

<http://doi.org/10.35784/iapgos.1834>

PERFORMANCE COMPARISON OF MACHINE LEARNING ALGORITHMS FOR PREDICTIVE MAINTENANCE

Jakub Gęca

Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Lublin, Poland

Abstract. *The consequences of failures and unscheduled maintenance are the reasons why engineers have been trying to increase the reliability of industrial equipment for years. In modern solutions, predictive maintenance is a frequently used method. It allows to forecast failures and alert about their possibility. This paper presents a summary of the machine learning algorithms that can be used in predictive maintenance and comparison of their performance. The analysis was made on the basis of data set from Microsoft Azure AI Gallery. The paper presents a comprehensive approach to the issue including feature engineering, preprocessing, dimensionality reduction techniques, as well as tuning of model parameters in order to obtain the highest possible performance. The conducted research allowed to conclude that in the analysed case, the best algorithm achieved 99.92% accuracy out of over 122 thousand test data records. In conclusion, predictive maintenance based on machine learning represents the future of machine reliability in industry.*

Keywords: machine learning, random forest, predictive maintenance, neural networks

PORÓWNANIE SKUTECZNOŚCI ALGORYTMÓW UCZENIA MASZYNOWEGO DLA KONSERWACJI PREDYKCYJNEJ

Streszczenie. *Skutki związane z awariami oraz niezaplanowaną konserwacją to powody, dla których od lat inżynierowie próbują zwiększyć niezawodność osprzętu przemysłowego. W nowoczesnych rozwiązaniach obok tradycyjnych metod stosowana jest również tzw. konserwacja predykcyjna, która pozwala przewidywać awarie i alarmować o możliwości ich powstawania. W niniejszej pracy przedstawiono zestawienie algorytmów uczenia maszynowego, które można zastosować w konserwacji predykcyjnej oraz porównanie ich skuteczności. Analizy dokonano na podstawie zbioru danych Azure AI Gallery udostępnionych przez firmę Microsoft. Praca przedstawia kompleksowe podejście do analizowanego zagadnienia uwzględniające wydobywanie cech charakterystycznych, wstępne przygotowanie danych, zastosowanie technik redukcji wymiarowości, a także dostrajanie parametrów poszczególnych modeli w celu uzyskania najwyższej możliwej skuteczności. Przeprowadzone badania pozwoliły wskazać najlepszy algorytm, który uzyskał dokładność na poziomie 99,92%, spośród ponad 122 tys. rekordów danych testowych. Na podstawie tego można stwierdzić, że konserwacja predykcyjna prowadzona w oparciu o uczenie maszynowe stanowi przyszłość w zakresie podniesienia niezawodności maszyn w przemyśle.*

Słowa kluczowe: uczenie maszynowe, losowy las, konserwacja predykcyjna, sieci neuronowe

Introduction

Today's industry is facing new problems associated with constant growth of production as well as higher accuracy and safety requirements. In addition, international market is very competitive in terms of prices. These prices are highly dependent on production speed and reliability. Machines and automats are very important parts of a manufacturing process. It means that if certain component fails, it will cause financial losses related to downtime of the production process. Moreover, some failures may lead to the safety violations, which of course are far more undesirable.

To avoid unwanted danger and financial losses, many maintenance strategies are used in the industry. According to the Susto et al. [23] maintenance approaches can be classified as follows:

- Corrective maintenance (also Run-to-Failure – R2F) – this method consists of replacing or fixing a certain component after it fails. It is the most straightforward approach, which is also the most ineffective one. It leads to the additional costs associated with downtime and unscheduled maintenance, often including spare parts delivery interval.
- Preventive maintenance (PvM) – where maintenance interventions are performed regularly to avoid unscheduled stoppages. Time duration between conservations is based on knowledge about certain system component, but do not grant full usage of their life. Thus scheduled maintenance may cause additional costs related to unnecessary repairs.
- Predictive maintenance (PdM) – the goal of PdM is to forecast failures before they occur. It is possible thanks to the monitoring and data acquisition systems, which provides useful information about history of the machine and its current state. Predictions are based on historical data, defined health factors, engineering approaches and statistical inference methods.

Machine learning algorithms are proved to be very effective in terms of failure prediction and remaining useful life (RUL) estimation [10,17,24]. They can also be used in wide range of industry applications such as engine soot emission prediction [18], gearbox failure prediction [11], robotic manipulation failures forecasting [20]. Moreover, predictive models are very popular in other fields of technology. In [22] authors used decision tree algorithm for hard disc drive failure prediction. Korvesis et al. [15] predicted

failures from post flight reports using random forest and support vector machine (SVM). Despite the fact, that machine learning methods are often utilized and gives good results, scientists are still working on some other interesting techniques [2,13]. Some of the forecasting tasks are complicated and struggle because of the missing maintenance history or other type of data so authors in [6] proposed a hybrid semi-supervised approach. Kanawaday and Sane [12] came up with idea to firstly predict production cycle parameters with ARIMA (AutoRegressive Integrated Moving Average) model and then feed supervised classifier with these values.

This paper presents a comprehensive approach to the predictive maintenance, where performance of eight machine learning algorithms with tuned parameters was compared. To the best of author's knowledge and according to [4] there is no such work in the literature.

1. Data structure and preprocessing

The data comes from Microsoft Azure AI Gallery and is dedicated for predictive maintenance modelling [25]. It consists of five datasets that contains useful information about a group of identical industrial machines. Every machine has its own identification number which indicates a model and age of the machine. First dataset includes real-time telemetry data, that is timestamp, voltage, rotation, pressure and vibration values. Error messages are in the second dataset. The rest of the datasets contains information about machines, maintenance history (timestamp and replaced component ID) and failures (timestamp, broken component ID).

Preprocessing starts with feature engineering, which is important to extract maximum of the useful information from the data. First of all, it should be determined how far back the algorithm should "look" in order to predict failures. It is so-called lookback parameter, because it is used to create lag features that constitute short term history of the machine. The width of this time window have to be discussed with an expert in a particular field. It is also very important to remember that if this time is too long, the data will be too noisy for algorithm to predict with satisfactory performance. On the other hand, if the time window is too small, it will contain too little information to determine the risk of

failure. Further research about lookback parameter is out of scope of this work but 24h time window was chosen.

Creating lag features for telemetry data consists of calculating mean and standard deviation for every third record in the dataset. Next, to capture a long term effect, mean and standard deviation of last 24 hours is also calculated.

The error dataset contains timestamp and error message ID number for every machine. The amount of errors of every type in 24h lag window have to be calculated in order to find out what impact on failure probability it has.

Maintenance history is one of the most important datasets, so it is crucial for company to build system that collects such data. It is used to calculate the amount of days since last replacement of a certain machine asset, which provides very useful information about its degradation level.

Finally all the datasets (including machine and failure information) are merged together and prepared for labelling i.e. marking as a class that says whether or not the fault has occurred. But that is not the only way to do it. The authors in [19] have considered each of the devices and their components separately, labelling them as faulty or not. Thibaux et al. [5] decided to distinguish between three classes: „impending failure detected”, „not impending failure detected” and “uncertain about future failure”. In the case of this work, it was decided to consider the issue as a multi-class classification problem where it will be anticipated which of the four components will fail or none. In addition, it was assumed that the prediction would take place 24 hours in advance, although this time should generally be chosen in terms of maintenance time and spare parts availability. It means that each data record located 24 hours before the fault is marked as “incoming failure of component number x” or “none” otherwise. Table 1 shows the structure of labelled data and sample values.

Table 1. Data structure

Feature	Example	
machine_ID	22	
datetime	2015-09-29 18:00:00	
volt_mean_3h	171.27	
rotate_mean_3h	493.40	
pressure_mean_3h	112.35	
vibration_mean_3h	39.654	
volt_std_3h	9.4917	
rotate_std_3h	10.984	
pressure_std_3h	6.4264	
vibration_std_3h	5.5001	
⋮		
number of errors	error1	0
	error2	0
	error3	0
	error4	0
	error5	0
days since last component replacement	component1	26
	component2	11
	component3	41
	component4	56
Model	1	
Age	14	
Failure	none	

Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network requires data input shape in the form of (batch_size, timesteps, features). In order to obtain such three-dimensionality it is necessary to run an algorithm that will generate a 24-hour machine history (as an additional dimension) for each labelled data record.

The next step is to process the categorical data, which consists of mapping the ordinal features and encoding nominal features as well as class labels. Then the dataset is split into training and test subsets. Each data record has a corresponding point in time, so no random splitting or random sampling method can be utilized. The reason for this is that past events cannot be predicted based on future events (which might happen when using some methods). It is also unacceptable to use data that has arisen later than the point under consideration. Hence, a time-dependent splitting method has been applied by selecting one point in time as a division point and

ignoring the records for 24 hours ahead. This eliminates the risk of information (created during labelling) leakage between these subsets. As a splitting point, 2015-07-31 1:00:00 was chosen and the 60:40 ratio between training and testing data was obtained.

It is proven that data preparation techniques such as normalization and standardization have a positive impact on the performance of prediction models [8]. Thus, data was standardized before model training.

2. Prediction models and validation

One of the breakthroughs in machine learning was the development of the perceptron learning rule :

$$\Delta w_i = \eta(y_i - y'_i)x_i \quad (2)$$

by F. Rosenblatt [21]. In the formula above, η is the learning rate (0÷1), y_i – true class label of the i-th sample, y'_i denotes the predicted class label and x_i is the corresponding input value.

Table 2. The best parameters

Classifier	Parameter	Best value
Logistic regression	dimensionality reduction	LDA
	C	10
	penalty	l2
	solver	newton_cg
	imbalance handling	ENN
Decision Tree	dimensionality reduction	GUS
	criterion	entropy
	max_depth	10
	max_features	None
	min_impurity_decrease	0.0001
SVM	dimensionality reduction	Tomek's links
	C	LDA
	kernel	10
	imbalance handling	linear
Random forest	dimensionality reduction	-
	criterion	GUS
	max_depth	entropy
	min_impurity_decrease	10
	n_estimators	0.0001
Gradient boosting	dimensionality reduction	500
	learning_rate	Tomek's links
	loss	RFE
	n_estimators	0.01
	subsample	deviance
ANN	dimensionality reduction	500
	topology	0.5
	units	-
	activation	one hidden layer
	kernel_initializer	50
	optimizer	relu
	loss	lecun_uniform
	epochs	adam
	batch_size	sparse_categorical_crossentropy
	imbalance handling	10
CNN	dimensionality reduction	32
	topology	-
	units in layer 1	two convolution layers
	units in layer 2	75
	layer 1 kernel_size	50
	layer 2 kernel_size	2
	activation	2
	kernel_initializer	elu
	optimizer	glorot_uniform
	loss	adam
LSTM	dimensionality reduction	sparse_categorical_crossentropy
	topology	10
	units	32
	activation	-
	recurrent_activation	one recurrent layer
	kernel_initializer	75
	recurrent_initializer	relu
	imbalance handling	sigmoid

The idea of updating weights has led researchers to develop more sophisticated models. In this article, the following classification algorithms have been used for failure prediction: logistic regression, support vector machines (SVM), decision tree, random forest, gradient boosting classifier, artificial neural network (ANN), convolutional neural network (CNN), long short-term memory (LSTM).

Initially, the performance of algorithms with default parameters was tested for a different splitting and dimensionality reduction method. It appeared that the training and test data proportions has only a small impact on the accuracy, so for further research the 60:40 split was used. The next step was to investigate the influence of dimensionality reduction techniques such as principal component analysis (PCA) [9], linear discriminant analysis (LDA), generic univariate select (GUS), and recursive feature elimination (RFE) on prediction accuracy. For each algorithm, the method giving the best results was selected for further research. As a consequence, each prediction model has been prepared for the parameter tuning process.

Selection of the best parameters values was made by applying a grid search algorithm, which consists of finding the best result for every parameter combination from the grid. This method usually involves using the k-fold cross-validation with random sampling, which is unacceptable in this case. Therefore, the time-series splitting method for grid search was used to avoid overestimating the performance. Furthermore, for neural networks, the best topology was chosen by manual testing and comparison. The best parameters for each model are listed in Table 2.

In predictive maintenance, there is another important factor affecting performance. Failures are very rare occurrences among telemetry data, which leads to imbalance in the label distribution. Hence, the classifier tends to perform better predicting majority class labels than the minority. There are many solutions to this problem. Among others, undersampling can be used as authors in [3]. Alternatively, class weighting can be applied to increase or decrease algorithm's sensitivity towards specific classes. In this work, the methods proposed in [16] were used, especially Tomek's links and edited nearest neighbours (ENN). However, oversampling was not utilized because the number of newly added samples was unreasonably large compared to the efficiency improvement and the model training time became too long.

3. Results

Table 3 summarises performance metrics of each model for default parameters and once the data preprocessing and parameter tuning have been applied. The aforementioned label distribution imbalance has also an impact on how the model is evaluated. Failure-free records represent the vast majority in the dataset [14] so the algorithm can predict only a few faults while still maintaining high accuracy. Therefore, other performance metrics such as precision, recall and f1-score should be taken into account. These metrics are based on a number of positive and negative hypotheses, so for multi-class predictions, their macro averages are calculated. The results for CNN and LSTM algorithms confirm the problem described above. The accuracy values exceed 99% in both cases but the other metrics are much lower.

On the basis of the presented results, it would seem that there is no need for sophisticated methods of data preparation and parameter selection, since the results are very good and their improvement is only a fraction of a percent. However, considering a company that would like to use such a system, any incorrect forecast can cost a lot of money, so it is reasonable to refine the algorithms to the perfection. The performance metrics of the gradient boosting (marked in the table) seem to be particularly interesting, as they have slightly deteriorated after the model improvement. Nonetheless, the overfitting has decreased, so the risk of worse behaviour towards new, previously unseen data will be lower.

Table 3. Performance metrics

Classifier	Initial metrics values				Final metrics values			
	accuracy	f1-score	precision	recall	accuracy	f1-score	precision	recall
Logistic regression	99.80	94.33	93.96	94.71	99.82	95.04	93.35	96.85
Decision Tree	99.87	96.32	95.55	97.21	99.88	96.49	96.09	96.97
SVM	99.83	95.72	95.20	96.33	99.85	96.07	94.24	98.12
Random forest	99.92	97.74	98.56	96.95	99.93	97.92	98.74	97.14
Gradient boosting	99.93	97.78	98.01	97.57	99.92	97.76	98.02	97.52
ANN	99.83	95.56	94.13	97.09	99.88	96.82	96.04	97.66
CNN	97.72	62.01	57.44	75.59	99.50	88.02	92.63	84.06
LSTM	98.82	68.86	78.66	63.65	99.61	90.38	94.59	86.69

Among the introduced methods, three that achieved the best results in prediction of defects were chosen and their confusion matrices were presented (Fig. 1-3). Comparison of these matrices and explicit selection of the best algorithm involves determining several requirements related to functioning of the system to which the application is dedicated. First of all, it is important to specify how expensive are the so-called false alarms, i.e. situations in which the model predicts a failure, when it does not actually occur. In addition, it is necessary to determine how harmful it will be to forecast failure of one component instead of another. Of course, not detecting the upcoming malfunction is the worst case scenario, because unscheduled maintenance is the most expensive and avoiding it is desirable.

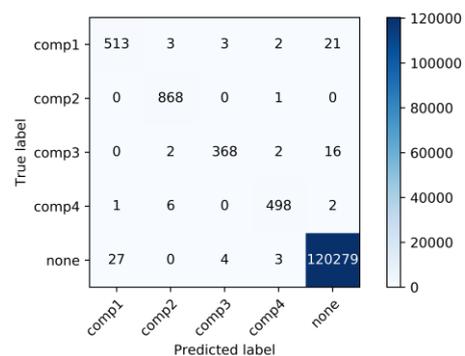


Fig. 1. Gradient boosting confusion matrix

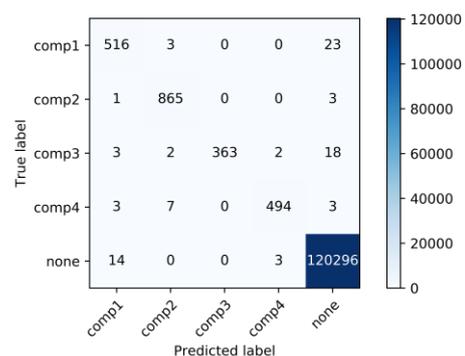


Fig. 2. Random forest confusion matrix

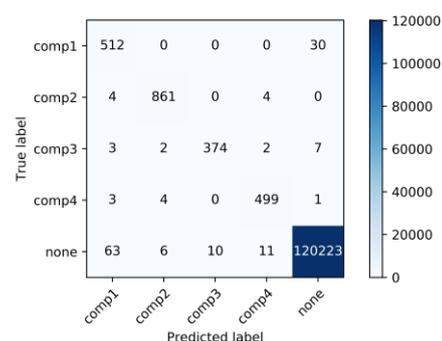


Fig. 3. ANN confusion matrix

It can be seen that the gradient boosting algorithm and the neural network have a similar number of undetected faults, but the latter has falsely alarmed up to 90 times. The least such situations occurred when using the random forest, but it was less effective in detecting faults. For this work, the gradient boosting algorithm has been chosen as the best because it provides the least undetected faults while maintaining a reasonable number of false alarms. The ROC (receiver operating characteristic) curves (Fig. 4) for this model show that failures of components 1 and 4 are detected with almost ideal efficiency, in contrast to failures of components 2 and 3, for which the area under the ROC curve (AUC) is even smaller than for none.

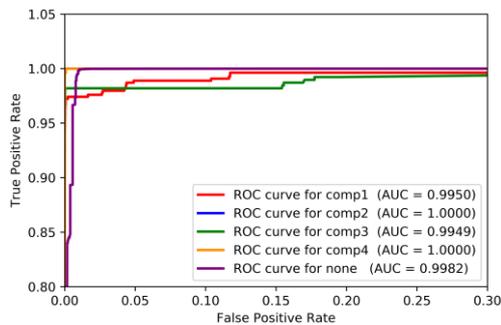


Fig. 4. Gradient boosting ROC curves.

4. Discussion

The conducted research shows that machine learning algorithms, in particular, gradient boosting, random forest and ANN, give the best results in prediction of industrial machines failures. In predictive maintenance, selection of the best algorithm is based on a financial analysis of the problem. In presented case, the gradient boosting was chosen. It obtained an accuracy of 99.92%, falsely alarmed 34 times and did not detect 39 faults out of over 122 thousand test data records. This leads to the conclusion that predictive maintenance based on machine learning is the future of many industry sectors. However, it requires an appropriate early warning system. An example of such application can be found in [1]. In addition, if maintenance of one component is much more expensive than the others, it will be possible to use a class weighting mechanism. As a result, the algorithm will be more sensitive to a failure of certain component.

The final performance of the production process depends on the proper data preparation, algorithm selection and parameter tuning. This article presents a comprehensive approach to the predictive maintenance issue considering all of these elements, as well as the evaluation criteria for such a system.

At the selection stage of artificial neural network topology, it was found out that the best results are obtained with one hidden layer. Thus, the classified data are approximately linearly separable, which is also confirmed by the fact that the SVM algorithm obtained the best results with a linear kernel. Nonetheless, in order to achieve the maximum possible predictive performance, the hidden data correlations need to be further examined.

References

- [1] Binding A., et al.: Machine Learning Predictive Maintenance on Data in the Wild. IEEE 5th World Forum on Internet of Things (Wf-Iot), 2019, 507–512.
- [2] Burnaev E.: On Construction of Early Warning Systems for Predictive Maintenance in Aerospace Industry. Journal of Communications Technology and Electronics 64/2019, 1473–1484, [https://doi.org/10.1134/S1064226919120027].
- [3] Campos J. R., et al.: Exploratory Study of Machine Learning Techniques for Supporting Failure Prediction. 14th European Dependable Computing Conference (EDCC), 2018, 9–16, [https://doi.org/10.1109/EDCC.2018.00014].
- [4] Carvalho T.P., et al.: A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance. Computers & Industrial Engineering 137/2019, 106024, [https://doi.org/10.1016/j.cie.2019.106024].

- [5] Chigurupati A., et al.: Predicting Hardware Failure Using Machine Learning. 2016 Annual Reliability and Maintainability Symposium (RAMS), 2016, 1–6, [https://doi.org/10.1109/RAMS.2016.7448033].
- [6] Cho S., et al.: A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future. Advances in Production Management Systems: Smart Manufacturing for Industry 4.0 – APMS 2018, 536/2018, 311–317, [https://doi.org/10.1007/978-3-319-99707-0_39].
- [7] Corazza A., et al.: A Machine Learning Approach for Predictive Maintenance for Mobile Phones Service Providers. Advances on P2P, Parallel, Grid, Cloud and Internet Computing 1/2017, 717–726, [https://doi.org/10.1007/978-3-319-49109-7_69].
- [8] Dzierżak R.: Comparison of the Influence of Standardization and Normalization of Data on the Effectiveness of Spongy Tissue Texture Classification. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 9/2019, 66–69, [https://doi.org/10.35784/iapgos.62].
- [9] Garcia S., et al.: Data Preprocessing in Data Mining. Data Preprocessing in Data Mining 72/2015, 1–320, [https://doi.org/10.1007/978-3-319-10247-4].
- [10] Gutsch C., et al.: Log-Based Predictive Maintenance in Discrete Parts Manufacturing. 12th Cirp Conference on Intelligent Computation in Manufacturing Eng. 79/2019, 528–533, [https://doi.org/10.1016/j.procir.2019.02.098].
- [11] Jiang R., et al.: Failure Prediction Method of Gearbox Based on Bp Neural Network with Genetic Optimization Algorithm. International Conference on Renewable Power Generation – RPG 2015, 2015, 1–3, [https://doi.org/10.1049/cp.2015.0444].
- [12] Kanawaday A., Sane A.: Machine Learning for Predictive Maintenance of Industrial Machines Using Iot Sensor Data. 2017, 87–90, [https://doi.org/10.1109/ICSESS.2017.8342870].
- [13] Khalil M.: Failure Prediction of Pv Inverters under Operational Stresses. IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2019, 1–5, [https://doi.org/10.1109/EEEIC.2019.8783241].
- [14] Kolokas N., et al.: Forecasting Faults of Industrial Equipment Using Machine Learning Classifiers. 2018 Innovations in Intelligent Systems and Applications (Inista), 2018, 6.
- [15] Korveis P., et al.: Predictive Maintenance in Aviation: Failure Prediction from Post-Flight Reports. IEEE 34th International Conference on Data Engineering (ICDE), 2018, 1414–1422, [https://doi.org/10.1109/ICDE.2018.00160].
- [16] Lemaitre G., Nogueira F., Aridas C.: Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. 18/2016.
- [17] Masani K.L., et al.: Predictive Maintenance and Monitoring of Industrial Machine Using Machine Learning. Scalable Computing-Practice and Experience 20(4)/2019, 663–668, [https://doi.org/10.12694/scpe.v20i4.1585].
- [18] Mishra K., et al.: Failure Prediction Model for Predictive Maintenance. 7th IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2018, 72–75, [https://doi.org/10.1109/ccem.2018.00019].
- [19] Parisi L., Ravi Chandran N.: Genetic Algorithms and Unsupervised Machine Learning for Predicting Robotic Manipulation Failures for Force-Sensitive Tasks. 4th International Conference on Control, Automation and Robotics (ICCAR), 2018, 22–25, [https://doi.org/10.1109/ICCAR.2018.8384638].
- [20] Rosenblatt F.: The Perceptron, a Perceiving and Recognizing Automaton Project Para. Cornell Aeronautical Laboratory, 1957. Report: Cornell Aeronautical Laboratory.
- [21] Rymarczyk T., et al.: Analysis of Data from Measuring Sensors for Prediction in Production Process Control Systems. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 9(4)/2019, [https://doi.org/10.35784/iapgos.570].
- [22] Schaub M.: Data-Based Prediction of Soot Emissions for Transient Engine Operation. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 9(4)/2019, [https://doi.org/10.35784/iapgos.29].
- [23] Suchatpong T., Bhummittipich K.: Hard Disk Drive Failure Mode Prediction Based on Industrial Standard Using Decision Tree Learning. 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014, 1–4, [https://doi.org/10.1109/ECTICon.2014.6839839].
- [24] Susto G.A., et al.: Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. IEEE Transactions on Industrial Informatics 11(3)/2015, 812–820, [https://doi.org/10.1109/TII.2014.2349359].
- [25] https://gallery.azure.ai/Experiment/Predictive-Maintenance-Implementation-Guide-Data-Sets-1 (available: 24.04.2020).

M. Sc. Jakub Gęca
e-mail: j.geca@pollub.pl

He is in the first year of doctoral studies at the Lublin University of Technology, where he also graduated with a master's degree. His master's thesis was written at the Faculty of Electrical Engineering and Computer Science. He is cooperating with the Department of Electric Drives and Machines and with the Department of Computer Science. His scientific interests include machine learning, industrial automation, photovoltaic and electric drive control systems

<https://orcid.org/0000-0003-0610-2975>

otrzymano/received: 8.05.2020

przyjęto do druku/accepted: 15.09.2020

