DOI: 10.48011/asba.v2i1.1092

An Autonomous Model to Classify Lathe's Cutting Tools Based on TSFRESH, Self-Organised Direction Aware Data Partitioning Algorithm and Machine Learning Techniques

Thiago E. Fernandes * **Guilherme P. C. de Miranda** * **Alexandre F. Dutra** * **Matheus A. M. Ferreira** * **Matheus P. Antunes** * **Marcos V. G. R. da Silva** * **Eduardo P. de Aguiar** *

* *Dept. of Industrial and Mechanical Engineering, Federal University de Juiz de Fora, Juiz de Fora, MG, (e-mail: {thiago.esterci, guilherme.pires, alexandre.dutra , matheus.malta , matheus.antunes, marcos.victor , eduardo.aguiar} @engenharia.ufjf.br*

Abstract: The machining processes are of major importance to industries, due to the fact that these processes take part in the manufacturing of a substantial portion of mechanical components. Hence, during these processes, operational interruptions and accidents induced by fault occurrence are likely to cause economic losses. Concerning these consequences, real-time monitoring can result in productivity and safety increase along with cost reduction. This paper aims to discuss an autonomous model based on self-organised direction aware data partitioning algorithm and machine learning techniques, including features extraction and selection based on hypothesis tests, to solve the adressed problem. The model proposed in this work was evaluated using a data set acquired in a real machining system at the Manufacturing Processes Laboratory of Federal University of Juiz de Fora (UFJF).

Keywords: Autonomous Learning; Empirical Data Analyses; Machine Learning; Machining Processes.

1. INTRODUCTION

The companies around the world, recognising the competitiveness increase in the manufacturing scenario, have directed their efforts to the optimisation of its production processes. Considering this optimisation the reliability and availability of its production equipment are crucial. Accordingly, Condition Based Maintenance (CBM) (Jardine et al., 2006; Mobley, 2002), has been broadly implemented in industrial processes and is one of the most efficient predictive maintenance approaches, owing to its success into decreasing the uncertainty associated with maintenance activities (Rastegari and Mobin, 2016).

Moreover, the manufacturing process by machining emerges as one of the most important, considering that the major part of the mechanical components, fabricated for industrial use, went through a machining process during its manufacturing. According to Trent and Wright (Trent and Wright, 2000), machining transforms about 10% of all metal production in chips and employs dozens of millions of persons worldwide, implying that the development of new technologies in this area is of fundamental importance.

The manufacturing environment has experienced considerable transformations in recent decades. The use of more efficient machines and staff reduction are strategies commonly adopted by industries to obtain cost savings. However, improvements in manufacturing time and product quality are also demanded (Byrne et al., 1995). Thus, these demands resulted in researches centred on the reduction of the human factor within the manufacturing processes. Consequently, the development of new technologies gained momentum. Regarding this scenario, the real-time monitoring of quantities, such as acoustic emission, power, voltage, vibration and current, emerged as a solution of major importance.

In the 1980s and 1990s, the implementation of adaptive inspection mechanisms supported the development of new tool replacement methods. Particularly, the methods based on monitoring cutting edge wear (Snr, 2000). Despite representing an enhancement to the traditional methods, the great number of variables involved make this type of technology to be expensive and subjective to use. Thus its use is not justified for real-time monitoring applications. Lately, to overcome these issues, computational intelligence tools have been used into the development of diagnosis, prognosis and monitoring systems for industrial processes. Regarding the implementation of CBM methods and monitoring systems, the artificial neural networks (ANN's) are the mainly applied tool, as presented in (Calderano et al., 2019; Lee et al., 2010). However, the information provided by monitoring systems applied in machining processes is acquired in the form of extremely dynamic data flows which have large dimensionality. Due to these data flow particularities, monitoring systems based on ANN's are not the best option in solving the proposed problem. Furthermore, the retraining systems are generally not an applicable option, since data flows arrive continuously (Bouchachia et al., 2014).

Consequently, the development of systems capable of supervising the machining process through real-time monitoring data gained momentum, as well as strategies that minimise human interference in the process. Additionally, The implementation of smarter maintenance procedures intends to replace the schedule-based maintenance by a condition-based maintenance (Calderano et al., 2019; Lee et al., 2010). The CBM allows predictive analysis in sensor monitored equipment based on the historical stored data previously to the fault's occurrence (Ellis, 2008). Furthermore, the CBM concerns in improve reliability as well as reduce the time of maintenance in industrial processes which leads to lower incurred costs. Considering this scenario, autonomous learning and machine learning techniques emerge as key tools to avoid faults, to prevent accidents and to reduce losses that may occur between overhauls.

Although the literature presents numerous fault diagnostic methods(Simon et al., 2014), autonomous learning techniques application in lathe's cutting toll classification holds unexplored potential. Therefore, a time series analyses method was applied in this work, owing to its increasing relevance for forecasting and control (Box et al., 2015). This approach aims to develop a model suitable for applying a CBM on machining processes.

The analysis of time series with large dimensionality is a task of high computational cost for several algorithms, particularly regarding the required execution time. Consequently, the dimensionality reduction through feature extraction is essential when considering real-time monitoring applications. Moreover, selecting pertinent and representative features from the data is one of the major challenges when analysing time series. To overcome these challenges, a methodology named TSFRESH was adopted in this work Christ et al. (2018). The TSFRESH algorithm extracts the features from the time series. Additionally, it applies feature selection by the means of hypotheses test.

Furthermore, the authors in (Gu et al., 2018) claim that traditional clustering techniques demand prior knowledge and handcrafting to operate, leading to a subjective result. Hence, aiming to minimise users interference in the model, the Self-Organised Direction Aware Data Partitioning Algorithm was adopted in this work. It is worth to mention that SODA algorithm permits a future improvement of the model presented in this work. Owing to its high efficiency of adapting to various types of data, along with its capability of processing streaming data, it allows the development of an online extension of the proposed model.

In this context, the main contributions of this work are summarised as follows:

- We apply the SODA algorithm (Gu et al., 2018) to a data-set recorded by Fluke 125 Industrial Scopemeter, in the UFJF's Manufacturing Processes Laboratory. The SODA is capable of self-adjusting the data-clouds structure and centres to follow the data patterns in an agile manner.
- An autonomous model for classifying cutting tools' wear state is presented for the first time in the literature.
- We propose a novel model which does not require prior expert knowledge and aims to improve the machining process reliability, by the means of classifying cutting tools' wear state.

And our major conclusions are:

- The proposed model is suitable for identifying the data patterns that separate the adequate condition from the inadequate condition of cutting tools' wear, obtaining satisfactory performances in all cases and allowing to avoid faulty pieces fabrication.
- The feature extraction and selection based on the scalable hypothesis (TSFRESH algorithm) (Christ et al., 2018) is completely applicable to the time series analyses of a lathes' three-phase motor, considering the satisfactory outcomes in our research.
- The development of a lathe's cutting tool diagnosis methodology based on the TSFRESH, SODA and Machine Learning Techniques is justified by the benefits of analysing the time series of lathes' motor, such as the capacity to manage uncertainties. The numerical examples in this paper exhibit that the proposed autonomous model provides high-quality classification results and has notable computational efficiency.

This paper is organised as follows: Section 2 states the problem formulation. Section 3 discusses the methods adopted in the proposed model. After that, Section 4 explains the numerical results. Finally, Section 5 closes the work and presents the conclusions concerning the stated propositions.

2. PROBLEM FORMULATION

Considering the attention gained by the machining cutting power in the literature, electric voltage and current of the lathe's motor were elected as variables of the time series analysed in this work. The adoption of these quantities in monitoring systems is supported by the implication that more energy is consumed when machining with a worn tool, than with a new tool (Shao et al., 2004).

The flank wear occurs when the portion of the tool in contact with the workpiece is eroded by their friction. The tool's flank wear evolution produces an enlargement of the contact area at the tool-workpiece interface. Consequently, the increase in power consumption is likely to be associated with the intensification of friction and machining forces. Furthermore, this work applies statistical and multivariate analysis tools, such as Principal Component Analysis (PCA) and TSFREH to select pertinent features by the means of reducing the computational complexity of the classification task.

The data-set used in this work was acquired from operations of a real machining system at the UFJF's Manufacturing Processes Laboratory. The acquisition process consisted of recording consecutive machining operations, starting with an adequate state tool and stopping when the cutting tool reaches an inadequate state. These two states were established considering ISO 3685/1993. Acknowledging the technical guidelines presented in ISO 3685/1993, the machining conditions defined were: depth of cut of $0.5mm$, spindle speed of $755rpm$ and feed rate of $0.156mm/rev$. The turning process was performed in a Timemaster Tb 350 universal lathe exposed presented in Figure 1.

Figure 1. Timemaster Tb 350 universal lathe

The lathe was equipped with a WEG's cfw500 frequency inverter and using a set of interchangeable carbide inserts.The operations data were collected by an acquisition system using Fluke 125 Industrial Scopemeter connected to the lathe's three-phase motor.

3. THE LATHE'S CUTTING TOOL PROGNOSTIC MODE

The structure of the model presented in this work is exhibited in Figure 2:

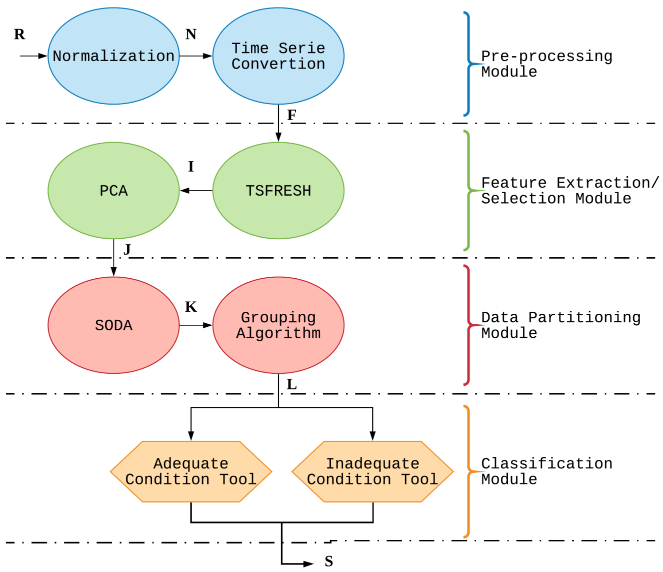


Figure 2. Proposed Model

The data set **R** consist of 185 time series recorded with Fluke 125 Industrial Scopemeterof in an acquisition rate of 100 measurements per second. Each time serie has 250 measurements of voltage and curent of the lathe's motor. Firstly, **R** is presented to the pre-processing stage, which commences with the normalisation of the time-series, represented by **F** in Figure 2. After the pre-processing stage, the data are presented to the Feature extracting/selecting stage and then to the data partitioning stage. Finally, the data are presented to the Classification stage in which a cutting tool can be classified as an adequate state tool or inadequate state tool.This classification output is represented by **S** in Figure 2.

3.1 Pre-Processing

Accounting for the magnitude discrepancy between our variables, the time series were normalised following the equation 1.

$$x_{norm} = \frac{x - x_{mim}}{x_{max} - x_{min}} - 1 \tag{1}$$

3.2 Feature Extraction on basis of Scalable Hypothesis tests (TSFRESH)

TSFRESH is a python package used to to extract characteristics from time series. It automatically calculates a large number of time series characteristics, the so called features. Further the package contains methods to evaluate the explaining power and importance of such characteristics for regression or classification tasks. (Christ et al., 2018)

The relevancy of a feature X (Radivojac et al., 2004; Christ et al., 2018) to a target Y is calculated as the difference between their conditional distribution and expressed as: $f_{x|y=y_1}$ and $f_{x|y=y_2}$ when $Y = y_1$ and $Y = y_2$ respectively. Therefore, feature X is relevant to estimate Y if, and only if,

$$\exists y_1, y_2 \;\; with \;\; f_y(y_1), f_y(y_2) > 0 : f_{x|y=y_1} \neq f_{x|y=y_2} \tag{2}$$

Equation 2 also corresponds to X and Y being statistically dependants. Feature X is irrelevant when:

$$\exists y_1, y_2 \;\; with \;\; f_y(y_1), f_y(y_2) > 0 : f_{x|y=y_1} = f_{x|y=y_2} \tag{3}$$

and it also means that X and Y are statistically independents.

The relevancy can also be investigated through hypothesis test (Christ et al., 2016). To the extracted features $X_1, X_2, ..., X_n$, a hypothesis test is applied independently, in order to investigate the following hypothesis:

$$\begin{aligned} H_0^i &= X_i \text{ is not relevant to } Y \\ and \;\; H_1^i &= X_i \text{ is relevant to } Y \end{aligned} \tag{4}$$

The result of each test is called p-value and corresponds to the probability of obtaining a measure of equality or inequality between the hypothesis test and the observed in the data based on the null hypothesis. In this work, the p-value measures if the analysed feature is relevant or not and small p-values show more relevant features.

The test applied in this paper is the Kolmogorov-Smirnov (KS) (Wilcox, 2005), considering the following hypotheses:

$$H_0^i = \left\{ f_{X_i|Y=y_1} = f_{X_i|Y=y_2} \right\}$$
$$H_1^i = \left\{ f_{X_i|Y=y_1} \neq f_{X_i|Y=y_2} \right\} \quad (5)$$

where $f_{X_i|Y=y_1}$ is the cumulative distribution function (CDF) of feature X considering the healthy operation and $f_{X_i|Y=y_2}$ is the CDF of feature X_i considering faulty operations.

The KS test considers the maximum difference between the CDF obtained from the features, a shown in Equation (6).

$$D = \sup |f_{X_i|Y=y_1} - f_{X_i|Y=y_2}|. \quad (6)$$

Therefore, the null hypothesis H_o^i is rejected if $D > D_{n,a}$, in which $D_{n,a}$ is a critical value that can be found in Appendix A.

The presence of irrelevant features can yield to a false-positive result. In addition, when multiple hypothesis and features are used, these errors are accumulated (Curran-Everett, 2000). Authors in (Benjamini and Yekutieli, 2001) propose to reject the hypothesis based on the p-values while controlling the False Discovery Rate (FDR) (Benjamini and Hochberg, 1995)

$$FDR = \frac{R}{V} \quad (7)$$

where, R is the total number of rejected hypothesis and V the number of true null hypothesis rejected. This approach seeks the first intersection of the p-values $p(i)$ and the following linear sequence:

$$r_i = \frac{i \cdot FDA}{n \sum_{k=1}^{i} k^{-1}} \quad (8)$$

where n is the total number of hypothesis.

3.3 Principal Component Analysis

Given a set of variables $X = x_i$, with $i = 1, 2, 3, ..., n$, it is possible to investigate a smaller set of variables, within X, in which their linear combination $\alpha_\mathbf{k}\mathbf{X}$ preserves a major part of the information available in X with maximum variance. These variable are called principal components and the first of the principal components is z_i, known for comprising the major variability of the data:

$$z_1 = \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n = \sum_{k=1}^{n} \alpha_{1k}x_k \quad (9)$$

The other components are calculated analogously and the j^{th} principal component must not be correlated to the previous components (Jolliffe, 1986):

$$z_j = \alpha_{j1}x_j + \alpha_{j2}x_j + ... + \alpha_{jn}x_n = \sum_{k=1}^{n} \alpha_{jk}x_k \quad (10)$$

3.4 Self-Organised Direction Aware Data Partitioning Algorithm(SODA)

In order to express this method, we must consider data space R^m and assume a data set as $\{x_1, x_2, x_3...\}$, where $x_i = [x_{i,1}, x_{i,2}, ..., x_{i,m}]^T \in R_m$ is a m dimensional vector, $i = 1, 2, 3, ...$; m is the dimensionality; subscript $i(i = 1, 2, 3, ...)$ indicate the time instances at which the i^{th} data sample arrives. Therefore, within the observed data set at the n^{th} time instance denoted by $\{x_1, x_2, ..., x_n\}$, we also consider the set of sorted unique values of data samples $\{u_1, u_2, ..., u_{n_u}\}$ $(u_i = [u_{i,1}, u_{i,2}, ..., u_{i,m}]^T \in R_m)$ from $\{x_1, x_2, ..., x_n\}$ and the corresponding normalised numbers of repeats $\{f_1, f_2, ..., f_n\}$, where $n_u(1 < n_u \leq n)$ is the number of unique data samples and $\sum_{i=1}^{n_u} f_i = 1$. The following derivations are conducted at the n^{th} time instance as a default unless there is a specific declaration (Gu et al., 2018).

Distance/Dissimilarity Components in SODA The SODA approach, in this work, employs (Gu et al., 2018):

i a magnitude component $d_M(x_i, x_j)$ based on the Canberra distance metric;
ii a angular $d_A(x_i, x_j)$ component based on the cosine similarity;

EDA Operators The recently introduced Empirical Data Analytics (EDA) (Gu et al., 2018) is an alternative methodology for machine learning which is entirely based on actual empirically observed data samples (Angelov, 2014; Angelov et al., 2017a,b).

The EDA operators include(Gu et al., 2018):

i. **Cumulative ProximityGu et al. (2018):**
The cumulative proximity, π of $x_i(i = 1, 2, ..., n)$ is defined as (Angelov, 2014; Angelov et al., 2017b):

$$\pi_n(x_i) = \sum_{j=1}^{n} d^2(x_i, x_j) \quad (11)$$

where $d(x_i, x_j)$ denotes the distance/dissimilarity between x_i and x_j.

ii. **Local Density Gu et al. (2018):**
Local density D is defined as the inverse of the normalised cumulative proximity and it directly indicates the main pattern of the observed data. The local density, D of i $x_i(i = 1, 2, ..., n; n_u > 1)$ is defined as follows (Angelov et al., 2017a,b):

$$D_n(x_i) = \frac{\sum_{j=1}^{n} \pi_n(x_j)}{2n\pi_n(x_i)} \quad (12)$$

In the proposed SODA data partitioning approach, since both components, the magnitude (metric) and the angular one are equally important, the local density of $x_i(i = 1, 2, ..., n; n_u > 1)$ is defined as the sum of the metric/Canberra-based local density $(D_n^M(x_i))$ and the angular-based local density $(D_n^A(x_i))$.

iii. **The Global Density Gu et al. (2018):**
The global density is defined for unique data samples together with their corresponding numbers of repeats in the data set/stream. It has the ability of providing multi-modal distributions automatically without the need of user decisions, search/optimisation procedures or clustering algorithms. The global density of a particular unique data sample, $u_i(i = 1, 2, ..., n_u; n_u > 1)$ is expressed as the product of its local density and its number of repeats considered as a weighting factor (Angelov et al., 2017b) as follows:

$$D_n^G(u_i) = f_i D_n(u_i) \quad (13)$$

As we can see from the above equations, the main EDA operators: cumulative proximity (π), local density (D) and global density (D^G) can be updated recursively, which shows that the proposed SODA algorithm is suitable for online processing of streaming data.

SODA Algorithm for Data Partitioning The main steps of the SODA algorithm include: firstly, form a number of DA planes from the observed data samples using both, the magnitude-based and angular-based densities; secondly, identify focal points, using the granularity γ of the clustering results and relates to the Chebyshev inequality (Angelov et al., 2017b), we used $\gamma = 2.0$ in this work; finally, use the focal points to partition the data space into data-clouds. The detailed procedure of the proposed SODA partitioning algorithm is presented by (Gu et al., 2018).

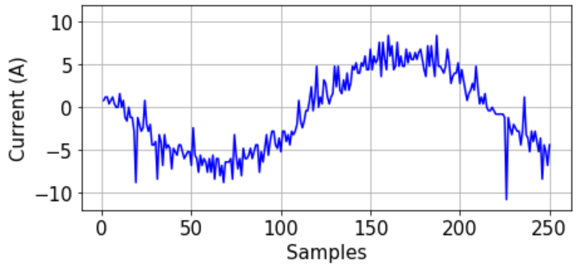
Grouping algorithm This algorithm gathers all data-clouds that contain data pertaining to the same group. The groups are adequate condition tool data ($Index = 0$) and inadequate condition tool data ($Index = 1$), as presented in Section 1. Accordingly, the grouping algorithm associates each data sample to a label that is used in the classification module. The output provided by SODA is a vector composed by the indexes that indicate from which data-cloud each data sample belongs. Taking the number of data samples into consideration for each data-cloud, the percentage of data relating to each group (0 or 1) was determined. Hence, if a data-cloud contains more than 67% of its data relating to one of the groups, the algorithm assigns the group's index as a label to each data sample of the data-cloud. The data-clouds with less than 67% of data pertaining to one of the groups was disregarded, in order to avoid misleading classifications.

4. EXPERIMENTAL RESULTS

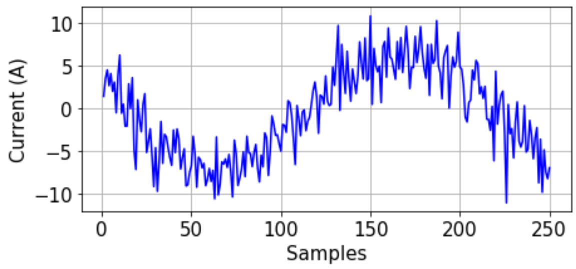
In this Section, all the algorithms were performed on a computer with Intel Core i5-7200U processor with clock frequency 3.10 GHz and 8 GB of RAM. The acquired data set is composed of 185 normalised time series of voltage and current. Additionally, 100 of those time series were recorded using an adequate condition tool and the other 85 used the inadequate condition tool. Moreover, different levels of additive white Gaussian noise (AWGN) were applied in the original data set in order to challenge the proposed model. Aiming to represent possible variations of the machine tool defects, three different intensities of AWGN were applied in all data set, resulting in signals with Signal Noise Ratio (SNR) = 1, 3 and $5dB$, as presented in Figure 3 and 4. After evaluating the performance with those values, SNR = $3dB$ was selected and used as a strategy to corrupt the original data set. The performance gains in terms of accuracy are similar for other values of SNR.

4.1 Classification

The first step of our classification problem was to extract and select the features to be used as inputs of the classifiers listed in Table 1. The TSFRESH algorithm was applied in this task, extracting and selecting 65 features for each time serie. Subsequently, the PCA method was applied to the TSFRESH output data aiming to reduce its dimensionality. Owing to the fact that using more components would

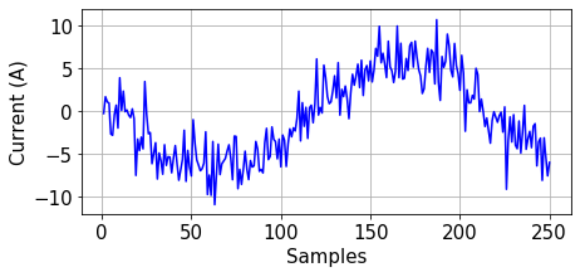


(a) Original Time Serie.

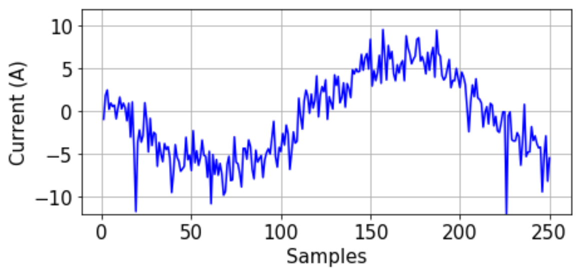


(b) Time Serie corrupted with SNR = 1 dB.

Figure 3. Demonstrative image considering a Current Time Serie, from an appropriate condition tool



(a) Time Serie corrupted with SNR = 3 dB.



(b) Time Serie corrupted with SNR = 5 dB.

Figure 4. Demonstrative image considering a Current Time Serie, from an appropriate condition tool

not increase significantly the representation of the previous data, the first 3 PC's were kept.

The data partitioning, performed by the SODA algorithm (Gu et al., 2018) has formed 4 data-clouds. The grouping algorithm divided these data-clouds into 2 groups as it follows: adequate condition tools' clouds, inadequate condition tools' clouds. Accordingly, the data samples, contained in each of the grouped data-clouds, was labelled as discussed in Section 2. Therefore, the labelled data samples were randomly divided using the proportion of

60% for the training phase and 40% for the testing phase. The classification was executed 33 times and the result obtained by each classifier is presented in Table 1.

Table 1. Classification Accuracy

Classifier	Average	Standard Deviation
Nearest Neighbors	91.89	0.0
Radial-basis function kernel SVM	91.89	0.0
Radial-basis function kernel Gaussian Process	91.89	0.0
Decision Tree	91.89	0.0
Random Forest	90,50	1.84
MLP Neural Network	91.89	0.00
AdaBoost	91.89	0.0
Gaussian Naive Bayes	91.89	0.0
Quadratic Discriminant Analysis	91.89	0.0

The classifiers used in this work were implemented trough scikit-learn (Pedregosa et al., 2011), an open-source machine learning library in python. Although other configurations were studied, the maximum accuracy of the classifiers was accomplished with the configurations exhibited in the example that follows `https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`, except the Random Forest, the MLP methods and the Decision Tree. In the Decision tree and in the Random Forest methods, the nodes are expanded until all leaves are pure or until all leaves contain less than 2 samples, since the maximum depth of the tree was not defined. In the MLP the maximum iterations number was set to 200.

5. CONCLUSION

This paper proposed an autonomous approach for classifying cutting tool's wear state based on TSFRESH, SODA and Machine Learning Techniques. The model is suitable for identifying the data patterns that separate the adequate condition from the inadequate condition of cutting tools' wear, obtaining satisfactory performances in all cases and allowing to avoid faulty pieces fabrication. The feature extraction and selection provided by TSFRESH algorithm is completely applicable to the time series analyses of a lathes' three-phase motor, considering the satisfactory outcomes in our research.

Moreover, SODA algorithm considers both spatial and angular divergence, resulting in a more accurate similarity recognition among the data than traditional clustering/partitioning methods. Additionally, it presents a great efficiency when applied to large-scale and high-dimensional situations and do not demands prior knowledge nor handcrafting to operate it. Hence, minimising human interference in the application of the proposed model and granting a high computational efficiency, which supports the machine learning techniques in the classification task. Owing to its effectiveness when adapting to various types of data, along with its capability of processing streaming data, an indication for future works is the development of an online extension of the proposed model using SODA algorithm. Furthermore, we will apply the model discussed in this work in other engineering problems (i.e rotating machines), preventing fault occurrences and classifying them in order to help professionals not only in the decision making processes but also to devise strategies in the industry.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Financing Code 001, National Council for Scientific and Technological Development - CNPq (process 433389 / 2018-4), Minas Gerais State Research Support Foundation - FAPEMIG (APQ-02922-18) and Federal University of Juiz de Fora - UFJF for financial support.

REFERENCES

Angelov, P. (2014). Outside the box: an alternative data analytics framework. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8(2), 29–35.

Angelov, P., Gu, X., and Kangin, D. (2017a). Empirical data analytics. *International Journal of Intelligent Systems*, 32(12), 1261–1284.

Angelov, P., Gu, X., and Príncipe, J.C. (2017b). A generalized methodology for data analysis. *IEEE transactions on cybernetics*, (99), 1–13.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, 289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 1165–1188.

Bouchachia, A., Lughofer, E., and Mouchaweh, M.S. (2014). Evolving soft computing techniques and applications. *Appl. Soft Comput.*, 14, 141–143.

Box, G.E., Jenkins, G.M., Reinsel, G.C., and Ljung, G.M. (2015). Time series analysis: forecasting and control. *John Wiley & Sons*.

Byrne, G., Dornfeld, D., Inasaki, I., Ketteler, G., König, W., and Teti, R. (1995). Tool condition monitoring (tcm)—the status of research and industrial application. *CIRP annals*, 44(2), 541–567.

Calderano, P.H.S., Ribeiro, M.G.C., Amaral, R.P.F., Vellasco, M.M.B.R., Tanscheit, R., and de Aguiar, E.P. (2019). An enhanced aircraft engine gas path diagnostic method based on upper and lower singleton type-2 fuzzy logic system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 41(2), 70.

Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A.W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package). *Neurocomputing*, 307, 72 – 77.

Christ, M., Kempa-Liehr, A.W., and Feindt, M. (2016). Distributed and parallel time series feature extraction for industrial big data applications. *arXiv preprint arXiv:1610.07717*.

Curran-Everett, D. (2000). Multiple comparisons: philosophies and illustrations. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 279(1), R1–R8.

Ellis, B.A. (2008). Condition based maintenance. *The Jethro Project*, 10, 1–5.

Gu, X., Angelov, P.P., Kangin, D., and Principe, J.C. (2018). Self-organised direction aware data partitioning algorithm. *Information Sciences*, 423, 80–95.

Jardine, A.K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7), 1483–1510.

Jolliffe, I.T. (1986). Principal component analysis and factor analysis. *Springer New York*, 115–128.

Lee, Y.K., Mavris, D.N., Volovoi, V.V., Yuan, M., and Fisher, T. (2010). A fault diagnosis method for industrial gas turbines using bayesian data analysis. *Journal of Engineering for Gas Turbines and Power-transactions of The Asme - J ENG GAS TURB POWER-T ASME*, 132.

Mobley, R.K. (2002). *An introduction to predictive maintenance*. Elsevier.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., J., Vanderplas, Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12, 2825–2830.

Radivojac, P., Obradovic, Z., Dunker, A.K., and Vucetic, S. (2004). Feature selection filters based on the permutation test. 334–346.

Rastegari, A. and Mobin, M. (2016). Maintenance decision making, supported by computerized maintenance management system. In *2016 annual reliability and maintainability symposium (RAMS)*, 1–8. IEEE.

Shao, H., Wang, H., and Zhao, X. (2004). A cutting power model for tool wear monitoring in milling. *International Journal of Machine Tools and Manufacture*, 44(14), 1503–1509.

Simon, D.L., Borguet, S., Léonard, O., and Zhang, X. (2014). Aircraft engine gas path diagnostic methods: public benchmarking results. *Journal of Engineering for Gas Turbines and Power*, 136(4), 041–201.

Snr, D.E.D. (2000). Sensor signals for tool-wear monitoring in metal cutting operations—a review of methods. *International Journal of Machine Tools and Manufacture*, 40(8), 1073–1098.

Trent, E.M. and Wright, P.K. (2000). *Metal cutting*. Butterworth-Heinemann.

Wilcox, R. (2005). Kolmogorov–smirnov test. *Encyclopedia of biostatistics*.

Appendix A. CRITICAL VALUES FOR THE KOMOLGOROV-SMIRNOV TEST

Table A.1. Critical values $D_{n,\alpha}$ of KS test for $\alpha = 0.05$ e $\alpha = 0.01$.

n	0.05	0.01	n	0.05	0.01	n	0.05	0.01	n	0.05	0.01
1	0.9750	0.9950	**26**	0.2591	0.3106	**51**	0.1866	0.2239	**76**	0.1534	0.1841
2	0.8419	0.9293	**27**	0.2544	0.3050	**52**	0.1848	0.2217	**77**	0.1524	0.1829
3	0.7076	0.8290	**28**	0.2499	0.2997	**53**	0.1831	0.2197	**78**	0.1515	0.1817
4	0.6239	0.7342	**29**	0.2457	0.2947	**54**	0.1814	0.2177	**79**	0.1505	0.1806
5	0.5633	0.6685	**30**	0.2417	0.2899	**55**	0.1798	0.2157	**80**	0.1496	0.1795
6	0.5193	0.6166	**31**	0.2379	0.2853	**56**	0.1782	0.2138	**81**	0.1487	0.1784
7	0.4834	0.5758	**32**	0.2342	0.2809	**57**	0.1767	0.2120	**82**	0.1478	0.1773
8	0.4543	0.5418	**33**	0.2308	0.2768	**58**	0.1752	0.2102	**83**	0.1469	0.1763
9	0.4300	0.5133	**34**	0.2274	0.2728	**59**	0.1737	0.2084	**84**	0.1460	0.1752
10	0.4092	0.4889	**35**	0.2242	0.2690	**60**	0.1723	0.2067	**85**	0.1452	0.1742
11	0.3912	0.4677	**36**	0.2212	0.2653	**61**	0.1709	0.2051	**86**	0.1444	0.1732
12	0.3754	0.4490	**37**	0.2183	0.2618	**62**	0.1696	0.2034	**87**	0.1435	0.1722
13	0.3614	0.4325	**38**	0.2154	0.2584	**63**	0.1682	0.2018	**88**	0.1427	0.1713
14	0.3489	0.4176	**39**	0.2127	0.2552	**64**	0.1669	0.2003	**89**	0.1419	0.1703
15	0.3376	0.4042	**40**	0.2101	0.2521	**65**	0.1657	0.1988	**90**	0.1412	0.1694
16	0.3273	0.3920	**41**	0.2076	0.2490	**66**	0.1644	0.1973	**91**	0.1404	0.1685
17	0.3180	0.3809	**42**	0.2052	0.2461	**67**	0.1632	0.1958	**92**	0.1396	0.1676
18	0.3094	0.3706	**43**	0.2028	0.2433	**68**	0.1620	0.1944	**93**	0.1389	0.1667
19	0.3014	0.3612	**44**	0.2006	0.2406	**69**	0.1609	0.1930	**94**	0.1382	0.1658
20	0.2941	0.3524	**45**	0.1984	0.2380	**70**	0.1597	0.1917	**95**	0.1375	0.1649
21	0.2872	0.3443	**46**	0.1963	0.2354	**71**	0.1586	0.1903	**96**	0.1368	0.1641
22	0.2809	0.3367	**47**	0.1942	0.2330	**72**	0.1576	0.1890	**97**	0.1361	0.1632
23	0.2749	0.3295	**48**	0.1922	0.2306	**73**	0.1565	0.1878	**98**	0.1354	0.1624
24	0.2693	0.3229	**49**	0.1903	0.2283	**74**	0.1554	0.1865	**99**	0.1347	0.1616
25	0.2640	0.3166	**50**	0.1884	0.2260	**75**	0.1544	0.1853	**100**	0.1340	0.1608