UDC 004.8

APPROACH TO DATA DIMENSIONALITY REDUCTION AND DEFECT CLASSIFICATION BASED ON VIBRATION ANALYSIS FOR MAINTENANCE OF ROTATING MACHINERY

Molchanova M. O. – Postgraduate student of the Department of Computer Sciences, Khmelnytskyi National University, Khmelnytskyi, Ukraine.

Didur V. O. – Student of the Department of Computer Sciences, Khmelnytskyi National University, Khmelnytskyi, Ukraine.

Mazurets O. V. – PhD, Associate Professor, Department of Computer Science, Khmelnytskyi National University, Khmelnytskyi, Ukraine.

ABSTRACT

Context. The actual problem of effective intelligent diagnostics of malfunctions of rotating equipment is solved. The object of study is the process of data dimensionality reduction and defect classification based on vibration analysis for maintenance of rotating machines. The subject of study is the methods of dimension reduction and defect classification by vibration analysis.

Objective. Development of an approach to data dimensionality reduction and defect classification based on vibration analysis for maintenance of rotating machines

Method. The comprehensive approach to data dimensionality reduction and defect classification based on vibration analysis is proposed, which solves the problem of data dimensionality reduction for training classifiers and defect classification, and also solves the problem of building a neural network classifier capable of ensuring the speed of fault classification without loss of accuracy on data of reduced dimensionality. The approach differs from the existing ones by the possibility of using optional union and intersection operators when forming a set of significant features, which provides flexibility and allows to adapt to different contexts and data types, ensuring classification efficiency in cases of large-dimensional data.

A denoising method allows to preserve important information, avoiding redundancy and improving the quality of data for further analysis. It involves calculating the signal-to-noise ratio, setting thresholds, and applying a fast Fourier transform that separates relevant features from noise. Applying the LIME method to a set of machine learning models allows to identify significant features with greater accuracy and interpretability. This contributes to more reliable results, as LIME helps to understand the influence of each feature on the final model solution, which is especially important when working with large datasets, where the importance of individual features may not be obvious. The implementation of optional operators of union and intersection of significant features provides additional flexibility in choosing an approach to defining important features. This allows the method to be adapted to different contexts and data types, ensuring efficiency even in cases with a large number of features.

Results. The developed method was implemented in software and examined when solving the problem of defect classification based on vibration analysis for maintenance of rotating machines.

Conclusions. The conducted experimental studies confirmed the high efficiency and workability of the proposed approach for reducing the dimensionality of data and classifying defects based on vibration analysis in the aspect of maintenance of rotating machines. Prospects for further research will be directed to the search for alternative neural network architectures and their training to reduce training time.

KEYWORDS: dimensionality reduction, significant features, defect detection, MobileNetV2.

ABBREVIATIONS

CBM is a Condition-based Maintenance;

PM is a Predictive Maintenance;

ISO is an International Organization for Standardization;

CNC is a computer numerical control;

CNN is a Convolutional Neural Network;

MSVM is a Multiclass Support Vector Machine;

PdM-CNN is a Predictive Maintenance using Convolutional Neural Network;

VMD is a Variational Mode Decomposition;

CWT is a Continuous Wavelet Transform;

SVM is a Support Vector Machine;

CWRU is a Case Western Reserve University;

KNN is a K-Nearest Neighbors;

LIME is a Local Interpretable Model-agnostic Explanations;

FTX is a FeaTure eXtractor;

NNT is a Neural Network Trainer.

NOMENCLATURE

x is a set of vibration characteristics of machines;

y is a corresponding classes that include "with fault" and "without fault";

F is a structure of the model, which allows to reduce the dimensionality of the vibration data;

w is a model parameters to be optimized;

x' is a selection of characteristics from the original set x;

xs is a represents individual vibrational characteristics;

y' is a corresponding classes for the training data subset;

ys is a corresponding class for characteristics xs;

$MLset$ is a set of machine learning models;

ML_1 is a first machine learning model in $MLset$;

ML_q is a q-th machine learning model in $MLset$;

SNR is a signal-to-noise ratio;

$SNRw$ is a percentage threshold value for denoising;

n is a number of models from $MLset$ to generate a list of numbers of significant features;

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

q is a total number of machine learning models;

nk is a total number of features in the sample xr;

μ is an arithmetic mean value of the features for the sample xr;

xr_1 is a first feature in the sample xr;

xr_i is a i-th feature in the sample xr;

xr_{nk} is a nk-th feature in the sample xr;

T is a threshold value;

FT is a Fourier transform function;

X is a signal with Fourier transform;

X' is a signal with Fourier transform filtered according to the T threshold;

w_1 is a weight of the first feature in the sample xr;

w_i is a weight of the i-th feature in the sample xr;

w_s is an average value by characteristics;

Sf is a set of numbers of significant signs;

Op is a set of optional operations;

Sf_1 is a first important feature;

Sf_j is a j-th important feature;

Sf_n is a last important feature;

Sf_{res} is a set of important features;

Accuracy is a metric the proportion of correct predictions made by the model across the entire dataset;

TP is a number of correctly identified positive instances;

TN is a number of correctly identified negative instances;

FP is a number of incorrectly identified positive instances;

FN is a number of incorrectly identified negative instances.

Precision is a metric the proportion of *TP* predictions among all positive predictions made by the model;

Recall is a metric also known as sensitivity or *TP* rate, metric the proportion of *TP* predictions among all actual positive instances;

F1 is a metric that balances is a metric that balances *Precision* and *Recall*.

INTRODUCTION

Complex mechanical systems such as rotating machines require expensive maintenance to prevent accidents that can cost lives or cause serious damage to the system itself [1]. Every year, the industry spends billions of dollars on maintenance, which can account for up to a third of production costs. Improved maintenance can significantly reduce the overall cost of the system over its lifetime. One of the modern approaches to maintenance in complex rotating machinery based on condition-based maintenance techniques is the use of vibration analysis [2]. In this approach, vibration sensors are installed near the rotating components of the system, and signal processing algorithms are used to detect faults and classify their sources.

The object of study is the process of data dimensionality reduction and defect classification based on vibration analysis for maintenance of rotating machines.

The subject of study is methods of dimensionality reduction and defect classification by vibration analysis.

The purpose of the work is to improve the quality of maintenance of rotating machines by reducing the dimensionality of the data, which will allow to reduce computational costs and improve the speed of data processing without deteriorating the accuracy of diagnostics.

1 PROBLEM STATEMENT

In today's rotating machinery maintenance industry, vibration analysis is critical for timely defect detection and accident prevention. However, processing large volumes of vibration data can be challenging due to the high dimensionality, which complicates the defect classification process.

The problem is formulated as follows: given a set of precedents $<x, y>$, it is necessary to develop a structure of the model $F()$, which allows to reduce the dimensionality of the vibration data, while preserving the information important for the classification of defects.

Optimize parameters w of the model based on $<x', y'>$, where $x' \subset \{xs\}$, $y' = \{ys \mid xs \in x'\}$, which makes it possible to achieve the best results in the classification of defects in two classes: "with a fault" and "without a fault".

Conduct a comparative analysis of model results to confirm the effectiveness of the data dimensionality reduction approach in defect classification tasks.

The task of this study is to develop an effective method of data dimensionality reduction, which will ensure the accuracy and speed of fault classification, thus improving the quality of maintenance of rotating machines.

2 REVIEW OF THE LITERATURE

Vibration measurements are critical for fault diagnosis in industrial equipment because they provide information about the condition of rotating equipment. A review [3] documents state-of-the-art deep learning methods for equipment health monitoring based on vibration signals. Numerous studies from two leading databases, Web of Science and Scopus, were selected for review. After careful analysis, 59 studies were selected and systematically analyzed. The purpose of the review is to provide researchers with an in-depth understanding of fault diagnosis techniques that use deep learning to process vibration signals.

A study [4] discusses the importance of creating an effective maintenance system, in particular CBM and PM that complement each other. The use of CBM and PM in the study focuses on the case study of three pumps in a chili sauce factory in Jakarta, Indonesia, demonstrating that vibration analysis using accelerometers is an effective method for condition monitoring of rotating machinery. Calculation of Root Mean Square and comparison of the result with ISO 10816 allows to determine the current state of the engine, and the fast Fourier transform helps in

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

grouping vibrations by frequencies for damage analysis. The study shows that technology-enabled service is affordable for small and medium-sized businesses, and the use of machine learning will improve future predictions.

In [5] and [6], the importance of bearings for CNC machines and their role in ensuring the reliable operation of machines is considered. Signals in the conditions of faulty and normal bearings are analyzed, emphasizing the importance of early diagnosis of faults. Despite the importance of the topic, the problem is complicated by the insufficiency of bearing fault databases and the presence of noise in the vibration and acoustic signals, which makes automatic fault identification difficult. A CNN was applied to extract high-level features from vibration and acoustic signals, which enabled the training of MSVM. Experimental analysis showed that the proposed method can diagnose bearing faults of CNC machines with a classification accuracy of 98.9% based on the combined features of vibration and acoustic signals.

The use of the PdM-CNN model is proposed for the automatic classification of rotating equipment malfunctions and recommendations on the need for maintenance [7]. The paper uses data from only one vibration sensor mounted on the bearing on the drive side of the motor, which is the most common location in industry. The study was carried out under controlled conditions with variations in rotational speed, load levels and fault intensities to test whether it is possible to build a model capable of classifying such faults using only one set of vibration sensors. The results showed that the accuracy of the PdM-CNN model was 99.58% and 97.3% when applied to two different public databases.

Research [8] proposed a hybrid bearing fault diagnosis method based on VMD, CWT, CNN, and SVM, which is suitable for processing small samples. First, the resampled data are subjected to VMD processing, after which the reconstructed IMF data are overlaid and sampled to obtain a 2D time-frequency image using CWT. Next, a CNN model is created with the selected hyperparameters, and training samples are fed into the CNN to train the model. A pre-trained CNN model is used to stepwise train test samples to extract fault features, and SVM is used instead of the Softmax function to identify and classify faults.

The effectiveness of the proposed method is confirmed using the vibration data of the CWRU bearings and the test bench for testing the spindle device, where the classification accuracy was on average 99.9% for the former and 90.15% for the latter.

However, the need for a large amount of training data for training neural networks can be noted as a problem for effective intelligent diagnosis of malfunctions of rotating equipment [9, 10]. In addition, this data may not contain samples to identify rare types of damage, so it is advisable to choose pre-trained neural networks that can take advantage of Zero-Shot Learning as fault detection models.

Another problem is the high dimensionality of the data, a large part of which is uninformative. This leads to an increase in the time of diagnosis and training of models.

3 MATERIALS AND METHODS

To implement the approach to data dimensionality reduction and defect classification based on vibration analysis for the maintenance of rotating machines, the following tasks must be solved:

1) data dimensionality reduction for classifier training and defect classification;

2) construction of a neural network classifier capable of ensuring the speed of fault classification without loss of accuracy on data of reduced dimensions.

The general scheme of the approach to data dimensionality reduction and defect classification based on vibration analysis is shown in Figure 1.

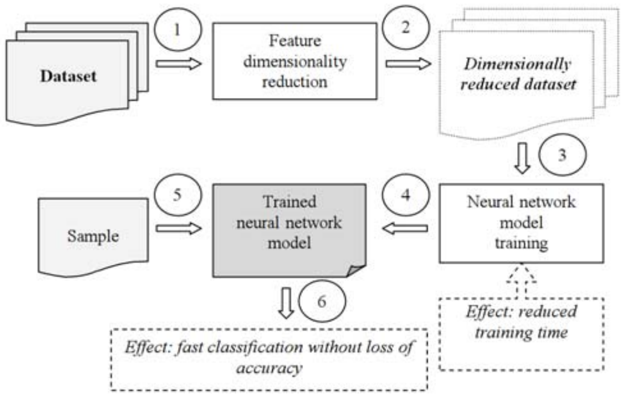

Figure 1 – Scheme of the proposed approach

Figure 1 shows the key points of the proposed approach to data dimensionality reduction and defect classification by vibration analysis. The approach primarily solves the problem of reducing the dimensionality of the data for further training of the neural network model, which allows to achieve the effect of reducing the time spent on training without loss of accuracy. In the future, the trained neural network is able to classify defects based on significant features based on vibration analysis.

To solve the problem of reducing the dimensionality of data for training classifiers and classifying defects, a method of reducing the dimensionality of data is proposed, the scheme of which is shown in Figure 2.

The input data of the dimensionality reduction method is the dataset provided within the framework of the competition “All-Ukrainian competition of young scientists in the field of intellectual IT” (https://zp.edu.ua/vkiit/), the set *MLset* (1):

$$MLset = \{ML_1,\dots,ML_q\}, \quad (1)$$

SNRw (determined empirically, by default equal to 0.9), an optional operation to create a dataset of significant features based on union or intersection, as well as *n* models from *MLset* to create a list of significant feature numbers.

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8
OPEN ACCESS

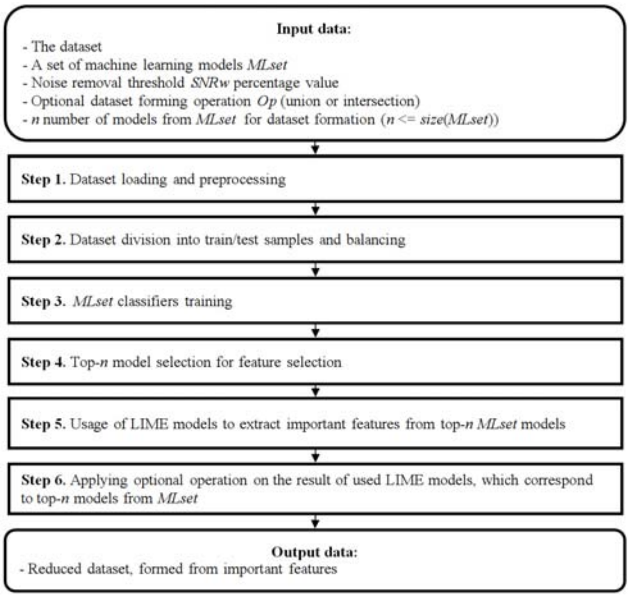

Figure 2 – The method of data dimensionality reduction

The following model architectures were used in the study: SVM, Naive Bayes, RandomForest, Gradient Boosting and KNN. Within these architectures, different sets of models can be created by changing the hyperparameters specific to each model.

In step 1, labels and their corresponding files with vibrations are loaded. Vibration signals are often expressed as noise from various sources, such as electrical interference, environmental factors, and measurement errors. To solve this problem, various methods of signal processing were developed, one of which is the Fourier transform [11]. The loaded dataset is preprocessed by denoising all samples using transformations (2–6). For each sample, a signal-to-noise ratio (2) is calculated:

$$SNR(x) = \frac{\frac{1}{nk}\sum_{i=1}^{nk} xr_i^2}{\frac{1}{nk}\sum_{i=1}^{nk}(xr_i - \mu)^2}, \qquad (2)$$

which is then divided by the user's desired ratio in order to set a threshold that would discard noise and not affect potentially important data.

The threshold value T is determined by transformation (3):

$$T = \begin{cases} -\infty, \, if \; SNRw = 0, \\ \min(|xr_i|) where \, xr_i \neq 0, \; if \; SNRw = 1, \\ \max(|xr_i|) \times SNR \times (1 - SNRw), \, if \; 0 < SNRw < 1. \end{cases} \qquad (3)$$

Next, the fast Fourier transform FT [12] (4) is calculated for each sample:

$$X(f) = FT(xr_i). \qquad (4)$$

After that, the coefficients are filtered according to the T threshold (5):

$$X'(f) = \begin{cases} X(f), if \, |X(f)| \geq T, \\ 0, \, if \, |X(f)| < T. \end{cases} \qquad (5)$$

For the filtered coefficients from (5), the inverse Fourier transform is calculated to obtain the filtered signal (6):

$$x'(i) = F^{-1}(X'(f)). \qquad (6)$$

In step 2, the dataset, where the transformation (2–6) was applied to each element, is divided into training and validation samples in the ratio of 80/20. If the dataset is unbalanced, SMOTE balancing is additionally applied to the training sample [13].

In step 3, models are trained from the set *MLset*. Parameters can be modified within each architecture.

In step 4, the top-n models are selected for further obtaining a list of significant features. The selection takes place according to the criterion F1 of the metric [14] in descending order.

In step 5, a set of LIME models, the number of which is equal to n, is used to obtain a list of significant features for the models selected in step 4 from the *MLset*, since LIME can be applied to any model, regardless of its type [15]. The LIME model focuses on explaining individual predictions by generating locally linear models around the prediction point [16]. This allows for detailed analysis of why the model made the prediction it did for a particular case, which is useful for identifying significant features. Significant features are selected according to the condition "if the weight of the feature w_i is greater than or equal to the average value w_s for the entire set of features, it is important", features with a lower weight are discarded. At this step, the set Sf is obtained from the numbers of significant features (7):

$$\{Sf\} = \{i, \mid w_i \geq w_s\}. \qquad (7)$$

In step 6, provided that $n >= 2$, an optional operation is performed on the result of the n LIME models used from the set $Op \in \{\cup, \cap\}$ – defined by the user at the input data stage. For all possible pairs of sets of LIME models, sets with numbers of important features are formed by transformation (8):

$$\{Sf_{res}\} = \mathop{Op}_{j-1}^{n} Sf_j. \qquad (8)$$

The output is a reduced dimensionality dataset, where each sample has $size\{Sf_{res}\}$, formed from a list of significant feature numbers. Accordingly, the created dataset will be used as input data to solve the second task of the research – the construction of a neural network

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8
OPEN ACCESS

classifier capable of ensuring the speed of fault classification without loss of accuracy.

The method of obtaining a fault classification model is shown in Figure 3.

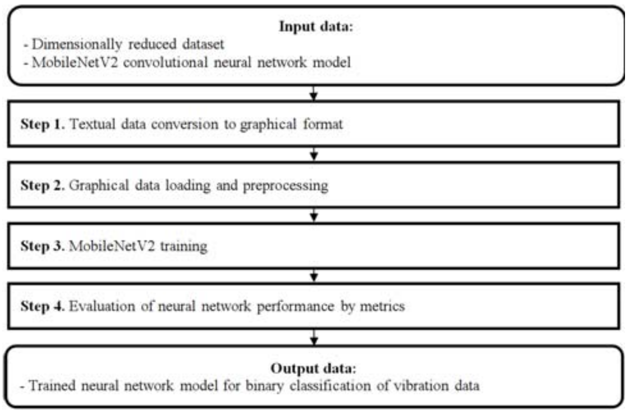

Figure 3 – The method of obtaining a fault classification model

The input data is a reduced dimensionality dataset and a pre-trained MobileNetV2 convolutional neural network model.

MobileNet-v2 is considered to be an efficient CNN model for mobile devices (or embedded systems) with the potential characteristics of small size, low latency, and low power consumption [17], which are important characteristics for vibration analysis defect classification for rotating machinery maintenance. MobileNet-v2 is optimized to achieve high accuracy with significantly reduced computational costs. This is achieved through the use of depth-distributed convolutional layers and inverted residual blocks.

In step 1, the reduced-dimensional text dataset is transformed into a graphical one by plotting the data of each sample on a graph, where the ordinal numbers of the features are on the X axis, and their values are on the Y axis (or in another way: time moments are on X, and powers of vibrations at given moments of time are on Y). An example of the transformation of step 1 is shown in Figure 4.

In step 2, the generated graph dataset is loaded and each sample is processed using the MobileNetV2 convolutional neural network image preprocessor (mobilenet_v2.preprocess_input).

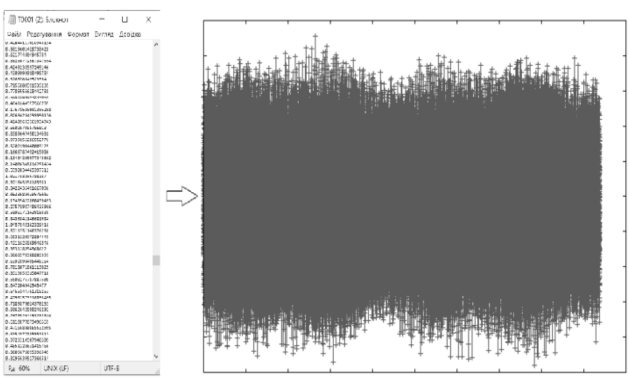

Figure 4 – Converting a text representation into a graphic representation

In step 3, the neural network is trained. MobileNetV2 model, using transfer learning. A neural network is retrained for the task of binary classification of vibration graphs. The pre-trained MobileNetV2 model is used only for feature extraction from images, instead additional GlobalAveragePooling2D and Dense layers are trained (Figure 5).

At step 4, the trained neural network model is evaluated according to the following metrics: Accuracy, Precision, Recall, F1 [18].

The *Accuracy* is calculated by formula (9):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{9}$$

The *Precision* is calculated by formula (10):

$$Precision = \frac{TP}{TP+FP}. \tag{10}$$

The *Recall* is calculated by formula (11):

$$Recall = \frac{TP}{TP+FN}. \tag{11}$$

And the $F1$ is calculated by formula (12):

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall}. \tag{11}$$

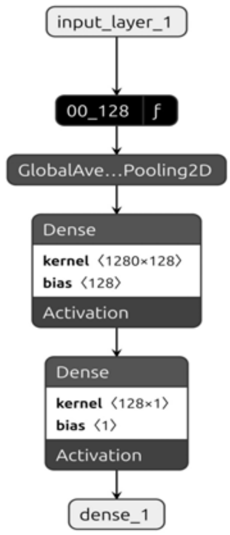

Figure 5 – Architecture of the neural network used

The output of the method is a trained model of the MobileNetV2 convolutional neural network for binary classification of vibration graphs.

So, within the framework of the developed approach to data dimensionality reduction and defect classification based on vibration analysis for the maintenance of rotating machines, the problem of data dimensionality reduction was solved by using agnostic LIME models,

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8
OPEN ACCESS
CC BY SA

and the problem of building a neural network classifier based on the MobileNetV2 architecture, capable of ensuring the speed of fault classification without loss of accuracy.

4 EXPERIMENT

Based on the proposed approach, a software complex was developed in the form of three applications with a command line interface and a web page. The software implementation and instructions for deployment and use are available on GitHub (https://github.com/Pravetz/mmdv_vkiit). The interaction between the components of the software complex and data flows is shown in Figure 6. The language for creating the components of the software complex is Python.

"FTX" is an application for the problem of reducing the dimensionality of the data set. It uses libraries sklearn (for training classifiers, dividing the dataset into samples) [19], imblearn (for SMOTE-balancing of the training sample, it happens by default and is not controlled by the user through console parameters) [20], lime (creating explanations, extracting important features) [21], scipy (auxiliary functions for the algorithm for extracting important features) and numpy (working with data arrays) [22].

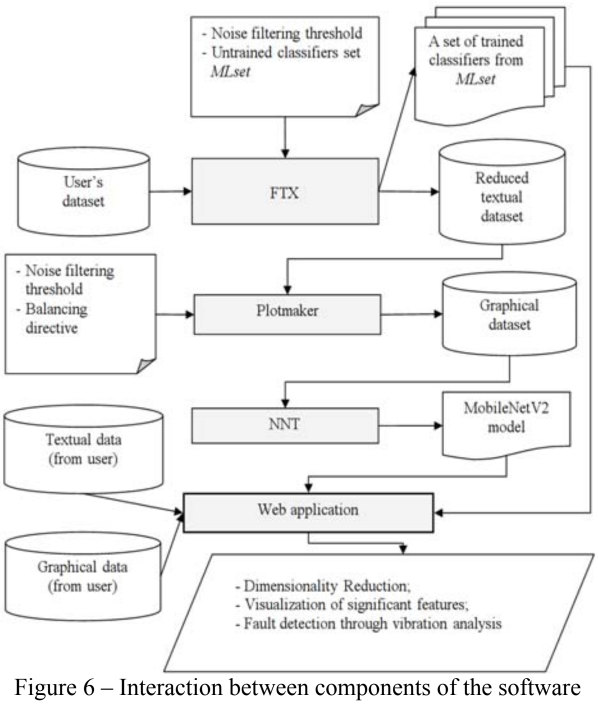

Figure 6 – Interaction between components of the software complex and data flows

The application for performing the task of building a neural network classifier (starting from step 2) "NNT" also does not have a graphical user interface. In the model training mode, the program loads the dataset from the specified path, processes it with the MobileNetV2 neural network image preprocessor, then trains the model for a user-specified number of epochs, evaluates its performance, and saves the trained model according to the specified path.

Libraries such as tensorflow (for training the MobileNetV2 model, loading the dataset) [23], sklearn (dividing the dataset into samples, evaluating the model according to the main metrics), numpy (for working with data arrays) [24] and matplotlib (for saving model training history) [25] are used by it.

The "Plotmaker" application for converting a text dataset into a graphic one performs step 1 of the task of building a neural network classifier.

This application is the smallest of all and performs the preparation of graphical data for MobileNetV2 training. From the libraries, imblearn is used here for SMOTE-balancing of the dataset, graphs are drawn by transferring drawing data using a pipeline to the gnuplot program [26].

The console applications described above were created to be used in the form of a "pipeline": first, "FTX" creates a dataset of reduced dimensionality, then the created dataset is passed to "Plotmaker", which creates a graphical data set, which in turn is passed to "NNT" for training a neural network classifier.

The web application (Figure 7) serves to demonstrate the results of the console applications and allows: to use trained versions of the MobileNetV2 model trained on different versions of the dataset (full, reduced by union of significant features, intersection of significant features) to determine the state of the machine component according to the vibration graph, provided by the user; view the important features for each of the classifiers from the *MLset* on an interactive graph, where for each feature its number and weight are written during decision-making.

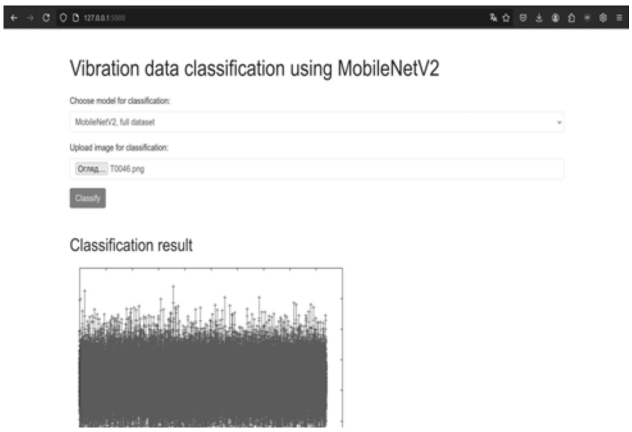

Figure 7 – Web application interface

Based on the developed software complex, a study of the effectiveness of the proposed approach to data dimensionality reduction and defect classification based on vibration analysis for maintenance of rotating machines will be conducted.

The plan for researching the effectiveness of the proposed approach:

1. Study the influence of the application of noise cleaning on the classification of defects of parts of rotating machines.

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

OPEN ACCESS

2. Study significant features for classifiers from the set *MLset*, and find the optimal top-*n* for the selected sample in which the F1 metric will be as close to 1 as possible.

3. Study the ability of MobileNetV2 neural network to learn on data with different numbers of significant features.

During the experiment *MLset* of length 5 was formed, which includes the following models: Gradient Boosting, Random Forest, Naive Bayes, SVM, KNN. These models were trained on the base dataset in 2 variants: without applying transformations (2)–(6) and with their application. The results by metrics are shown in Table 1.

The transformation (7)–(8) is applied to each model from *MLset*, which is included in the top-*n*. The results for values of *n* from 2 to 4 are shown in Table 2.

5 RESULTS

Within the dimensionality reduction task, a set of *MLset* classifiers was trained, the data of the experiment according to the Accuracy, Precision, Recall, F1 metrics, as well as the training time in seconds are shown in Table 1.

Table 1 – Evaluation of classifiers from *MLset* by metrics

Metrics	Gradient Boosting	Random Forest	Naive Bayes	SVM	KNN
Original dataset					
Accuracy	0.784	0.776	0.780	0.797	0.224
Precision	0.761	0.602	0.829	0.776	0.05
Recall	0.784	0.776	0.780	0.797	0.224
F1	0.711	0.678	0.688	0.780	0.082
Training time (seconds)	3670	6	2	109	1
Datasets with transformations (2)–(6) applied					
Accuracy	0.978	0.974	0.961	0.853	0.970
Precision	0.98	0.977	0.967	0.861	0.970
Recall	0.978	0.974	0.961	0.853	0.970
F1	0.979	0.975	0.962	0.829	0.970
Training time	3643	6	2	56	1

Table 2 shows the data of the dimensionality reduction experiment with the original dataset (without applying transformations (2)–(6)). Number of features in the input dataset: 93752.

For *n*=4. Top-4 models by F1 score:

1. SVM.
2. Gradient Boosting.
3. Naive Bayes.
4. Random Forest.

The minimum score of the top-4 model according to F1 metric: 0.678. Table 2 shows the pairwise application of the optional operators “intersection” and “union” between the classifiers included in the top 4 on the dataset with noise.

Table 2 – Application of “intersection” and “union” operators to reduce dimensionality (without (2)–(6))

	SVM	Gradient Boosting	Naive Bayes	Random Forest
For “Intersection” operation				
SVM	39930	16875	16951	16995
Gradient Boosting	16875	39627	16946	16804
Naive Bayes	16951	16946	39838	16962
Random Forest	16995	16804	16962	39758
For “Union” operation				
SVM	39930	62682	62817	62693
Gradient Boosting	62682	39627	62519	62581
Naive Bayes	62817	62519	39838	62634
Random Forest	62693	62581	62634	39758

The intersection of the top 2 models (SVM, Gradient Boosting) reduced the number of significant features to 16875. The intersection of the top 3 models (SVM, Gradient Boosting, Naive Bayes) reduced the number of significant features to 7221. The intersection of the top 4 models (SVM, Gradient Boosting, Naive Bayes, Random Forest) reduced the number of significant features to 3097. At the same time, the application of the “union” operator for the top 2 models (SVM, Gradient Boosting) reduced the number of features to 62682. The union of the top 3 models (SVM, Gradient Boosting Boosting and Naive Bayes) reduced the number of features to 75844. Combining the top 4 models (SVM, Gradient Boosting, Naive Bayes and Random Forest) reduced the number of features to 83390.

Table 3 shows the data of the dimensionality reduction experiment with the denoised dataset (using transformations (2)–(6)). Number of features in the input dataset: 93752.

Table 3 – Application of “intersection” and “union” operators to reduce dimensionality (with (2)–(6))

	Gradient Boosting	Random Forest	KNN	Naive Bayes
For “Intersection” operator				
Gradient Boosting	39676	16835	16728	17655
Random Forest	16835	39730	16844	17643
KNN	16728	16844	39868	17659
Naive Bayes	17655	17643	17659	41704
For “Union” operator				
Gradient Boosting	39676	62571	62816	63725
Random Forest	62571	39730	62754	63791
KNN	62816	62754	39868	63913
Naive Bayes	63725	63791	63913	41704

Top-4 models by F1 score:

1. Gradient Boosting.
2. Random Forest.
3. K-Nearest Neighbors.
4. Naive Bayes.

The minimum rating of the top-4 model according to the F1 metric: 0.962. Table 3 shows the application of the “intersection” and “union” operators among the top 4 classifiers on the noise-free dataset.

The intersection of the top 2 models (Gradient Boosting and Random Forest) reduced the number of

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8
OPEN ACCESS

significant features to 16835. The intersection of the top 3 models (Gradient Boosting, Random Forest and Naive Bayes) reduced the number of significant features to 7061. The intersection of the top 4 models (Gradient Boosting, Random Forest, K-Nearest Neighbors and Naive Bayes) reduced the number of significant features to 3102. Combining the top 2 models (Gradient Boosting and Random Forest) reduced the number of features to 62571. Combining the top 3 models (Gradient Boosting, Random Forest and Naive Bayes) reduced the number of features to 75928. Combining the top 4 models (Gradient Boosting, Random Forest, K-Nearest Neighbors and Naive Bayes) reduced the number of features to 83834.

The next experiment was a study of learning the MobileNetV2 neural network on the obtained datasets.

The performance of MobileNetV2 at the intersection of the top-*n* models on the dataset with noise (without transformations (2)–(6)) and without noise, with and without SMOTE balancing is shown in Table 4. These results highlight MobileNetV2's robustness under varying noise conditions and balancing techniques, offering insights into its reliability across different data scenarios.

The performance of MobileNetV2 on the full dataset with and without noise, with and without SMOTE balancing is shown in Table 5. Number of significant features: 93752.

Table 4 – Evaluation of MobileNetV2 with intersection and union operators of top-*n* models by metrics

***n*=4**	**Dataset without (2)–(6)**				
		"Intersection" operator, size{Sfres}: 3097		*"Union" operator, size{Sfres}: 83390*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.8996	0.9071	0.9986	1.0000
	Precision	0.9012	0.8333	0.9985	1.0000
	Recall	0.8818	0.7570	0.9971	1.0000
	F1	0.9215	0.9267	1.0000	1.0000
	Training time (sec)	33.41	23.57	34.24	23.57
	Dataset with (2)–(6)				
		"Intersection" operator, size{Sfres}: 3102		*"Union" operator, size{Sfres}: 83834*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.8996	0.9071	0.9892	0.9946
	Precision	0.9012	0.8333	0.9891	0.9892
	Recall	0.8818	0.7570	0.9884	0.9913
	F1	0.9215	0.9267	0.9898	0.9871
	Training time (sec)	34.60	24.19	34.63	24.57
***n*=3**	**Dataset without (2)–(6)**				
		"Intersection" operator, size{Sfres}: 7221		*"Union" operator, size{Sfres}: 75844*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.9487	0.9579	0.9993	1.0000
	Precision	0.9493	0.9099	0.9993	1.0000
	Recall	0.9327	0.9801	0.9985	1.0000
	F1	0.9666	0.8491	1.0000	1.0000
	Training time (sec)	31.75	24.20	32.56	24.27
	Dataset with (2)–(6)				
		"Intersection" operator, size{Sfres}: 7061		*"Union" operator, size{Sfres}: 75928*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.9487	0.9546	0.9451	0.9968
	Precision	0.9473	0.9005	0.9416	0.9935
	Recall	0.9681	1.0000	0.9984	0.9957
	F1	0.9273	0.8190	0.8910	0.9914
	Training time (sec)	34.43	23.91	33.86	24.30
***n*=2**	**Dataset without (2)–(6)**				
		"Intersection" operator, size{Sfres}: 16875		*"Union" operator, size{Sfres}:62682*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.9393	0.9741	0.9884	1.0000
	Precision	0.9395	0.9483	0.9886	1.0000
	Recall	0.9588	1.0000	0.9943	1.0000
	F1	0.9209	0.9016	0.9831	1.0000
	Training time (sec)	34.57	23.96	32.30	23.23
	Dataset with (2)–(6)				
		"Intersection" operator, size{Sfres}: 16835		*"Union" operator , size{Sfres}: 62571*	
	Metrics	Balanced data	Unbalanced data	Balanced data	Unbalanced data
	Accuracy	0.8844	0.9526	0.9566	0.9569
	Precision	0.8750	0.9076	0.9560	0.9123
	Recall	0.9790	0.9310	0.9939	0.9811
	F1	0.7910	0.8852	0.9209	0.8525
	Training time (sec)	33.10	24.42	34.77	24.06

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

An experiment was also conducted to study the effectiveness of the dimensionality reduction method with the smallest number of features obtained – 3102 with noise removal (n=4, classifiers: Gradient Boosting, Random Forest, K-Nearest Neighbors, Naive Bayes)). The experimental data are shown in Table 6.

All experiments were conducted on a system equipped with an Intel Core i7 processor and 16 GB of RAM, supported by a 32 GB swap file to manage memory-intensive operations. The software environment included Python version 3.10.15, utilizing libraries such as sklearn, imblearn, tensorflow, keras, lime, flask for data processing, modeling, and web functionalities, and Gnuplot 6.0.0, all running on Linux Mint 22 Xfce OS (kernel version 6.8.0–38-generic).

Table 5 – Evaluation of MobileNetV2, noisy and noise-free dataset by metrics

Dataset without transformations (2)–(6)		
Metrics	Balanced data	Unbalanced data
Accuracy	0.9942	1.0000
Precision	0.9944	1.0000
Recall	0.9888	1.0000
F1	1.0000	1.0000
Training time (sec)	35.15	25.70
Dataset with transformations (2)–(6)		
Accuracy	0.9566	0.9612
Precision	0.9575	0.9280
Recall	0.9602	0.9062
F1	0.9548	0.9508
Training time (sec)	35.18	24.73

Table 6 – Evaluation of classifiers from *MLset* by metrics with the number of significant features 3102

Metrics	Gradient Boosting	Random Forest	Naive Bayes	KNN
Original dataset				
Accuracy	0.99	0.98	0.96	0.97
Precision	0.99	0.98	0.96	0.97
Recall	0.99	0.98	0.96	0.97
F1	0.99	0.98	0.96	0.97
Training time (sec)	115	<1	<1	<1

The “Discussion” section details the obtained data from Tables 1–6.

6 DISCUSSION

As can be seen from Table 1, before applying transformations (2)–(6), the metrics show results of less than 83% on all metrics for all models from the *MLset*. The worst results are observed for the KNN model. This is due to the susceptibility of KNN to problems in high-dimensional spaces where the distance between points becomes less informative.

After applying transformations (2)–(6), the metrics significantly improved their performance for all classifiers from the *MLset* set. After denoising for the KNN classifier, the distance between the points became more informative, which allowed us to obtain metrics at the level of 97%.

However, taking into account the specificity of the problem of defect classification by vibration analysis, the obtained accuracy indicator is not a benchmark and needs improvement.

It is known, that neural networks better track complex dependencies, therefore the MobileNetV2 neural network was chosen as a classifier, which was studied on the obtained data sets by applying the optional operators “intersection” and “union” of significant features selected using LIME models without transformations (2)–(6) (table 2) and with their application (table 3).

The smallest number of features obtained using the optional intersection operator is 3097 without denoising and 3102 with denoising, which is a 30.27-fold reduction in dimensionality.

From the conducted experiment with n=4 for classifiers from the *MLset* (table 6), it was observed that reducing the dimensionality by more than 30 times does not worsen the result, but on the contrary, for most classifiers it improves the values of the metrics (compared to the results of table 1). Gradient Boosting managed to improve all metrics to 0.99, while reducing training time from 3643 seconds to 115 (31.67 times).

However, this reduction in dimensionality leads to a loss of accuracy of the MobileNetV2 neural network classifier (table 4).

The corresponding results of the experiment are shown in Table 4. From which it can be seen that, compared to the best F1 indicators of 0.99 for Gradient Boosting, it was possible to raise not only the metrics to 1 on the validation and training data, but also to reduce the training time to 23.23 seconds (compared to 115 seconds).

Such metric results were achieved when applying the optional “union” operator for n=4, 3 and 2 in the absence of SMOTE-balancing on the dataset without applying transformations (2)–(6). This is explained by the fact that the MobileNetV2 neural network is capable of reducing the dimensionality and determining meaningful data. However, with n=2 for the “union” operator, the number of significant features was reduced to 62,682 (compared to the initial number of 93,752), which in turn contributed to the reduction of the learning time to 23.23 (table 5), compared to 25.70 with the initial dataset in 93752 features.

Compared with analogues, it was possible to improve the accuracy from 0.989 in [6] and from 0.996 in [7] to 1.

CONCLUSIONS

The actual problem of effective intelligent diagnostics of malfunctions of rotating equipment is solved.

The scientific novelty consists in creating a method of reducing the dimensionality of data by applying a complex approach that combines:

– cleaning from noise (transformation (2)–(6)), allows to save important information, avoiding redundancy;

– use of the LIME method on *MLset* to determine significant features, which allows to obtain interpreted re-

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

sults and ensure greater reliability in determining important features;

– the use of optional operators, which provides flexibility in choosing an approach to determining important features, which can be useful in different contexts and for different types of data, especially in large datasets with a large number of features.

The proposed approach differs from the existing ones by the possibility of using optional operators of union and intersection when forming a set of significant features, which provides flexibility and allows to adapt to different contexts and types of data, ensuring the efficiency of classification in cases of large-dimensional data.

On the basis of the created datasets of reduced dimension, a neural network classifier was built, which reached the values of 1 according to the Accuracy, Precision, Recall and F1 metrics.

The practical significance is that software has been developed that implements the proposed approach to data dimensionality reduction and defect classification based on vibration analysis for maintenance of rotating machines. The results of the experiments show that the developed approach is effective and allows reducing the dimensionality of the data by more than 30 times without losing classification accuracy.

Prospects for further research will be aimed at finding alternative neural network architectures to reduce training time.

ACKNOWLEDGEMENTS

The work was carried out as part of the All-Ukrainian competition-hackathon of scientific works of young scientists in the field of intellectual information technologies in 2024 at Zaporizhzhya National Technical University.

REFERENCES

1. Matania O., Bachar L., Bechhoefer E., Bortman J. Signal Processing for the Condition-Based Maintenance of Rotating Machines via Vibration Analysis: A Tutorial, *Sensors*, 2024, Vol. 24, No. 2, P. 454. DOI: 10.3390/s24020454.
2. Mongia C., Goyal D., Sehgal S. Vibration response-based condition monitoring and fault diagnosis of rotary machinery, *Materials Today: Proceedings*, 2022, Vol. 50, pp. 679–683. DOI: 10.1016/j.matpr.2021.04.395.
3. Tama B. A., Vania M., Lee S., et al. Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals, *Artificial Intelligence Review*, 2023, Vol. 56, pp. 4667–4709. DOI: 10.1007/s10462-022-10293-3.
4. Nata C., Laurence N., Hartono N., Cahyadi L. Implementation of Condition-based and Predictive-based Maintenance using Vibration Analysis, *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, 2021, pp. 90–95. DOI: 10.1109/IC2IE53219.2021.9649400.
5. Iqbal M., Madan A. K., Ahmad N. Vibration and acoustic signal-based bearing fault diagnosis in CNC machine using an improved deep learning, *Iran Journal of Computer Science*, 2024. DOI: 10.1007/s42044-024-00205-9.
6. Althubaiti A., Elasha F., Teixeira J.A. Fault diagnosis and health management of bearings in rotating equipment based on vibration analysis, *Journal of Vibroengineering*, 2022, Vol. 24, No. 1, pp. 46–74. DOI: 10.21595/jve.2021.22100.
7. Souza R.M., Nascimento E.G.S., Miranda U.A., Silva W.J.D., Lepikson H.A. Deep learning for diagnosis and classification of faults in industrial rotating machinery, *Computers & Industrial Engineering*, 2021, Vol. 153, P. 107060. DOI: 10.1016/j.cie.2020.107060.
8. Gu J., Peng Y., Lu H., Chang X., Chen G. A novel fault diagnosis method of rotating machinery via VMD, CWT and improved CNN, *Measurement*, 2022, Vol. 200, P. 111635. DOI: 10.1016/j.measurement.2022.111635.
9. Di Maggio L.G., Brusa E., Delprete C. Zero-Shot Generative AI for Rotating Machinery Fault Diagnosis: Synthesizing Highly Realistic Training Data via Cycle-Consistent Adversarial Networks, *Applied Sciences*, 2023, Vol. 13, No. 22, P. 12458. DOI: 10.3390/app132212458.
10. Ma C., Wang X., Li Y., Cai Z. Broad zero-shot diagnosis for rotating machinery with untrained compound faults*, Reliability Engineering & System Safety*, 2024, Vol. 242, P. 109704. DOI: 10.1016/j.ress.2023.109704.
11. Liang R., Ran W., Chen Y. et al. Fault Diagnosis Method for Rotating Machinery Based on Multi-scale Features, *Chin. J. Mech. Eng.,* 2023, Vol. 36, P. 141. DOI: 10.1186/s10033-023-00966-7.
12. Fast Fourier Transform (FFT) [Electronic resource]. Access mode: https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter24.03-Fast-Fourier-Transform.html.
13. Krak I., Zalutska O., Molchanova M., Mazurets O., Manziuk E., Barmak O. Method for neural network detecting propaganda techniques by markers with visual analytic, *CEUR Workshop Proceedings*, 2024, Vol. 3790, pp. 158–170 [Electronic resource]. Access mode: https://ceur-ws.org/Vol-3790/paper14.pdf.
14. Krak I., Zalutska O., Molchanova M., Mazurets O., Bahrii R., Sobko O., Barmak O. Abusive Speech Detection Method for Ukrainian Language Used Recurrent Neural Network, *CEUR Workshop Proceedings*, 2023, Vol. 3387, pp. 16–28. DOI: 10.31110/COLINS/2024-3/002.
15. Hasan M.M. Understanding Model Predictions: A Comparative Analysis of SHAP and LIME on Various ML Algorithms*, Journal of Scientific and Technological Research*, 2024, Vol. 5, No. 1, pp. 17–26. DOI: 10.59738/jstr.v5i1.23(17–26).eaqr5800.
16. Baniya A., Vinod A., Chinta H.S., Vishnu P., T.S. Intrusion Detection based on clustering with ML algorithm and LIME Insights, *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS),* Raichur, India, 2024, pp. 1–6. DOI: 10.1109/ICICACS60521.2024.10498512.
17. Pham M.T., Kim J.-M., Kim C.H. Deep Learning-Based Bearing Fault Diagnosis Method for Embedded Systems, Sensors, 2020, Vol. 20, pp. 6886. DOI: 10.3390/s20236886.
18. Slobodzian V., Molchanova M., Kovalchuk O., Sobko O., Mazurets O., Barmak O., Krak I. An Approach Based on the Visualization Model for the Ukrainian Web Content Classification, *2022 12th International Conference on Advanced Computer Information Technologies (ACIT),* 2022, pp. 400–405. DOI: 10.1109/ACIT54803.2022.9913162.
19. Scikit-learn: machine learning in Python – Scikit-learn 1.6.0 documentation [Electronic resource], Scikit-learn: machine learning in Python – Scikit-learn 0.16.1

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

documentation. Mode of access: https://scikit-learn.org/stable/.
20. Imbalanced-learn documentation – Version 0.13.0 [Electronic resource], *Welcome to imbalanced-learn documentation.* Mode of access: https://imbalanced-learn.org/stable/.
21. Local Interpretable Model-Agnostic Explanations (lime) – lime 0.1 documentation [Electronic resource], *Local Interpretable Model-Agnostic Explanations (lime) – lime 0.1 documentation*. Mode of access: https://lime-ml.readthedocs.io/en/latest/.
22. SciPy [Electronic resource], *SciPy*. Mode of access: https://scipy.org/.
23. TensorFlow [Electronic resource], *TensorFlow*. Mode of access: https://www.tensorflow.org.
24. NumPy [Electronic resource], *NumPy*. Mode of access: https://numpy.org/.
25. Matplotlib – Visualization with Python [Electronic resource], *Matplotlib – Visualization with Python*. Mode of access: https://matplotlib.org/.
26. Gnuplot homepage [Electronic resource], *Gnuplot homepage.* Mode of access: https://gnuplot.sourceforge.net/.

Received 02.12.2024.
Accepted 24.01.2025.

УДК 004.8

ПІДХІД ДО ЗМЕНШЕННЯ РОЗМІРНОСТІ ДАНИХ ТА КЛАСИФІКАЦІЇ ДЕФЕКТІВ ЗА АНАЛІЗОМ ВІБРАЦІЙ ДЛЯ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ ОБЕРТОВИХ МАШИН

Молчанова М. О. – аспірант кафедри комп’ютерних наук Хмельницького національного університету, м. Хмельницький, Україна.

Дідур В. О. – студент кафедри комп’ютерних наук Хмельницького національного університету, м. Хмельницький, Україна.

Мазурець О. В. – канд. техн. наук, доцент кафедри комп’ютерних наук Хмельницького національного університету, м. Хмельницький, Україна.

АНОТАЦІЯ

Актуальність. Вирішується актуальна проблема ефективної інтелектуальної діагностики несправностей обертового обладнання. Об’єктом дослідження є процес зменшення розмірності даних та класифікації дефектів за аналізом вібрацій для технічного обслуговування обертових машин. Предметом дослідження є методи зменшення розмірності та класифікації дефектів за аналізом вібрацій.

Мета роботи – створення підходу до зменшення розмірності даних та класифікації дефектів за аналізом вібрацій для технічного обслуговування обертових машин

Метод. Запропоновано комплексний підхід до зменшення розмірності даних та класифікації дефектів за аналізом вібрацій, що вирішує задачу зменшення розмірності даних для навчання класифікаторів та класифікації дефектів, а також вирішено задачу побудови нейромережевого класифікатора, що спроможний забезпечити швидкість класифікації несправностей без втрати точності на даних зменшеної розмірності. Підхід відрізняється від існуючих можливістю застосування опційних операторів об’єднання та перетину при формуванні множини значущих ознак, що надає гнучкість та дозволяє адаптуватись до різних контекстів та типів даних, забезпечуючи ефективність класифікації у випадках даних великої розмірності.

Метод для очищення від шуму дозволяє зберігати важливу інформацію, уникаючи надлишковості та покращуючи якість даних для подальшого аналізу. Він передбачає обчислення співвідношення сигнал/шум, встановлення порогових значень та застосування швидкого перетворення Фур’є, що забезпечує виокремлення релевантних ознак від шумів. Використання методу LIME до множини моделей машинного навчання дозволяє визначити значущі ознаки з більшою точністю та інтерпретованістю. Це сприяє отриманню надійніших результатів, оскільки LIME допомагає зрозуміти вплив кожної ознаки на кінцеве рішення моделі, що особливо важливо при роботі з великими датасетами, де важливість окремих ознак може бути неочевидною. Впровадження опційних операторів об’єднання та перетину значущих ознак надає додаткову гнучкість у виборі підходу до визначення важливих ознак. Це дозволяє адаптувати метод до різних контекстів та типів даних, забезпечуючи ефективність навіть у випадках з великою кількістю ознак.

Результати. Розроблений метод реалізовано програмно і досліджено при вирішенні задачі класифікації дефектів за аналізом вібрацій для технічного обслуговування обертових машин.

Висновки. Проведені експериментальні дослідження підтвердили високу ефективність та працездатність запропонованого підходу для зменшення розмірності даних та класифікації дефектів за аналізом вібрацій в аспекті технічного обслуговування обертових машин. Перспективи подальших досліджень будуть направлені на пошуки альтернативних нейромережевих архітектур та їх навчання для зниження часу навчання.

КЛЮЧОВІ СЛОВА: зменшення розмірності, значущі ознаки, виявлення дефектів, MobileNetV2.

ЛІТЕРАТУРА

1. Signal Processing for the Condition-Based Maintenance of Rotating Machines via Vibration Analysis: A Tutorial / [O. Matania, L. Bachar, E. Bechhoefer, J. Bortman] // Sensors. – 2024. – Vol. 24, No. 2. – P. 454. DOI: 10.3390/s24020454.
2. Mongia C. Vibration response-based condition monitoring and fault diagnosis of rotary machinery / [C. Mongia, D. Goyal, S. Sehgal] // Materials Today: Proceedings. – 2022. – Vol. 50. – P. 679–683. DOI: 10.1016/j.matpr.2021.04.395.
3. Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals / [B. A. Tama, M. Vania, S. Lee et al.] // Artificial Intelligence Review. – 2023. – Vol. 56. – P. 4667–4709. DOI: 10.1007/s10462-022-10293-3.
4. Implementation of Condition-based and Predictive-based Maintenance using Vibration Analysis / [C. Nata, N.

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8

Laurence, N. Hartono, L. Cahyadi] // 2021 4th International Conference of Computer and Informatics Engineering (IC2IE). – 2021. – P. 90–95. DOI: 10.1109/IC2IE53219.2021.9649400.
5. Iqbal M. Vibration and acoustic signal-based bearing fault diagnosis in CNC machine using an improved deep learning / M. Iqbal, A. K. Madan, N. Ahmad // Iran Journal of Computer Science. – 2024. DOI: 10.1007/s42044-024-00205-9.
6. Althubaiti A. Fault diagnosis and health management of bearings in rotating equipment based on vibration analysis – a review / A. Althubaiti, F. Elasha, J. A. Teixeira // Journal of Vibroengineering. – 2021. – Vol. 24, No. 1. – P. 46–74 DOI: 10.21595/jve.2021.22100.
7. Deep learning for diagnosis and classification of faults in industrial rotating machinery / [R. M. Souza, E. G. S. Nascimento, U. A. Miranda et al.] // Computers & Industrial Engineering. – 2021. – Vol. 153. – P. 107060. DOI: 10.1016/j.cie.2020.107060.
8. A novel fault diagnosis method of rotating machinery via VMD, CWT and improved CNN / [J. Gu, Y. Peng, H. Lu et al.] // Measurement. – 2022. – Vol. 200. – P. 111635. DOI: 10.1016/j.measurement.2022.111635.
9. Di Maggio L.G. Zero-Shot Generative AI for Rotating Machinery Fault Diagnosis: Synthesizing Highly Realistic Training Data via Cycle-Consistent Adversarial Networks / L.G. Di Maggio, E. Brusa, C. Delprete // Applied Sciences. – 2023. – Vol. 13, № 22. – P. 12458. DOI: 10.3390/app132212458
10. Ma C. Broad zero-shot diagnosis for rotating machinery with untrained compound faults / C. Ma, X. Wang, Y. Li, Z. Cai // Reliability Engineering & System Safety. – 2024. – Vol. 242. – P. 109704. DOI: 10.1016/j.ress.2023.109704.
11. Fault Diagnosis Method for Rotating Machinery Based on Multi-scale Features / [R. Liang, W. Ran, Y. Chen et al.] // Chinese Journal of Mechanical Engineering. – 2023. – Vol. 36. – P. 141. DOI: 10.1186/s10033-023-00966-7.
12. Fast Fourier Transform (FFT) – Python Numerical Methods [Electronic resource]. – Mode of access: https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter24.03-Fast-Fourier-Transform.html.
13. Method for neural network detecting propaganda techniques by markers with visual analytic / [I. Krak, O. Zalutska, M. Molchanova et al.] // CEUR Workshop Proceedings . – 2024. – Vol. 3790. – P. 158–170 [Electronic resource]. – Access mode: https://ceur-ws.org/Vol-3790/paper14.pdf.
14. Abusive Speech Detection Method for Ukrainian Language Used Recurrent Neural Network / [I. Krak, O. Zalutska, M. Molchanova et al.] // CEUR Workshop Proceedings. – 2023. – Vol. 3387. – P. 16–28. DOI: 10.31110/COLINS/2024-3/002.
15. Hasan M. M. Understanding Model Predictions: A Comparative Analysis of SHAP and LIME on Various ML Algorithms / M. M. Hasan // Journal of Scientific and Technological Research. – 2024. – Vol. 5, No. 1. – P. 17–26. DOI: 10.59738/jstr.v5i1.23(17–26).eaqr5800.
16. Intrusion Detection based on clustering with ML algorithm and LIME Insights / [A. Baniya et al.] // 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS). – Raichur, India, 2024. – P. 1–6. DOI: 10.1109/ICICACS60521.2024.10498512.
17. Pham M. T. Deep Learning-Based Bearing Fault Diagnosis Method for Embedded Systems / M. T. Pham, J.-M. Kim, C. H. Kim // Sensors. – 2020. – Vol. 20, No. 23. – P. 6886. DOI: 10.3390/s20236886.
18. An Approach Based on the Visualization Model for the Ukrainian Web Content Classification / [V. Slobodzian, M. Molchanova et al.] // 2022 12th International Conference on Advanced Computer Information Technologies (ACIT). – 2022. – P. 400–405. DOI: 10.1109/ACIT54803.2022.9913162.
19. Scikit-learn: machine learning in Python – Scikit-learn 1.6.0 documentation [Electronic resource] // Scikit-learn: machine learning in Python – Scikit-learn 0.16.1 documentation. – Mode of access: https://scikit-learn.org/stable/.
20. Imbalanced-learn documentation – Version 0.13.0 [Electronic resource] // Welcome to imbalanced-learn documentation. – Mode of access: https://imbalanced-learn.org/stable/.
21. Local Interpretable Model-Agnostic Explanations (lime) – lime 0.1 documentation [Electronic resource] // Local Interpretable Model-Agnostic Explanations (lime) – lime 0.1 documentation. – Mode of access: https://lime-ml.readthedocs.io/en/latest/.
22. SciPy [Electronic resource] // SciPy. – Mode of access: https://scipy.org/.
23. TensorFlow [Electronic resource] // TensorFlow. – Mode of access: https://www.tensorflow.org.
24. NumPy [Electronic resource] // NumPy – Mode of access: https://numpy.org/.
25. Matplotlib – Visualization with Python [Electronic resource] // Matplotlib – Visualization with Python. – Mode of access: https://matplotlib.org/.
26. Gnuplot homepage [Electronic resource] // Gnuplot homepage. – Mode of access: https://gnuplot.sourceforge.net/.

© Molchanova M. O., Didur V. O., Mazurets O. V., 2025
DOI 10.15588/1607-3274-2025-1-8