Vibration Based Milling Diagnostics Using Artificial Intelligence

Guillaume VINCHON[1,4], Dominique KNITTEL[1,2], Mohammed NOUARI[1]
Hamid MAKICH[3], Abderrahmane YAKOB[4], Daniel ROMANI[5]

[1]Univ Lorraine, LEM3, UMR7239, F-88100 St Dié des Vosges, France
[2]Univ Strasbourg, Faculty P&I, F-67000 Strasbourg, France
[3]Institut Mines-Telecom, LEM3, UMR7239, F-88100 St Dié des Vosges, France
[4]iDEMoov, F-67124 Entzheim, France
[5]md6, F-67124 Entzheim, France

abderrahmane.yakob@idemoov.fr
knittel@unistra.fr

Abstract

In industrial machining processes, tool failures may result in losses in surface and dimensional accuracy of a finished part, or possible damage to both the work piece and the machine. Consequently, tool condition monitoring has become essential to achieve high-quality machining as well as cost-effective production. Moreover, cutting tool degradation may vary considerably under different operation conditions and materials behaviour. Therefore, real time identification of the tool state during machining, before it reaches its failure stage, is critical.

In this study the vibrations of the cutting tool and of the workpiece material are online measured. Features are then calculated in time domain from the raw signals. Transient zones, when the cutting tool enters or exits the material, can be considered or not. All the calculated features are normalized and stored in a table. It is then necessary to make a dimensional reduction of that feature table in order to avoid overfitting and to reduce the computing time of the learning algorithm. In this study, 54 milling experiments were conducted from which features were calculated and then split into two groups: 80% for the machine learning model training, 20% for the test phase. The first part of this study proposes an analysis on the impact of the features on the robustness of the models, and a second part focuses on a real-time data driven prognostics and health management (PHM) approach for tool condition monitoring, based on supervised machine learning techniques (i.e. the model training needs labeled data). The fusion of decision coming from several machine learning algorithms (kNN, decision trees (DT) and random forest (RF)) is then used to predict the tool quality in real time. All parameters and configurations of the algorithms are optimized in order to maximize the real time diagnosis accuracy. The experimental results show that our proposed approach achieves good accuracy and real time performances in dry milling operations. Results of our study are implemented in real tool wear diagnosis, and thus give new opportunities toward realizing Industry 4.0.

1 Introduction

Prognostics and Health Management (PHM) is offering insights on the health of the assets and acts as a decision support for the different actors of the maintenance and production [1, 2]. In fact, predictive maintenance benefits greatly from the health indicators provided by PHM. Consequently, an effective predictive maintenance strategy provides information leading to an optimized on-time replacement of the faulty part.

The three main tasks of the PHM consist of fault detection (detection of abnormal or unknown behaviour), diagnosis / fault identification (State of the system, origin / cause of the faulty behaviour), fault prognostic (Prediction of the system's future sate).

In this paper, we focus on the supervised machine learning based approaches, based on the training of predictive models with labeled data. Features are computed from raw data, generating a dataset used for the model training. Once trained, the model can perform either a classification task or a regression task based on the chosen learning algorithm. The performances of such approaches are heavily reliant on the quality of the data both during the training and the application stage [9].

Several Machine learning based PHM approaches have been applied in the industry, Zhaoyi Xu *et al*. (2021) give a comparative study between machine learning classification and regression methods applied to the industry [6]. Sylvain Chabanet *et al.* (2021) applied an Active Learning method to optimize the lumber productions in sawmill industries. Their strategy relied upon the smart selection of instances, using a "digital twin", a simulation coupled with a kNN based Machine Learning model. At each new instance, they calculate criterion to determine when the simulation, which is more demanding in resources, must be called to make the prediction [10].

More specifically, machine learning algorithms have been used to monitor milling operations, and to diagnostic the tool's state. For example, Emiliano Traini *et al.* (2019) used machine learning algorithms such as Random Forest (RF), Boosted Decision Trees (BDT) and Artificial Neural Network (ANN) for the diagnosis (classification) and prognosis (regression) of the tool's condition during milling operations [7]. In another study, T. Mikołajczyk *et al.* developed an Artificial Neuronal Network (ANN) for tool-life prediction in machining with a high level of accuracy, especially in the range of high wear levels, which meets the industrial requirements [8].

However, the implementation of a good data acquisition method/system can become expensive and time consuming [16]. Saeed Shurrab *et al.* (2021) reduced instrumentation to a maximum by using CNC available measurements and G-code elements such as actual position, velocity, and acceleration as well as various electrical values from the different axis motors. They also exploit other information contained in the G-code such as the executed subprogram or code lines. They tested different classification algorithms, the DT (decision trees) was performing the best [15].

Another problematic would be the adaptation of the models. Most of the methodologies used to generate predictive models do not deal with unknown cases. The work presented in [10] proposes an auto generation of LSTM models for CNC machine type diagnosis. The performances can be improved when models are iteratively generated, adding or subtracting classes [16].

2 Experimental setup

All experimental milling tests illustrated in this paper were carried out on a vertical machining center. For assessing the performance of the machining process, we monitored and measured the cutting forces and torques generated during cutting, by using the Kistler dynamometer model 9129AA [11]. The Kistler table is mounted below the metal sample to measure the three components of the machining force as shown in figure 1. During the measurements, the x-axis of the dynamometer is aligned with the feed direction of the milling machine. The three orthogonal components of machining force (Fx, Fy and Fz) were measured according to figure 1 using the Kistler table.

Fig 1: Experimental milling setup (located at LEM3, St Dié des Vosges)

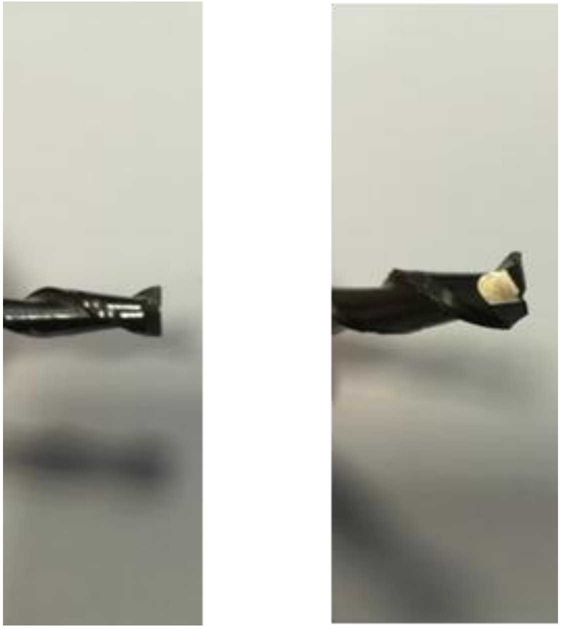

Fig. 2. Good milling tool (left) and damaged milling tool (right)

Many milling experiences have been made in our study, by setting different spindle speeds and feed rates. All measured signals have been stored and analyzed. For example, figure 3 and figure 4 show respectively the milling force and torque along the X-axis, for a given experiment.

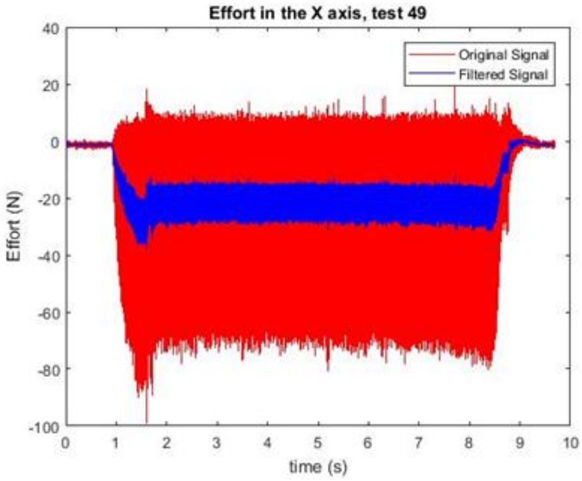

Fig. 3. Measured effort in the x- axis

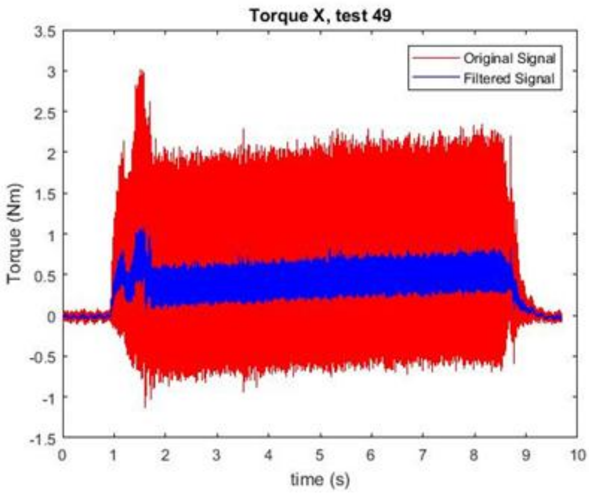

Fig. 4. Measured torque in the x-axis

3 Offline milling diagnosis using machine learning approaches

Supervised learning techniques are implemented in our approach, with labeled measurements for the model training. The labels here correspond to the milling tool's quality. Features associated to the measurements were calculated for this approach.

Machine learning techniques can be separated mainly in two categories [12, 14]:
- Unsupervised approaches: techniques based on unlabeled input data, where the goal is to form groups (clustering).
- Supervised approaches based on labeled inputs. Supervised learning algorithms can be split in two categories [14]: Classification models which partition observations in categorical groups (leads to a predictive model for discrete responses), Regression models which describe the relationship between outputs and variables through mathematical functions (leads to a predictive model for continuous responses).

Here, the data have been labeled : "obtained signals with a good milling tool", "obtained signals with a damaged milling tool". Features are then calculated offline, resulting in a table linking a label to a set of features. All the experiments are then split into two groups: 80% to train the machine learning model (training set) and 20% to test the model (test set). In this experiment, the ratio between the labels was respected in the training set as well as the test set.

3.1 Features calculation

The features are calculated only in time domain from the raw signals (effort and torque time series) in all 3 axes. In our study, transient zones, when the cutting tool enters or exits the material, are not considered. Other used features are deduced from the envelope of the signal. In fact, features are not calculated in frequency domain (i.e. by using Fast Fourier transformation) because the signal can have nonlinearities.

All the calculated features are then normalized and stored in a table whose lines and columns respectively represent the experimental number (also called sample or instance), the label corresponding to associated feature values.

3.2 Features reduction

The obtained normalized features table contains several hundred calculated features and 3 input values for each experiment. The input values are the tool rotation speed, the cutting speed, and the depth of cut. To avoid overfitting, a dimensional reduction was necessary. To reduce the size of a set of features, there exists two kinds of methods: features reduction (for example with the PCA algorithm [17, 18]) and features selection. For the features selection, an optimal subset of features is chosen according to an objective function. In this work, we opted for the Relieff algorithm [20].

3.3 Applied supervised learning algorithms

In this work, several classification algorithms have been implemented in the Matlab software environment [13] and in Python in real time) : k-Nearest Neighbor (kNN), Decision tree (DT), Support Vector Machine (SVM), Naïve Bayes. Several variants of each model have been implemented (various distances, different kernels, etc.).

- The kNN algorithm was implemented by keeping the default Euclidean distance, with a limited number of neighbors k=3. The kNN algorithm has also been modified by using the Chebyshev distance.
- The Decision Tree algorithm is a fitted binary classification decision tree. The tree has been pruned to a level we optimized.
- Two SVM algorithms have been implemented using different kernel functions. The first one is a linear SVM which is the default function for a two-class data set. The second one is the Gaussian SVM algorithm which is a normalized polynomial kernel.

To compare the efficiency of the algorithms, the accuracy (1) is calculated (for the test signals) for each algorithm and compared:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

with TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative)

For this study, 54 milling experiments were conducted from which features were calculated and then split into two groups: 80% for the machine learning model training, 20% for the test phase. All combinations are explored, by keeping the same ratio for the two labels "with a good milling tool" and "with a damaged milling tool": we obtained 27 allocation tables (with 80% - 20 % distribution). Then each machine learning algorithm is applied to each allocation table and the accuracy is deduced. Thus, for each algorithm, 27 accuracies (corresponding to 27 allocation tables) are calculated and only the worst cases are compared, for a given number of features (This step was made for the same weighted classified features). The number of the weighted classified features is then increased, and the procedure repeated until all the features are included in the building of the model. In the next step of the experiment, white noise was added to each signal before calculating the features to test the robustness of the process.

The computation for the whole procedure is heavy, and therefore it needed a long time to yield its results. The results of the process currently described in this section took several hours for a recent computer to be delivered. It follows that such a process can only be applied to offline applications. With this procedure the robustness (to the different splits) is insured. Although the mean accuracy can be useful as an indicator, the most interesting results to estimate the robustness of the models would be the worst cases.

Accuracies for the features from EZ and MY signals without noise are shown on figure 5:

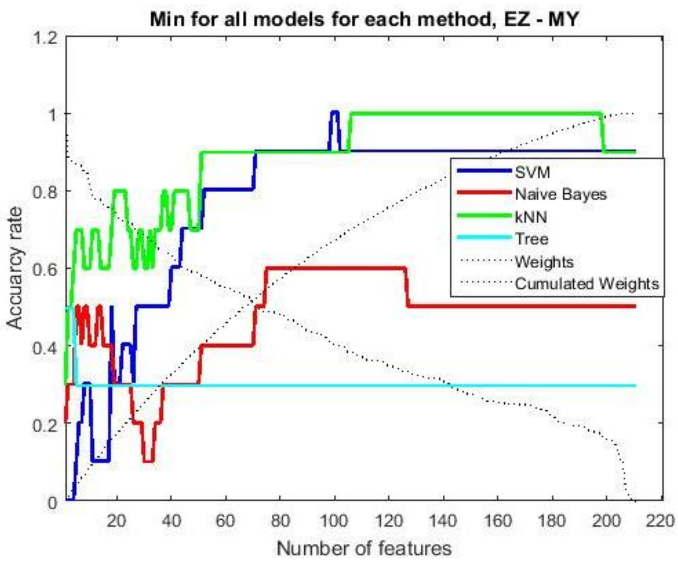

Fig. 5: Worst accuracy result for each algorithm, depending on the number of classified features

The figure 5 presents the worst accuracy (worst of all the 27 allocation tables) for each algorithm, depending on the order-classified features. The weights and cumulated weights of the features are also indicated on this figure. The objective is to select, in a robust manner, the best algorithm and the adequate features for the machine learning approach. We can see on figure 5 that the most robust results are obtained with the kNN algorithm with nearly 100 features. Good accuracy can also be obtained with only 50 features. The approach indicates which features must be selected to achieve a reliable and robust performance of the models.

As well as the robustness against the training / test partitions, the models' robustness was also tested against noisy inputs. On the effort signals is applied a Gaussian noise of 20dBW, and on the Torque signals a Gaussian noise of amplitude -10dBW. Accuracies for the features from EZ and MY signals with noise are shown on figure 6:

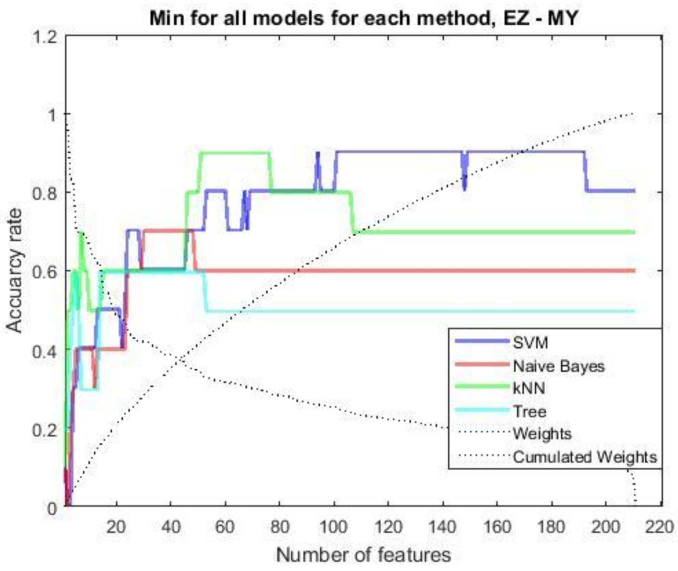

Fig. 6: Worst accuracy result for each algorithm, depending on the number of classified features

As expected, with a small gaussian noise added to the signals, the performances are slightly lower. Even with enough features the models never reach the best accuracies obtained by models without noise.

By studying mean accuracy values, instead worst accuracy, less of features are needed, but the robustness is not guaranteed. Therefore, improving the worst-case scenario is important to find a robust combination of algorithm and/or features that guaranties correct predictions (i.e good accuracy) whether the resulting model's inputs are disturbed.

3.4 Conclusion of the offline application

It is known that the milling's performance is qualified by evaluating the roughness of the resulted surface. In this work, different supervised machine learning algorithms have been implemented and compared. To do this, features were firstly calculated from measured milling forces and torques and then each machine learning model has been trained with the labeled set of features. From the prediction results, kNN algorithm seems to be the most efficient diagnosis algorithm in this application of aluminum material milling. The robust developed diagnosis approach can also be applied to the milling of other materials.

4 Real time milling diagnosis using machine learning approaches

Based on the same measurements, a solution for real time diagnostics was developed: a python development environment has been used with the functionalities of scikit-learn library. Contrary to the offline version of the diagnostics, which provided a one-shot prediction for a given signal part, now the model analyses the whole signal through a "sliding window". Multiple labels are contained within the signal to analyze.

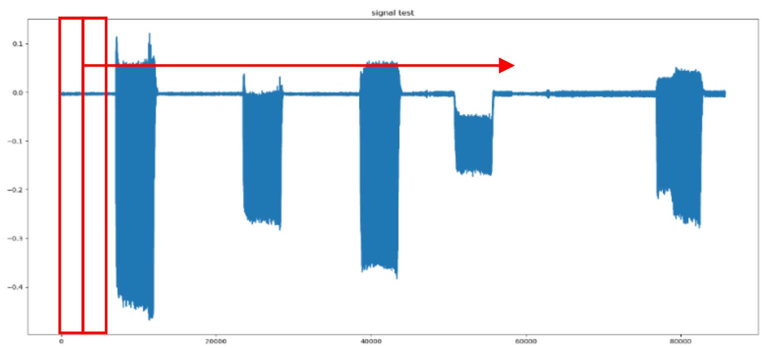

Fig 7: Example of a sliding window going through the signal

This method is divided into 2 steps: first a predictive model needs to be trained, then it can be applied in real time on the signal in step 2. Studied signals are divided into 3 parts: the training set, the validation set and the test set.

4.1 Training phase

To calculate instances (features vector and labels), a sliding window is applied to the labeled of the training signals, and time domain features are then calculated for each window. Window sizes are fixed, there is no overlap, but its size will be shortened to avoid an overlap on 2 different labels. Once the dataset has been obtained, different learning algorithms are tuned to optimize their performances. Models are thus trained with different partitioning for different hyper-parameters thanks to the cross-validation method. The models obtaining the best precision are then selected to be applied in real time. Three learning algorithms have been tested: Decision Trees (DT), Random Forest (RF), kNN.

4.2 Application in real time

Once the models are trained, they are applied on the test signal. Like the training phase, a sliding window is running on the signal. The same data processing done during the training phase (signal filtering, features calculation and normalization, …) must obviously be reproduced. The new instance (features vector) is then ready to be fed, the model outputs being the predictive labels (diagnostics).

4.3 Real time results

To evaluate and visualize the performances of the diagnostic models running in real time, a Human Machine Interface (HMI) was built in the Grafana environment.

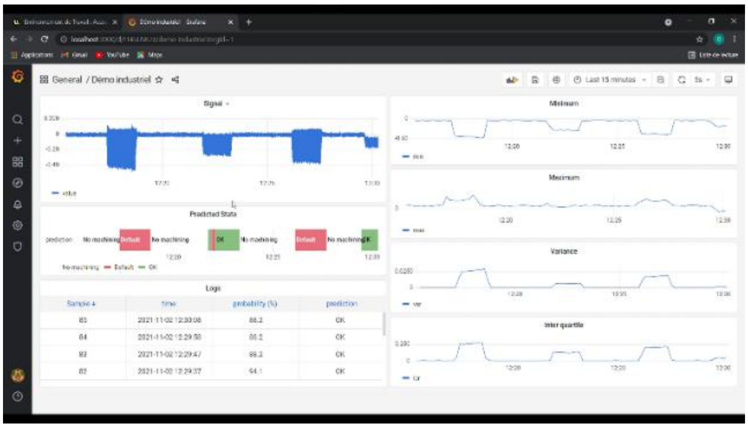

Fig 8: Example of a supervision destined to the operator or the production / maintenance manager

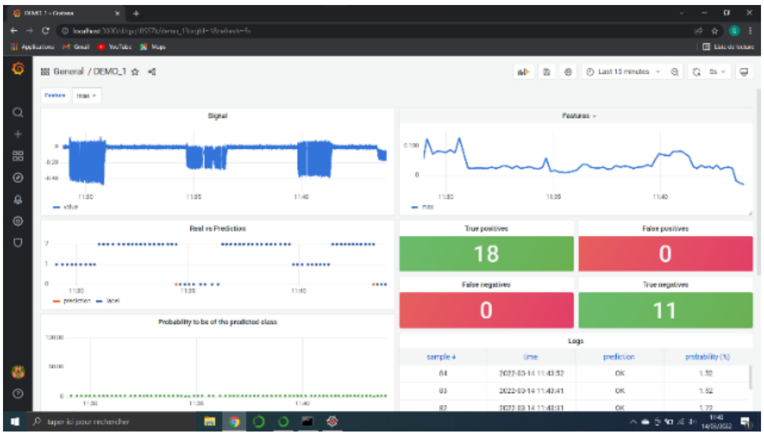

Fig 9: Example of a supervision destined to the model developers

The different features and the diagnostics results were computed periodically (depending on the window size) and transferred to a database (MySQL). Then the results can be accessed through the Grafana platform. Two versions were made, displaying different results according to the user's profile: an industrial version (figure 8) and a laboratory version (figure 9) for research purpose. The diagnostic result (predicted labels) is compared online with the real label (if it is known), and a confusion matrix is online visualized.

5 Conclusion

The features selection approaches show that the reduction of the number of features lead to better diagnostic models (i.e. with higher accuracy) and more robust to signal noise. The results of the first part of this study have then been implemented in a real time diagnostics platform. Once trained, the different machine learning models predicted online the milling tool's state. To evaluate and visualize the performances of the models running in real time, a Human Machine Interface (HMI) was built in the Grafana environment. Two versions were built, displaying different results according to the user's profile: an industrial version and a laboratory version for research purpose.

Acknowledgments

The authors wish to thank the iDEMoov Company and the LEM3 laboratory for the support.

References

[1] Gianpaolo Di Bona, Vittorio Cesarotti, Gabriella Arcese, Tommaso Gallo, "Implementation of Industry 4.0 technology: New opportunities and challenges for maintenance strategy", Procedia Computer Science, Volume 180, 2021, Pages 424-429, ISSN 1877-0509

[2] Luca Silvestri, Antonio Forcina, Vito Introna, Annalisa Santolamazza, Vittorio Cesarotti, "Maintenance transformation through Industry 4.0 technologies: A systematic literature review", Computers in Industry, Volume 123, 2020, 103335, ISSN 0166-3615

[3] Omar D. Mohammed, Matti Rantatalo, Gear fault models and dynamics-based modelling for gear fault detection – A review, Engineering Failure Analysis, Volume 117, 2020, 104798, ISSN 1350-6307

[4] Omar Djebili, Contribution à la maintenance prédictive par analyse vibratoire des composants mécaniques tournants. Application aux butées à billes soumises à la fatigue de contact de roulement, 2013.

[5] Dalila Belmiloud, "Contribution à l'étude de l'endommagement des matériaux, constituants de machines tournantes, en fonction des paramètres de température et fréquence de rotation. Application aux roulements", juin 2019

[6] Zhaoyi Xu, Joseph Homer Saleh, Machine learning for reliability engineering and safety applications: Review of current status and future opportunities, Reliability Engineering & System Safety, Volume 211, 2021, 107530, ISSN 0951-8320

[7] Emiliano Traini, Giulia Bruno, Gianluca D'Antonio, Franco Lombardi, Machine Learning Framework for Predictive Maintenance in Milling, IFAC-PapersOnLine, Volume 52, Issue 13, 2019, Pages 177-182, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2019.11.172.

[8] Mikołajczyk T., Nowicki K., Bustillo A. and Pimenov D., "Predicting tool life in turning operations using neural networks and image processing", Mechanical Systems and Signal Processing, 104: 503-513, 2018.

[9] N. Omri, Z. Al Masry, N. Mairot, S. Giampiccolo, N. Zerhouni, "Towards an adapted PHM approach: Data quality requirements methodology for fault detection applications", Computers in Industry, Volume 127, 2021, 103414, ISSN 0166-3615

[10] Sylvain Chabanet, Hind Bril El-Haouzi, Philippe Thomas, Coupling digital simulation and machine learning metamodel through an active learning approach in Industry 4.0 context, Computers in Industry, Volume 133, 2021, 103529, ISSN 0166-3615

[11] https://www.kistler.com/files/document/000-709e.pdf

[12] The MathWorks Inc., "Mastering Machine Learning A Step-by-Step Guide with MATLAB", Matlab Ebook section 1 to 4, ©2018.

[13] Kubat M., “An introduction to machine learning”, Second Edition, Springer, 2017
[14] Shalev-Shwartz S. and Ben-David S., “Understanding Machine Learning: From Theory to Algorithms”, Published by Cambridge University Press, 2014.
[15] S. Shurrab, A. Almshnanah and R. Duwairi, "Tool Wear Prediction in Computer Numerical Control Milling Operations via Machine Learning," 2021 12th International Conference on Information and Communication Systems (ICICS), 2021, pp. 220-227
[16] K. A. Masalimov, "A Machine Learning based Approach to Autogenerate Diagnostic Models for CNC machines," 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2020, pp. 1358-1360.
[17] Vidal R., Ma Y., Sastry S.S. “Generalized Principal Component Analysis”, Springer 2016.
[18] Yang F., Yun Z., Haiyu Q., Dequn L., Huamin Z. and Jürgen L., “Analysis of feature extracting ability for cutting state monitoring using deep belief networks”, Conference on Modelling of Machining Operations (CIRP), 31: 29-34, 2015.
[20] Ryan J. Urbanowicz, Melissa Meeker, William LaCava, Randal S. Olson, Jason H. Moore, “Relief-Based Feature Selection: Introduction and Review”, November 2017, Journal of Biomedical Informatics 85