output before applying the activation function. Typically, the model output (after an activation function) can be a certain event probability thought. For example, if the sensor failure probability is the 0.9 value model outputs, this means that the model is 90 % confident that it will fail. To determine the i-th sensor failure complex probability and the i-th sensor failure location complex probability, the activation function is used for the model's corresponding outputs. If the model output for the i-th sensor failure is $z_i^{sensor\ failure}$, and for the i-th sensor failure location is $z_i^{localization}$, then the probabilities are calculated as:

$$P(sensor\ failure)=\sigma\left(z_i^{sensor\ failure}\right), \tag{42}$$

$$P(localization)=\sigma\left(z_i^{localization}\right). \tag{43}$$

It is noted that in the BNN case, the sigmoidal activation function is often used because it has the limiting values property in the range from 0 to 1, which corresponds to probabilities, which is an important characteristic for Bayesian models. Also, the sigmoid function is smooth and differentiated over the entire range, which makes it possible to effectively use optimization methods during network training. In addition, the sigmoid activation function provides a nonlinear relation good approximation between input and output data, allowing the model to more accurately adapt to complex data relations. Therefore, the sigmoidal activation function use is justified in the BNN context as it provides favourable properties for efficient model training and use.

Analytical expressions (42), and (43) allow us to the i-th sensor failure probability determine and the sensor failure locating probability based on the model outputs transformed [45, 46] using the activation function σ. Fig. 10 shows the changes in dynamics in the probability values P(*sensor failure*) (a) and P(*localization*) (b) over 1000 training epochs.

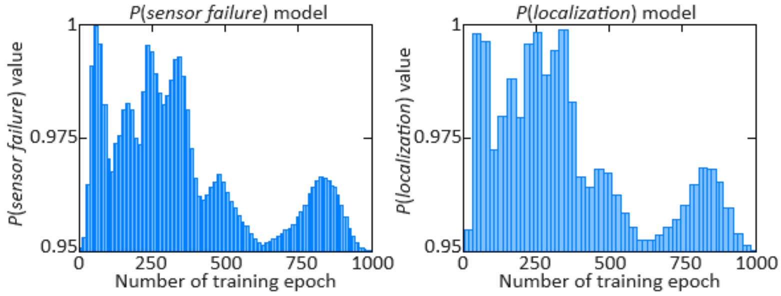


Fig.10. Histogram of the complex probability values distribution of the i-th sensor failure (a) and the complex probability of failure location of the i-th sensor (b) over 1000 training epochs (author's research)

Fig. 10 demonstrates that employing a BNN trained with the proposed algorithm allows for determining and localizing sensor failures with at least 95 % confidence. This indicates the chosen data processing and analysis approach has high efficiency, which is a significant step toward developing reliable helicopter TE monitoring and diagnostic systems. Future steps in this direction could involve optimizing the model to reduce computational costs and enhance accuracy, as well as adapting it to the sensors and engineering systems of other types, thereby improving the technological processes' overall reliability and safety.

4. Results

To simulate the standard helicopter TE sensor (using the TV3-117 TE as an example) failure, a virtual installation was created in the LabVIEW software package. This setup models a system with three sensors: a gas-generator rotor r.p.m. sensor (D-2M), a free rotor speed sensor (D-1M), and a gas temperature sensor in front of the compressor turbine (14 dual thermocouples T-102) [47] (Fig. 11). This research focuses on the gas-generator rotor r.p.m. sensor (D-2M) failure. Fig. 11, the “NN enabled” activation button indicates the gas-generator rotor r.p.m. sensor (D-2M) failure and the neural network information restoration system activation (Fig. 2). Once activated, the neural network system analyzes signals from other sensors to compensate for data loss and gas-generator's condition provide continuous monitoring. Therefore, even if a critical sensor fails, the system ensures reliable diagnostics and engine operation support, enhancing the helicopter operations' overall reliability and safety. Additionally, the virtual installation allows for testing various failure scenarios and their impact on system performance, aiding developers in improving neural network restoring algorithms. It also provides personnel with training in realistic conditions, enhancing their technical skills and readiness to respond to potential failures in actual operations.

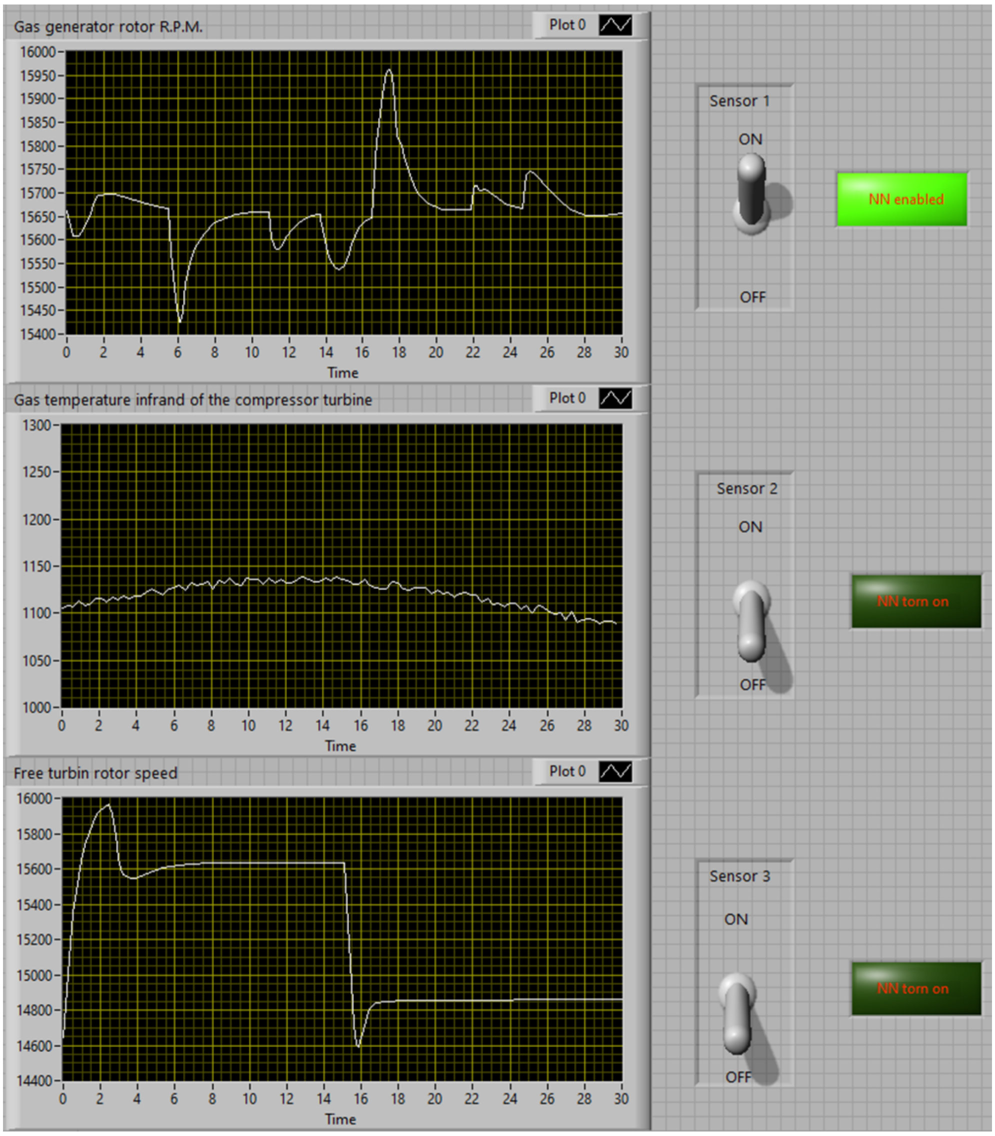
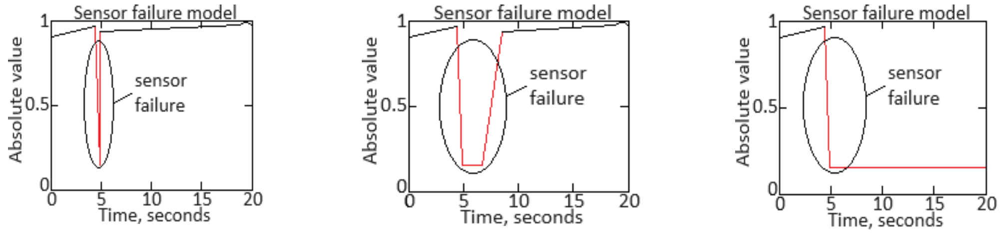


Fig.11. A virtual installation in the LabVIEW software complex that simulates a system with three sensors (author's research)

Fig. 12 shows a generalized diagram for modelling a universal neural network diagram for the helicopter TE parameters (Fig. 2) values restoring in the gas-generator rotor r.p.m. sensor (D-2M) instantaneous failure event (a), the gas-generator rotor r.p.m. sensor (D-2M) short-term failure of the gas-generator rotor r.p.m. sensor (D-2M) (c). As can be seen, if the sensor fails, through analytical redundancy, that is, the BNN use (r_{ii}), it is possible to indirectly measure the ith thermogas-dynamic parameters, which allows for maintaining the control processes required quality.

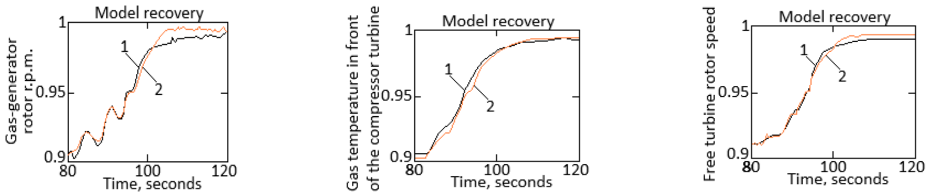


a b c

Fig.12. The gas-generator rotor r.p.m. sensor (D-2M) n_{TC} failure numerical modelling results (author's research)

In the helicopter TE parameters values restoring universal neural network diagram experimental research course (on a TV3-117 TE), the n_{TC}, T_G^*, and n_{FT} parameters sensor failures recorded on board the helicopter were simulated, and the measurement signals were replaced with signals obtained by calculation in it. When a fault signal is input, the fault detection algorithm is first run. According to this algorithm, over some time equal to five calculation cycles, the last sensor reading before its failure is recorded. After this, the sensor value is replaced by the neural network model ($NN_1 \ldots NN_i$) corresponding parameter calculated value.

Fig. 13 shows the TV3-117 TE characteristics in steady-state and transient modes, obtained by operating its self-propelled control system with control parameters from real sensors and replacing them with parameters calculated in the helicopter TE parameters values restoring universal neural network diagram.

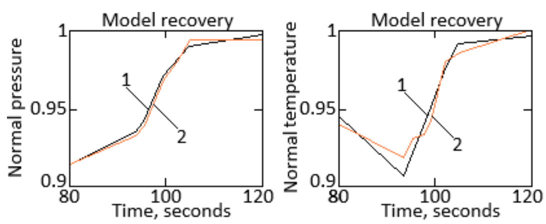


a b c

Fig.13. Diagram of helicopter turboshaft engines thermogas-dynamic parameters sensors failures (1 – measured with working sensors, 2 – restored by a neural network model): a – parameter n_{TC}; b – T_G^* parameter; c – n_{FT} parameter (author's research)

The presented results indicate the engine control acceptable quality when working with the helicopter TE parameters values restoring the universal network diagram. In steady-state conditions, the greatest discrepancy is observed in the idle mode (up to 2...3 %). The error magnitude varies across different experiments due to differences in the engine degree warming in this mode. The greatest difference between the measured and calculated values is observed in the T_G^* parameter and can reach 10...20 K at the 600...700 K temperature level. Transient processes during the helicopter TE parameters values restoring universal network diagram operation are also close to the processes with operating sensors. Differences in flow rates in individual areas do not exceed 3...5 %. The gas-generator rotor r.p.m. and free turbine rotor speed n_{TC} and n_{FT} configuration processes vary slightly.

To the T_N and P_N parameter sensors failures compensate, according to [48], a method is used that is based on introducing into the circuit (Fig. 2) an additional loop with feedback to restore information is the red line in Fig. 2. In this circuit, the parameters T_N and P_N act as the engine control factors. The measured engine parameters (n_{TC}, T_G^*, n_{FT}) calculated values are used as adjustment parameters, with their measured values serving as the regulator settings. This updating method computational research and experimental testing on an engine were performed. The T_N and P_N parameters updating information process is shown in Fig. 14.



a b

Fig.14. Diagram of helicopter turboshaft engines thermogas-dynamic parameters sensors failures (1 is the measured with working sensors, 2 is the restored by a neural network model): a – P_H parameter; b – T_H parameter (author's research)

At time t_0, a sensor failure is simulated. After the failure is recognized, the information update algorithm is activated. The maximum information update time (when the T_N and P_N values are reset to zero at the failure moment) does not exceed 3...5 seconds, which is acceptable for these parameters. The error in updating information when using a universal neural network scheme for restoring the helicopter TE parameters values does not exceed 0.005 according to T_N and P_N under conditions $M = 0$, $H = 0$.

It is important to ensure sufficient accuracy of the operation of a universal neural network diagram for restoring the helicopter TE parameters values under various flight conditions and when engine characteristics change during its operation [49]. To identify the helicopter TE parameters values restoring universal network diagram in the ACS sensors failures compensating task solving context, a method for correcting the calculated parameters obtained using this model was used. This method corresponds to the following mathematical relations:

$$x_{adj} = x_{calc} + \Delta x\left(x_{giv}\right), \tag{44}$$

where x_{adj} is the parameter x adjusted value; x_{calc} is the parameter x calculated value, Δx is the correction value specified as the corresponding parameter given value x_{giv} function [50].

The dependence $\Delta x(x_{giv})$ is formed and updated during the helicopter TE and its ACS operation, while according to [48], the Δx value is determined from the relation:

$$\Delta x = x_{meas} - x_{calc} + \Delta x, \tag{45}$$

where x_{meas} is the x parameter measured value.

During the helicopter TE normal operation, the difference Δx values between the n_{TC}, T_G^*, n_{FT}, etc. parameters values, are measured and calculated using the helicopter TE parameters values restoring universal network diagram, as well as the corresponding given parameters calculated values, are entered into special error arrays. At the same time, to obtain maximum accuracy, the data array formation and updating that determine the dependence $\Delta x(x_{giv})$ is calculated only in steady-state engine operating modes.

If any parameter sensor failure is detected, the neural network model training adjustments cease. The error arrays obtained to date remain unchanged and are used to generate corrections to the calculated model data [51–53].

For example, when adjusting the T_G^* information from the n_{FT} sensor can be used. In this case, the adjusted calculated T_G^* value is determined by the expression:

$$T^*_{G\,adj} = T^*_{G\,meas} + K_{T_G^*} \cdot \Delta n_{FT}, \tag{46}$$

where $\Delta n_{FT} = n_{FT\,calc}$ is the difference between the measured n_{FT} and the $n_{FT\,calc}$ dispersion in the helicopter TE parameters values restoring universal network diagram by the n_{FT}, and $K_{T_G^*}$ is a constant coefficient connecting the value T_G^* and n_{FT}.

5. Discussion

After completing the BNN training, the model performance was assessed on test data (Table 4). For this aim, the following quality metrics are used [54–58]:

- Accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{47}$$

where TP (true positive) is the true positive results number; FN (false negative) is the false negative results number; TN (true negative) is the true negative results number; FN (false negative) is the false negative results number (Table 3).

Table 3. Error matrix (author's research)

Classification		Dataset class labels	
		Positive grade label	Negative grade label
Class labels generated by the BNN	Positive grade label	*TP*	*TN*
	Negative grade label	*FP*	*FN*

- Precision:

$$Precision = \frac{TP}{TP + FP}. \tag{48}$$

- Recall:

$$Recall = \frac{TP}{TP + FN}. \tag{49}$$

- F1-score:

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{50}$$

- False positive rate:

$$FPR = \frac{FP}{FP + TN}. \tag{51}$$

- Specificity coefficient:

$$Specificity = \frac{TN}{TN + FP}. \tag{52}$$

- Precision positive rate:

$$PPV = \frac{TP}{TP + FP}. \tag{53}$$

- Negative implied value ratio:

$$NPV = \frac{TN}{TN + FN}. \tag{54}$$

- Pearson correlation coefficient:

$$\rho_{pearson} = \frac{4 \cdot \left(TP^2 + TN^2 + FP^2 + FN^2\right) - \left(TP + TN + FP + FN\right)^2}{\sqrt{\left(4 \cdot \left(TP^2 + TN^2 + FP^2 + FN^2\right) - \left(TP + TN + FP + FN\right)^2\right)^2}}. \tag{55}$$

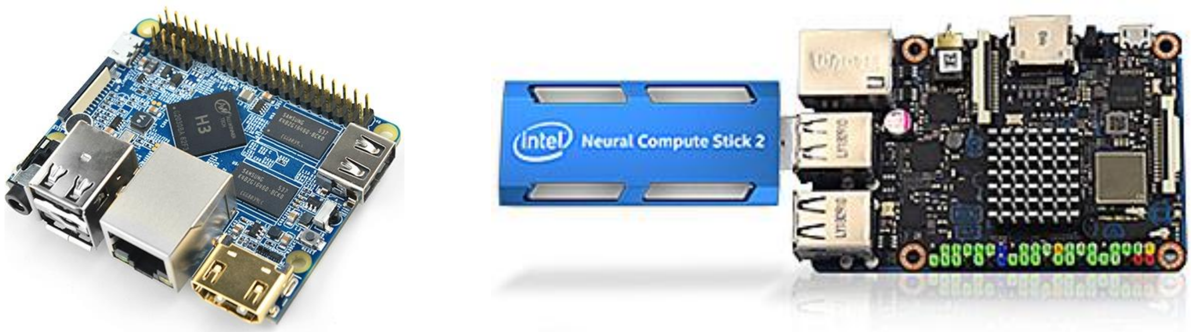
The calculating quality metrics (47)–(55) results are shown in Table 4. The obtained model results indicate high accuracy, recall and balance in detecting sensor failures and determining their location. The model minimizes both the false positives and false negatives numbers, which indicates its high efficiency in dealing with failures. The accuracy, precision, recall, F1-score, specificity, and expected positive and negative values characteristics confirm the model's high quality in identifying failures and their locations. Additionally, the high Pearson correlation coefficient demonstrates a strong linear relationship between the estimated and observed values, underscoring the model's effectiveness in predicting results. This comprehensive approach enables sensor failures accurate localization reliable identification, making the model an effective tool for diagnostic tasks in helicopter TE.

The results underscore the developed model's significant practical value for on-board implementation in helicopters. With 99.7 % accuracy, the model demonstrates exceptional capability in correctly identifying sensor failures and their locations, critical for maintaining operational safety and reliability. High precision (99.6 %) and recall (99.4 %) indicate its efficacy in minimizing both false positives and missed failures, which is crucial for preventing unnecessary alarms and ensuring accurate diagnostics. The balanced F1-score (99.5 %) further supports its robustness in both failure detection and localization. The very low false positive rate (0.3 %) and high specificity (98.9 %) reflect the model's effectiveness in accurately distinguishing between failure and non-failure scenarios. Positive predictive value (99.1 %) and negative predictive value (99.2 %) confirm its reliability in assessing sensor status and failure locations. The Pearson correlation coefficient (0.963) highlights a strong alignment between predicted and actual outcomes, affirming the model's effectiveness in practical applications. This combination of high-performance metrics makes the model highly suitable for enhancing onboard sensor failure restoring systems in helicopters, thereby improving overall safety and operational efficiency.

Table 4. Quality metrics calculation results (author's research)

Metric	Value	Conclusion on metrics
Accuracy	0.997	The model indicates obtained accuracy and a high ability to correctly determine each sensor failure probability and the failure location.
Precision	0.996	The model's resulting accuracy indicates a high ability for each sensor failure to correctly identify and its failure location to accurately determine, minimizing the false positives number.
Recall	0.994	The model's resulting recall indicates a high ability to detect almost all sensor failures and accurately determine their failure location, minimizing the number of missed failures number.
F1-score	0.995	The model resulting F1-measure indicates its high balanced ability to both detect sensor failures and accurately locate them, taking into account accuracy and recall.
FPR	0.003	The resulting false positive rate indicates that the model has an incorrectly identifying very low probability of a sensor failure or the sensor failure location, minimizing the false alarm number.
Specificity	0.989	The model-obtained specificity indicates a high ability to correctly identify the sensor failure absence and accurately identify locations without failures, minimizing the false positives number.
PPV	0.991	The model resulting positive estimated value indicates its high accuracy in detecting sensor failure and accurately locating sensor failure, minimizing the false positives number.
NPV	0.992	The model obtained negative estimated value indicates its high accuracy in the sensor failure absence identifying and the accuracy in identifying locations without failures, minimizing the false negatives number.
$\rho_{pearson}$	0.963	The Pearson correlation coefficient value indicates that there is a strong linear relationship between the estimated and observed values. This high correlation coefficient confirms the model's effectiveness in predicting the results.

Similarly to [23–25, 59], an assessment was conducted on the computational expenses associated with a neural network algorithm for the helicopter TE mathematical model implementing. This assessment was based on the Raspberry Pi NanoPi M1 Plus microprocessor controller (Fig. 15, a) and the Intel Neural Compute Stick 2 neuro controller (Fig. 15, b), with their technical specifications detailed in Table 5.



a b

Fig.15. Appearance: a – microprocessor controller Raspberry Pi NanoPi M1 Plus; b – neuro controller Intel Neural Compute Stick 2 (author's research)

Table 5. Microcontrollers technical characteristics (author's research, published in [25])

Parameter	Raspberry Pi NanoPi M1 Plus (processor– Allwinner H3) [59]	Intel Neural Compute Stick 2 (processor – Intel Atom Z3735F)
Clock frequency	1,2 GHz	1,33 GHz
Maximum execution time for an addition command	0,833 ns	0,7519 ns
Maximum execution time for a multiply instruction	8,33 ns	0,7519 ns
Maximum execution time for a division instruction	16,67 ns	0,7519 ns
Bit depth	32/64	32/64
Internal RAM capacity	1 Gb	2 Gb

Read/write memory access time is defined as:

$$T_{TAM} = T_{BAT} + T_{DBT} + T_{BCT}, \qquad (56)$$

where T_{TAM} is memory access; T_{BAT} – address bus readiness time; T_{DBT} – data bus readiness time; T_{BCT} – bus readiness time for control.

By employing the Raspberry Pi NanoPi M1 Plus microprocessor controller and the Intel Neural Compute Stick 2 neurocontroller specifications (Table 5), we derived:

$$T_{TAM}^{Intel} = 0.7519 + 0.7519 + 0.7519 = 2.2557 \text{ (ns)}, \; T_{TAM}^{Raspberry} = 0.833 + 8.33 + 16.67 = 25.833 \text{ (ns)}.$$

The memory quantity needed for a neural network in RAM is determined as follows:

$$V_{NN} = U \cdot \left(\sum_{i=1}^{3} N_i + N_c + \sum_{i=1}^{2} N_i \cdot N_{i+1} \right), \tag{57}$$

where U is the digit number required to record the neuron displacement (4 bytes); $\sum_{i=1}^{2} N_i \cdot N_{i+1}$ is the neural network weighting coefficients number; N_i is the neural network i-th layer neurons; N_c is the neural network layers number [23–25].

For a BNN with a 4–6–3 structure (Fig. 4), we obtained:

$$V_{NN} = 4 \cdot ((4 + 6 + 3) + 3 + (4 \cdot 6 + 6 \cdot 3)) = 97 \text{ (byte)}.$$

During neural network operations, various mathematical (addition, multiplication, division) and logical (comparison) operations are executed. This process involves calculating a neuron's input signals weighted sum, adding a bias, and then passing the result to the neuron's activation function. On the Intel Neural Compute Stick 2 neuro controller, activation functions are implemented in hardware, whereas on the Raspberry Pi NanoPi M1 Plus microcontroller, they are handled in software. For a sigmoid activation function, the operations include addition and division. According to [23–25], the operations executed overall number during the BNN functioning is:

$$OS = N_1 + \sum_{i=1}^{2} N_i \cdot N_{i+1} + \sum_{i=1}^{3} N_i, \tag{58}$$

$$OU = N_1 + \sum_{i=1}^{2} N_i \cdot N_{i+1} + \sum_{i=1}^{3} N_i; \tag{59}$$

$$T_f = \sum_{i=1}^{3} N_i; \tag{60}$$

where OS is the addition operations quantity; OU is the multiplication operations quantity; T_f is the activation functions number; N_i is the neural network i-th layer neurons number.

For a BNN with a 4–6–3 structure (Fig. 4), according to (58)–(60), the following is obtained:

$$OS = 4 + (4 \cdot 6 + 6 \cdot 3) + (4 + 6 + 3) = 59,$$

$$OU = 4 + (4 \cdot 6 + 6 \cdot 3) + (4 + 6 + 3) = 59,$$

$$T_f = 13.$$

Given that all operations on the Intel Neural Compute Stick 2 neuro controller, including neuron activation functions, are completed within a single clock cycle, and factoring in the bit capacity, memory access speed, and algorithm parallelism, the operations' total number is calculated as follows:

$$O_{\text{Intel}} = \frac{59}{2} + \frac{59}{2} + \frac{4}{2} + \frac{6}{2} + \frac{3}{2} = 65.5.$$

For the Raspberry Pi NanoPi M1 Plus microcontroller, the total time for executing all operations (including the time required to access data from memory) is determined as [23–25]:

$$T_{Raspberry} = O_d \cdot t_d + OS \cdot t_p + OU \cdot t_m + T_f \cdot t_p + V_{nn} \cdot t_{mem}, \tag{61}$$

where t_d is the division command execution time; t_p is the addition command execution time; t_m is the multiplication command execution time; t_{mem} is the time, required to retrieve data from memory.

Then, according to (57) for the Raspberry Pi NanoPi M1 Plus, we obtain:

$$T_{Raspberry} = 9 \cdot 16.67 + 59 \cdot 0.833 + 59 \cdot 8.33 + 13 \cdot 0.833 + 97 \cdot 0.833 = 782.277 \text{ (ns)}.$$

Considering the processor (32 bits) and operands (64 bits) differing bit sizes, the calculation time will be doubled:

$$T_{Raspberry} = 782.277 \cdot 2 = 1564.554 \text{ (ns)}.$$

For the Intel Neural Compute Stick 2 neuro controller, this time is:

$$T_{Intel} = 65.5 \cdot 0.7519 + 97 \cdot 0.7519 = 122.184 \text{ (ns)}.$$

This shows that the dedicated Intel Neural Compute Stick 2 neuro controller implements a BNN 12.8 times faster than a Raspberry Pi NanoPi M1 Plus-based computer.

Table 6 provides the results from comparing the neural network method with an analytical (piecewise linear) engine model to a polynomial regression model, which is a ninth-degree polynomial implemented using the least squares method.

Table 6. The information restoring task solving various approaches comparative analysis results (author's research, based on [25])

Models	Calculation time, μs	Occupied memory amount, bytes	Calculation error, %		
			n_{TC}	T_G^*	n_{FT}
Piecewise linear	46.3	3192	3.677	2.829	6.266
Polynomial	51.4	486	1.118	1.515	2.539
Neural network	3.6	394	0.719	0.762	0.725

Table 6 analysis shows that the helicopter TE parameters values restoring the universal network diagram more effectively solve this task: the maximum identification error does not exceed 0.762 %, which is 8.6 times less than the piecewise linear model maximum error, and 3.7 times less than the polynomial engine model.

At the same time, in [60] this authors' team developed restoring lost information method from sensors based on auto-associative neural networks. This enables the lost information to be restored during sensor failures with an accuracy exceeding 99 %. A modified training approach for an auto-associative neural network was proposed, incorporating regularization coefficients into the loss function to develop a more robust and generalizable model. This approach performs effectively on the training dataset and shows promise for yielding favourable results with new data. Carrying out additional comparative analysis with [60] is an avenue for further research.

6. Conclusions

This work's primary contribution is the helicopter turboshaft engine parameters values restoring universal network diagram development and validation, achieving up to 99.7 % accuracy, 95 % confidence in sensor failure localization, and significantly lower identification error compared to existing models, with practical implementation demonstrated through a high-performance Bayesian neural network and virtual simulation tools.

A mathematical task is formulated to determine the helicopter turboshaft engine sensor failure location occurrence and localization using a multi-class Bayesian classification model taking into account a priori knowledge and updating probabilities based on new information, which made it possible to increase the detection accuracy to 0.996...0.997 (99.6...99.7%) sensor failures, their failures localization, as well as the thermogas-dynamic parameters values restoring. The proposed approach ensures the system's timely diagnosis and correction, which helps to increase the helicopter operations' reliability and safety, the emergencies and technical downtime risk minimization.

The Bayesian neural network use is proposed as the helicopter turboshaft engine parameters values restoring universal network diagram basis, which made it possible to determine and localize sensor failure with at least 95 % confidence.

The Bayesian neural network training algorithm has been improved, which differs from the traditional one in that due to the variational approximation use to estimate the neural network parameters before distribution, the direct likelihood maximization "evidence lower bound" parameter instead, as well as the log-likelihood gradients and "evidence lower bound" calculation when updating network parameters, adding warning regularization terms, distributions and uncertainty estimates for interpreting results, ensuring balanced data accounting, effective training, which is confirmed by almost 100 % training accuracy on both training and validation samples, and the model better understanding, which is confirmed by training losses, which did not exceed 2.5 % on both the training and validation samples.

A virtual installation has been developed in the LabVIEW software package, simulating a system with three helicopter turboshaft engine sensors: a gas-generator rotor r.p.m. sensor (D-2M), a free turbine rotor speed sensor (D-1M), a gas temperature sensor in front of the compressor turbine (14 double thermocouples T-102). The developed virtual system use made it possible to conduct a computational experiment in the helicopter turboshaft engine's thermogas-dynamic parameters values restoring task solving when the corresponding sensors simulate failures. It has been experimentally confirmed that the reconstructed parameter values do not exceed 3...5 % of the experimental ones, which indicates the proposed approaches high accuracy.

The approaches proposed in the work adequacy is confirmed by the corresponding accuracy and quality metrics high values: Accuracy is 0.997, Precision is 0.996, Recall is 0.994, F1-score is 0.995, FPR is 0.003, Specificity is 0.989, PPV is 0.991, NPV is 0.992, $\rho_{pearson}$ is 0.963.

The Intel Neural Compute Stick 2 neuro controller using feasibility for implementing the developed helicopter onboard neural network system has been analytically demonstrated. The Intel Neural Compute Stick 2 executes a Bayesian neural network 12.8 times faster than a system based on the Raspberry Pi NanoPi M1 Plus.

Experimental results show that the proposed universal neural network model for gas turbine engine parameter recovery achieves a 0.782 % maximum identification error. This error is 8.6 times lower than that of the piecewise linear model and 3.7 times lower than the polynomial model.

To enhance the research effectiveness and relevance in helicopter turboshaft engine parameter restoration, future work should focus on the following areas [61]. First, further improvement is needed for the universal neural network used in parameter restoration, aiming to increase accuracy to 0.999, ensuring reliable solutions during sensor failures. Second, the multi-class Bayesian classification model with advanced mechanisms development for incorporating prior knowledge and dynamic probability updates should be pursued to improve failure localization accuracy to 0.999. Third, new methods should be introduced to increase confidence in Bayesian neural network results to 98 % and optimize training algorithms using adaptive regularization and expanded uncertainty estimation methods, achieving near-perfect accuracy on training and validation samples. Fourth, the expansion of virtual simulators in LabVIEW is recommended to test proposed approaches across a broader range of failure scenarios. Finally, exploring the integration of developed models with neuro controllers like Intel Neural Compute Stick 2 is essential for enhancing performance and real-time processing speed.

References

[1] H. Sheng, Q. Chen, J. Li, W. Jiang, Z. Wang, Z. Liu, T. Zhang, and Y. Liu, "Research on dynamic modeling and performance analysis of helicopter turboshaft engine's start-up process", *Aerospace Science and Technology*, vol. 106, 106097, 2020. https://doi.org/10.1016/j.ast.2020.106097

[2] N. Trapani, and L. Longo, "Fault Detection and Diagnosis Methods for Sensors Systems: a Scientific Literature Review", *IFAC-PapersOnLine*, vol. 56, issue 2, pp. 1253–1263, 2023. https://doi.org/10.1016/j.ifacol.2023.10.1749

[3] A. Yin, Z. Sun, and J. Zhou, "Hypergraph construction using Multi-Sensor for helicopter Tail-Drive system fault diagnosis", *Measurement*, vol. 231, 114586, 2024. https://doi.org/10.1016/j.measurement.2024.114586

[4] S. de Gioia, F. Adamo, F. Attivissimo, D. Lotano, and A. Di Nisio, "A design strategy for performance improvement of capacitive sensors for in-flight oil-level monitoring aboard helicopters", *Measurement*, vol. 208, 112476, 2023. https://doi.org/10.1016/j.measurement.2023.112476

[5] S.M. Schlanbusch, and J. Zhou, "Adaptive predictor-based control for a helicopter system with input delays: Design and experiments", *Journal of Automation and Intelligence*, vol. 3, issue 1, pp. 50–56, 2024. https://doi.org/10.1016/j.jai.2024.02.001

[6] J. Hu, N. Hu, Y. Yang, L. Zhang, and G. Shen, "Nonlinear dynamic modeling and analysis of a helicopter planetary gear set for tooth crack diagnosis", *Measurement*, vol. 198, 111347, 2022. https://doi.org/10.1016/j.measurement.2022.111347

[7] K. Sun, A. Yin, and S. Lu, "Domain distribution variation learning via adversarial adaption for helicopter transmission system fault diagnosis", *Mechanical Systems and Signal Processing*, vol. 215, 111419, 2024. https://doi.org/10.1016/j.ymssp.2024.111419

[8] S. Vladov, R. Yakovliev, O. Hubachov, K. Mykolenko, S. Drodova, and Y. Stushchanskyi, "Neural Network Method for Helicopters Turboshaft Engines Dynamic Efficiency Monitoring", in: Proceedings of the 2023 IEEE 4 th KhPI Week on Advanced Technology, Kharkiv, Ukraine, October 02–06, 2023. pp. 160–165. https://doi.org/10.1109/KhPIWeek61412.2023.10312883

[9] M. Kordestani, M. Mousavi, A. Chaibakhsh, M. E. Orchard, K. Khorasani, and M. Saif, "A New Compressor Failure Prognostic Method Using Nonlinear Observers and a Bayesian Algorithm for Heavy-Duty Gas Turbines", IEEE Sensors Journal, vol. 23, issue 4, pp. 3889–3900, 2023. https://doi.org/10.1109/JSEN.2022.3233585

[10] T. Lu, and S. Hou, "A Two-Layered Malware Detection Model Based on Permission for Android", in: Proceedings of the 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, August 18–20, 2018. https://doi.org/10.1109/CCET.2018.8542215

[11] X. Zhu, and D. Li, "Robust fault estimation for a 3-DOF helicopter considering actuator saturation", *Mechanical Systems and Signal Processing*, vol. 155, 107624, 2021. https://doi.org/10.1016/j.ymssp.2021.107624

[12] J. Lan, R. J. Patton, and E. Punta, "Fault-Tolerant Tracking Control for A 3-DOF Helicopter with Actuator Faults and Saturation", *IFAC-PapersOnLine*, vol. 50, issue 1, pp. 5250–5255, 2017. https://doi.org/10.1016/j.ifacol.2017.08.465

[13] J. Y. Yassuda, C. M. Agulhari, and E. R. P. da Silva, "Sampled-data robust control of a 2-DoF helicopter modeled using a quasi-LPV framework", *Control Engineering Practice*, vol. 145, 105870, 2024. https://doi.org/10.1016/j.conengprac.2024.105870

[14] W. El Khatiri, R. Cherif, K. El Bikri, and N. Atalla, "Experimental study of component-based transfer path analysis hybrid methods applied to a helicopter", *Applied Acoustics*, vol. 210, 109431, 2023. https://doi.org/10.1016/j.apacoust.2023.109431

[15] Z. Gu, Q. Li, S. Pang, W. Zhou, J. Wu, and C. Zhang, "Turbo-shaft engine adaptive neural network control based on nonlinear state space equation", *Chinese Journal of Aeronautics*, vol. 37, issue 4, pp. 493–507, 2024. https://doi.org/10.1016/j.cja.2023.08.012

[16] B. Li. and Y.-P. Zhao, "Group reduced kernel extreme learning machine for fault diagnosis of aircraft engine", *Engineering Applications of Artificial Intelligence*, vol. 96, 103968, 2020. https://doi.org/10.1016/j.engappai.2020.103968

[17] S. Pang, Q. Li, and B. Ni, "Improved nonlinear MPC for aircraft gas turbine engine based on semi-alternative optimization strategy", *Aerospace Science and Technology*, vol. 118, 106983, 2021. https://doi.org/10.1016/j.ast.2021.106983

[18] S. Vladov, Y. Shmelov, and R. Yakovliev, "Modified Method of Identification Potential Defects in Helicopters Turboshaft Engines Units Based on Prediction its Operational Status", in: Proceedings of the 2022 IEEE 4th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, October 20–22, 2022, pp. 556–561. https://doi.org/10.1109/MEES58014.2022.10005605

[19] S.S. Talebi, A. Madadi, A.M. Tousi, and M. Kiaee, "Micro Gas Turbine fault detection and isolation with a combination of Artificial Neural Network and off-design performance analysis", Engineering Applications of Artificial Intelligence, vol. 113, 104900, 2022. https://doi.org/10.1016/j.engappai.2022.104900
[20] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, and S. Drozdova, "Neural Network Method for Helicopters Turboshaft Engines Working Process Parameters Identification at Flight Modes", in: Proceedings of the 2022 IEEE 4th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, October 20–22, 2022, pp. 604–609. https://doi.org/10.1109/MEES58014.2022.10005670
[21] Y. Shen, and K. Khorasani, "Hybrid multi-mode machine learning-based fault diagnosis strategies with application to aircraft gas turbine engines", *Neural Networks*, vol. 130, pp. 126–142, 2020. https://doi.org/10.1016/j.neunet.2020.07.001
[22] J. Zeng, and Y. Cheng, "An Ensemble Learning-Based Remaining Useful Life Prediction Method for Aircraft Turbine Engine", *IFAC-PapersOnLine*, vol. 53, issue 3, pp. 48–53, 2020. https://doi.org/10.1016/j.ifacol.2020.11.009
[23] V. Vasiliev, S. Zhernakov, and I. Muslukhov, "On-board algorithms for monitoring gas turbine engine parameters based on neural network technology", *Bulletin of USATU*, vol. 12, No. 1 (30), pp. 61–74, 2009.
[24] S. Zhernakov, "Algorithms for control and diagnostics of aviation gas turbine engines in the conditions of on-board implementation based on neural network technology", *Bulletin of USATU*, vol. 14, No. 3 (38), pp. 42–56, 2010.
[25] S. Vladov, K. Kotliarov, S. Hrybanova, O. Husarova, and L. Chyzhova. "On-board information restoring method in case of failure of one of the sensors of the aircraft engine TV3-117 based on neural network technologies", *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*, issue 6/2019 (119), pp. 91–98, 2019. https://doi.org/10.30929/1995-0519.2019.6.91-98
[26] S. Vasiliev, R. Badamshin, S. Valeev, V. Vasiliev, V. Gvozdev, M. Guzairov, S. Zhernakov, B. Ilyasov, S. Kusimov, R. Munasypov, Y. Raspopov, A. Frid, and L. Chernyakhovskaya, Intelligent control and monitoring systems for gas turbine engines, pp. 327–390, 2008.
[27] O. Dehtiarov, O. Zaporozhets, and T. Ovcharova, "Identification of nonlinear transform function using artificial neural network", *Ukrainian Metrological Journal*, no. 2, pp. 4–8, 2013.
[28] X. Liu, Y. Chen, L. Xiong, J. Wang, C. Luo, L. Zhang, and K. Wang, "Intelligent fault diagnosis methods toward gas turbine: A review", *Chinese Journal of Aeronautics*, vol. 37, issue 4, pp. 93–120, 2024. https://doi.org/10.1016/j.cja.2023.09.024
[29] J. Bill, B.A. Cox, and L. Champagne, "A comparison of quaternion neural network backpropagation algorithms", *Expert Systems with Applications*, vol. 232, 120448, 2023. https://doi.org/10.1016/j.eswa.2023.120448
[30] S. Vladov, R. Yakovliev, O. Hubachov, J. Rud, and Y. Stushchanskyi, "Neural Network Modeling of Helicopters Turboshaft Engines at Flight Modes Using an Approach Based on "Black Box" Models", *CEUR Workshop Proceedings*, vol. 3624, pp. 116–135, 2024.
[31] S. Babichev, V. Lytvynenko, J. Skvor, and J. Fiser, "Model of the objective clustering inductive technology of gene expression profiles based on SOTA and DBSCAN clustering algorithms", Advances in Intelligent Systems and Computing, vol. 689, pp. 21–39, 2018. https://doi.org/10.1007/978-3-319-70581-1_2
[32] V. Lytvynenko, D. Nikytenko, M. Voronenko, N. Savina, and O. Naumov, "Assessing the Possibility of a Country's Economic Growth Using Dynamic Bayesian Network Models", in: Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, September 23–26, 2020, pp. 36–39. https://doi.org/10.1109/CSIT49958.2020.9321995
[33] M. Voronenko, D. Nikytenko, Jan Krejci, N. Savina, V. Lytvynenko, "Assessing the Possibility of a Country's Economic Growth Using Static Bayesian Network Models", *CEUR Workshop Proceedings*, vol. 2608, pp. 462–473, 2020.
[34] M. Impraimakis, "A Kullback–Leibler divergence method for input–system–state identification", *Journal of Sound and Vibration*, vol. 569, 117965, 2024. https://doi.org/10.1016/j.jsv.2023.117965
[35] L. Karpukov, V. Voskoboynik, G. Kozina, O. Parshyna, Y. Parshyn, and A. Shapoval, "Pulses Dispersion Distortion Modeling in a Microstrip Line", in: Proceedings of the 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, October 19–23, 2023. https://doi.org/10.1109/CSIT61576.2023.10324298
[36] J. Li, S. Tong, Z. Tong, H. Li, F. Cong, W. Cheng, and G. Dong, "Investigating the vibration response and modulation mechanism for health monitoring of wind turbine planetary gearboxes using a tribodynamics-based analytical model", *Measurement Science and Technology*, vol. 34, number 3, 035119, 2022. https://doi.org/10.1088/1361-6501/aca927
[37] S. Vladov, Y. Shmelov, R. Yakovliev, and M. Petchenko, "Neural Network Method for Parametric Adaptation Helicopters Turboshaft Engines On-Board Automatic Control", *CEUR Workshop Proceedings*, vol. 3403, pp. 179–195, 2023.
[38] S. Vladov, R. Yakovliev, O. Hubachov, and J. Rud, "Neuro-Fuzzy System for Detection Fuel Consumption of Helicopters Turboshaft Engines", *CEUR Workshop Proceedings*, vol. 3628, pp. 55–72, 2024.
[39] S. Vladov, R. Yakovliev, M.Bulakh, and V. Vysotska, "Neural Network Approximation of Helicopter Turboshaft Engine Parameters for Improved Efficiency", *Energies*, vol. 17, issue 9, 2233, 2024. https://doi.org/10.3390/en17092233
[40] S. Vladov, Y. Shmelov, and R. Yakovliev, "Helicopters Aircraft Engines Self-Organizing Neural Network Automatic Control System", *CEUR Workshop Proceedings*, vol. 3137, pp. 28–47, 2022. https://doi.org/10.32782/cmis/3137-3
[41] S. Vladov, Y. Shmelov, R. Yakovliev, M. Petchenko, and S. Drozdova, "Helicopters Turboshaft Engines Parameters Identification at Flight Modes Using Neural Networks", in: Proceedings of the IEEE 17th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, November 10–12, 2022, pp. 5–8. https://doi.org/10.1109/CSIT56902.2022.10000444
[42] F.S. Corotto, "Appendix C - The method attributed to Neyman and Pearson", *Wise Use of Null Hypothesis Tests*, pp. 179–188, 2023. https://doi.org/10.1016/B978-0-323-95284-2.00012-4
[43] F.V. Motsnyi, "Analysis of Nonparametric and Parametric Criteria for Statistical Hypotheses Testing. Chapter 1. Agreement Criteria of Pearson and Kolmogorov", *Statistics of Ukraine*, no. 4'2018 (83), pp. 14–24, 2018. https://doi.org/10.31767/su.4(83)2018.04.02
[44] S. Babichev, J. Krejci, J. Bicanek, and V. Lytvynenko, "Gene expression sequences clustering based on the internal and external clustering quality criteria", in: Proceedings of the 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, September 05–08, 2017. https://doi.org/10.1109/STC-CSIT.2017.8098744
[45] H. Khebbache, M. Tadjine, S. Labiod, and A. Boulkroune. "Adaptive sensor-fault tolerant control for a class of multivariable

uncertain nonlinear systems", *ISA Transactions*, vol. 55, pp. 100–115, 2015. https://doi.org/10.1016/j.isatra.2014.10.001

[46] Z.B. Shi, T. Liu, and D. Sun, "Reconfiguration for Sensor Failure of Aero-Engine Electronic Control System Based on the MRAC", Applied Mechanics and Materials, vol. 602–605, pp. 1367–1371, 2014. https://doi.org/10.4028/www.scientific.net/AMM.602-605.1367

[47] S. Vladov, L. Scislo, V. Sokurenko, O. Muzychuk, V.Vysotska, S.Osadchy, and A. Sachenko, "Neural Network Signals Integration from Thermogas-dynamic Parameters Sensors for Helicopters Turboshaft Engines at Flight Operation Conditions", *Sensors*. vol. 24, issue 13, 4246, 2024. https://doi.org/10.3390/s24134246

[48] F. Golberg, O. Gurevich, and A. Petukhov, "Mathematical model of an ACS GTE in a gas turbine engine to improve reliability and control quality", *Electronic journal "Proceedings of MAI"*, no. 58, pp. 1–13, 2012.

[49] M.A. Contreras-Cruz, J.P. Ramirez-Paredes, U.H. Hernandez-Belmonte, and V. Ayala-Ramirez, "Vision-Based Novelty Detection Using Deep Features and Evolved Novelty Filters for Specific Robotic Exploration and Inspection Tasks", *Sensors*, vol. 19, issue 13, 2965, 2019. https://doi.org/10.3390/s19132965

[50] O.A. Lysenko, "Neural Network NARX Flow Identifier for a Pumping Unit With an Induction Motor", in: Proceedings of the 2023 Dynamics of Systems, Mechanisms and Machines (Dynamics), November 14–15, 2023. https://doi.org/10.1109/Dynamics60586.2023.10349512

[51] A. Sohoub, S. Sati, and M. Eshtawie, "Optimal Cluster Size for Wireless Sensor Networks", *International Journal of Wireless and Microwave Technologies (IJWMT)*, vol. 13, no. 1, pp. 36–44, 2023. https://doi.org/10.5815/ijwmt.2023.01.04

[52] H.H. Shia, M.A. Tawfeeq, and S.M. Mahmoud, "High Rate Outlier Detection in Wireless Sensor Networks: A Comparative Study", *International Journal of Modern Education and Computer Science*, vol. 11, no. 4, pp. 13–22, 2019.

[53] S. Gudla, and K.N. Rao, "Reliable Data Delivery Using Fuzzy Reinforcement Learning in Wireless Sensor Networks", *International Journal of Computer Network and Information Security*, vol. 15, no. 4, pp. 96–107, 2023.

[54] J.-H. Li, X.-Y. Gao, X. Lu, and G.-D. Liu, "Multi-Head Attention-Based Hybrid Deep Neural Network for Aeroengine Risk Assessment", *IEEE Access*, vol. 11, pp. 113376–113389, 2023. https://doi.org/10.1109/ACCESS.2023.3323843

[55] S. Yi, Z. Chen, L. Yi, and F. She, "CAS: Breast cancer diagnosis framework based on lesion region recognition in ultrasound images", *Journal of King Saud University - Computer and Information Sciences*, vol. 35, issue 8, 101707, 2023. https://doi.org/10.1016/j.jksuci.2023.101707

[56] Z. Hu, O. Shkurat, and M. Kasner, "Grayscale Image Colorization Method Based on U-Net Network", *International Journal of Image, Graphics and Signal Processing*, vol. 16, no. 2, pp. 70–82, 2024.

[57] Z. Hu, I. Dychka, K. Potapova, and V. Meliukh, "Augmenting Sentiment Analysis Prediction in Binary Text Classification through Advanced Natural Language Processing Models and Classifiers", *International Journal of Information Technology and Computer Science*, vol. 16, no. 2, pp. 16–31.

[58] R. Peleshchak, V. Lytvyn, I. Peleshchak, D. Dudyk, and D. Uhryn 2, "Data Clustering by Chaotic Oscillatory Neural Networks with Dipole Synaptic Connections", *International Journal of Modern Education and Computer Science*, vol. 16, no. 3, pp. 27–38, 2024.

[59] V. Sahana, R. Shashidhar, R. Bindushree, and A.N. Chandana, "IOT Based Burglar Detection and Alarming System Using Raspberry Pi", *International Journal of Engineering and Manufacturing*, vol. 13, no. 6, pp. 23–37, 2023.

[60] S. Vladov, R. Yakovliev, V. Vysotska, M. Nazarkevych, and V. Lytvyn, "Restoring lost information method from complex dynamic object sensors based on auto-associative neural networks", *Applied System Innovation*, vol. 7, issue 3, 53, 2024. https://doi.org/10.3390/asi7030053

[61] Y. Shmelov, S. Vladov, Y. Klimova, and M. Kirukhina, "Expert system for identification of the technical state of the aircraft engine TV3-117 in flight modes", in: Proceedings of the System Analysis & Intelligent Computing : IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, 08–12 October, 2018, pp. 77–82. https://doi.org/10.1109/SAIC.2018.8516864

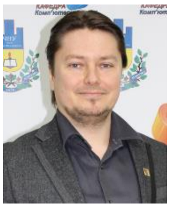
Authors' Profiles

Serhii Vladov is a Head of the Department of Scientific Work Organization and Gender Issues at Kremenchuk Flight College of Kharkiv National University of Internal Affairs. He is a Candidate in Technical Sciences. His current research and the fields of specialization – development of applied intelligent systems (including intelligent automatic control systems) of complex dynamic objects, in particular, helicopter turboshaft engines. He has nearly 200 scientific papers, among them more than 30, which are indexed in the Scopus scientific database.

Ruslan Yakovliev is a director of Kremenchuk Flight College of Kharkiv National University of Internal Affairs. He is the organizer of the training of flight and engineering personnel for units of the Ministry of Internal Affairs of Ukraine. He is the organizer of the training of flight and engineering personnel for units of the Ministry of Internal Affairs of Ukraine. His research interests are related to improving helicopter safety through the introduction of intelligent information technologies.

Victoria Vysotska is Associate Professor at the Information Systems and Networks Department of Lviv Polytechnic National University. She defended her Doctoral degree in Technical Science, speciality in «structural, applied and mathematical linguistics» in 2023 on the topic "Analysis and synthesis of computational linguistic systems for processing Ukrainian textual content" Also, she received a PhD degree in Information Technologies from Lviv Polytechnic National University in 2014. She has currently published more than 500 publications. Her main research interests are focused on the identification of disinformation/fakes/propaganda, detection of sources of disinformation and inauthentic behaviour (bots) of coordinated groups based on artificial intelligence, NLP, computer linguistics, data science, system analysis, information technologies, machine learning cyber security.

Dmytro Uhryn graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is a Doctor of Technical Sciences and associate professor at Yuriy Fedkovych Chernivtsi National University. He has currently published more than 140 publications. His research interests are data mining, information technologies for decision support, swarm intelligence systems, and industry-specific geographic information systems. His areas of scientific interest are decision support information technologies, swarm intelligence systems and industry geoinformation systems.



Yuriy Ushenko is a Professor at the Computer Science Department, Chernivtsi National University, Chernivtsi, Ukraine. Research Interests: Data Mining and Analysis, Computer Vision and Pattern Recognition, Optics & Photonics, Biophysics. His research interests include information systems design, data mining, pattern recognition and digital image processing, artificial neural networks, laser polarimetry and interferometry.

How to cite this paper: Serhii Vladov, Ruslan Yakovliev, Victoria Vysotska, Dmytro Uhryn, Yuriy Ushenko, "Universal On-board Neural Network System for Restoring Information in Case of Helicopter Turboshaft Engine Sensor Failure", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.4, pp.65-87, 2024. DOI:10.5815/ijcnis.2024.04.05