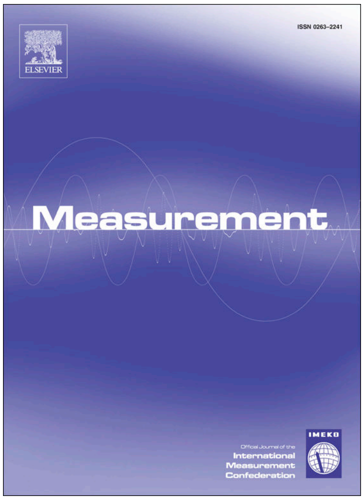


High-accuracy classification of thread quality in tapping processes with ensembles of classifiers for imbalanced learning

Jose Francisco Diaz-Pastor, Alain Gil Del Val, Fernando Veiga, Andres Bustillo

PII: S0263-2241(20)30865-4
DOI: https://doi.org/10.1016/j.measurement.2020.108328
Reference: MEASUR 108328

To appear in: *Measurement*

Received Date: 13 May 2020
Revised Date: 31 July 2020
Accepted Date: 2 August 2020

High-accuracy classification of thread quality in tapping processes with ensembles of classifiers for imbalanced learning

Jose Francisco Diaz-Pastor[1], Alain Gil Del Val[2], Fernando Veiga[2], Andres Bustillo*,[1]

[1] Department of Civil Engineering, University of Burgos, Burgos, Spain

[2] TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Científico, Parque Científico y Tecnológico de Gipuzkoa, E20009, Donostia-San Sebastián, Spain

* Corresponding author: email abustillo@ubu.es,

Abstract

Industrial threading processes that use cutting taps are in high demand. However, industrial conditions differ markedly from laboratory conditions. In this study, a machine-learning solution is presented for the correct classification of threads, based on industrial requirements, to avoid expensive manual measurement of quality indicators. First, quality states are categorized. Second, process inputs are extracted from the torque signals including statistical parameters. Third, different machine-learning algorithms are tested: from base classifiers, such as decision trees and multilayer perceptrons, to complex ensembles of classifiers especially designed for imbalanced datasets, such as boosting and bagging decision-tree ensembles combined with SMOTE and under-sampling balancing techniques. Ensembles demonstrated the lowest sensitivity to window sizes, the highest accuracy for smaller window sizes, and the greatest learning ability with small datasets. Fourth, the combination of models with both high Recall and high Precision resulted in a reliable industrial tool, tested on an extensive experimental dataset.

Keywords: Bagging, imbalanced datasets, threading, cutting taps, quality assessment.

1. Introduction

Tapping is a widely used thread cutting operation in industrial applications, such as the manufacture of automotive components, domestic appliances, energy production, and ship building. Industrial tapping uses a Computer Numerical Control (CNC) machine equipped with a cylindrical tool tapered to a conical head. The use of CNC machines is crucial, due to the complex synchronization that is necessary between the feed-in and rotational tool trajectories, which is aggravated at high speeds [2]. The tapping process is performed in a blind hole (perforation on only one side of a piece) or in a hole that pierces a piece on both sides. Both cutting taps and spiral taps can be used, to obtain the thread profile in the pre-drilled hole.

The complexity of the operation is due to the large number of cutting edges [3] and can mean that some threads are outside tolerance margins. Those errors can be classified by the following causes: (a) wear of the tapping tool; (b) misalignment between axis (hole/tap) and poor hole preparation; (c) under/oversized predrilled hole diameters. A sample tap will usually be inspected by the operator, to prevent the batch production of reject threads. Several competitive-cost solutions have been developed, so that an on-line monitoring and diagnosis system could complete the same task as the operator. There are three possible model-based strategies for tapping classification: mechanical-based models, statistical models extracted from experimental data, and machine-learning-based models. The work of Oeskaya and Biermann [37] belongs to the first strategy, in which a Finite

Element Method (FEM) software module is developed, to determine relative torque values for tapping processes with various tapping tools and diameters. Following the same FEM strategy, Hsia et al. [23] evaluated the quality of sheet metal tapping. The work of Monka et al. [35] studied screw quality following tap failure in machining C45. Finally, an example of the third strategy was found in the work of Gil Del Val et al. [19], who proposed a monitoring and diagnosis strategy for tapping processes based on torque parameters using statistical process control and Principal Component Analysis (PCA). More recently, Bustillo et al. [9] proposed a combination of Rotation Forest with unpruned Regression Trees for the prediction of form tap wear.

However, in comparison with other machining processes, such as milling [41]) and turning [34], tapping presents evident over-dispersion in relation to non-deterministic patterns of tool wear, reminiscent of burr formation patterns in drilling [17]. The high variety of behaviors between tools means that the application of machine-learning algorithms plus statistical analysis is, therefore, a more common solution for quality prediction of the threading process. The most common strategy is based on a monitoring analysis of the torque and force signals. In the three-pioneering works, a diagnosis of three common tapping faults was based on conditional probability functions [13], on diagnostic feedback neuronal system [30] and on neural networks [31], respectively. The outputs of all the studies were the level of tool wear, a measure of the misalignment and a measure of the under/oversized pre-drilled diameters of the holes. Taking into account the acquired knowledge of these three studies, Liu et al. [32] evaluated the same faults using an adaptive neuro-fuzzy interface system to classify thread quality. Currently, Ying Chao Ma et al. [46] are developing a dynamic model with lateral and torsional/axial vibration inputs from tapping operations. Moreira et al. [36] have recently proposed the use of new neural network families, such as convolutional neural networks, for online prediction of threading quality. Teti et al. [45] proposed a machine-learning system based on an artificial neural network to make smart decisions such as tool change time during a drilling automation process. Finally, some approaches based on failure modes and effects analysis and a hybridized genetic algorithm have studied certain machining systems including tapping operations, to improve risk factors and scalability during those manufacturing processes [10][42].

This research forms part of the third approach. It a new strategy based on experimental tests and machine-learning modeling, to overcome the strong dispersion of the experimental data for tapping processes. It takes an extensive list of statistical measures into account as extra-inputs to the experimental data for different windows of previous tapping processes. It likewise searches for the optimal machine-learning model for different industrial strategies (from total accuracy to minimization of false positives). This strategy, based on considering different windows of previous tapping processes, has a clear advantage: knowing that past windows will help to classify the present behavior of the tool; but it has also one important limitation: the first threads of each tool cannot be used in the dataset, because the sliding window is not available, and those instances must therefore be removed from the dataset, reducing its size. In this study, window size and its effect on the accuracy of the prediction model is therefore analyzed.

Some clear novelties in relation to previous studies will be presented in this research. First, only non-intrusive measurements, such as torque measurements, easily extracted from existing tapping machines will be considered. Second, a broad variety of up to 5 tapping tool coatings will be included in an extensive dataset of 35 tapping tools, to show the wide range of tool behaviors between tools from the same manufacturers. Third, an extensive number of different types of machine-learning algorithms will likewise be evaluated, in order to identify the most suitable family and the advantages and disadvantages of each family of algorithms, considering datasets measured under industrial conditions. As this industrial task is clearly imbalanced (there will be a small number of faulty taps in any tapping process), special machine-learning algorithms for imbalanced datasets will also be taken into account, such as SMOTE and under-sampling. Finally, to bring this research closer to its real application in manufacturing, different applications for the prediction models of industrial interest will be considered: such as improvements to traceability in the use of tapping tools and an extensive analysis of the effects of tapping tool behavior and its high dispersion on algorithmic prediction capabilities.

The paper will be organized as follows. In Section 2, the experimental procedure will be presented for the tapping tests, including torque signals and thread-quality evaluation. Then, the dataset extracted from the tapping tests will be presented with the machine-learning techniques to model the dataset for testing. Special attention will be paid to techniques designed exclusively for imbalanced datasets. The results of such modelling and the industrial use of the best model will be presented in Section 4. Finally, the most relevant results will be summarized in the Conclusions, pointing to future lines of research.

2. Experimental set up

A CNC machining center was used to perform the tapping process at High-Speed Cutting (HSC). The plate material of nodular cast iron (GGG50) measured 250x450mm, with a drilled thickness of 20 mm, as illustrated in Figure 1. The holes were of the appropriate diameter, measured to a micrometric resolution of 0.01mm. The tapping tool was equipped with three flutes for metric threads of M10x1.5mm with different coatings.

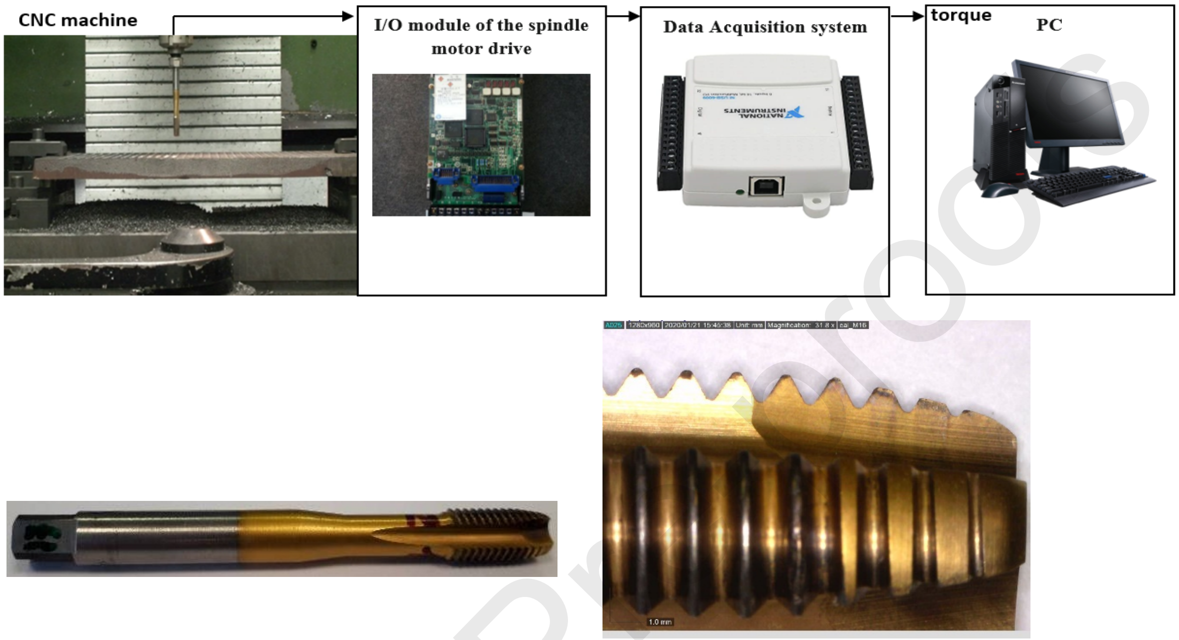


Fig. 1. Set-up configuration, tap tool and geometry of tap coated with TiN.

The tapping process was performed under high-speed cutting (HSC) conditions (65 m/min), without a coolant. Table 1 summarizes the tapping cutting process conditions and the GGG50 chemical composition of plates.

Table 1. Tapping cutting process conditions and chemical composition of GGG50.

Tapping Cutting conditions	
Tool diameter (mm)	10
Number of flutes	3
Cutting velocity (m/min)	65
Thread pitch (mm)	1.5
Plate thickness (mm)	20
No coolant used	
Chemical composition GGG50 (*in wt.%*)	
C	3.76
Si	2.4
Mn	0.14
P	0.021
S	0.007
Mg	0.063
Cu	0.76
Ni	1.9
Mo	0.24

Balance iron

In this study, five coatings were employed: TiN, TiCN, Steam, AlCrN and TiN plus Steam. Titanium Nitride (TiN), the first PVD protection against wear, provided effective reduction of abrasion and adhesion wear at low temperatures. Titanium carbonitride leaves a hard and smooth finish, which offers improved wear and built up edge resistance. Steam refers to the metal-originating hydroxide and oxide film on metal substrates. The goal of this thermal treatment process is to improve resistance to corrosion and oxidation. In comparison with TiN-based coatings, AlCrN (Aluminum Chromium Nitride) coating has been reported to show higher oxidation resistance, because both chromium and aluminum can form protective oxides, which suppress the diffusion of oxygen, yielding an extremely wear resistant coating with excellent hot hardness and thermal shock stability at high cutting speeds. Bilayer TiN+Steam tap coatings were also tested.

The thread quality is assessed by a "go–nogo" gauge inspection. Thread quality is considered faulty when nogo passes through the hole or when the go side cannot pass through. Table 2 shows the number of tap tools for each coating and the total number of threads and the threads that correctly passed the quality inspection (OK threads). As can be seen from the mean and standard deviation values of the sample, considering either the different coatings independently or the total number of specimens, there is a case of over-dispersion related to the non-deterministic wear behavior of the tools. Besides, the percentage of faulty threads oscillates between 8 and 23%, the mean value of which is 15%, knowing that the tool has a complex geometry (see Fig. 1) and the process works under high speed conditions to reduce the time costs. The variety of behaviors between tools means that the application of the machine learning strategy for the prediction of insufficient quality in the threading process is of great interest.

Table 2. Summary of the tapping process test conditions.

Coating	*Number of tap tools*	*Total number of threads*	*Total OK threads*	*% Faulty threads*	*Mean OK threads*	*Standard Deviation (SD)*
TiN	15	2290	1763	23%	117.53	85.92
Steam	7	1751	1509	14%	215.57	58.37
TiCN	5	1148	1047	9%	209.40	110.57
AlCrN	5	1519	1359	11%	271.80	111.73
TiN+Steam	4	407	374	8%	93.50	42.98
TOTAL	36	7115	6052	15%	182.75	116.38

In this paper, the torque signal was measured directly from the electrical cabinet of the CNC machine in the I/O module of the spindle motor drive, in order to monitor thread quality as can be seen in Figure 1. The proposed tap monitoring system was therefore a non-intrusive solution. The current signal was compared with the torque measured with a rotating multicomponent dynamometer (type 9123C1111; sensitivity: 54.03 mV/N m [0–200 N m]) (Gil Del Val A. et al. 2013). Once measured, the signal was transformed into the torque signal and the current signals were fitted to the torque signals, to find the sensitivity value. Fig. 1 illustrates the tapping process and, as an example, the first thread was selected. The tap starts its rotatory movement on a Reference Plane (RP), then it goes through the hole to a Stop Plane (SP) where the thread is finished. The reverse movement is done from the SP to the RP where both movements stop, and the machine moves to the next hole position for tapping.

Focusing on the calculation of the area values, the procedure derives the rotational velocity of the spindle motor, by calculating spindle acceleration. Figure 2 illustrates the spindle speed and acceleration of a single thread.

The second stage of the procedure is to find the integrated points (eight points; t_1, t_2…,t_8) using the change of gradient in the acceleration signal, as can be seen in Fig. 2.

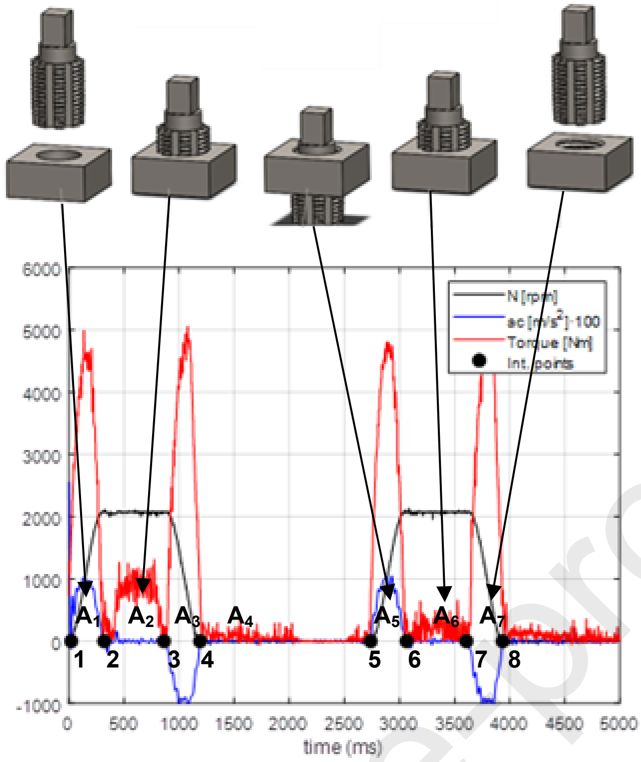


Fig. 2. Torque, rotational velocity, acceleration signals and integrated points in a tapping cycle.

Having estimated the eight integrated points, the area parameter is calculated in the torque signal, according to the following formula:

$$A_i = \int_{t_i}^{t_{i+1}} Torque(t) \cdot dt; being\ i = 1, 2,\dots 7 \quad (1)$$

The indicators selected for monitoring are the areas under the curve of the torque signal in the seven phases which define the tap process. Table 3 describes the definition of the areas. The first area, A1, is the area of spindle rotation acceleration on the RP. A2 is the area during the cutting of the thread when the tap and the RP rotation both stop (A3) and the angular and the travel positions are synchronized (A4). Finally, spindle acceleration begins again on the RP (A5), and both feed and rotation continue while traveling back to the SP (A6), once the tap tool is on the SP, the rotation stops (A7).

Table 3. Definition of areas calculated from the torque signal.

Monitoring parameter	*Name*	*Definition*
Area A1	Acceleration Torque	A1 parameter represents the motor torque evolution area required for accelerating the spindle from zero speed to the tapping angular speed while the tap bottom moves in the acceleration period
Area A2	Cutting torque area	Cutting torque area during the tapping operation itself when the chamfer teeth
Area A3	Deceleration torque area on the stop plane (SP)	Deceleration torque area required for stopping the main spindle. While the spindle motor decelerates, there is contact between some full teeth
Area A4	Dwelling torque area	Feedback signals to both motor regulators and, because the CNC attempts to maintain synchronicity between both, there are minute movements (at small angles)

Area A5	Acceleration torque area on the stop plane (SP)	Changes to torque during the tap acceleration, to invert the spindle rotation and to begin to exit the hole
Area A6	Torque tap reverse area	Tap torque evolution area during tap reverse
Area A7	Deceleration torque area on the reference plane (RP)	When the tap base is close to the RP, the spindle motor decelerates to arrive at the RP at zero speed.

3. Modeling

3.1 Dataset description

From the experimental tests described in Section 2, a dataset was extracted considering mainly the torque information related to each tapping process. The dataset included, in a first instance, for each thread, the seven areas of the torque signal previously presented in Table 3 (A1-A7). Besides, each thread was identified with another three attributes: 1) the coating of the tap tool (*coating*); 2) a numerical identifier of each tap tool (*ToolID*); and, 3) the thread number for the corresponding tap tool (*number*). The coating attribute can take 5 different nominal values (from 1 to 5 for TiN, TiCN, Steam, AlCrN and TiN+Steam coatings, respectively), because 5 different tap tool coatings were tested, as previously summarized in Table 2. The numerical identifier of each tap tool can take 36 possible values (from 1 to 36), because 36 different tap tools were used in the experimental tests (15 TiN; 7 Steam; 5 TiCN; 5 AlCrN, and 4 TiN+Steam). These two attributes, *coating* and *ToolID*, were considered nominal, because ordering their possible values made little or no industrial sense. Finally, the thread number takes a value of 1 for the first thread drilled with a certain tap tool, 2 for the second thread, and so on. The thread number was considered a numeric attribute. Although the original dataset included the tool identifier and the thread number, neither attribute was used for training and validation of the prediction model. The reason is due to the cross-validation structure and will be discussed in Section 3.4. The dataset output was the thread quality, a class with only two possible quality-inspection values (0: thread pass and 1: thread non-pass). Table 4 summarizes the inputs and output, their units and the range of values presented in the dataset; the output variable is shown in bold. The dataset has been included as supplementary material.

Table 4. Dataset attributes and output with their variation range

Variable	Abbreviation	Range
Coating	*Coating*	1-5
Tool identifier	*ToolID*	1-36
Thread number	*Number*	1-519
Torque signal areas (A1-A7)	*A1-A7*	0.0-17.3
Thread quality inspection	**Pass, not pass**	**1, 2**

As the dataset includes 1053 non-pass threads and 6063 correct threads, the minority class represents only 14.8% of the instances in the dataset. This low rate can clearly define the dataset as strongly imbalanced. A more detailed analysis of the imbalanced nature of the dataset is included in Section 3.3.

Finally, as will be presented in Section 3.3, a sliding window approach was considered to take into account the previous behavior of the tap tool. For this sliding window, a statistical feature can be extracted for each torque signal area (A1-A7): the slope of the regression fit, the standard error of the regression fit, and the True Strength Index (STI). These new features will be used to extend the number of attributes in the dataset.

3.2 Machine-learning techniques

While artificial intelligence aims to develop systems that have intelligent behavior, machine learning is the area within artificial intelligence that studies the creation of systems that learn by themselves, directly from the data.

In a classification problem, after the training stage in which the classifier has been fed with data, it receives both the inputs and the expected output (or class), the classifier is then able to generalize and to predict the category/class to which new data should be assigned. Ensembles of classifiers are combinations of classifiers, that use different techniques to obtain better predictive performance than could be obtained from any of the individual classifiers that form the ensemble [38]. It sometimes happens that the number of examples belonging to one class is much greater than the number of examples belonging to another. Problems of that sort are known as imbalanced problems. In this context the minority class is called “Positive” and the majority class is called “Negative”. Imbalanced problems are common in engineering and manufacturing, for example defect identification and diagnosis [48], automatic visual inspection [29][15], and fault detection [33].

Imbalanced problems are, for several reasons, more difficult than standard problems [44]: a) standard classifiers are driven by accuracy, implying that any examples belonging to the minority class will be ignored; b) the training data of standard classifiers is assumed to be a faithful representation of the data distribution of the problem to be modelled (not always the case); and, c) in imbalanced problems, not all errors are of the same importance, having to distinguish between false positives and false negatives.

Thousands of methods exist to work with imbalanced sets. In Galar et al. [18] these methods are categorized into four groups:

1. *Algorithm level approaches.* Groups composed of standard algorithms that are modified to have a bias towards accurate learning of the minority class.
2. *Data-level approaches.* Groups composed of pre-processing techniques that transform the data. The alternatives are to increase the size of the minority class creating artificial examples or to decrease the size of the majority class or to do both pre-processing at the same time. A common technique to decrease the size of the majority class is random under-sampling [4]. Some of the techniques used most often to increase the size of the minority class are as follows:
    - Random Oversampling [5], consisting of adding copies of some of the examples of the minority class, in order to reduce the imbalance between classes.
    - SMOTE [11] -Synthetic Minority Over-sampling Technique- is a technique that generates artificial instances from other existing data. To generate an example from an instance *i*, first the nearest *k* neighbors belonging to the same class as *i* are selected, then one of those neighbors is randomly chosen (called instance *j*) and finally an artificial instance is generated at a random point in the segment that joins instance *i* with instance *j*.
    - Borderline-SMOTE [21] or ADASYN [22], which are variants of the method described above.
3. *Cost-sensitive learning approaches.* A group formed by cost-sensitive versions of existing algorithms. A cost-sensitive algorithm assigns a different error cost to each class.
4. *Ensemble learning approaches.* An ensemble is a combination of multiple classifiers or regressors. An ensemble usually provides a better result than any of its individual members. One common strategy for building ensembles for imbalanced learning is to combine conventional ensembles, such as Bagging or Boosting, with data level approaches. Some examples that use this strategy are SMOTEBoost [12] and RUSBoost [43]. This approach is more versatile than approaches 1 and 3. The capability of ensembles to obtain better results than any of their constituent classifiers means that the algorithms in this category are the most recommendable.

From among all these families of techniques, individual classifiers, classical ensembles and ensembles for imbalanced datasets were used in this study. First, the following algorithms were used as individual classifiers:

- K-Nearest neighbors [1]: a classical Instance-based learning algorithm. It returns the most frequent class in the group formed by the k-nearest instances of the instance to be predicted. This algorithm was selected, because it is often used as a baseline due to its simplicity, however the algorithm produces poor results when the number of attributes is very high or several of them are irrelevant. The k value was set at 3 for the experiments. In the results tables, the method is identified as KNN3.
- Sequential minimal optimization (SMO) [26] is a Multi-class implementation of the well-known SVM classifier. This method is appropriate when it has a large number of attributes and is one of the most

widely used baseline methods, however it can yield poor results in the presence of noise or overlapping between classes.

- The Multilayer Perceptron is a feedforward artificial neural network, composed of at least 3 layers of artificial neurons: an input layer, a hidden layer and an output layer. This algorithm can approximate any nonlinear function and is robust in the presence of noisy data, although it can be slow in big datasets, and its performance is dependent on the chosen parameters [25]. It is a tried and tested method in any study that uses industrial data, due to its robustness to noise. In the results tables, this method is identified as MLP.

- Decision Trees. Due to the problem of class imbalance, decision trees were used without pruning and collapsing, but with Laplace smoothing at the leaves. A C4.5 tree with these options is called C4.4 [39]. In the results tables, this method is identified as TREE U.

As decision trees are fast to compute and are unstable (small changes in the data produce large changes in the classifier and different predictions), they function ideally as the base classifier of the ensemble. Decision Trees were therefore used as base classifiers in all the ensembles. The following algorithm was used as an example of a typical ensemble:

- Bagging [6] is one of the most common ensemble methods. This algorithm builds a set of base classifiers, each one trained from a random sample of the original training data. In the results tables, this method is identified as BAG. The foremost advantages of this method are as follows: it reduces overfitting and it performs well with a high-dimensional dataset (it is easily parallelizable). The negative aspects are that it is not in any way designed to deal with the imbalance. This algorithm is interesting because several imbalanced ensembles are based on bagging.

- Random Forest [20] is a fast and robust ensemble method based on Bagging. In this algorithm the diversity of the ensemble is increased using a random selection of subsets of attributes at each tree node; it can be seen as a combination of Bagging with Random Trees. This method is capable of working with data sets with a large number of attributes, without the need to perform a previous selection of attributes and it is very resistant to overfitting [8]. However, it is not designed for imbalanced data sets. In the results tables, this method is identified as RF.

Finally, the following ensemble algorithms especially adapted for imbalanced datasets were tested:

- Bagging+SMOTE. Bagging combined with SMOTE in each iteration. Several configurations were tested.
    - BAG100 means that the number of instances generated with SMOTE in each iteration is equal to the number of instances of the minority class.
    - BAG300, BAG500. The number of instances generated with SMOTE in each iteration is equal to 3 or 5 times the number of instances of the minority class.
    - BAGSM. Generating as many instances as needed to match the size of the majority class.

- Bagging+Random Under-sampling. Bagging combined with random under-sampling, in which random under-sampling is applied in each iteration, until the size of the two classes (correct/fault) are the same.

- Bagging+Random Balance. Bagging combined with Random Balance [16] in each iteration. The optimal amount of SMOTE or under-sampling is dependent on each problem. In Random Balance, that problem is avoided by relying on randomness and repetition. Each base classifier of the ensemble is constructed with a different data set, with the same size as the original, but with a random ratio between classes, resulting from applying SMOTE and Undersampling with random proportions. In the results tables, this method is identified as BAG-RB.

- SMOTEBoost. In boosting-based ensembles, each base classifier is trained with a weighted sample of instances that over-represents those instances failed by the previous base classifier. SMOTEBoos [12] is a modification of AdaBoost.M2, which combines the change of weights of the instances, performed by boosting, with the creation of instances of the minority class performed by SMOTE. In the results tables, this method is identified as SMB.

- RAMOBoost [14]. A variant of SMOTEBoost, it uses the ADASYN oversampling technique instead of SMOTE. ADASYN adaptively shifts the decision boundary to difficult-to-learn and majority instances. In the results tables, this method is identified as RAMOB.
- RUSBoost. Another boosting-based ensemble, RUSBoost [43] combines boosting with random undersampling. In the results tables, this method is identified as RUSB.
- RB-Boost [16]. A Random Balance modification on AdaBoost.M2. In each iteration, first the weights of the instances are modified according to their difficulty for the previous classifier, and then the ratio between the classes is altered, in a random way, using SMOTE and Random Undersampling. In the results tables, this method is identified as RB-B.

Table 5 summarizes the machine-learning methods tested in this research and their main parameters.

Table 5. Summary of machine learning techniques used in this study and their optimized parameters.

Individual classifiers		
KNN3	K-Nearest Neighbors	Parameter k value was set at 3
MLP	Multilayer Perceptron	Number of neurons = (nº attribs. + nº classes) / 2 Momentum = 0.3 Learning Rates values tested = 0.1, 0.3, 0.5
SMO	Sequential minimal optimization	Polynomial Kernel and C values ranging from 0.1 to 1
TREE U	Decision Tree	Weka default parameters except: Unpruned = False, Collapsing = True, Laplace Smoothing = True
Classical Ensembles		
BAG	Bagging	Number of base classifiers set to 100
RF	Random Forest	Number of base classifiers set to 100
Ensembles for imbalanced data		
BAGSM100	Bagging+SMOTE 100%	Number of base classifiers set to 100. SMOTE using k = 5 Number of synthetic instances equal to number of minority instances
BAGSM300	Bagging+SMOTE 300%	Number of base classifiers set to 100. SMOTE using k = 5 Number of synthetic instances equal to 3 times the number of minority instances
BAGSM500	Bagging+SMOTE 500%	Number of base classifiers set to 100. SMOTE using k = 5 Number of synthetic instances equal to 5 times the number of minority instances
BAGSM	Bagging+SMOTE	Number of base classifiers set to 100. SMOTE using k = 5 As many synthetic instances as necessary to reach majority class size
BAGRUS	Bagging+Random Undersampling	Number of base classifiers set to 100
BAGRB	Bagging+Random Balance	Number of base classifiers set to 100
SMB	SMOTEBoost	Number of base classifiers set to 100. SMOTE using k = 5
RAMOB	RAMOBoost	Number of base classifiers set to 100. ADASYN using k= 5
RUSB	RUSBoost	Number of base classifiers set to 100

RB-B	RB-Boost	Number of base classifiers set to 100. SMOTE using k = 5

3.3 Sliding window approach for feature extraction

The dataset presented in Section 3.1 was expanded with additional attributes that are the result of processing a sliding window with data on tool behavior in the previous threads. Sliding window-based approaches are widely used to detect errors in data streams from sensors. In this strategy, a set of features are extracted from a dataset consisting of the most recent data; as time passes, old samples are discarded, and new samples are inserted into the window [49]. In this way, if a sliding window of 7 threads is considered, the areas of the 7 threads that have been tapped immediately before the thread that is under consideration are used to calculate some new statistical features that help to provide a comparison between tool behavior in the immediate-past and its subsequent behavior. This strategy has a clear advantage: knowing the past behavior will help to classify the present behavior of the tool. But it has also one important disadvantage: the first threads of each tool cannot be used in the dataset because the sliding window is not available, therefore these instances must be removed from the dataset, reducing its size (e.g. in the previous example the first 7 threads of each tool should be removed from the dataset).

Several features can be extracted from the sliding window, one for each torque signal area (A1-A7):

1. Slope: slope of the regression line computed over the values of the sliding window. Wear usually plays a principal role in manufacturing processes, so torque or motor consumption will tend to grow as wear increases.
2. Error: Standard error of the regression line. This attribute is useful for discriminating whether the value of the slope is reliable, as very noisy slopes might not be a reliable indicator of real wear.
3. True Strength Index (TSI). The analysis of trends is a very frequent problem in forecasting the behavior of the stock market. Technical analysis forecasts price fluctuations through the study of past market data. A series of features based on technical analysis led to TSI, a measure that represents how much and how fast the area has changed. The TSI output is bound between +100 and −100.

The first two sets of sliding window attributes (Slope, Standard error of the regression line) were obtained using the Scipy Python Library. The third set of attributes was obtained using the Python wrapper for TA-Lib (http://ta-lib.org/).

An important parameter in the calculation of these attributes was the size of the sliding window. In the experimental study, 5 different sliding window sizes were evaluated. Table 6 summarizes the number of attributes in each dataset used in this research. Attribute numbers increased after the three kinds of statistical parameters had been added and a sliding window had been introduced. Thus, the number of attributes rose from 8 in the original dataset to 29 after the three kinds of statistical parameters had been added.

Table 6: Number of dataset attributes

Datasets	**Number of Attributes**
Original (Coating + Areas)	8
Original + S (Slopes)	15
Original + S + E (Errors)	22
Original + S + E + TSI	29

Finally, the imbalance ratio should be evaluated, because it might have a considerable effect on the performance of the machine-learning algorithms and their suitability for this industrial task. Table 7 summarizes the imbalance ratio and the dataset size of each dataset used in this research. The dataset size changes, as previously explained, because, for each tapping tool, as many of the first experiments should be deleted from the dataset as from the sliding window size. Table 7 shows that the imbalance ratio is quite stable between the different datasets (varying from 0.1479 to 0.1567), although the sliding window varies from 0 to 11 threads. In all cases, the dataset can be considered partially imbalanced and special techniques for imbalanced datasets might be tested, to identify the most accurate technique from the industrial point of view.

Table 7: Number of instances of the datasets, **W**: windows size, **NC:** number of correct samples, **NF:** number of fault samples, **IR:** imbalance ratio.

Datasets	**W**	**NC**	**NF**	**IR**
Original (Coating + A1-A7)	-	6063	1053	0.1479
Original + Sliding Window	5	5883	1053	0.1518
Original + Sliding Window	7	5811	1053	0.1534
Original + Sliding Window	9	5739	1053	0.1550
Original + Sliding Window	11	5667	1053	0.1567

3.4 Training/validation methodology

Two principal decisions should be taken, before the prediction models can be evaluated. First, the decision over which is the most suitable training/validation scheme. In this case, as the datasets are quite large, but not too large, a 10x10 cross validation scheme appeared to be the best option. In this scheme the dataset was randomly divided into 10 folds of equal size; then a first model was built using a training set composed of 90% of the instances (9 of the 10 folds) and the quality indicators of the instances included in the remaining fold were evaluated. The construction of the prediction model was done 10 times; in each of them, the validation set included a different fold and the averaged quality indicators were calculated. In this way, the prediction model was not over-optimistic, as it might otherwise have been, had the model been tested with the same instances used in the training stage. So, the generalization capabilities of the prediction model to deal with new instances can be properly evaluated. The process was repeated 10 times with different divisions in the folds and the mean values of the quality indicators were the quality indicators under consideration, to ensure that the dataset split into folds had no influence.

However, as already outlined in Section 3.1, this scheme uses, on average, 90% of the threads performed with each tapping tool to evaluate the performance of the other 10%. In the 90% of instances used in the training process, the prediction model is expected to find some threads that took place before each thread of the evaluation fold and some threads that took place after it. By doing so, the prediction model will predict the complete wear process of the tapping tool during the training step and will predict any thread in the validation step to a very high degree of accuracy. The thread number and the tool identifier should be deleted from the dataset, to blind the model to the temporal evolution of the wear process. In that way, the prediction model will be unable to classify the instances by their link to any well-identified tapping tool, and learning will be generalized. Besides, this reality will be the prevailing reality in most small workshops, where a tapping tool is used during a certain time and then stored before it is needed again, losing any traceability with regard to the previously drilled threads.

Both SMOTE and Random Undersampling were used on the training sets, rather than on the entire data set prior to cross validation, to ensure that there was no overfitting in the model's training procedure, Besides, these preprocessing techniques were combined with bagging and boosting strategies, so the amount of randomness introduced into the data sets with which the base classifiers were trained was quite high. The data sets used by the base classifiers were quite different from each other, and the bias introduced by each preprocessing technique was also very different.

Second, the quality indicators of the prediction model should be selected. Basically, accuracy is firstly considered when evaluating a classifier. Accuracy is the number of correctly labeled instances divided by the total number of instances. However, when working on binary (two class) and imbalanced problems, it is necessary to analyze the type of error. In an extreme case, if 99% of the instances belonged to the majority class and a classifier completely ignored the minority class, its accuracy would be around 99%, but it would be completely useless from an industrial point of view. Therefore, in binary and imbalanced problems there are four possible outcomes: 1) the instance is positive and it is predicted as positive; 2) the instance is negative and it is predicted as positive; 3) the instance is negative and it is predicted as negative; and, 4) it is positive and it is predicted as negative. These outcomes are summarized in Table 8.

Table 8: Measures derived from the confusion matrix

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

From the previous outcomes, other additional measures can be evaluated:

$$\text{Recall} = TP / (TP+FN).$$

$$\text{Precision} = TP / (TP+FP)$$

Recall, also called a True Positive Rate, is a measure of the sensitivity of the system at detecting a certain event. It is the metric to maximize, if the priority is to detect system failure. However, by examining only the recall, an algorithm with many false positives (which Recall will not register) might be selected, so it would also be necessary to evaluate precision. All these metrics were considered for the evaluation of the prediction models in this study.

Finally, Weka only provided the option for parametrical optimization for each fold, in terms of model Accuracy to optimize the parameters of the model, the quality of which will depend strongly on their parameters such as MLP and SMO (SVM). As accuracy might be a quality measure of low interest in imbalanced datasets, a different strategy was followed to optimize the parameters of the model: the testing of many different combinations of different parametrical values of the model and the selection of the model with the best performance in the averaged validation subsets of the cross validation.

4. Results and industrial implementation

Firstly, this section discusses the results of the prediction models generated from the experimental dataset and, then, the best way to extract information from those models of use to industry.

4.1 Machine-Learning Modeling results

The results were obtained with Weka [20], using the default parameters unless otherwise specified. The size of all ensembles was set at 100. First, the prediction models were built for the base classifiers under consideration, taking the original dataset, the datasets with the different sliding windows, and the slope attributes. Table 9 summarizes the results for the different base classifiers using the Recall metric; the most accurate models are highlighted in bold. Decision Trees and MLPs were the most accurate models depending on the sliding window. Table 9 also shows the effect of adding the slope of the torque areas to the dataset and the effect of the different window sizes. There was no increased accuracy for K-Nearest neighbors and Decision Trees following the addition of the slope attributes. Besides, only decision trees improved their accuracy (around 10%) as the window size increased. Finally, as MLPs and decision trees have similar accuracy, both classifiers can be considered as base classifiers for the ensemble models. Decision trees are the preferred option for such tasks, as they are easily tuned and are simpler than MLPs. In the case on MLPs and SMOs, multiple configurations were tested using cross-validation, the values shown in Table 9 are the recall values for the best configurations.

Table 9: Recall of the base classifiers with different sliding window sizes. Decision Tree (DT), K-Nearest neighbors (KNN3), Support Vector Machine (SMO), and Multilayer Perceptron (MLP)

Dataset	**Window Size**	**DT**	**KNN3**	**SMO**	**MLP**
Original (Coating + Area)	-	**0.654**	0.618	0.495	0.636
Original + Slope	5	0.653	0.583	0.540	**0.771**
Original + Slope	7	0.680	0.579	0.571	**0.742**
Original + Slope	9	0.699	0.552	0.587	**0.730**
Original + Slope	11	**0.714**	0.578	0.580	**0.714**

Subsequently, the prediction models based on ensemble techniques were built. Table 10 shows the recall of different ensembles of decision trees trained with different sets of attributes (using a sliding window size 11). Standard ensembles, those that are not specially designed for imbalanced problems, are marked in Table 10 with an asterisk. Bagging was the only ensemble not specially designed for imbalanced problems that is clearly better than decision trees for almost all the datasets under consideration. Besides, the ensembles for imbalanced datasets greatly improved on the best results obtained by the best single classifiers (MLP and decision trees) that have previously been presented. Table 10 shows that the recall model is improved, by adding the adjustment error of the linear regression and the RSI as attributes (last columns of Table 10). Taking recall as the quality indicator, the best ensemble was bagging combined with random undersampling (in bold in Table 10) for all the datasets considered alongside a sliding windows size of 11.

Table 10: Recall of decision-tree ensembles and MLPs with a sliding window size of 11.

Dataset	**Original**	**Original + Slope**	**O + S + Error**	**O+S+E+TSI**	**Average**
Decision Tree	0.654	0.714	0.796	0.852	0.754
MLP	0.636	0.714	0.829	0.867	0.762
BAG*	0.681	0.725	0.836	0.886	0.782
RF*	0.671	0.717	0.806	0.833	0.757
BAGSM100	0.777	0.823	0.896	0.922	0.854
BAGSM300	0.833	0.876	0.921	0.941	0.893
BAGSM500	0.864	0.899	0.930	0.948	0.910
BAGSM	0.866	0.899	0.930	0.946	0.910
BAGRUS	**0.930**	**0.935**	**0.955**	**0.960**	**0.945**
BAG-RB	0.889	0.873	0.916	0.935	0.903
SMB	0.728	0.790	0.903	0.940	0.840
RUSB	0.857	0.863	0.933	0.954	0.901
RB-B	0.733	0.799	0.890	0.925	0.837
RAMO	0.782	0.830	0.922	0.948	0.870
Average	*0.779*	*0.818*	*0.890*	*0.918*	

Third, as the recall model will not compute the number of False Positives and, therefore, an algorithm with a high recall will minimize False Negatives, but will predict multiple correct instances as faulty, an evaluation of precision is required. Table 11 shows the precision of the previous ensembles of decision trees trained with different sets of attributes (once again using a sliding window of 11). The ensembles for imbalanced datasets obtained better results than the classic ensembles. In this case, the best results were obtained by SMOTEBoost (in bold). Besides, the inclusion of the new attributes (the adjustment error of the linear regression and the RSI) in all cases improved the Recall model.

Table 11: Precision of the ensembles of decision trees and MLPs with a sliding window of 11.

Dataset	**Original**	**Original + Slope**	**O + S + Error**	**O+S+E+TSI**	**Average**

Decision Tree	0.701	0.767	0.819	0.867	0.788
MLP	0.617	0.672	0.829	0.867	0.746
BAG*	0.768	0.800	0.874	0.892	0.834
RF*	0.758	0.785	0.859	0.863	0.816
BAGSM100	0.698	0.738	0.819	0.847	0.775
BAGSM300	0.647	0.672	0.763	0.812	0.724
BAGSM500	0.621	0.649	0.736	0.788	0.698
BAGSM	0.628	0.664	0.749	0.794	0.709
BAGRUS	0.548	0.596	0.673	0.709	0.632
BAG-RB	0.603	0.672	0.764	0.790	0.707
SMB	**0.711**	**0.760**	**0.866**	**0.893**	**0.808**
RUSB	0.627	0.689	0.792	0.825	0.733
RB-B	0.705	0.786	0.866	0.879	0.809
RAMO	0.670	0.721	0.844	0.879	0.779
Average	*0.670*	*0.715*	*0.800*	*0.833*	

Having analyzed recall and precision, the accuracy metric was taken also into account, although this metric plays a limited role in the problem, due to the significant imbalance level. Table 12 shows the results of the different ensembles using accuracy as a metric. It may be concluded from the table that all the machine learning algorithms under consideration achieved high levels of accuracy, which is a major industrial requirement. With this metric, the best results were obtained by SMOTEBoost (in bold) for most of the datasets and the average of the four datasets. Besides, the inclusion of the new attributes (the adjustment error of the linear regression and the RSI) in all cases improved the accuracy of the model.

Table 12: Decision-tree ensemble accuracy and MLPs with a sliding window size of 11.

Dataset	Original	Original + Slope	O + S + Error	O+S+E+TSI	Average
Tree U	90.683	90.833	93.214	94.821	92.388
MLP	91.667	91.994	94.092	95.685	93.360
BAG	**92.229**	92.827	95.521	96.518	94.274
RF	91.948	92.470	94.866	95.298	93.646
BAGSM100	91.709	92.634	95.253	96.146	93.935
BAGSM300	90.767	91.310	94.271	95.640	92.997
BAGSM500	90.149	90.744	93.646	95.164	92.426
BAGSM	90.388	91.250	93.988	95.283	92.727

BAGRUS	87.605	89.018	91.979	93.199	90.450
BAG-RB	89.685	91.265	94.226	95.074	92.563
SMB	91.596	92.768	**96.280**	**97.277**	**94.480**
RUSB	90.289	91.696	95.060	96.071	93.279
RB-B	91.484	**93.408**	96.101	96.801	94.448
RAMO	91.062	92.277	96.086	97.128	94.138
Average	*90.804*	*91.750*	*94.613*	*95.722*	

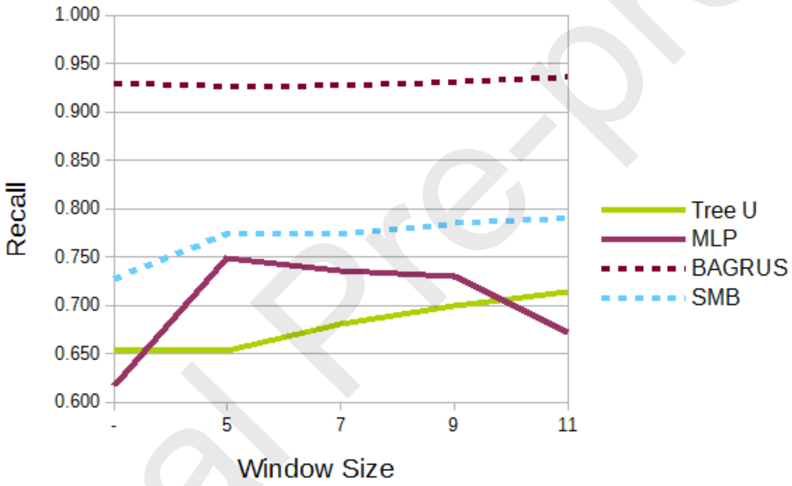
The last quality indicator of high interest in imbalanced datasets is AUC (Area under the ROC curve). Table 13 shows the results of the different algorithms using AUC. This metric is insensitive to changes in class distribution and is very popular for tasks relating to extremely imbalanced data. According to this metric, the best results were obtained (in general) by SMOTEBoost and RAMOBoost (in bold). The inclusion of the new attributes also improved the models according to this metric.

Table 13: Decision-tree ensemble and MLPs AUC with a sliding window size of 11.

Dataset	**Original**	**Original + Slope**	**O + S + Error**	**O+S+E+TSI**	**Average**
Tree U	0.949	0.952	0.966	0.976	0.961
BAG*	0.962	0.969	0.985	0.991	0.977
RF*	0.957	0.968	0.983	0.988	0.974
BAGSM100	0.962	0.97	0.985	0.99	0.977
BAGSM300	0.962	0.968	0.983	0.989	0.976
BAGSM500	0.961	0.967	0.982	0.989	0.975
BAGSM	0.961	0.968	0.983	0.989	0.975
BAGRUS	0.961	0.965	0.979	0.985	0.973
BAG-RB	**0.963**	0.969	0.984	0.989	0.976
SMB	0.96	0.972	**0.991**	**0.995**	**0.980**
RUSB	0.962	0.971	0.989	0.992	0.978
RB-B	0.96	**0.974**	0.99	0.993	0.979
RAMO	0.961	0.972	0.991	**0.995**	**0.980**
Average	*0.960*	*0.968*	*0.983*	*0.989*	

Finally, Figure 3 is presented below, as the information in Tables 10 to 13 cannot be used to evaluate the effect of the sliding window size on model performance. The figure shows the progressive performance of the recall metric for the best two basic classifiers: MLPs (unlike in the tables, which show the result of the best combination of parameters, which may be different for each case, the figure shows the configuration with a learning rate = 0.3, which is the best average configuration) and decision trees, and the two best ensembles: Bagging+Random Undersampling (BAGRUS) and SMOTEBoost. Higher window sizes could not be tested, due to the small size of some of the experimental tests performed with some tools (the extension of the window size reduced the size of the dataset and the level of imbalance in the dataset, because, as the first instances were non-fault instances for any tool, the use of a window deleted as many non-fault instances as fitted in the window size and none of the fault instances from the dataset). It may be appreciated from this figure that the ensembles were less sensitive to the window size, while obtaining, in any case, better performance than the basic classifiers. Besides, MLPs performed poorly with small dataset sizes (and therefore a small number of training instances) and were unable to achieve stable behavior, due to the small dataset sizes of each tool, as the window sizes were extended. This argument also supports the use of decision trees as base classifiers for the ensembles, because the high sensitivity of MLPs to the training dataset might otherwise reduce the accuracy of the ensemble models.

Fig. 3. Evolution of the recall metric with the sliding window size for different machine-learning models.



4.2 Industrial implementation

The identification of the most accurate prediction model might close the research from the computer science point of view, but the process engineer in the factory would expect a visual implementation of the prediction model for direct use [27]. From the industrial point of view, the recall model should be maximized, as false negatives are in every way undesirable. However, an algorithm with a high false positive rate is also undesirable and the precision indicator can help to maximize the sensitivity to false positive cases. Therefore, both precision and recall should be simultaneously considered as indicators.

Considering both quality indicators, the best prediction model in terms of recall, bagging combined with random undersampling (light blue), and the best prediction model in terms of precision, SMOTEBoost (dark red), are represented in Figure 4. All the experiments for a sliding window of 11 for the 35 tapping tools are shown (Tap Tool 1 to Tap Tool 35 in the figure labels). Each figure in Figure 4 refers to one of the tapping tools. Appendix I of this article shows high-quality images of the 35 figures for a detailed analysis. The X-axis refers to the threads for the corresponding tap tool in a random order, but locating all the positive (pass) threads first and then all the negative (non-pass) threads, in such a way that, ideally, a perfect model would generate a step function graph for each tap tool (0 at the left side and 1 at the right side). The prediction for each thread by both prediction models, shown on the Y-axis, was expressed as the probability of a thread quality inspection: positive (pass: 0) or negative class (no pass: 1). The prediction for each thread was computed for cases where the thread belonged to the validation fold in the cross-validation scheme. The models could not therefore have any information on thread behavior during the training stage. The thin black horizontal line can be considered a threshold between pass and no pass states. It was fixed at a probability of 50%, the standard value in classification tasks. The thick black line shows the real value of the output in the Y-axis: the 0 value represents a pass-thread and the value of 1 represents a non-pass thread. In summary, if a model provides a probability higher than 50% (0.5 in Figure 4), the engineer will expect a non-pass thread; therefore, although the figures look very noisy, because each instance has a different prediction probability, the process engineer can consider them only as binary values (0 or 1), taking into account the 50%-threshold.

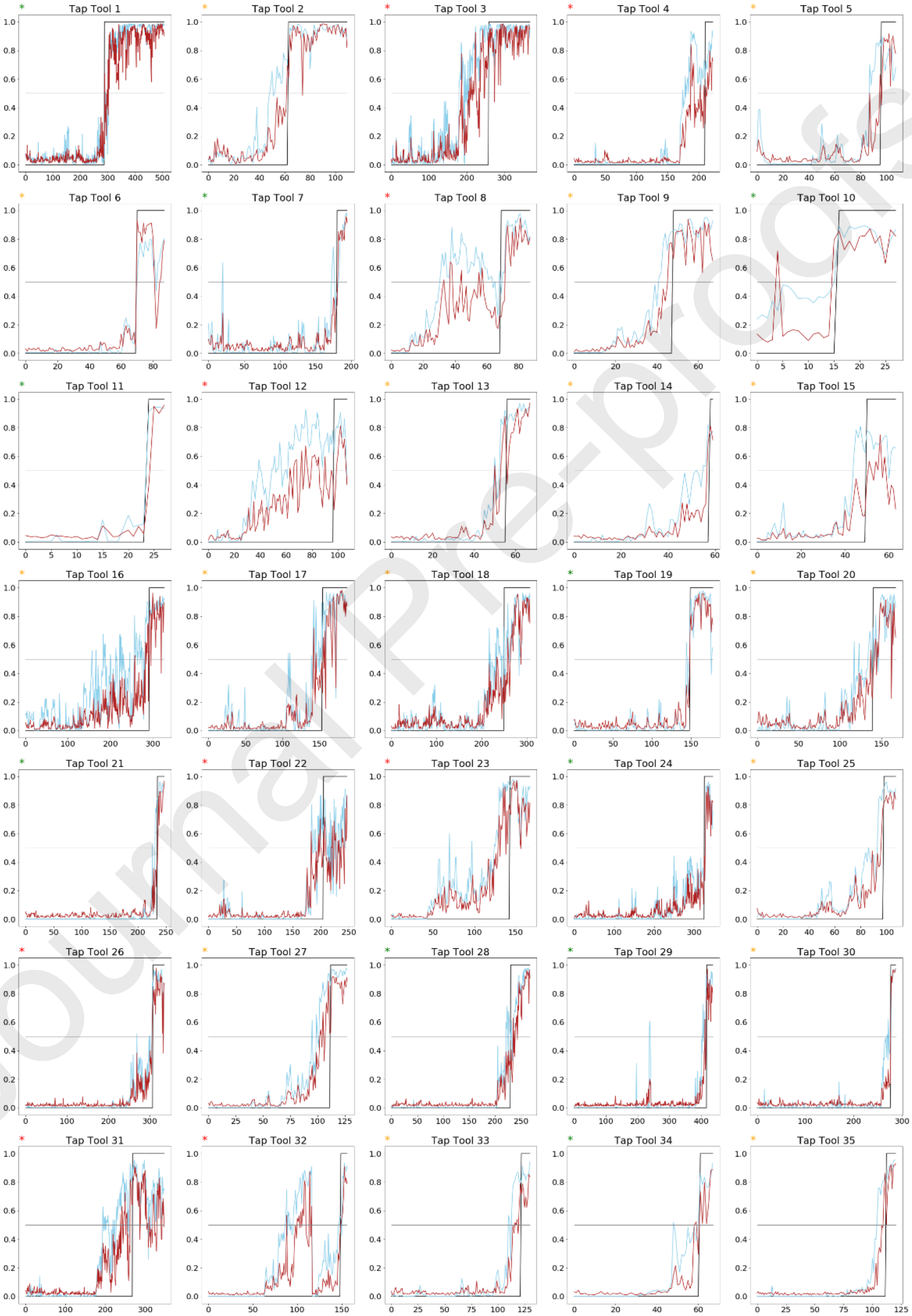


Fig. 4. Probability of pass and non-pass quality of all the threads in the dataset predicted with bagging combined with random undersampling (light blue) and SMOTEBoost (dark red).

This figure requires close analysis to extract all the information that may be of use to the process engineer. Firstly, both algorithms were only capable of classifying all the threads of a few tapping tools (numbers 1, 11 and 24). In other cases (tools number 7, 10, 19, 21, 26, 28, 29, and 34), one of the algorithms was able to classify almost all the threads correctly (over 99.5% of the cases). Besides, it was clear from the differences in their behavior that Bagging+Random Undersampling detected all the faults and could anticipate failure (although it can commit false positives) and SmoteBoost made much more conservative predictions, committing far fewer false positives at the expense of not detecting all the failures. This result can clearly be seen in tapping tool number 15, where the orange line shows probabilities under 50% for all the failures and the green line provides probabilities over 50% for all the right threads. Besides, Figure 4 shows that the tapping process is a noisy process and, although with good results, the analysis of the torque signals might not be enough to predict the quality of the threads that are produced with 100% certainty. Some clear cases are tool numbers 3, 4, 8, 12, 22, 23, 26, 31, and 32. But in all cases the proposed strategy, mixing the predictions of a model with high Recall and a model with high Precision, helps to take a decision on the changes to the tool.

5. Conclusions

An extensive experimental stage of internal thread production has been conducted in this study using cutting taps under high-speed cutting conditions with no coolants. Secondly, different machine-learning strategies have been tested to build a reliable prediction model for thread quality. The following conclusions can be extracted from this study:

- If only easy-to-measure process variables under industrial conditions, such as tool coating and torque signals were considered, immense effort would be required to achieve accurate prediction models, although these conditions are basic requirements in small workshops where no traceability of tapping tools is implemented.
- The torque signal is a key parameter, because it describes the essential information in the tapping operation independent of the coatings. The huge dispersion between similar tools is due to high-speed cutting under dry conditions.
- Despite the highly different coatings tested in this research, the thread-quality classification system managed to work independently of the coatings. Therefore, this approach could work in an automated manufacturing production line with tools from different providers.
- The effect of adding some statistical variables to the dataset, such as the slope of the regression line, the standard error of the regression fit and the true strength index in all cases improved the accuracy of the machine-learning models.
- Bagging+Random Undersampling provided the best model when searching to detect all non-pass threads although it can classify pass threads as non-passes. The following quality indicators were achieved by the best model (the original dataset plus the statistical inputs): Recall 0.960; Precision 0.709; Accuracy 93.2%; and, AUC, 0.985.
- The SMOTEBoost provided the best model when searching to detect all pass threads although it can classify non-pass threads as passes. The following quality indicators were achieved by the best model (the original dataset plus the statistical inputs): Recall 0.940; Precision 0.893; Accuracy 97.3%; and, AUC 0.995.
- Both ensemble-based models showed a lower sensitivity than base classifiers to the window size with higher accuracy for smaller window sizes, demonstrating their higher learning ability with small datasets. Besides, even when the traceability of tapping tool was available (equivalently a 0-size window can be used), ensembles continued to show high prediction accuracy.

- The graphic combination of the predictions of both models, Bagging+Random Undersampling and SMOTEBoost, provided a reliable industrial tool for proper classification of thread quality, although a conservative decision might be taken, considering the highly dispersed behavior between similar tapping tools, a natural limitation in tapping modeling. In any one study of a tap tool, this strategy will help to take decisions on changes to the tool, providing reliable visual information on tap tool behavior.

Future lines of work will focus on the extension of this classification strategy to different workpiece materials, tap diameters, and cutting conditions, to extend the final use of the prediction models. In addition, it will be tested in manufacturing cells, to assess thread quality during the tapping processes. The combination of these classification models with automatic image-processing systems can improve system reliability and reduce the true negatives as previous works have outlined for other solutions [27], so that risky tool wear may be detected. Finally, this classification strategy will be customized to warn when thread quality is out of control or requires manual evaluation, because the tapping tool is close to the non-pass thread area.

Acknowledgments

This investigation was partially supported through Projects TIN2015-67534-P (MINECO/FEDER, UE) of the Ministerio de Economía Competitividad of the Spanish Government and project BU085P17 (JCyL/FEDER, UE) of the Junta de Castilla y León, both co-financed through European-Union FEDER funds and project IT-2005/00201 of the INNOTEK Program of the Basque Government. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. Machine learning, 6(1), 37-66.

[2] Ahn J. H., Lee D. J., Kim S. H., Cho K. K. (2003) Effects of synchronizing errors in cutting performance in the ultrahigh-speed tapping. Annals of the CIRP 52(1):53–56.

[3] Armarego M. N. & Chen P. (2002) Predictive models for the forces and torque in machine tapping with straight flute taps. Annals of CIRP 51(1):75–78.

[4] Barandela, R., Valdovinos, R. M., & Sánchez, J. S. (2003). New applications of ensembles of classifiers. Pattern Analysis & Applications, 6(3), 245-256.

[5] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 6(1), 20-29.

[6] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[7] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[8] Bustillo, A., Pimenov, D. Y., Matuszewski, M., & Mikolajczyk, T. (2018). Using artificial intelligence models for the prediction of surface wear based on surface isotropy levels, Robotics and Computer-Integrated Manufacturing, 53, 215-227.

[9] Bustillo, A., López de Lacalle, L.N., Fernández-Valdivielso, A., & Santos, P. (2016). Data-mining modeling for the prediction of wear on forming-taps in the threading of steel components. J. Comput. Des. Eng., 3, 337–348

[10] Chang, K. H., Chang, Y. C. & Lai, P. T. (2014). Applying the concept of exponential approach to enhance the assessment capability of FMEA, J Intell Manuf 25: 1413. https://doi.org/10.1007/s10845-013-0747-9

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[12] Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In European conference on principles of data mining and knowledge discovery (pp. 107-119). Springer, Berlin, Heidelberg.

[13] Chen Y. B., Sha J. L., Wu S. M. (1990). Diagnosis of tapping process by information measure and probability voting approach. J Eng Ind 112:319–325

[14] Chen, S., He, H., & Garcia, E. A. (2010). RAMOBoost: ranked minority oversampling in boosting. IEEE Transactions on Neural Networks, 21(10). 1624-1642.

[15] Díez-Pastor, J. F., García-Osorio, C., Barbero-García, V., & Blanco-Álamo, A. (2013). Imbalanced learning ensembles for defect detection in X-ray images. In International Conference on Industrial,

Engineering and Other Applications of Applied Intelligent Systems (pp. 654-663). Springer, Berlin, Heidelberg.

[16] Díez-Pastor, J. F., Rodríguez, J. J., García-Osorio, C., & Kuncheva, L. I. (2015). Random balance: ensembles of variable priors classifiers for imbalanced data. Knowledge-Based Systems, 85, 96-111.

[17] Ferreiro S., Sierra B., Irigoien I., & Gorritxategi E. (2011). Data mining for quality control: Burr detection in the drilling process, Computers & Industrial Engineering, 60(4), 801-810.

[18] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 463-484.

[19] Gil Del Val A., Fernández J., del Castillo E., Arizmendi M., & Veiga F. (2013). Monitoring of thread quality when tapping nodular cast iron with TiN-coated HSS cutting taps. Int J Adv Manuf Technol 69:1273–1282. https://doi.org/10.1007/s00170-013-5078-7

[20] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18

[21] Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Springer, Berlin, Heidelberg.

[22] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 1322-1328). IEEE.

[23] Hsia, S.-Y., Chou, Y.-T., Lu, G.-F. (2016). Analysis of sheet metal tapping screw fabrication using a finite element method. Applied Sciences (Switzerland), 6 (10), art. no. 300.

[24] Hsu, C.-C., Yeh, S.-S., & Lee, J.-I. (2016). Effect analysis and optimal combination of cutting conditions on the cutting torque of tapping processes using Taguchi methods. IEEE International Conference on Automation Science and Engineering, 2016-November, art. no. 7743544, pp. 1215-1218.

[25] Juez-Gil, M., Erdakov, I. N., Bustillo, A., Pimenov, D. Y. (2019). A regression-tree multilayer-perceptron hybrid strategy for the prediction of ore crushing-plate lifetimes, Journal of Advanced Research, 18, 173-184.

[26] Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. Neural computation, 13(3), 637-649.

[27] Krolczyk, G. M., Maruda, R. W., Krolczyk, J. B., Wojciechowski, S., Mia, M., Nieslony, P., & Budzik, G. (2919). Ecological trends in machining as a key factor in sustainable production – A review, Journal of Cleaner Production, 218, 601-615, https://doi.org/10.1016/j.jclepro.2019.02.017.

[28] Li W., Li D., & Ni J. (2002). Diagnosis of tapping process using spindle motor current. Int J Mach Tool Manuf 43:73–79.

[29] Liao, T. W. (2008). Classification of weld flaws with imbalanced class data. Expert Systems with Applications, 35(3), 1041-1052.

[30] Liu T. I., Ko E. J., & Sha S. L. (1990). Intelligent monitoring of tapping tools. J Mater Shaping Tech 8: 249. https://doi.org/10.1007/BF02833821.

[31] Liu T., Ko E. J., & Sha S. L. (1991). Diagnosis of tapping processes using an AI approach. J Mater Shaping Tech 9(1):39–46.

[32] Liu T-I., Lee J., Liu G., & Zhang W. (2012). Monitoring and diagnosis of the tapping process for product quality and manufacturing. Int J Adv Manuf Tech. 64 (5–8):1169–1175.

[33] Martin-Diaz, I., Morinigo-Sotelo, D., Duque-Perez, O., & Romero-Troncoso, R. D. J. (2017). Early fault detection in induction motors using AdaBoost with imbalanced small data and optimized sampling. IEEE Transactions on Industry Applications, 53(3), 3066-3075.

[34] Mia, M., Krolczyk, G., Maruda, R. (2019). Intelligent Optimization of Hard-Turning Parameters Using Evolutionary Algorithms for Smart Manufacturing, Materials 12(6), art. no 879.

[35] Monka, P., Monkova, K., Modrak, V., Hric, S., Pastucha, P. (2019). Study of a tap failure at the internal threads machining. Eng. Fail. Anal., 100, 25–30.

[36] Moreira, G. R., Lahr, G. J. G., Boaventura, T., Savazzi, J. O., & Caurin, G. A. P. (2018). Online prediction of threading task failure using Convolutional Neural Networks IEEE International Conference on Intelligent Robots and Systems, art. no. 8594501, pp. 2056-2061.

[37] Oezkaya, E. & Biermann, D. (2018). Development of a geometrical torque prediction method (GTPM) to automatically determine the relative torque for different tapping tools and diameters Int J Adv Manuf Technol 97: 1465-1479. https://doi.org/10.1007/s00170-018-2037-3

[38] Polikar, R. (2006). Ensemble based systems in decision making. IEEE Circuits and systems magazine, 6(3), 21-45.

[39] Provost, F., & Domingos, P. (2003). Tree induction for probability-based ranking. Machine learning, 52(3), 199-215.

[40] Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. IEEE transactions on pattern analysis and machine intelligence, 28(10), 1619-1630.

[41] Sanchez-Egea, A.J., & Lopez de Lacalle, L.N. (2018) Machines, processes, people and data, the keys to the 4.0 revolution. *DYNA*, 93(6), 576-577

[42] Shao, H., Li, A., Xu, L. et al. (2019). Scalability in manufacturing systems: a hybridized GA approach J Intell Manuf 30: 1859. https://doi.org/10.1007/s10845-017-1352-0

[43] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40(1), 185-197.

[44] Visa, S., & Ralescu, A. (2005). Issues in mining imbalanced data sets-a review paper. In Proceedings of the sixteen midwest artificial intelligence and cognitive science conference (Vol. 2005, April, pp. 67-73). sn.

[45] Teti, R., Segreto, T., Caggiano, A., & Nele, L. (2020). Smart Multi-Sensor Monitoring in Drilling of CFRP/CFRP Composite Material Stacks for Aerospace Assembly Applications. Appl. Sci., 10, 758.

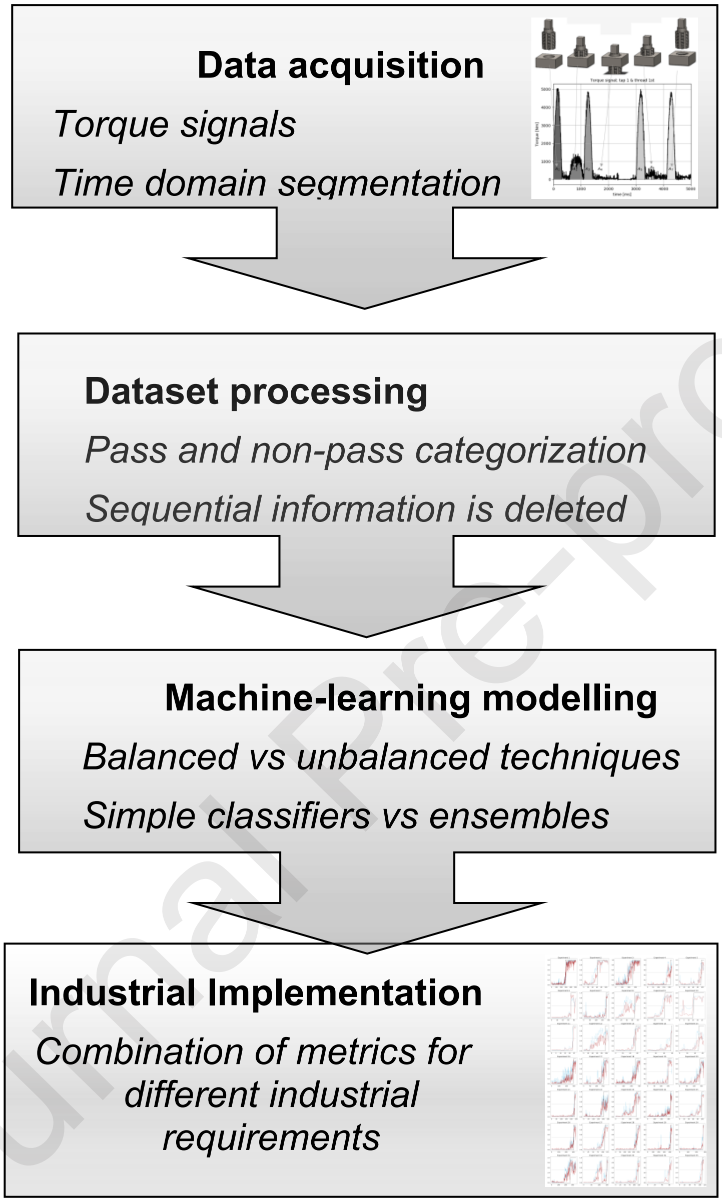
[46] Ying-Chao Ma, MinWan, YunYang, & Wei-Hong Zhang (2019). Dynamics of tapping process. International Journal of Machine Tool and Manufacture, 140: 34-47. https://doi.org/10.1016/j.ijmachtools.2019.02.002

[47] Yonghong Peng (2004). Intelligent condition monitoring using fuzzy inductive J Intell Manuf, 15: 373-380 https://doi.org/10.1023/B:JIMS.0000026574.95637.36

[48] Wang, J., Liu, S., Gao, R. X., & Yan, R. (2012). Current envelope analysis for defect identification and diagnosis in induction motors. Journal of Manufacturing Systems, 31(4), 380-387.

[49] Zhang, L., Lin, J., & Karim, R. (2017). Sliding window-based fault detection from high-dimensional data streams. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(2), 289-303.

Graphical abstract



Data acquisition

Torque signals

Time domain segmentation

Dataset processing

Pass and non-pass categorization

Sequential information is deleted

Machine-learning modelling

Balanced vs unbalanced techniques

Simple classifiers vs ensembles

Industrial Implementation

Combination of metrics for different industrial requirements

Highlights:

- An extensive industrial dataset of threads processed with different coated tools
- A new approach to predict threads quality considering tool coating and torque signals
- Different machine-learning techniques for balanced & imbalanced datasets were tested
- Ensembles are the most accurate, easily-optimized & industrially-applicable models

CRediT author statement

Andres Bustillo: Conceptualization, Methodology, Writing- Original draft preparation. **Alain Gil del Val.**: Data curation, Writing- Original draft preparation, Visualization, Investigation. **Fernando Veiga**: Supervision, Writing- Reviewing and Editing**: Jose Francisco Diaz-Pastor**: Software, Validation, Writing- Original draft preparation.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: