UDC 621.43, 004.032.26

METHOD FOR ANALYZING INPUT DATA FROM GEAR VIBRATIONS

Shalimov O. Y. – Student of the Department of Automation and control in technical systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Moskalchuk O. O. – Student of the Department of Automation and control in technical systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Yevseienko O. M. – PhD, Associate Professor, Associate Professor of the Department of Automation and control in technical systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

ABSTRACT

Context. The paper considers the problem of analyzing large data vectors for analyzing helicopter engine performance. This issue is crucial for improving the reliability and efficiency of modern aviation technologies.

Objective. To create a method for analyzing engine vibration data to achieve accurate classification of engine states based on vibration signals**.**

Method. The input data is analyzed, and a decision is made to create a neural network that is trained to recognize the class of the input vector. The neural network can work immediately and be configured for further training based on similar data. The program was implemented using a classical neural network method. The optimal weights and offsets are calculated with derivatives to minimize the loss function. The stochastic gradient descent (SGD) algorithm was used for optimization, and different activation functions were tested to find the best configuration. Choosing the right activation functions ensured maximum performance.

Results. The graphs of the input vectors show that vectors from the first class had more peaks, which helped facilitate classification. After applying this method, the accuracy was about 70–75%, which was insufficient for the task. To improve this, we enhanced the model structure and reconfigured the activation functions. With the new method, the neural network can classify the input vector with 100% accuracy.

Conclusions. This study presents an approach to analyzing engine vibration data for assessing performance. The scientific novelty lies in adapting a multilayer perceptron (MLP) for classifying vibration signals. The research shows that high accuracy can be achieved without deep architectures by optimizing the MLP. This method is universally applicable, eliminating additional model adaptation costs, which is crucial for industrial use. The practical significance is demonstrated through software and experiments, proving the effectiveness of the MLP for performance monitoring when model parameters and activation functions are properly adjusted.

KEYWORDS: 1D convolutional neural networks**,** multilayer perceptron, stochastic gradient descent, loss function, engine vibration analysis, helicopter performance, signal classification, neural network, machine learning.

ABBREVIATIONS

CNN is a convolutional neural network
FCNN is a fully connected neural network;
ICA is an independent component analysis;
MLP is a multilayer perceptron;
NN is a neural network;
PCA is a principal component analysis;
RAM is a random-access memory;
SGD is a stochastic gradient descent;
ZSL is a zero-shoot learning.

NOMENCLATURE

$a^{(l)}$ is an activation of the previous layer (or the input vector if it is the first layer);
$b^{(l)}$ is an offset vector for the l-th layer;
f is an filter size;
i is an iteration index;
L is a number of the layer on which the calculation is performed;
R is a set of numbers in a single vector;
s is a step size;
$W^{(l)}$ is a matrix of weights for the l-th layer;
X a set of vectors with input data;
x a value of the input layer;
y this is an outer layer;
$Z^{(l)}$ is an intermediate layer;
Δ is a gradient of a function;
δ is an error for layer l;
Θ is an extracted feature set
λ is a regularization hyperparameter;
$\sigma(x)$ is a sigmoid function.

INTRODUCTION

Any working component has a so-called 'service life'. After a certain amount of time or a specific period of operation, the component requires a mandatory technical inspection to ensure continued reliability. For instance, in the context of a helicopter flight, safety is paramount. To guarantee flight safety, we must implement a robust method of training a neural network, which should ultimately provide accurate assessments of engine conditions.

Predicting whether an inspection is truly necessary before examining a component is a significant challenge. Vibrations can be read and analyzed, but interpreting them is not straightforward. This task is inherently complex, making it difficult to achieve reliable results through traditional methods. Therefore, we need an alternative approach to analyze vibration data in a way that is both efficient and accessible.

Modern technology offers such an alternative in the form of neural networks. Neural networks can process large volumes of data at high speeds and handle multitasking beyond the capabilities of human or traditional computer algorithms. However, we cannot simply use any

DOI 10.15588/1607-3274-2025-2-13

neural network and expect it to perform the task effectively. A neural network must be carefully trained to understand the specifics of the problem it is intended to solve. Only with appropriate training can it provide the accurate and reliable results we need.

The object of study is the process of training a neural network using the Zero Shot method. ZSL for neural networks allows models to classify objects they have not seen during training. This is achieved by using semantic information such as object descriptions or attributes to generalize knowledge into new classes.

The purpose of the work is to use the available methods to classify vectors into two classes and automate this process. This is necessary because you have to work with large amounts of data, so you need to work with methods that are suitable for working with big data. The goal is to implement a method for classifying data samples.

Given a large set of data, namely 1158 sets, and a file that stores shortcuts (classes) for each of the vectors, it is necessary to develop a tool that, using the given data, will learn to diagnose which files are good and which have a damaged component within a given data sample. Class 1 represents the correct operation of the engine, while Class 0 indicates that the engine has a malfunction.

1 PROBLEM STATEMENT

We have a dataset of 1158 vectors taken from the helicopter gear and a file containing information about the class of each vector. Construct an algorithm that determines the corresponding label Y (class 1 or 0) for a new vector $x \in R$. It will calculate the necessary parameters for the vector classification algorithm. Increase the classification accuracy and minimize the loss function.

2 REVIEW OF THE LITERATURE

Traditional fault detection contains three main steps: data acquisition, features extraction, fault detection and classification [1]. As a common practice the data is collected with the help of numerous sensors. Feature extraction is typically implemented through linear or nonlinear transformations and data decomposition. Linear methods include PCA [2] and ICA [3]. Nonlinear techniques, such as kernel-based methods, have been shown to perform better than their linear counterpart, PCA [4]. Reducing the size or dimensionality of input data is also useful, as it is difficult to work with large signals or images and because training time increases. Straightforward methods, such as max pooling, have been actively used to address this problem [5].

In recent years, 1D convolutional neural network has been actively used to extract the most significant features from raw data for vibration-based fault detection. For example, Hsueh et al. [6] proposed a method in which they used a wavelet transform to convert a signal into a two-dimensional grayscale image and then trained a deep CNN on it to extract robust features. This data is then used to train a classification neural network that determines the health status of the system.

Another example of sophisticated feature extraction is presented in [7], where 1D vibration signals are transformed into 2D time-frequency images. While this transformation adds an additional layer of computation and complexity, the classification results are significantly better than those obtained from simple max pooling.

An uncommon solution for fault detection can be a simple MLP [8] or FCNN. However, it is important to note that the training time can be unacceptable, as an FCNN must be able to process large amounts of data [9]. As mentioned previously, many data extraction and decomposition techniques exist that help reduce these limitations.

3 MATERIALS AND METHODS

The method has been designed and tailored to address the problem. The first step was to accept signals and distribute them into vectors according to the peaks. The second step is feature extraction. Feature extraction helps reduce the dimensionality of input data while retaining the information that has the most significant impact on solving the task. A large data vector is compressed into a smaller one to save resources and speed up the process using the formulas below in this section [9]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \exp(-2\pi i \frac{kn}{N}), k = 0,1,2,\ldots,N-1, \quad (1)$$

$$\theta : R^n \to R^m, m = 469, \quad (2)$$

$$\text{MaxPool}(x \mid f,s)_i = \max(x_{is:is+f}), \quad (3)$$

$$\text{MaxPool}(x \mid 100,100)_i = \max(x_{i\cdot 100:i\cdot 100+100}). \quad (4)$$

In the third step, we start training our neural network. For the sake of simplicity and clarity, we provide an algorithm for one vector. However, in the problem, we use it for 100 files and divide them into 10 mini-batches. We run the neural network once with randomly set parameters for weights and offsets. In general, it looks like this: For each hidden layer, activation is performed using the sigmoid function according to the formula:

$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)}). \quad (5)$$

For the layer, denoted L, softmax is used to calculate the probabilities using the following formulas:

$$z(L) = (W^{(l)} a^{(l)} + b^{(l)}), \quad (6)$$

$$y = soft\max z(L), \quad (7)$$

where the function softmax is determined by the formula:

$$soft\max(z(L)) = \frac{e^{z_i}}{\sum_j e^{z_i} j}. \quad (8)$$

Backward pass: we calculate an error of the output layer using the cross-entropy loss function [13] with softmax by the formula:

$$\sigma^{(l)} = a^{(l)} - y. \quad (9)$$

DOI 10.15588/1607-3274-2025-2-13

Gradients for the displacement weights of the output layer L are calculated using the formulas:

$$\nabla b(L) = \sigma(L), \tag{10}$$

$$\nabla w^{(L)} = \delta^{(L)} \cdot (a^{(L-1)})^T. \tag{11}$$

Neural network regularization helps to avoid over-training, in this case, adding a penalty for large values of model weights [11]. This prevents the model from "re-learning" [12] on training data.

$$L = L_0 + \lambda \|W\|^2. \tag{12}$$

Recursive calculation of errors for previous layers:

For each layer l (moving from the original layer to the first layer):

Define the derivative of the sigmoid function by the formula:

$$\sigma`(z) = \sigma(z)(1 - \sigma(z)). \tag{13}$$

The error for layer l is calculated through the error of the next layer and the derivative of the activation function by the formula:

$$\delta^{(L)} = ((W^{(L+1)})^T \circ \sigma`(z^{(L)}), \tag{14}$$

where $\circ$ denotes an elementary multiplication (Adamar multiplication)

Gradients for weights and offsets in hidden layers: For each layer l by the formulas:

$$\nabla b^{(1)} = \sigma^{(l)}, \tag{15}$$

$$\nabla w^{(L)} = \delta^{(L)} \cdot (a^{(L-1)})^T. \tag{16}$$

Formulas 15 and 16 will be implemented later within this function, named **_backprop**. This function **_backprop** calculates the gradients of weights and biases for each layer, as well as the value of the loss function for the current example.

4 EXPERIMENTS

After analyzing the data, it was clear that using full vectors for learning would not be completely appropriate. The file that stored the data had weight while the program was running, and this amount of information was stored in RAM, making the training of such a neural network very resource-intensive and inefficient for performance.

To gain a better understanding, we built graphs and noticed that the graphs for different classes were distinct. This is shown in Figure 1.

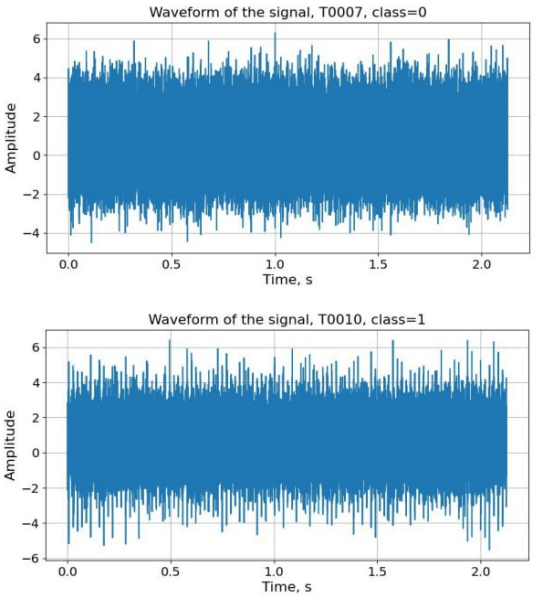

Figure 1 – Example of peaks for different classes

The graphs corresponding to class = "1" had more peaks, which were used to try to classify the vectors. First, using the discrete Fourier transform [10] with the formula (1), we converted the signal into its amplitude. We also noticed a difference in the amplitude graphs, which is shown in Figure 2.

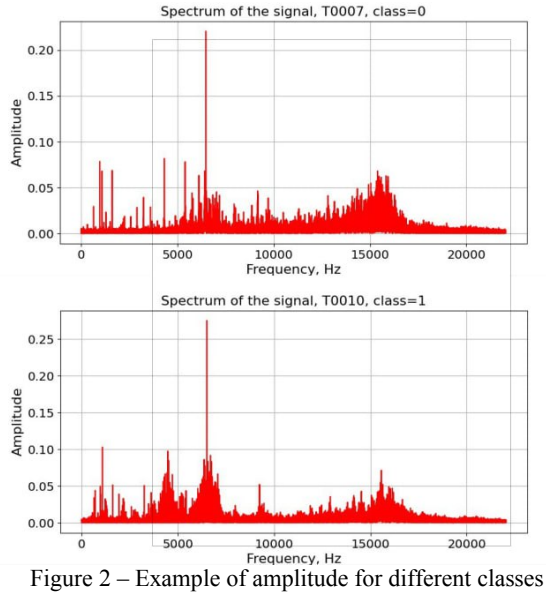

Figure 2 – Example of amplitude for different classes

It was decided to take the modulus of the amplitude vector, shift it by 0.1 values down to account for values less than zero, and retain only the maximum peaks. This is shown in Figure 3.

DOI 10.15588/1607-3274-2025-2-13

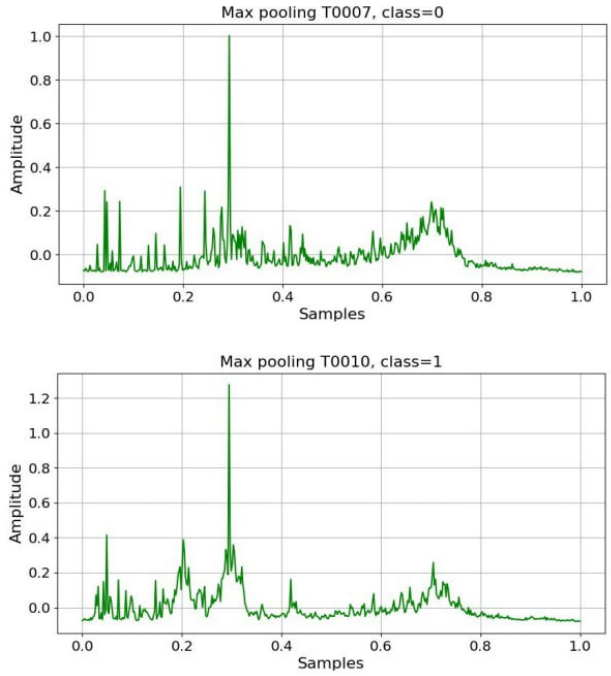

Figure 3 – Example of modulus of the amplitude for different classes

We clarified that it is possible to classify vectors by peaks. To facilitate the process, we decided to compress the data. Such manipulations with the input data allowed the input vector to have not 93.752 values, but 469 by formulas (2–4). This did not affect the training of the neural network but helped reduce the time and resources needed for training. Next, the multilayer perceptron method was used, which is a common instance of the error propagation algorithm.

The input data is permuted by the Hadamard multiplication, using a randomly generated weight vector, which is combined with the input vector using a specific displacement vector (also randomly generated). This process is repeated a specific number of times, producing an output value, which is compared with the standard (given to us by class 1 or 0). We then calculate the error, and from this error, we proceed in reverse order, gradually calibrating weights and biases. Repeating this over a certain number of files generates specific weight and bias vectors. This dataset will be used to train the neural network, which, using these vectors and the multilayer perceptron method, will be able to classify a vector with a certain probability.

5 RESULTS

The results obtained from the analysis of engine vibration data using the proposed multilayer perceptron method show promising improvements in classification accuracy. The accuracy of the initial model was 70–75%, but after further adjustments to the model structure and optimization of the activation functions, the accuracy reached nearly 100%. This significant enhancement underscores the effectiveness of the chosen approach in classifying vibration signals, which is essential for performance monitoring in helicopter engines.

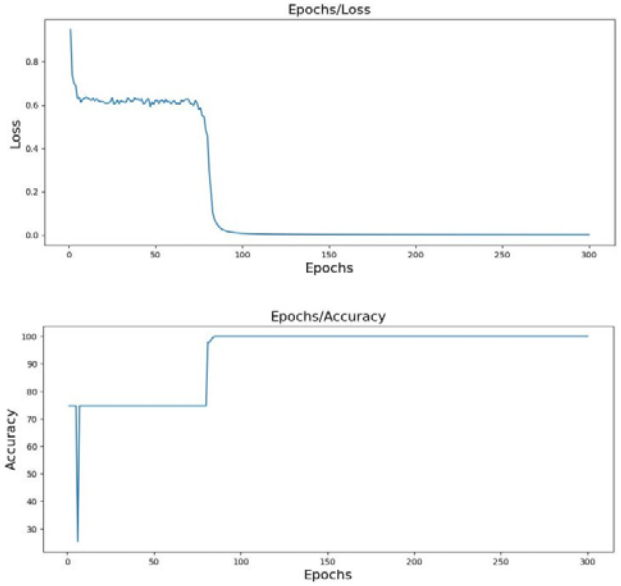

Figure 4 – Results of the learning neural network

The graphs presented in the study clearly illustrate the improvement in model accuracy after optimization. One graph shows a comparison of the accuracy between the initial and optimized models, highlighting a significant increase in accuracy after adjusting the network structure and selecting optimal activation functions. Another graph visualizes the learning process, where the reduction in error on both the training and test datasets signals good model consistency. Additionally, when comparing with other approaches, such as convolutional neural networks, it is evident that our model, despite its simplicity, yields competitive results with lower computational costs. These visualizations help to better understand the effectiveness of the applied method and emphasize its potential for real-world applications, where processing speed and resource efficiency are critical.

The comparison with similar studies reveals both similarities and differences in the methodology and results. For example, Hsueh et al. [6] used a wavelet transform followed by a CNN for fault detection. While this approach also provided robust feature extraction, it added complexity and computational overhead. In contrast, our approach using MLPs, despite its simplicity, demonstrated high accuracy without requiring deep architectures. This is particularly beneficial for real-time applications where computational resources and processing speed are critical.

6 DISCUSSION

As is evident from Figure 4, increasing the number of epochs in the formed sample improves classification accuracy. The time spent on learning is optimized. Even an increase in input data has almost no effect on the speed of diagnosis, maintains acceptable accuracy, and does not significantly increase the training time.

There is an increase in speed compared to similar methods, such as 1D Convolutional Neural Networks. Compared to the original model, the accuracy improve-

DOI 10.15588/1607-3274-2025-2-13

ment is nearly 100%. This confirms the feasibility of using the proposed method.

It should also be noted that the training method qualitatively affects the speed and effectiveness of training compared to similar methods. This has made it possible to improve the training process itself.

The limitations of the study primarily stem from the necessity of adjusting model parameters (e.g., weights and activation functions), which can introduce challenges in scaling the model to larger datasets. However, the proposed method's ability to work with a limited number of layers without compromising accuracy is a key advantage in real-world applications, where training time and computational efficiency are often constrained.

Practical applications of this method are significant. The approach demonstrated here could be directly applied to performance monitoring and predictive maintenance in aviation systems, where it could reduce downtime and prevent potential failures. This has important implications for both safety and cost-effectiveness in the aviation industry.

While the study provides a solid foundation for vibration-based fault detection, further research is required to enhance the model's adaptability to varying conditions and sensor configurations. Exploring hybrid models that combine the strengths of both convolutional networks and fully connected networks could lead to even more robust solutions. Additionally, the integration of real-time data collection and analysis in the model would make the method more dynamic and responsive to operational changes.

CONCLUSIONS

The problem of providing automation for vibration analysis is an urgent one. One of the most effective ways to address this is by using a neural network. Neural networks come in different types and functionalities, which allows them to be applied to various tasks. In our case, we decided to develop and train an MLP-based neural network. The use of a neural network developed based on the MLP method and SGD training is a clear and universal solution for solving similar classification problems.

Due to the combination of methods and their high-quality implementation, we obtained satisfactory results. Unfortunately, the results may vary slightly depending on the number of epochs and initial conditions. However, this should not be considered a flaw, and this method can be regarded as a correct solution for similar problems. Compared to methods that use convolution, our method is more accurate and efficient. This allows for a more accurate understanding of vibrations and the data derived from them.

The prospects for further research involve studying the capabilities of the neural network to learn from and analyze similar data. Pilot testing with similar methods and further analysis are also planned. Practical application will aim to improve and adjust the neural network.

ACKNOWLEDGEMENTS

We would like to thank Oleksiy Tielnov for his help and support throughout the project, his suggestions and links to articles were very useful and essential for our work.

REFERENCES

1. Pan J. et al. LiftingNet: a novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification, *IEEE Transactions on Industrial Electronics,* 2017, Vol. 65, Iss. 6, pp. 4973–4982.
2. Abdi H., Williams L. J. Principal component analysis, *Wiley Interdisciplinary Reviews Computational Statistics,* 2010, Vol. 2, Iss. 4, pp. 433–459.
3. Lee T. W. Independent Component Analysis, Independent Component Analysis : Theory and Applications. Boston, MA, 1998, pp. 27–66.
4. Lee J-M. et al. Nonlinear process monitoring using kernel principal component analysis, *Chemical Engineering Science,* 2004, Vol. 59, Iss. 1, pp. 223–234.
5. Lin M., Chen Q., Yan S. Network in Network, *Proceedings of the 2nd International Conference on Learning Representations, ICLR, 2014, 14–16 Apr.* Banff, Canada, 2014. Access mode: https://www.semanticscholar.org/reader/5e83ab70d0cbc003471e87ec306d27d9c80ecb16.
6. Hsueh Yu-Min et al. Fault diagnosis system for induction motors by CNN using empirical wavelet transform, *Symmetry*, 2019, Vol. 11, Iss. 10, pp. 1–15.
7. Xu Juan et al. Zero-shot learning for compound fault diagnosis of bearings [Electronic resource], *Expert Systems with Applications,* 2022, Vol. 190. Access mode: https://doi.org/10.1016/j.eswa.2021.116197.
8. Lichtner-Bajjaoui A. Neural Networks and Applications [Electronic resource] : Advanced Mathematics Master Program, Universitat de Barcelona. Barcelona, 2020, 51 p. Access mode: https://diposit.ub.edu/dspace/bitstream/2445/180441/2/tfm_lichtner_bajjaoui_aisha.pdf/.
9. Nielsen M. Neural Networks and Deep Learning [Electronic resource] : textbook. Access mode: http://neuralnetworksanddeeplearning.com/.
10. Scornet E. Convolutional Neural Networks [Electronic resource]. Access mode: https://erwanscornet.github.io/teaching/CNN.pdf.
11. Kriesel D. A Brief Introduction to Neural Networks [Electronic resource]. Access mode: https://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf .
12. Haykin S. Neural Networks and Learning Machines [Electronic resource]. 3-rd ed. New York [et al.], 2009, 938 p. Access mode: https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf..
13. Roch S. Mathematical methods in data science (with Python) [Electronic resource]. Access mode: https://mmids-textbook.github.io/index.html.

Received 30.01.2025.
Accepted 04.04.2025.

DOI 10.15588/1607-3274-2025-2-13

УДК 621.43, 004.032.26

МЕТОД АНАЛІЗУ ВХІДНИХ ДАНИХ З ВІБРАЦІЙ ЗУБЧАСТИХ МЕХАНІЗМІВ

Шалімов О. Є. – студент кафедри Автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут».

Москальчук О. О. – студент кафедри Автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут».

Євсеєнко О. М. – канд. техн. наук, доцент, доцент кафедри Автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут».

АНОТАЦІЯ

Актуальність. Розглянуто задачу аналізу векторів даних великого обсягу для аналізу працездатності двигуна гелікоптерів. Ця проблема є критично важливою для покращення надійності та ефективності сучасних авіаційних технологій.

Мета роботи. Створити метод для аналізу вібраційних даних двигуна з метою точного класифікування станів двигуна на основі вібраційних сигналів.

Метод. Проаналізовано вхідні дані, після чого було прийнято рішення створити нейромережу для розпізнавання класу вхідного вектора. Нейромережа може працювати одразу або бути налаштованою для подальшого навчання на подібних даних. Програма була реалізована з використанням класичного методу нейромереж. Оптимальні ваги та зміщення обчислюються за допомогою похідних для мінімізації функції втрат. Для оптимізації було використано алгоритм стохастичного градієнтного спуску (SGD), а також було протестовано різні функції активації для вибору найкращої конфігурації. Вибір правильних функцій активації забезпечив максимальну ефективність.

Результати. На графіках вхідних векторів видно, що вектори з першого класу мали більше піків, що полегшило процес класифікації. Після застосування цього методу точність досягла 70–75%, що було недостатньо для задачі. Для покращення результатів була змінена структура моделі та переналаштовані функції активації. З новим методом нейромережа здатна класифікувати вхідні вектори з точністю 100%.

Висновки. У цьому дослідженні представлено підхід до аналізу вібраційних даних двигуна для оцінки його працездатності. Наукова новизна методу полягає в адаптації багатошарового перцептрону (MLP) для класифікації вібраційних сигналів. Дослідження показало, що навіть без глибоких архітектур можна досягти високої точності, оптимізувавши MLP. Цей метод є універсальним, що дозволяє уникнути додаткових витрат на адаптацію моделі, що важливо для промислового використання. Практичне значення підтверджується програмним забезпеченням та експериментами, що доводять ефективність MLP для моніторингу працездатності, коли параметри моделі та функції активації налаштовані належним чином.

Перспективи подальших досліджень полягають у вивченні можливостей нейромережі для навчання та аналізу подібних даних, а також у пілотних тестуваннях із використанням схожих методів і подальшому аналізі.

КЛЮЧОВІ СЛОВА: 1D Convolutional Neural Networks, Multilayer Perceptron, Stochastic Gradient Descent, Loss Function, аналіз вібрацій двигуна, класифікація сигналів, нейромережа, машинне навчання.

ЛІТЕРАТУРА

1. LiftingNet: a novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification / J. Pan [et al.] // IEEE Transactions on Industrial Electronics. – 2017. – Vol. 65, Iss. 6. – P. 4973–4982.
2. Abdi H. Principal component analysis / H. Abdi, L. J. Williams // Wiley Interdisciplinary Reviews Computational Statistics. – 2010. – Vol. 2, Iss. 4. – P. 433–459.
3. Lee T. W. Independent Component Analysis / T. W. Lee // Independent Component Analysis : Theory and Applications / Lee T. W. – Boston, MA, 1998. – P. 27–66.
4. Nonlinear process monitoring using kernel principal component analysis / J-M. Lee [et al.] // Chemical Engineering Science. – 2004. – Vol. 59, Iss. 1. – P. 223–234.
5. Lin M. Network in Network / Min Lin, Qiang Chen, Shuicheng Yan // Proceedings of the 2nd International Conference on Learning Representations, ICLR, 2014, 14–16 Apr. – Banff, Canada, 2014. – Access mode: https://www.semanticscholar.org/reader/5e83ab70d0cbc003471e87ec306d27d9c80ecb16.
6. Fault diagnosis system for induction motors by CNN using empirical wavelet transform / Yu-Min Hsueh [et al.] // Symmetry. – 2019. – Vol. 11, Iss. 10. – P. 1–15.
7. Zero-shot learning for compound fault diagnosis of bearings [Electronic resource] / Juan Xu [et al.] // Expert Systems with Applications. – 2022. – Vol. 190. – Access mode: https://doi.org/10.1016/j.eswa.2021.116197.
8. Lichtner-Bajjaoui A. Neural Networks and Applications [Electronic resource] : Advanced Mathematics Master Program / Aisha Lichtner-Bajjaoui ; Universitat de Barcelona. – Barcelona, 2020. – 51 p. – Access mode: https://diposit.ub.edu/dspace/bitstream/2445/180441/2/tfm_lichtner_bajjaoui_aisha.pdf.
9. Nielsen M. Neural Networks and Deep Learning [Electronic resource] : textbook / Michael Nielsen. – Access mode: http://neuralnetworksanddeeplearning.com/.
10. Scornet E. Convolutional Neural Networks [Electronic resource] / E. Scornet. – Access mode: https://erwanscornet.github.io/teaching/CNN.pdf.
11. Kriesel D. A Brief Introduction to Neural Networks [Electronic resource] / David Kriesel. – Access mode: https://www.dkriesel.com/_media/science/neuronalenetzeen-zeta2-2col-dkrieselcom.pdf.
12. Haykin S. Neural Networks and Learning Machines [Electronic resource] / Simon Haykin. – 3-rd ed. – New York [et al.], 2009. – 938 p. – Access mode: https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf.
13. Roch S. Mathematical methods in data science (with Python) [Electronic resource] / Sebastien Roch. – Access mode: https://mmids-textbook.github.io/index.html.

DOI 10.15588/1607-3274-2025-2-13