Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 79 (2019) 528–533

www.elsevier.com/locate/procedia

12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy

Log-based predictive maintenance in discrete parts manufacturing

Clemens Gutschi[a,]*, Nikolaus Furian[a], Josef Suschnigg[a], Dietmar Neubacher[a], Siegfried Voessner[a]

[a]*Department of Engineering- and Business Informatics, Graz University of Technology, Kopernikusgasse 24/III, Graz 8010, Austria*

* Corresponding author. Tel.: +43-316-873-8007; fax: +43-316-873-108007. *E-mail address:* clemens.gutschi@tugraz.at

Abstract

The performance of discrete parts manufacturing systems is heavily influenced by unplanned machine breakdowns. Predictive maintenance allows for the conversation of unplanned machine breakdowns to scheduled corrective maintenance actions. We present a data-driven approach for estimating the probability of machine breakdown during specified time interval in the future. Machine learning algorithms are utilized for a specific use-case which is based on real-world data-sets including machine log messages, event logs and operational information. The paper describes applied data-mining, feature-extraction and machine learning methods and concludes with results indicating that machine failures can be reliably predicted up to 168 hours in advance.

Keywords: Predictive maintenance; Probability of failure estimation; Remaining useful life; Random forrest; Ensemble prediction

1. Introduction

Manufacturing industry operate in global markets dealing with strong competition, fluctuating market conditions, and new technologies. To be profitable, companies are undertaking much effort to improve quality and productivity, as well as to secure a high level of reliability in order to reduce costs. Therefore, flexible production systems and the task of production planning and control are becoming more and more relevant. In terms of improvements and a company's performance, maintenance strategies play a key role. An effective maintenance strategy improves equipment availability and extends equipment life. In contrast, poor maintenance operations decrease equipment availability by allowing more frequent breakdowns, which lead to delays in production scheduling and may result in scrap. Summarizing, the performance of a company is significantly influenced by its maintenance strategies in place. Historically, reactive maintenance or 'fire-fighting', where equipment is just repaired when it breaks down, was the main maintenance approach deployed by most companies. In contrast, the scope of a proactive maintenance strategy is to prevent equipment to suffer from breakdowns. This includes preventive and predictive maintenance strategies. Breakdowns are avoided by monitoring equipment information (e.g. deterioration) to assess the equipment state and executing maintenance operations before the equipment state exceeds a predefined threshold (e.g. a wear limit). [1, 2]

Preventive maintenance (PM) strategies trigger maintenance operations in predefined intervals or evaluate prescribed criteria in order to reduce the probability of failures. While time-based maintenance monitors the operating time or produced parts since the last maintenance operation, condition-based maintenance uses certain measurement information about the physical condition of the equipment such as vibration, temperature, or noise. [2, 3, 5]

Predictive maintenance (PdM) strategies apply prognostic models to forecast the equipment's condition. Sufficiently accurate forecasting models of the failure modes are derived from repeated analysis and evaluation of collected data. Such models are trained to predict the equipment's health, probability of failures, or a remaining useful lifetime. Collected

2212-8271 © 2019 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering.
10.1016/j.procir.2019.02.098

data may include sensor data used for PM or event-log information produced by a machine control or IT-system [3, 5].

In general, maintenance strategies can be applied on any type of equipment. However, the area of application of high-level proactive strategies has practical limitations, as operating costs and costs for gathering knowledge and data for prediction models increase rapidly. While the generation of knowledge about statistical life information is relatively simple, generating accurate mathematical or physical models for predictions is very work- and time-intensive or even not possible. [4]

This paper presents a data-driven approach for estimating the probability of milling machine breakdowns during a specified time interval in the future and proofs its validity on a real-world use case. The steps described give insights on data processing, feature extraction, machine learning, and estimations of the probability of machine breakdowns. The estimations are generated by an ensemble prediction approach, whose design and influence on the estimations is proposed in this paper.

This paper is organized as follows: section 2 presents commonly used approaches for PdM. Section 3 describes the log-based PdM approach containing data preparation, applied feature selection, different methods for remaining useful lifetime (RUL) estimation, and algorithms applied for evaluation. Section 4 deals with applied experiments and results. Finally, the paper concludes with a discussion of results and suggestions for further research.

2. Data-driven PdM in use

Following Peng et al. [7], Schwabacher & Goebel [8], and Lee et al. [9], prognostic models can be classified as model-based, data-driven or a combination of both. Furthermore, human expert knowledge may be added to any of these models. Like in condition based maintenance, model-based approaches are utilizing physics or expert knowledge to generate degradation models of a given equipment. The process of the equipment's degradation is monitored by sensors and maintenance is triggered when exceeding a given threshold. As sensor-based approaches can be relatively easy implemented on the basis of existing condition-based maintenance strategies, they are commonly used. Data-driven approaches use historic data and machine learning technologies fitting mathematical models to reproduce a system's behavior. In contrast to model-based approaches, data-driven approaches require high amount or complex data. [6, 7, 8, 9]

Data-driven models can be classified in sensor-based, log-based, and hybrid approaches. Sensor-based approaches make use of time-series signals of single or multiple sensors without physical models to assess the remaining useful lifetime (RUL). Sensor-based approaches are often applied in mechanical engineering, mainly on rotatory machines or components [10, 11, 12] such as bearings or gear-boxes. Log-based approaches use historical event-log data (instead of sensor data) to train machine learning algorithms. RUL estimation is implemented by a predetermined level of failure probability and applying the model on real-time event-log data. Event-log data can either be aggregated data from sensor streams or extracted data from log-messages like system messages, alarm codes, numerical values, or keywords. As only relatively small portions of these large datasets are related to PdM, data preprocessing with techniques to extract relevant features that can be used for breakdown prediction are crucial for success. Log-based approaches are mainly applied in IT-systems like ATM's [13, 14, 15].

Concluding, the major difference of log-based and sensor-based is the data source and consequently the system's behavior. Sensor-based PdM is a bottom-up approach which mainly monitors single components equipped with sensors. Whereas, log-based PdM is more like a top-down approach where many components can be monitored with the same data-stream. However, there may be components without any possibility to monitor. Furthermore, features mined for log-based PdM are often warnings which occur a certain period before a breakdown, thus this approach acts more like an early warning system than a continuous monitoring system.

3. A log-based PdM approach in discrete parts manufacturing

The log-based PdM approach introduced in this paper is designed for processing messages of programmable logic controllers (PLC) without using any additional sensors data. The equipment in scope are milling machines within a production line. The machine learning model to estimate the probability of machine breakdowns is based on historical PLC-data and the documentation of breakdowns.

The main steps of a log-based PdM approach are shown in Figure 1. The methodology for data preparation, feature selection, and the creation of the mathematical model will be described in this section. This section will conclude with presenting two different possibilities of RUL estimation.

3.1. Data preparation and feature extraction

Data preparation and feature extraction is performed in a two-step aggregation, a time window aggregation and a rolling window aggregation. Each aggregation step contains mathematical functions f for extracting features of the raw and first aggregated signals.

For the time window aggregation, all raw signals $\mathbf{r}$ of different time resolution are aggregated into equally sized time windows $\mathbf{tw}$ for all timespans ts within a dataset.

$$\mathbf{tw}_i = \left(f_\alpha(\mathbf{r}_1,\ldots,\mathbf{r}_n)\right)_{\alpha\in\{\text{isRaising},\,\text{countRaising},\,\text{isFalling},\,\text{countFalling}\}} \quad (1)$$

i … indices representing a ts within the dataset
n … number of raw signals in a time window

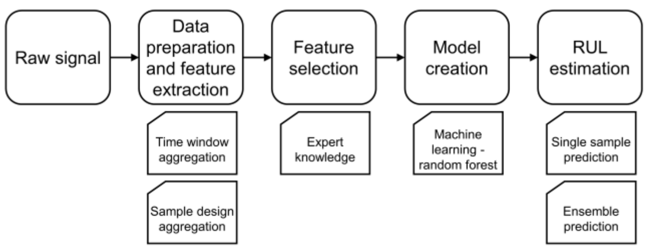

Fig. 1: Main steps of a log-based PdM approach; following [9, 10]

Equation (1) describes how raw signals are aggregated into **tw**. The given operators f_α, for example isRaising or isFalling are defined to be 1 if a log-message is opened or closed within the **tw**. In some cases, a message is occurring multiple times in a **tw**, therefore countRaising and countFalling evaluate the number of messages opened and closed within that **tw**.

The goal of the sample design aggregation is to construct datasets **S** ready for machine learning with related target values **y**. A single sample $\mathbf{s}_i$ is generated by aggregating time windows **tw** into a certain amount of concatenated rolling windows **rw**:

$$\mathbf{s}_i = \left[\mathbf{rw}_0^i, \ldots, \mathbf{rw}_{n_{rw}}^i\right], \tag{2}$$

$$\mathbf{rw}_k^i = \left(f_\beta\left(\mathbf{tw}_j\right)\right)_{\beta \in \{\min, \max, \text{average}, \text{diff}\}}, \tag{3}$$

with

$$lb_k^i \leq j < ub_k^i. \tag{4}$$

n_{rw} … number of rolling windows

The operators f_β in equation (3) aggregate **tw** signals into **rw** of larger size by the simple mathematical operations min, max, average and difference. Furthermore, rolling windows are unequally sized and as they are serial concatenated, time windows for aggregation are taken from previous timespans. The idea of non-equal sized rolling windows is implemented to weight their influence by their distance to the target value.

The target values y_i for each sample $\mathbf{s}_i$ are defined by RUL-classes (*RC*). Each *RC* represents a determined timespan of time to failure (*TTF*) from the sample $\mathbf{s}_i$ to the next breakdown,

$$y_i = RC\left(TTF_i\right). \tag{5}$$

3.2. Feature selection

For feature selection, two strategies were implemented. The first is selecting the 300 best features based on the ANOVA f-value. The second is based on expert knowledge of maintainers and manually examining data for clusters of messages occurring before incidents.

3.3. Model creation

A random forest (RF) is used as the machine learning method to create a classification model. RF is an ensemble learning method to construct a collection of individual classification or regression trees. For construction of a single classification or regression tree, the algorithm selects a subset of samples of the whole dataset **S**. Besides sampling on the dataset, trees are randomized by using bagging and boosting techniques to generate splits, see [16, 17].

3.4. RUL estimation

The scope of predictions is to estimate the probability of a breakdown *POB* in a determined timespan *RC*. This section presents the metric applied for evaluation and the RUL estimation methods.

Table 1: Overview of two different methods of RUL estimation

RUL estimation method	Scope of prediction	applied number of samples
single sample method	probability prediction	1 (sample $\mathbf{s}_i$)
ensemble method	majority votes of ensemble prediction	*ne* (samples $\mathbf{s}_i, \ldots, \mathbf{s}_{i-(ne-1)}$) *ne* … number of samples in the ensemble

Table 1 give an overview of the two methods.

3.4.1. Metric for evaluation

The proposed algorithm computes the probability of a breakdown in a future timespan *RC*. To evaluate its accuracy, we introduce a threshold th_{POB} to this probability above at which the algorithm is assumed to actually indicate a breakdown. Hence, we compute the hit-rate *h* as the number of correctly predicted breakdown's (true positives (*TP*)) divided by the number of actually occurred breakdowns $n_{breakdowns}$, and the precision *p* as the *TP* divided by all predicted breakdowns, i.e. including false positive (*FP)* [18]. *TP* and *FP* are computed with respect to the chosen *RC* and representing the number of positive predictions where a real breakdown occurred within the duration of the lower and twice the upper bound of *TTF* of the suggested *RC* or not,

$$p = \frac{TP}{TP + FP}, \tag{6}$$

$$h = \frac{TP}{n_{breakdowns}}. \tag{7}$$

True negatives and false negatives are not evaluated because the training data design is just using target values for upcoming breakdowns and not using target values for non-breakdowns. Thus, the prediction condition can just be positive. [18]

3.4.2. RUL estimation by the single sample method

This classical prediction method uses only a single sample $\mathbf{s}_i$ for prediction. $\mathbf{s}_i$ is built as given in equations (2-4). A RF predicts the class probabilities of all predefined *RC*'s. They are computed as the mean class probability *MCP* of all trees in the forest. Thus, the *POB* for a predefined *RC* is defined as the related *MCP* of the prediction result.

$$POB_i^{RC} := MCP^{RC}(\mathbf{s}_i) \tag{8}$$

3.4.3. RUL estimation by the ensemble method

This approach is based on RF, which generates many predictions for a single sample by using a large amount of trees generated by ensemble learning [17]. The ensemble method extends RF by generating predictions with a set or ensemble of samples along time. The idea behind creating an ensemble of samples is to include more information of previous samples $\mathbf{s}_{i-x}$ which are not concerned by the single sample method using $\mathbf{s}_i$ only. Furthermore, the prediction result along time is

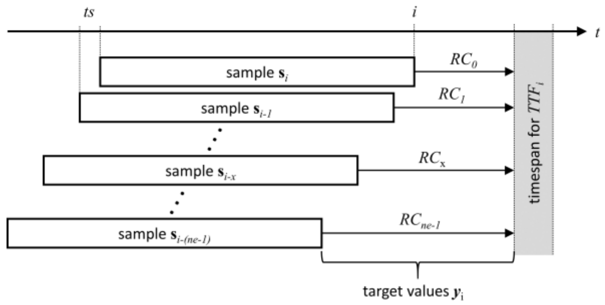

Fig. 2: Architecture of samples for the ensemble method

supposed to have less noise and fewer outliners by smoothening, which is done by averaging majority votes MV of single samples.

Fig. 2 shows the architecture of an ensemble consisting of ne samples $\mathbf{s}_{i-x}$ with corresponding target values. In contrast to the single sample method where the target values are the same for all samples along i, the target values within an ensemble have to be updated for each sample $\mathbf{s}_{i-x}$. The POB_i of the ensemble prediction method is evaluated by a majority vote MV of each sample $\mathbf{s}_{i-x}$ of an ensemble. The MV of a sample $\mathbf{s}_{i-x}$ is defined to be 1 if the decision of the RF meets the supposed RC_x and 0 otherwise:

$$MV^{RC_x}(\mathbf{s}_{i-x}) := \begin{cases} 1 \text{ if } MV(s_{i-x}) = RC_x \\ 0 \text{ otherwise} \end{cases}, \qquad (9)$$

$$POB_i^{RC} := \frac{1}{n_e}\sum_{x=0}^{n_e-1} MV^{RC_x}(\mathbf{s}_{i-x}). \qquad (10)$$

4. Use Case

The presented log-based PdM approaches of section 3 are applied on real-world data from a discrete parts manufacturer now. Therefore, seven identical milling machines were chosen as the data source with a data history of 180 days. The data collected are breakdown documentations and messages of PLC's. In total, 92 breakdowns were documented by the PLC and data included about 200.000 logged messages (of 2.540 different message types).

In the following, we describe all necessary steps of data preprocessing and RUL estimation. Furthermore, the experimental design and results will be outlined and discussed.

For data preparation, all messages are aggregated into time windows $\mathbf{tw}_i$ using a constant timespan $ts=2h$. The design of a sample $\mathbf{s}_i$ by concatenated rolling windows $\mathbf{rw}_k^i$ is given in Table 2 with $n_{rw}=5$ rolling windows inside a sample.

Table 2: Definition of rolling window sizes and positions for $\mathbf{rw}_k^i$ in hours

k	rolling window size	lower bound lb_k	upper bound ub_k
0	12	0	12
1	36	12	48
2	72	48	120
3	120	120	240
4	216	240	456

Table 3: Aggregation operators for feature extraction

Combination	f_α	f_β
1	isRaising	max
2	isRaising	average
3	countRaising	min
4	countRaising	max
5	countRaising	average
6	isFalling	max

Table 4: Definition of RUL-classes in hours

RC	TTF greater than [h]	TTF smaller [h]
1	0	24
2	24	48
3	48	84
4	84	168
5	168	336
6	336	∞

Applied aggregation operators for feature extraction are defined in Table 3 (see equation (1) and (3)). In total there are 76.200 features (2.540 message types; 6 aggregation operators; 5 rolling windows) created from all raw signals $\mathbf{r}$ (types of messages) along 2.160 timespans per machine. Knowledge-based feature selection converts 18 selected types of messages resulting in 108 features.

The number of RUL-classes and their bounds defined by TTF are set in Table 4.

4.1. Experiment design

This section provides information about data usage for model construction, experimental sets, applied metrics for model evaluation, and varied parameter for RF-optimization.

The dataset was divided for model construction and evaluation. Model validation was not executed because of the small number of machines and breakdowns. The dataset was split as following: the data of two machines were used for model construction and the data of the remaining five machines were used for evaluation.

The experiment-sets are defined by the applied RUL-estimation method and the feature selection method as determined in Table 5.

Each single experiment-set is evaluated 48 times by varying parameters for model construction. These are the number of trees n_{Trees} in the RF and the minimum number of samples

Table 5: Definition of methods for experiment-sets

		feature selection	
		ANOVA f-value	knowledge-based
RUL-estimation	single sample method	set a1	set k1
	ensemble method ne=120	set a2	set k2
	ensemble method ne=30	set a3	set k3

Table 6: Parameter variation within each experiment-set

parameter	variation
n_{Trees}	100, 200, 300
$min_{samples\_leaf}$	8, 4, 2, 1
th_{POB}	0.3, 0.4, 0.5, 0.6

needed to be a leaf $min_{samples\_leaf}$. Furthermore, each set is tested with different thresholds th_{POB} applied on the metric for evaluation. Within an experiment-set, all possibilities of combinations of all parameters are evaluated. The variation of all parameters is given in Table 6.

All experimental-sets are evaluated by $RC=3$, predicting failures between $48h \leq TTF < 86h$ in the future. Furthermore, the best set is evaluated for $RC=\{1, 2, 3, 4, 5\}$ to define the maximum forecasting horizon.

4.2. Results

In the following we show a comparison of results by the single sample and ensemble method, results generated by all experiment-sets for $RC = 3$ and the possible range of RC for estimating the POB.

Figure 3 illustrates the different POB-curves for the single sample and ensemble method. As suggested, the ensemble method gives a smoother curve. This is a result of averaging the majority votes of each sample.

Figure 4 and figure 5 present the results for RC=3 separated for ANOVA f-value and knowledge-based feature selection. The results of ANOVA f-value feature selection show a hit-rate h in between 0 to 0.5 and also a high variance on the precision p. This means, the results are very dependent on n_{Tree}, $min_{samples\_leaf}$, and th_{POB}. Increasing the threshold th_{POB} result in a drop of the hit rate, but there were no patterns found for differences in n_{Tree} and $min_{samples\_leaf}$. The dependence on th_{POB} can be explained by the higher possibility for exceeding a threshold at a lower value, as visualized in Figure 3. Experiment-set a2 (ensemble method with ne=120) is most independent of parameters and hence most stable. Furthermore, the hit-rate of set a2 is creating true positive estimations at any configuration (h>0.17) and the precision show a much smaller variance compared to set a1 and set a3. The results of knowledge-based feature selection show better performance in

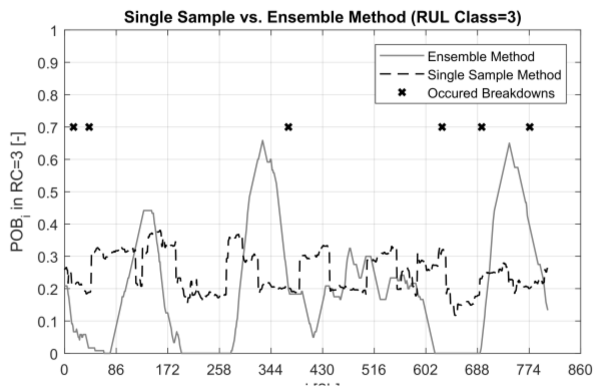

Fig. 3: Schematic POB of the single sample and ensemble method for $RC = 3$, $n_{Trees} = 200$ and $min_{samples\_leaf} = 8$

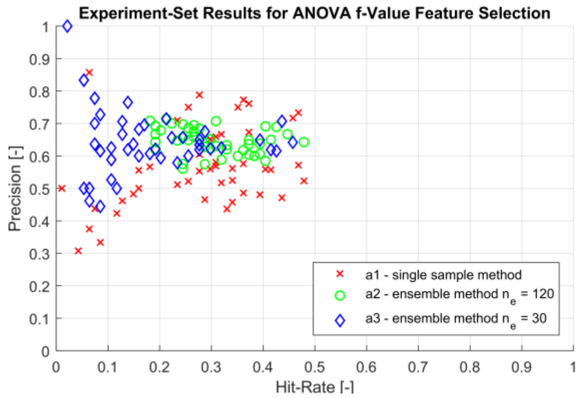

Fig. 4: Experiment results for ANOVA f-value feature selection

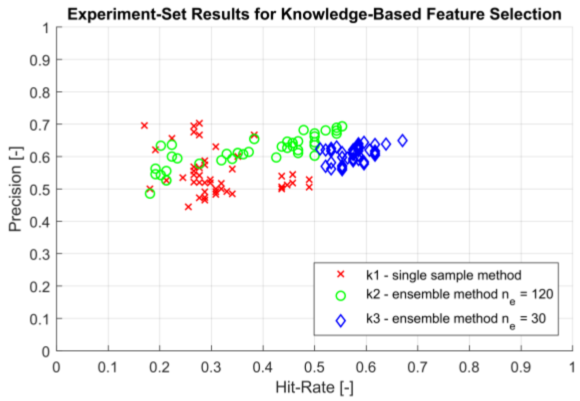

Fig. 5: Experiment results for knowledge-based feature selection

minimum and maximum hit-rate but perform worse in maximum precision.

The best result of all experiment-sets was found with k3 (ensemble prediction using ne=30). The minimum hit rate of k3 is even higher than the maximum hit rate of all experiments with ANOVA f-value feature selection but resulting in a little worse mean precision. Furthermore, this setting was very stable because of the low variance in hit rate and precision.

The maximum forecasting horizon is evaluated by the ensemble method using knowledge-based feature selection only. Figure 6 shows the result of the predictability with a drop of the hit-rate after RC=4, representing a TFF between 84h and 168h.

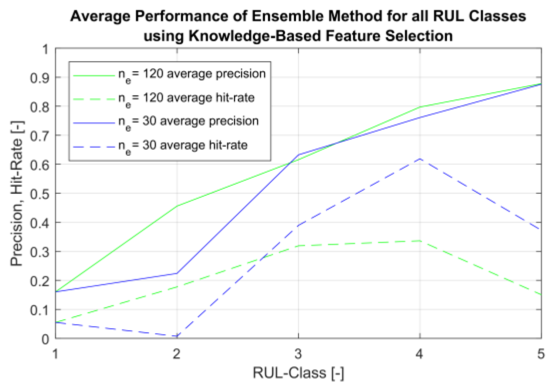

Fig. 6: Average performance of the ensemble method using knowledge-based feature selection

5. Conclusion and further research

This paper presents a data-driven log-based predictive maintenance approach which was applied and validated on a real-world use case. The performance of the introduced ensemble prediction method is in general more stable than a single sample method and in some cases outperforms it significant. The presented application on milling machines illustrates, that major breakdowns of these machines are predictable up to 168 hours in the future. Thus, many reactive maintenance operations can be precisely scheduled and resources as for example spare parts can be provided in the right place at the right time. Furthermore, maintenance costs can be optimized and costs for installing sensors and creating physical models can be avoided. On the other hand, the study showed that creating stable data-driven log-based models is very time consuming and its profitability has to be proven yet.

The ensemble prediction method utilizes a random forest as the machine learning algorithm. In future, adoptions to combine this method with other machine learning algorithms may be of scientific interest. Changing the aggregation of the *POB* is going to enable optimization on the number of samples in the ensemble across different machine learning algorithms or even combinations of them.

Comparing feature selection techniques showed a significant impact on the results regardless of the applied prediction method. Although, feature selection was not in the major scope of this paper, log-based PdM is highly influenced by the applied feature selection algorithm. Thus, further investigations should also consider feature selection in combination with other prediction methods into account.

References

[1] Tambe PP, Kulkarni MS. A superimposition based approach for maintenance and quality plan optimization with production schedule, availability, repair time and detection time constraints for a single machine. Journal of Manufacturing Systems 2015; Volume 37/1: 17–32.

[2] Swanson L. Linking maintenance strategies to performance. International Journal of Production Economics 2001; Volume 70/3: 237–244.

[3] Ojanen V. Maintenance innovations - Types, patterns and emerging trends. IEEE International Conference on Management of Innovation and Technology 2014; 321–326.

[4] Tran VT, Yang BS, Oh MS, Tan A. CC. Machine condition prognosis based on regression trees and one-step-ahead prediction. Mechanical Systems and Signal Processing 2008; Volume 22/5: 1179–1193.

[5] DIN EN 13306:2018-02, Maintenance - Maintenance terminology; Trilingual version (EN 13306:2017)

[6] Lee J, Wu F, Zhao W, Ghaffari M, Liao L, Siegel D. Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. Mechanical Systems and Signal Processing 2014; Volume 42/1–2: 314–334.

[7] Peng Y, Dong M, Zuo MJ. Current status of machine prognostics in condition-based maintenance: a review. The International Journal of Advanced Manufacturing Technology 2010; Volume 50/1-4: 297–313.

[8] Schwabacher M, Goebel K. A survey of artificial intelligence for prognostics. Association for the Advancement of Artificial Intelligence 2007; AAAI Fall Symposium 2007: 107–114.

[9] Lee J, Ni J, Djurdjanovic D, Qiu H, Liao H. Intelligent prognostics tools and e-maintenance. Computers in Industry 2006; Volume 57/6: 476–489.

[10] Benkedjouh T, Medjaher K, Zerhouni N, Rechak S. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Engineering Applications of Artificial Intelligence 2013; Volume 26/7: 1751–1760.

[11] Hameed Z, Hong YS, Cho YM, Ahn SH, Song CK. Condition monitoring and fault detection of wind turbines and related algorithms: A review. Renewable and Sustainable Energy Reviews 2009; Volume 13/1: 1–39.

[12] Kandukuri ST, Klausen A, Karimi HR, Robbersmyr KG. A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management. Renewable and Sustainable Energy Reviews 2016; Volume 53: 697–708.

[13] Wang J, Li C, Han S, Sarkar S, Zhou X. Predictive maintenance based on event-log analysis: A case study. IBM Journal of Research and Development 2017; Volume 61/1: 11:121-11:132.

[14] Wang C, Vo HT, Ni P. An IoT Application for Fault Diagnosis and Prediction. IEEE International Conference on Data Science and Data Intensive Systems 2015; 726-731.

[15] Zhang K, Xu J, Min MR, Jiang G, Pelechrinis K, Zhang H. Automated IT system failure prediction: A deep learning approach. IEEE International Conference on Big Data (Big Data) 2016; 1291-1300.

[16] Liaw A, Wiener M. Classification and Regression by randomForest. R News 2002; Volume 2/3: 18-22.

[17] Dietterich TG. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning 2000; Volume 40/2: 139–157.

[18] Powers DMW. Evaluation: From precision, recall and f-measure to roc., informedness, markedness and correlation. Journal of Machine Learning Technologies 2011; Volume 2/1: 1–24.