



Use of artificial intelligence techniques in characterization of vibration signals for application in agri-food engineering

Amalia Luque^{1,2}  · Daniel Campos Olivares³ · Mirko Mazzoleni² · Antonio Ferramosca² · Fabio Previdi² · Alejandro Carrasco³

Accepted: 1 March 2025
© The Author(s) 2025

Abstract

Bottling machinery is a critical component in agri-food industries, where maintaining operational efficiency is key to ensuring productivity and minimizing economic losses. Early detection of faulty conditions in this equipment can significantly improve maintenance procedures and overall system performance. This research focuses on health monitoring of gripping pliers in bottling plants, a crucial task that has traditionally relied on analyzing raw vibration signals or using narrowly defined, application-specific features. However, these methods often face challenges related to limited robustness, high computational costs, and sensitivity to noise. To address these limitations, we propose a novel approach based on generic features extracted through basic signal processing techniques applied to vibration signals. These features are then classified using a random forest algorithm, enabling an effective analysis of health states. The proposed method is evaluated against traditional approaches and demonstrates clear advantages, including higher accuracy in detecting and classifying faulty conditions, greater robustness against random perturbations, and a reduced computational cost. Additionally, the method requires fewer training instances to achieve reliable performance. This study highlights the potential of artificial intelligence and signal processing techniques in predictive maintenance, offering a scalable and efficient solution for fault detection in manufacturing processes, particularly within the agri-food sector.

Keywords Bottling systems · Preventive maintenance · Health condition monitoring · Feature extraction

1 Introduction

Maintenance strategies play a critical role in ensuring the reliability and efficiency of industrial systems. However, traditional approaches often fail to fully address the complexity of modern industrial environments, where real-time monitoring and predictive insights are essential. Key challenges include the high dimensionality of monitoring data, the imbalanced nature of datasets due to infrequent fault occurrences, and the lack of cost-effective solutions for comprehensive Health Monitoring (HM).

Defining a proper maintenance strategy of devices and systems or, in a broader sense, an Asset Management (AM), is a very hot topic with a huge impact on industry operation as it enhances performance, increases reliability, decreases time-down, and saves cost. An up-to-date state of the art of this topic can be found in [1].

For any type of organizations, developing an effective AM is a key factor to achieve value by balancing risks and opportunities, to attain an optimum trade-off between costs, risks, and performance [2].

In the context of AM, up to five levels in the maintenance of devices and systems are usually considered [3–5]: Reactive Maintenance (level 1), running until a failure and then repairing; Preventive Maintenance (level 2), cyclic inspection and conservation activities based on time or use; Condition-Based Maintenance (level 3), triggered by some rules (e.g. exceeding a certain threshold) based on objective measures obtained by several sensors; Predictive Maintenance (level 4), where actual sensor data about the equipment and its operating condition are extensively and continuously analyzed for a very early detection of potential malfunctioning, suggesting equipment or operating conditions modifications and predicting its Remaining Useful Life (RUL); and Risk-Based Maintenance (level 5), for an integral and comprehensive approach, considering assets performance, reliability, and

Extended author information available on the last page of the article

costs. These maintenance maturity levels are depicted as a pyramid in Fig. 1.

This paper focuses on the three upper levels of the pyramid (highlighted in green), where Health Monitoring is essential. HM involves capturing data such as vibrations (accelerations), sounds, temperatures, voltages, and electric currents. These measurements are typically recorded as temporal signals or time series data, often sampled at a constant rate.

The field of industrial Health Monitoring (HM) has seen considerable advancements in maintenance strategies, encompassing rule-based systems, traditional Machine Learning (ML) techniques, and specialized feature extraction methods. Despite these advancements, existing methodologies face significant challenges, particularly in addressing the dynamic and resource-constrained nature of industrial environments. Rule-based systems [6], while conceptually simple, lack the adaptability to handle complex and variable fault patterns, often resulting in suboptimal performance in dynamic scenarios. Similarly, traditional ML techniques [7] typically rely on extensive labeled datasets to achieve high accuracy [8, 9]. However, the collection of such datasets in HM applications is time-consuming, costly, and impractical, as faults occur infrequently and datasets are often small and imbalanced.

Recent studies have explored data-driven and ML-based approaches as state-of-the-art solutions for analyzing high-dimensional HM signals [10]. However, few have addressed the critical trade-off between feature extraction complexity and performance in resource-constrained environments. For example, methods reliant on deep learning architectures often achieve superior performance but at the cost of significant computational and data requirements, which limits their practical applicability in real-world settings [11]. Similarly, many existing techniques fail to account for the robustness needed to handle noisy environments and diverse fault types, resulting in reduced reliability across applications.

A major limitation of existing data-driven methods is their inability to balance computational efficiency with classification accuracy. Approaches based on raw features, such as vibration signals comprising thousands of samples, yield high-dimensional datasets imposing excessive computational demands and often lead to overfitting when training ML models [12]. These issues are further exacerbated by the need for large and representative datasets, which are challenging to obtain in industrial contexts where faulty states are rare.

Conversely, specialized feature extraction methods are frequently designed for specific fault detection tasks [13], requiring deep domain expertise and significant computational resources. When a deep knowledge of the signal interpretation is available, very specialized features can be extracted applying specific signal processing algorithms. For instance, MPEG-7 [14] parameters or fine-tuned Mel Frequency Cepstrum Coefficients [15] for audio signals, State-of-Charge and State-of-Health related Key Performance Indicators (KPI) for battery description [16], or kurtogram-based attributes for rolling bearings [17]. While these methods may achieve high accuracy in controlled environments, their scalability and generalizability to diverse industrial settings are limited.

On one hand, describing signals with raw features is the simplest extracting procedure but it yields to high-dimensional datasets, requiring a large number of instances and very demanding classification algorithms. On the other hand, characterizing signals by very specialized features drives to low-dimensional reduced datasets which are more easily classified, but at the cost of a large extraction effort which requires a deep engineering knowledge.

To address these limitations, this study introduces a novel methodology leveraging generic features for fault detection and classification. Generic features provide a middle ground between raw and specialized features, offering computa-

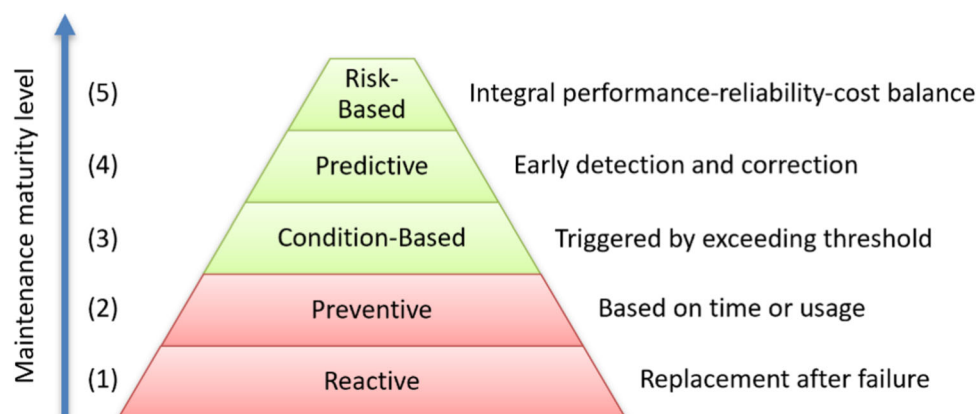


Fig. 1 Maintenance maturity pyramid

tional efficiency without sacrificing classification accuracy. This trade-off is depicted in Fig. 2. Unlike specialized features, they require minimal domain-specific knowledge and are inherently adaptable to various fault types. This innovation significantly reduces data requirements, computational complexity, and costs, making the methodology particularly well-suited for real-world applications where scalability, robustness, and efficiency are paramount. Moreover, the proposed approach demonstrates strong resilience to noise and variability, which are common in industrial environments, thereby enhancing its practicality and reliability.

In this approach, vibration signals are summarized by a reduced set of generic features, which are commonly extracted from the signal using time, frequency, or time-frequency domain analysis. Some more specific methods have also been proposed, based on Cepstrum [18], Wavelet Analysis [19], Hilbert-Huang Transform [20], Empirical Mode Decomposition [21], Linear Predictive Coding [21], Time Series Analysis [22], etc. A review of these feature extraction procedures has been described for different applications, such as mechanical vibrations [23], sounds [24] or biomedical signals [25].

The present paper studies the compromise between classification and feature extraction when the three previous defined types of features are used to detect faults in a particular industrial context. In the research, we focus on the input gripping pliers of the blow molding machine in a plastic bottling plant (detailed in Section 2.1), where the features must be extracted from vibration signals acquired by accelerometers (extraction process described in Sections 2.2, 2.3, and 2.4).

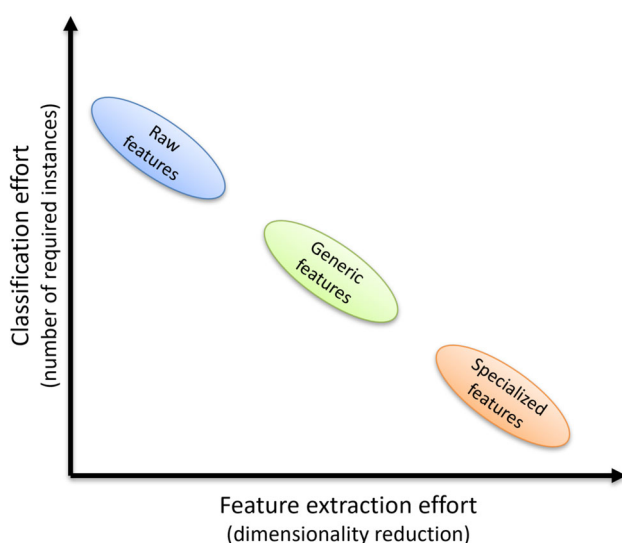


Fig. 2 Trade-off between classification and feature extraction efforts

The analysis of the input gripping pliers serves as a critical case study, as their malfunction can disrupt the production line and contribute to inefficiencies. Understanding the relationship between feature extraction and classification accuracy in this context not only aids in fault detection but also underscores the broader implications for optimizing production systems. By addressing these challenges, this research aims to highlight how improved Health Monitoring (HM) can directly impact key performance indicators like overall equipment effectiveness (OEE), which remains a pressing issue in many industries.

The importance of this study relies on one hand, in the fact that the OEE in production companies is about 60% [26, 27]. The 40% loss of efficiency is mainly due to five factors: (i) planned stoppages, for planned maintenance or cleaning activities; (ii) machine failures, due to unexpected damages; (iii) short process failures (up to a few minutes), caused by jams or lacking products; (iv) process slowdown, where the equipment is not running at its optimum production rate; and (v), initialization procedures, to prepare the machine for a certain production process. Most of these factors can be significantly reduced with an adequate AM strategy, increasing the OEE and enhancing the overall economic results.

Also, the studied application is very significant as bottling plants are an essential element in the production phase of the beverage [28] and other agri-food products life cycle [29]. The food industry includes all activities (production, processing, manufacturing, sale, and distribution) of food, dietary supplements, and beverages [30]. In other words, the food industry includes all the stages of the process, from design to delivery of the product to the customer, through design, construction, and maintenance, in the animal nutrition industry and the food industry [31].

The primary contribution of this research is the demonstration that, in the studied industrial setting, a set of generic features offers superior performance compared to both raw, unprocessed features and highly specialized, knowledge-based attributes. Unlike raw or highly specialized features, generic features achieve a significantly higher classification accuracy with much lower computational demands. This advantage is particularly noteworthy in fault type identification, where generic features outperform specialized attributes, which, while effective for general fault detection, show limited capability in pinpointing specific fault causes.

Furthermore, generic features prove to be more resilient to noise, maintaining classification performance even in datasets with random noise contamination, a limitation observed in raw and specialized features. Another important contribution is the reduction in dataset size requirements; classifiers trained with generic features need far fewer data points, making this approach more practical and cost-effective for industrial applications.

Additionally, the research highlights the advantage of dividing vibration signals into smaller segments, thereby increasing the number of training examples and improving classification performance, with an optimal segment length identified to balance representational accuracy and instance availability.

By advancing the understanding and application of generic features, this study contributes to bridging critical gaps in HM methodologies. It offers a scalable, efficient, and robust solution that addresses the key shortcomings of existing approaches while enabling more effective fault detection and classification. These contributions mark a significant step forward in the field, particularly for resource-constrained industrial systems requiring practical, high-performance HM solutions.

The remaining of the paper is organized as follows: Section 2 describes the engineering application of the research (a bottling system), their health conditions, the dataset generation process and the classification algorithm proposed; Section 3 presents the results obtained by applying three types of features (raw, specialized and generic); Section 4 discusses these results analyzing the impact of perturbations on datasets, the number of training instances, the length of vibration signals and the required computational effort; and, finally, Section 5 states the main conclusions of the article.

2 Methodology

2.1 The bottling system

A bottling plant for filling beverages is a set of several very different types of specialized machines and conveyors that automatically handles the entire process from pallets supplied with boxes containing empty bottles to the final output of pallets with (clean) boxes and filled and labeled bottles [32].

The bottling process can be abstracted as a discrete sequence of operations aimed at transforming raw materials into a finished product ready for distribution. Let $S = s_1, s_2, \dots, s_n$ represent the set of sequential stages in the bottling system, where s_i denotes the i -th process, and n is the total number of stages. Each stage involves an input state b_{i-1} , a transformation $f_i(b_{i-1})$, and an output state b_i . Mathematically, the system can be expressed as:

$$b_i = f_i(b_{i-1}), i = 1, \dots, n \quad (1)$$

where b_0 is the raw material input and b_n represents the final output, i.e., packaged and palletized bottles. The goal of this work is to implement a health monitoring system for critical components $C \subset S$, ensuring operational reliability and efficiency.

Specifically, the bottling system under investigation consists of the following processes:

1. Input Stage (s_1): A conveyor belt delivers raw plastic preforms (b_0) to the system (state b_1).
2. Heating Stage (s_2): An oven heats the preforms to a temperature sufficient for blow molding, transforming the preforms into a deformable state b_2 .
3. Transfer Stage (s_3): Gripping pliers transfer the heated preforms to the blow molding machine (state b_3).
4. Blow Molding Stage (s_4): The blow molding machine applies a shaping force to form the final bottle structure b_4 .
5. Intermediate Transfer (s_5): Gripping pliers transfer the bottles to the filling machine (state b_5).
6. Filling and Capping Stage (s_6): The bottles are filled with a target liquid volume and capped (state b_6).
7. Packaging Stage (s_7): The bottles are grouped into packages (state b_7).
8. Palletizing Stage (s_8): Packages are stacked onto pallets for distribution, ensuring a spatial arrangement that maximizes stability during transport (state b_8).

The health monitoring system, indicated in gray in Fig. 3, focuses on critical stages $C = \{s_3, s_4, s_5\}$, particularly the gripping pliers, which are prone to wear and mechanical failures. These components operate under thermal and mechanical conditions that are monitored using specialized sensors. Fig. 3 illustrates the system layout, highlighting cold parts (blue), hot parts (orange), and health monitoring system integrations (gray). By generalizing the system's operations using the above formulations, the methodology can be adapted to a wide range of bottling processes beyond the specific implementation described here.

The gripping pliers are embedded in a mechanical system with several moving parts, where the plier's position is controlled by a rotating motor linked through a bearing, and the opening of the plier's arms is driven by a cam coupled by a spring. Four gripping pliers, at a 90° angle, are linked to the rotating motor through a rod (see Fig. 4). The bottles arrive at a nominal rate of 7200 pieces per hour (120 per minute) which means that motor rotates at 30 rpm (4 bottles are managed in each revolution). The operation at an increased rate of 9200 bottles per hour (bph) has also been tested (the motor rotating at 38.3 rpm).

In this arrangement the most mechanically demanded elements are the bearings (supporting a continuous high-speed friction) and the springs (suffering repetitive cycles of stretching and compression). Additionally, the input gripping pliers (showed in Fig. 5) suffer increased wear and tear as they operate at slightly higher temperatures than the output counterparts. Therefore, the focus of this research will be in

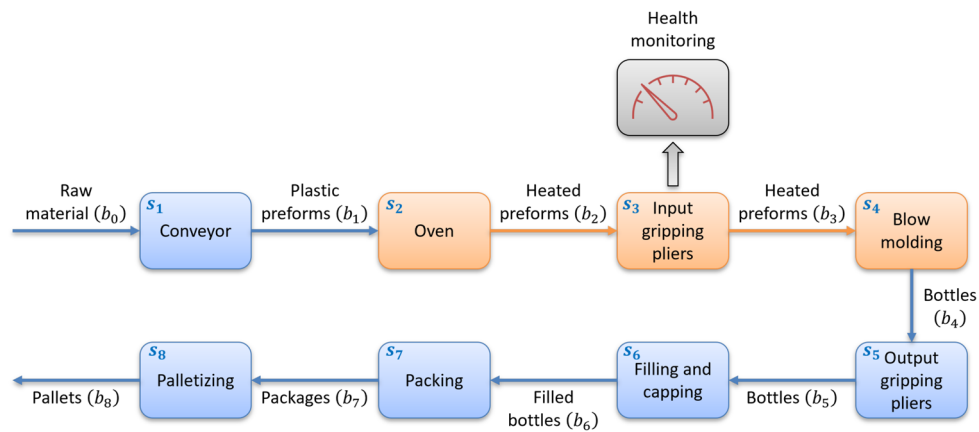


Fig. 3 Bottling system diagram. Cold parts (blue), hot parts (orange), and health monitoring (gray)

those elements. A more detailed description of the bottling system and the studied elements can be found in [33].

The hypothesis of this work is that possible damages in any of the elements under supervision (bearings or springs), will yield a different pattern of mechanical vibrations. To explore this postulate, a single axis accelerometer is attached to the pliers’ structure measuring de vibrations in the radial direction (orthogonal to the rotation axis). More complex faults may necessitate monitoring across multiple axes; however, these scenarios are beyond the scope of our study. Since the vibration signals (both healthy and faulty) have a bandwidth of approximately 5.5 kHz, sensor values are acquired at a rate of 12.8 kHz. This rate exceeds twice the bandwidth and is among the sampling options available in the data acquisition equipment described in [33].

A bottling machine, typical of those used in bottling plants, was tested across 13 experiments conducted over multiple days under controlled conditions. The machine was evaluated under three health conditions: in four experiments (1 to 4), it operated in a healthy state; in five experiments (5 to 9), a spring damage (SD) fault was introduced; and in the remaining four experiments (10 to 13), a bearing damage (BD) fault was induced. Each experiment lasted approximately 3 minutes.

2.2 Raw features

Considering each experiment an instance (example) in a machine learning approach, their number (4 or 5 of each

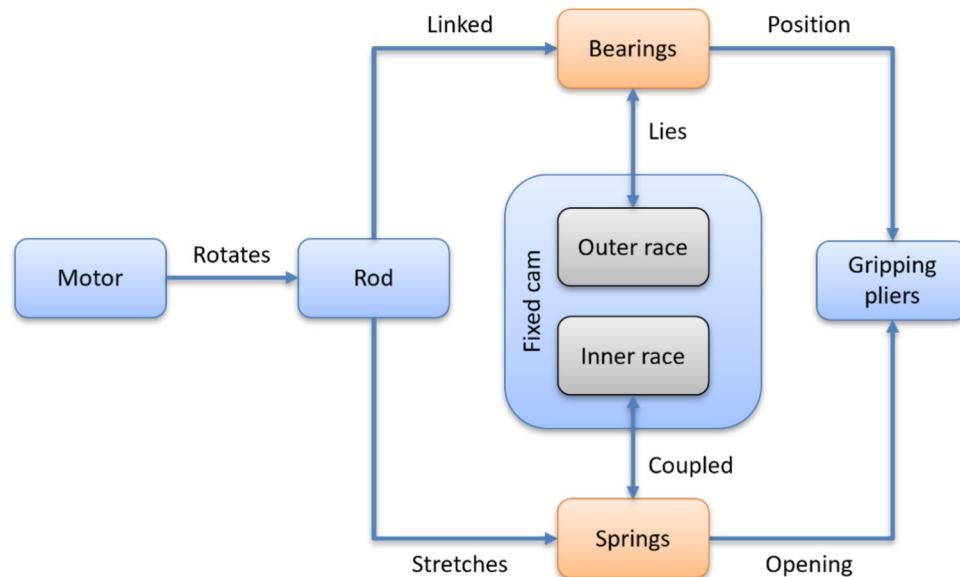


Fig. 4 Diagram of the mechanical system controlling position and opening of the input gripping pliers. The most mechanically demanded elements are colored in orange

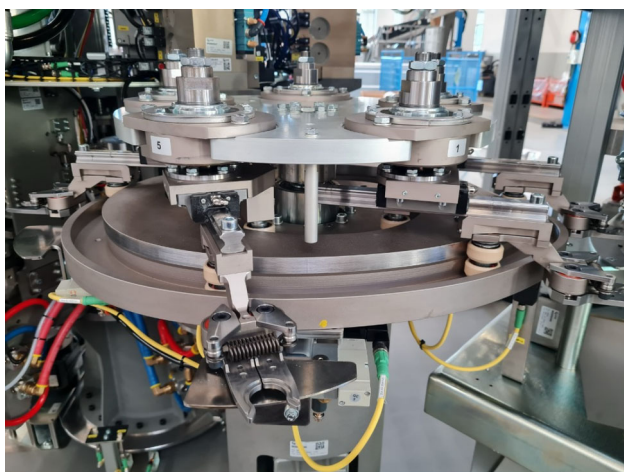


Fig. 5 Input gripping pliers of a bottling system

class) are clearly insufficient for a data-driven classification. Therefore, each experiment is split in 5-seconds segments which obtains 497 hundred instances (instead of the original 13). The number of segments in each class is detailed in Table 1. The value chosen for the segment duration (five seconds) is discussed in Section 4.5.

The number of values (features in ML terminology) characterizing each 5-seconds segment is 64000, a figure directly derived from the 12.8 kHz sampling rate. This is a high dimensional vector that poses a special challenge for classifier algorithms.

A graphical representation of the vibration signals is depicted in Fig. 6, for examples of each health state. This time-domain representation seems quite confusing and, at a first glance, no definitive conclusion about the correspond-

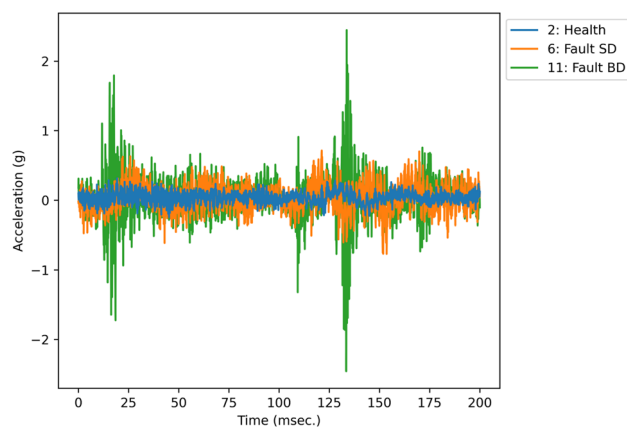


Fig. 6 Examples of vibration signals for each health state. Acceleration units are expressed in terms of g (9.81 m/s^2)

ing health state can be easily reached. Then, another set of features representing vibrations must be explored.

2.3 Specialized features

As the high dimensionality of raw vectors may cause special complexities to the ML algorithms, specially is the number of instances is not too large, alternative features are considered. In a general case, the deep knowledge of the engineering process yielding the vibration signals, can be used to design a very short set of highly specialized features that summarize the health state of the system. For the input gripping pliers of the bottling machine, these summarizing features are derived from the hypothesis—initially inspired by [33]—that the system behaves similarly to a rolling bearing, an assumption later validated through this research. This approach lever-

Table 1 Experiments and segments for each health state

Health State	Experiment	Speed (bph)	# Segments	# Segments (per class)
Healthy	1	7200	31	139
	2	7200	38	
	3	9200	42	
	4	9200	28	
Fault SD (spring damaged)	5	9200	43	213
	6	9200	43	
	7	9200	42	
	8	7200	42	
	9	7200	43	
Fault BD (bearing damaged)	10	7200	38	145
	11	9200	41	
	12	9200	28	
	13	7200	38	

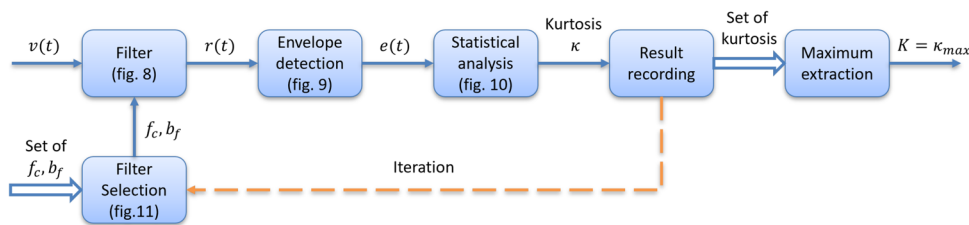


Fig. 7 Overview of the process to extract the kurtosis-based specialized feature

ages the extensive body of knowledge on rolling bearings, which have been widely studied for health monitoring purposes [34]. Specifically, the specialized features proposed here are primarily based on kurtosis, as defined in [35] and [17].

The process to obtain these features begins with band-pass filtering of the vibration signal, followed by envelope detection and calculation of the statistical kurtosis of the resulting values. This sequence is repeated across a range of filters, selecting the filter that produces the highest kurtosis as the optimal choice, with this kurtosis value serving as the feature characterizing the vibration. To provide a clearer understanding of the workflow and its connection to the paper’s outcomes, Fig. 7 offers an overview of the process and references Figs. 8 to 11 to highlight their interrelation.

In more detail, for each vibration segment, defined as a signal $v(t)$, a band-pass filtered signal $r(t)$ is obtained for a filter in the frequency range $[f_c - b_c, f_c + b_f]$, where f_c is the central frequency and b_c the semi-bandwidth of the filter. Then, the envelope $e(t)$ of the filtered signal $r(t)$ is derived. Since envelope detection of a signal is some sort of demodulation of an Amplitude Modulated signal, the envelope $e(t)$ is centered at zero-frequency and has bandwidth b_f . The spectrum of the vibration, filtered, and envelope signals are depicted in the left panel of Fig. 8 for a healthy experiment. In the right panel, the envelope’s spectra of different health conditions are compared.

The time-domain representation of the envelope signal is depicted in Fig. 9 for segments with different health conditions.

Once the envelope signal $e(t)$ has been obtained, its samples are statistically analyzed and its kurtosis derived using the following expression,

$$k(f_c, b_f) = \frac{1}{N} \sum_{i=1}^N \left(\frac{e_i - \mu_e}{\sigma_e} \right)^4 \tag{2}$$

where e_i is the value of the i^{th} sample of the $e(t)$, μ_e and σ_e are the mean and standard deviation of the values of $e(t)$, and N is the total number of samples in the segment. The probability density functions of the samples of $e(t)$ for different health states are depicted in Fig. 10. They have been obtained using a kernel density estimation technique.

The value of the kurtosis is a function of the parameters of the filter, f_c and b_c . This relationship is drawn as a heat-map in Fig. 11 for a certain segment.

The kurtosis feature K is defined as the maximum value of the kurtosis $k(f_c, b_f)$, that is,

$$K = \max_{f_c, b_c} k(f_c, b_c) \tag{3}$$

For the segment represented in Fig. 11, the maximum value is $K = 174$, obtained for an optimum band-pass

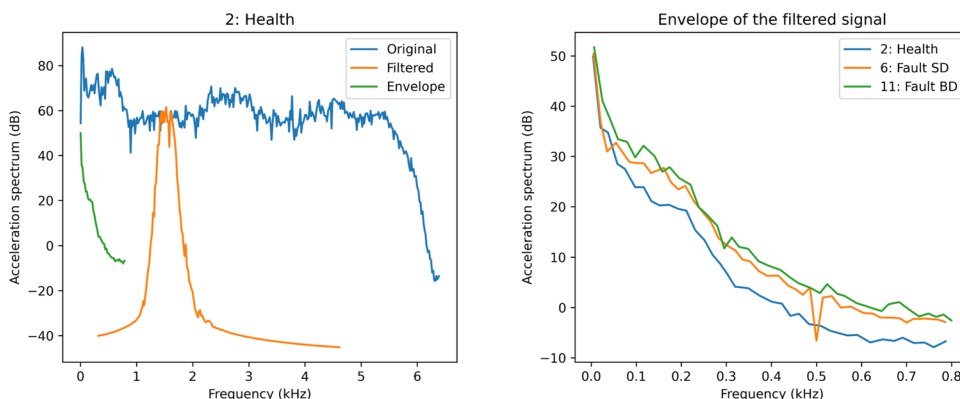


Fig. 8 Example of spectra for a vibration signal. Left: Original healthy signal (blue), and the result after filtering (orange), and envelope detection processes (green). Right: Spectra of the envelope signal corresponding to three different health states

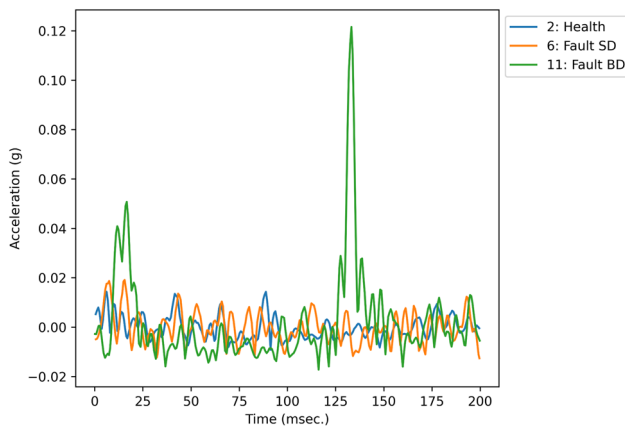


Fig. 9 Envelope signals for vibrations corresponding to different health conditions

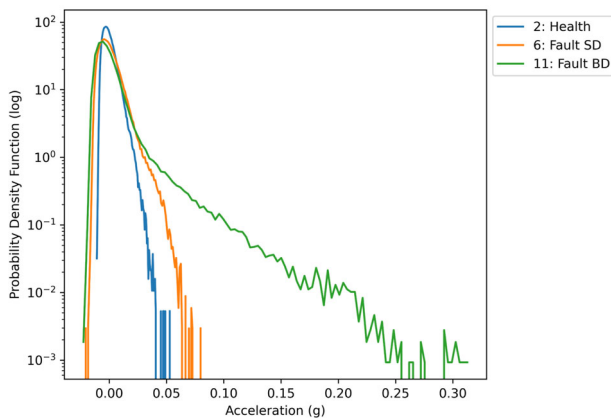


Fig. 10 Statistical distribution of the samples of the envelope signals, for vibrations corresponding to different health conditions

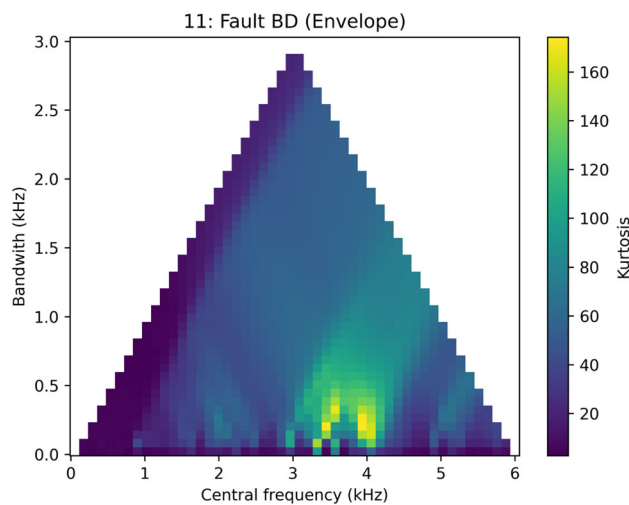


Fig. 11 Kurtosis of the envelope of a filtered vibration signal, for different values of the filter's parameters: central frequency f_c , and bandwidth b_f

filter centered at $f_c^* = 3936\text{Hz}$ with a semi-bandwidth $b_f^* = 202\text{Hz}$. When the optimum filter is applied to a segment, an optimum filtered signal and an envelope signal are obtained, denoted as $r^*(t)$ and $e^*(t)$ respectively.

From the previous process, a companion feature can also be obtained, the Root Mean Square (RMS) of the optimum envelope $e^*(t)$. It can be defined as:

$$R = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^{*2}} \quad (4)$$

Using this approach, the number of features characterizing each segment is just two (kurtosis and RMS), which is a very low dimensional vector, quite convenient for ML algorithms.

2.4 Generic features

The specialized features described in the previous section obtains a very low-dimensional description of the vibrations but requires a significant extraction process (computationally expensive) and, even more important, a deep knowledge of the underlying engineering process. In fact, the proposed features have been derived from the field of rolling bearings, but do not consider the unknown engineering details of other faults, such as those concerning the springs.

An alternative, intermediate between raw and specialized features, is to use a reduced set of generic features which are not tailored for any known engineering process. In the case of time signals, they are derived analyzing the vibrations both in the time-domain and in the frequency-domain. The set of generic features used in the paper are inspired in those used for some other authors [36–38].

The process of feature extraction assumes that the vibration signal $v(t)$ is sampled at a certain rate obtaining a sequence of N ordered values, where the i^{th} sample is denoted as v_i . Then, in the time-domain, up to 24 features are defined, as it is described in Table 2.

In this time-domain set of features, T_{12} resembles the specialized kurtosis defined in (2) and (3), and T_7 looks like a scaled version of the RMS defined in (4) but they are not the same, since generic features are obtained from the original vibration signal while the specialized attributes are derived from the envelope signal.

Applying the Fast Fourier Transform to a vibration signal, its power spectrum is obtained in the form of a sequence of M ordered values, where the j -th element denoted as V_j , corresponds to the power of the harmonic component at the frequency f_j . By the Nyquist-Shannon sampling theorem, the number of values in the power spectrum is one half of the number of samples, $M = N/2$. Then, in the frequency-domain, up to 9 features have been defined, as it is described in Table 3.

Table 2 Definition of time-domain generic features

T_1	Mean	$\frac{1}{N} \sum_{i=1}^N x_i$	T_{13}	Skewness Index	$\frac{T_{11}}{T_{10}^3}$
T_2	Median	$\frac{v_N + 1}{2}$	T_{14}	Kurtosis Index	$\frac{T_{12}}{(\sqrt{T_{10}})^4}$
T_3	Range	$\max_i v_i - \min_i v_i$	T_{15}	Zero Crossing	$\frac{1 - \text{sgn} v_i \cdot \text{sgn} v_{i+1}}{2}$
T_4	Maximum Absolute Value	$\max_i v_i $	T_{16}	Root Amplitude	$(\frac{1}{N} \sum_{i=1}^N \sqrt{ v_i })^2$
T_5	Minimum Absolute Value	$\min_i v_i $	T_{17}	Crest Factor	$\frac{T_4}{T_8}$
T_6	Absolute Range	$T_5 - T_4$	T_{18}	Shape Factor	$\frac{T_8}{\frac{1}{N} \sum_{i=1}^N v_i }$
T_7	Cumulative Square	$\sum_{i=1}^N v_i^2$	T_{19}	Pulse Indicator	$\frac{T_4}{T_1}$
T_8	Root Mean Square (RMS)	$\sqrt{\frac{1}{N} T_7}$	T_{20}	Entropy	$-\sum_{i=1}^{N-1} \text{prob}(v_i) \cdot \log_2 \text{prob}(v_i)$
T_9	Variance	$\sum_{i=1}^N (v_i - T_1)^3$	T_{21}	Clearance Factor	$\frac{T_8}{T_{16}}$
T_{10}	Standard Deviation	$\sqrt{T_9}$	T_{22}	Waveform Length	$\sum_{i=1}^{N-1} v_{i+1} - v_i $
T_{11}	Skewness	$\frac{1}{N-1} \sum_{i=1}^N (v_i - T_1)^3$	T_{23}	Willson Amplitude	$\sum_{i=1}^{N-1} [\lg(v_{i+1} - v_i)]^1$
T_{12}	Kurtosis	$\frac{1}{N-1} \sum_{i=1}^N (v_i - T_1)^4$	T_{24}	Slope Sign Change	$\sum_{i=2}^{N-1} \frac{1 - \text{sgn}(v_{i+1} - v_i) \cdot \text{sgn}(v_i + 2 - v_{i+1})}{2}$

¹Take into account:

$$g(x) = \begin{cases} 1 & x > \epsilon \\ 0 & x < \epsilon \end{cases}$$

Table 3 Definition of frequency-domain generic features

F_1	Cumulative Power	$\sum_{i=1}^M V_j$	F_6	Power Spectrum Deformation	$\frac{F_4}{F_2}$
F_2	Spectral Centroid	$\frac{\sum_{i=1}^M f_j V_j}{F_1}$	F_7	Spectral Skewness	$\frac{\sum_{j=1}^M (f_j - F_2)^3 V_j}{F_1 F_5^3}$
F_3	Median Frequency	$f_a, \sum_{i=1}^a = \sum_{i=a+1}^M V_j = \frac{\sum_{i=1}^a V_j}{2}$	F_8	Spectral Kurtosis	$\frac{\sum_{j=1}^M (f_j - F_2)^4 V_j}{F_1 F_5^4}$
F_4	Frequency RMS	$\sqrt{\frac{\sum_{j=1}^M f_j^2 V_j}{F_1}}$	F_9	Spectral Crest	$\frac{\max_j V_j}{\frac{1}{M} F_1}$
F_5	Spectral Spread	$\sqrt{\frac{\sum_{j=1}^M (f_j - F_2)^2 V_j}{F_1}}$			

Using this approach, the number of generic features characterizing each segment is 33. This means that the dimensionality of the vector describing each vibration segment, while not low, is certainly quite moderate.

2.5 Design matrices

The segments of vibrations obtained from the experiments described in Table 1, can be characterized using three different strategies: raw, specialized, and generic features. So, the design matrices corresponding to each approach has the same number of rows (497, one for each 5-seconds length segment), but different number of columns (features).

To tune the ML model, the rows of the design matrix are split into 3 groups using a hold-out technique: training (60%, 298 instances), to obtain the parameters of the ML model; validation (20%, 99 instances), to select the hyperparameters of the model; and testing (20%, 100 instances), to generalize the results of the model to data not used in its construction. The number of rows in each design matrix corresponds to the count of observations. The number of columns, however, depends on the feature set used in each scenario: 64,000 columns for raw features (5-second recordings sampled at 12.8 kHz), 2 columns for specialized features (kurtosis and RMS), and 33 columns for generic features (as outlined in Tables 2 and 3). The structure of the corresponding design matrices is summarized in Table 4.

For a given vibration segment, the values of its corresponding features can significantly differ from each other, leading to a convergence problem in many ML algorithms. To avoid it, the design matrix is usually normalized by column before its use for training. In this research a z-score normalization has been employed. For a design matrix X of dimension $(n \times d)$, the value of the i -th row and j -th column

is normalized using the expression:

$$Z_{i,j} = \frac{X_{i,j} - \mu_{xj}}{\sigma_{xj}} \quad (5)$$

where μ_{xj} and σ_{xj} are the mean and standard deviation of the values in the j -th column, that is,

$$\mu_{xj} = \frac{1}{n} \sum_{i=1}^n X_{i,j}, \sigma_{xj} = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_{i,j} - \mu_{xj})^2} \quad (6)$$

To extend the research to more demanding scenarios, some synthetic design matrices are derived from the original ones by adding some random noise. The value of the i -th row and j -th column of the noisy synthetic design matrix is obtained using the expression, $W(i, j) = Z(i, j) + G(i, j)$, where G is a random gaussian matrix of dimension $(n \times d)$, null mean, and standard deviation σ_N , also known as noise level.

2.6 Classification algorithm

The ML algorithm requires, not only a design matrix, but also a target (column) vector, y , of dimension $(n \times 1)$, where the value at the i -th row corresponds to the health state of the i -th

Table 4 Structure of the design matrices

			Columns (# of features)		
			Raw	Specialized	Generic
			64000	2	33
Rows (# of segments)	Training	298			
	Validation	99			
	Testing	100			
	Total	497			

vibration segment. As there are only three different values of the health state (see Table 1) and they are nominal (no order between them can be established), the ML algorithm must face a classification problem.

Many classification algorithms have been proposed. As the purpose of this research is to compare different ways of describing vibrations and not classification algorithms, a particular classifier has been selected: the random forest classifier [39], which can work in non-linear problems, has a very limited number of hyperparameters, outperforms many other algorithms when datasets have a low number of instances, and its predictions can be very well generalized as it contains an ensemble of underlying decision trees.

The computing tasks in the research have been coded in Python, while the ML algorithms and, in particular, the random forest classifier, have been programmed using the scikit-learn library [40]. The random forest classifier uses the default options in that library but two of them are used to tune the model. These two hyperparameters are: the number of decision trees in the forest; and the maximum depth of every tree. This later one plays the role of regularization hyperparameter, avoiding overfitting to training data.

The performance of a certain classification algorithm can be revealed by the absolute confusion matrix, where the element A_{kq} of this matrix represents the number of instances belonging to the k -th class, that has been classified as belonging to the q -th class. It is also common to represent the algorithm's performance in terms of classification ratios by using the relative confusion matrix, where the element CM_{kq} is obtained as

$$CM_{kq} = \frac{A_{kq}}{n_k} \quad (7)$$

being n_k the number of instances belonging to the k -th class.

To summarize the confusion matrix in a single value, several classification metrics have been proposed. Probably the most used one is the classification's accuracy ACC defined as

$$ACC = \frac{1}{n} \sum_{k=1}^C n_k CM_{kk} \quad (8)$$

where C is the total number of classes. This metric is easy to understand, has a good behavior under slightly imbalanced datasets [41] and it is not affected by a change in the order of the classes [42].

3 Results

3.1 Classification using raw features

As it was described in Section 2.2 and summarized in Table 4, each vibration segment can be featured using 64000 raw values. This high-dimensional vectors can be graphically represented by applying a Principal Component Analysis (PCA) [43]. This technique rotates the vectors using a linear transformation that maximize statistical variance in the first components of the new vectors. Then the transformed vector is projected onto the plane of the two first components, obtaining a 2D representation. The result, indicating with colors the different health states, is shown in Fig. 12. From this figure, we should expect a quite good separation of classes but with some overlaps leading to a few misclassifications.

To select the two hyperparameters of the random forest classifier several combinations of the number of trees and their maximum depth have been explored. The classification accuracy on the validation dataset for these combinations is depicted in Fig. 13. An ensemble of 75 trees with a maximum depth of 30 steps between the root node and the leaf node have been selected as the hyperparameters of the classifier.

After training the model with these hyperparameters, the classification results obtained on the testing dataset are shown as a confusion matrix shown in Fig. 14, with an overall accuracy of 83%.

3.2 Classification using specialized features

In the case of using the two specialized features described in Section 2.3, , kurtosis and RMS, the graphic representation of the segments in the dataset is straightforward, as it is shown in Fig. 15. As the range of values for kurtosis is very large,

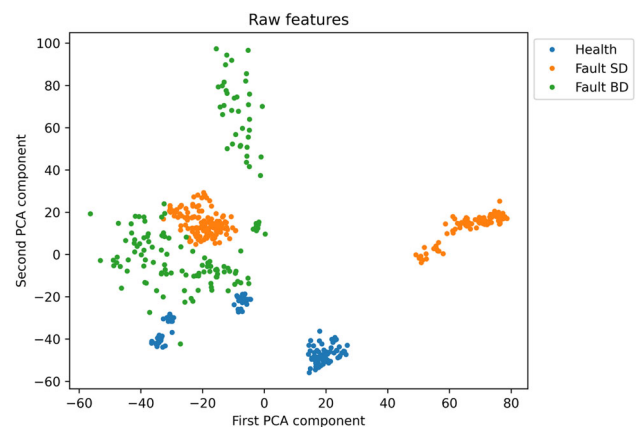


Fig. 12 Bidimensional representation of the vibration dataset using raw features and a PCA dimensionality reduction technique

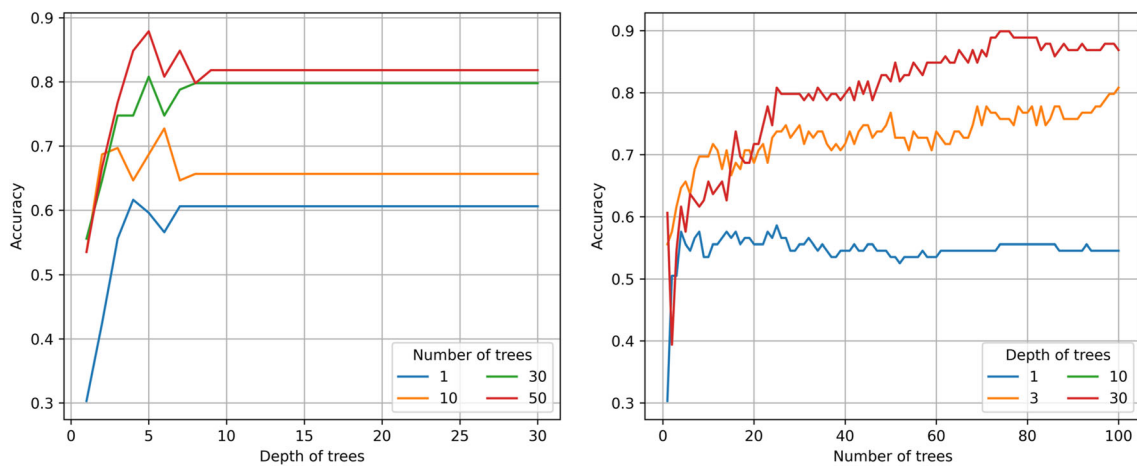


Fig. 13 Classification accuracy as a function of the values of the hyperparameters: maximum depth of trees (left) and number of trees (right)

they have been represented in logarithmic scale. From this figure, a very good identification of healthy vibrations should be expected. In fact, using only the kurtosis feature, most of the healthy segments are correctly separated. However, the two types of damages (on bearing and spring) are very overlapped, suggesting a difficult separation of these two faulty classes.

A clearer view of the overlapping of the feature values can be seen using the probability density function (PDF) of these values as they are depicted in Fig. 16. By using just kurtosis, healthy segments appear very well separated from faulty ones, but the pdfs of the faulty conditions are significantly overlapped. On the other hand, RMS values could help to separate the faulty segments, but a limited success should be expected as their PDFs are still considerably overlapped.

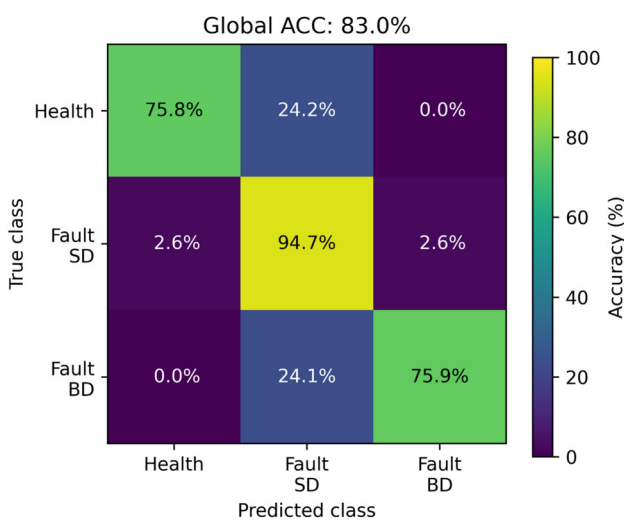


Fig. 14 Confusion matrix for the random forest classifier with segments defined by raw features

A random forest classifier has been selected and trained using these specialized features. The best results on the validation datasets have been obtained for the same values of the hyperparameters indicated in the previous Section. And the classification results are drawn as a confusion matrix in Fig. 17, with an overall accuracy of 88%. It can be seen that healthy segments are successfully identified in a 100% of the cases, but identifying the type of fault is a much harder task for this classifier.

3.3 Classification using generic features

Finally, if the vibration segments are defined using the 33 generic features defined in Section 2.4, the dataset can be graphically represented using PCA as it is depicted in Fig. 18. A clearer separation of classes can be observed, which suggest that classification algorithms should obtain very good results.

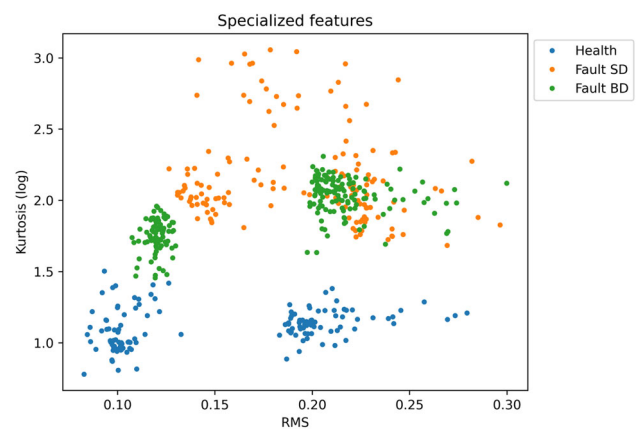


Fig. 15 Bidimensional representation of the vibration dataset using specialized features. Kurtosis values are in logarithmic scale

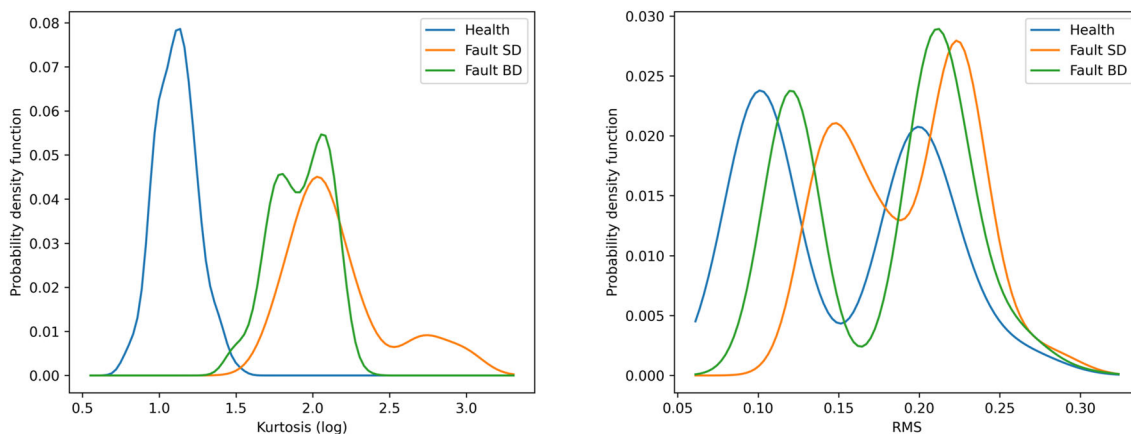


Fig. 16 Probability density functions corresponding to the specialized features: kurtosis (left) and RMS (right)

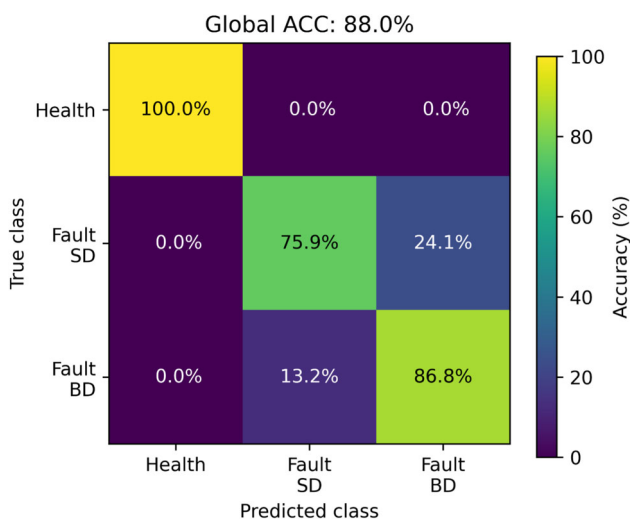


Fig. 17 Confusion matrix for the random forest classifier with segments defined by specialized features

The classification results obtained by the random forest classifier using the generic features are represented in Fig. 19. A 100% success rate have been obtained clearly overperforming the other two ways of featuring vibration signals.

4 Discussion

The results described through Section 3 show that, for the analyzed dataset, the description of vibrations by generic features clearly outperforms the other two alternatives: raw or specialized features.

In this section the impact on performance of four issues will be discussed: synthetically generated noisy features, number of training instances, length of vibration segments and computational effort of each method.

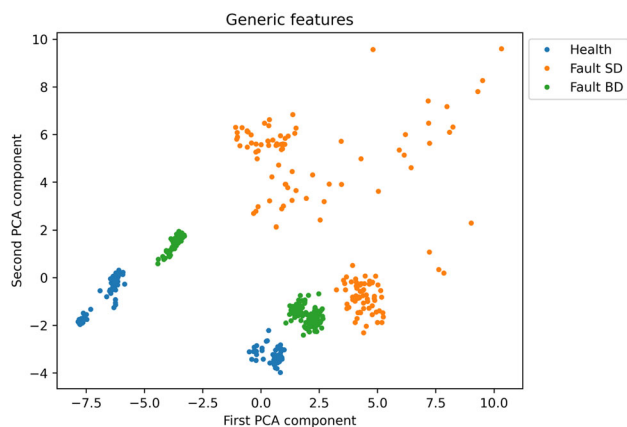


Fig. 18 Bidimensional representation of the vibration dataset using generic features

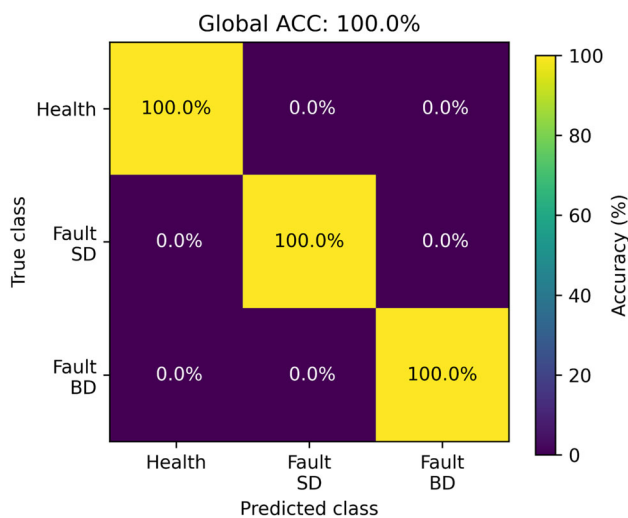


Fig. 19 Confusion matrix for the random forest classifier with segments defined by generic features

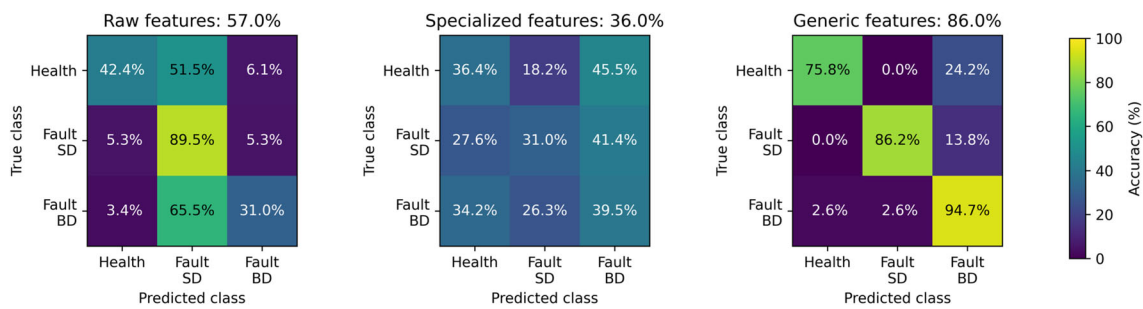


Fig. 20 Confusion matrices for the random forest classifier with segments defined by three types of noisy features (noise level $\sigma_N = 1$): raw (left), specialized (center) and generic (right)

4.1 Noisy features

To extend the study's conclusions beyond the original datasets, a synthetic modification of the design matrices, as described in Section 2.5, was performed. First, each feature value was normalized to achieve a mean of zero and a standard deviation of $\sigma = 1$. Gaussian noise with a mean of zero and a standard deviation of σ_N was then added. For σ_N , classification performance significantly decreased: overall accuracy dropped by 26 percentage points (from 83% to 57%) for raw features, 52 percentage points (from 88% to 36%) for specialized features, and 14 percentage points (from 100% to 86%) for generic features.

Confusion matrices for these three types of features are shown in Fig. 20. It must be noted that a classifier making a random guess of one out of the three possible classes should obtain on average an accuracy of 33.3%, a close value to the result obtained when specialized features are used.

It can be seen that the use of generic features not only obtains the best accuracy results in the original datasets, but they are also the most robust features under noise per-

turbations. These results can be extended to different noise levels showing that accuracy of models using generic features worsen more slowly than those based on raw or specialized features, as it is depicted in Fig. 21.

To evaluate the noise robustness of each feature set, we define the Half-Accuracy Noise Threshold (HANT, $\sigma_{N_{50}}$) as the noise level that reduces the accuracy by half. For each line in Fig. 21, which depicts the function $acc = (\sigma_N)$ for a given feature set, it holds that $acc = (\sigma_{N_{50}}) = acc(0)/2$. The values obtained for each feature set are shown in Table 5. The results indicate that generic features demonstrate greater resilience to noise compared to specialized features, with raw features occupying an intermediate position between the two.

4.2 Alternative techniques

As part of the research, various alternative approaches have been explored, including strategies to address dataset imbalance, the evaluation of different sets of generic features, and an analysis of the impact of using various classifiers.

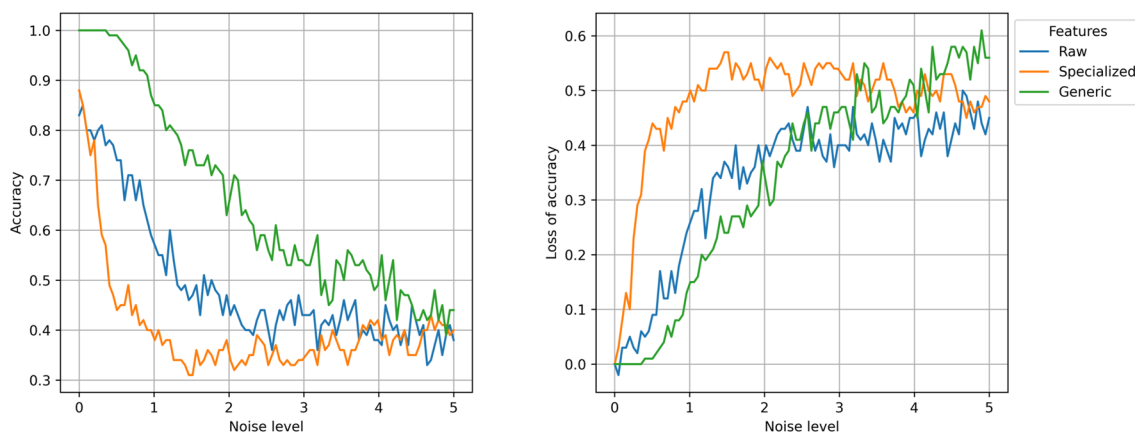


Fig. 21 Classification performance as a function of the noise level added to the original dataset: accuracy (left); loss of accuracy (right)

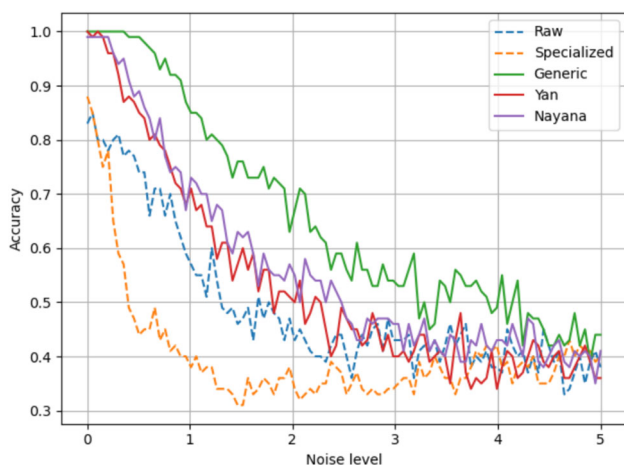
Table 5 Noise robustness for each feature set

Feature set	Half-Accuracy Noise Threshold ($\sigma_{N_{50}}$)
Raw	2.12
Specialized	0.45
Generic	3.18

First, to address the slight class imbalance in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) [44] was initially applied to increase the instance count of each class to match that of the most populated class. However, as the imbalance was small and SMOTE yielded no significant improvement, the study proceeded without synthetic instances.

Second, in addition to the generic features listed in Tables 2 and 3, two additional feature sets were analyzed: one proposed by Nayana [45], based solely on time-domain characteristics, and another introduced by Yan [46], which incorporates statistical metrics from both time and frequency domains. The results, presented in Fig. 22, show that all generic feature sets outperform both specialized and raw features, with the feature set proposed in this research achieving the best performance among all.

Finally, additional classifiers were tested, including logistic regression, support vector machines (SVM), and various configurations of feedforward neural networks. While these models showed similar accuracy for both specialized and generic features, the high dimensionality of raw features presents a significant challenge for classification. As a result, performance varies noticeably, depending heavily on the classifier type and its hyperparameters. Among these algorithms, logistic regression had the lowest computational cost, followed by SVM and random forest with moderate requirements, while the neural network configurations were the most computationally demanding.

**Fig. 22** Accuracy vs. noise level added for other generic feature sets

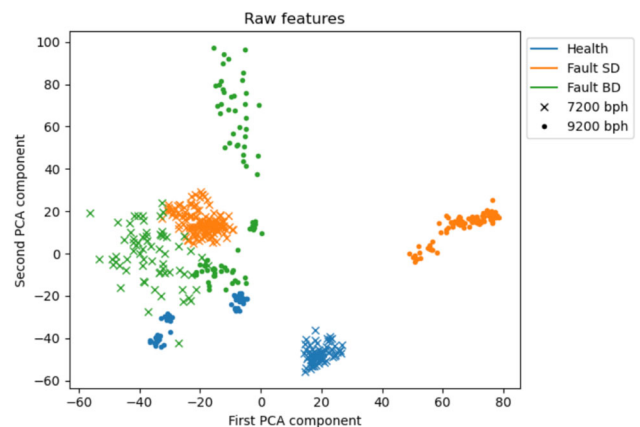
4.3 Bottling speed

All acquired vibration signals include bottling speed as associated information. However, this information was intentionally excluded as a feature in the design matrices for the three classification approaches (using raw, specialized, and generic features). Instead, only features derived from the vibration signal itself were used. This choice was based on the observation that bottling speed information is inherently embedded in the other features. For example, a PCA representation of the dataset using raw features, as shown in Fig. 23, reveals that instances with different bottling speeds occupy distinct regions, indicating that just two principal components are sufficient to differentiate the corresponding bottling speeds.

Classification accuracy under varying noise levels was also evaluated for the three feature types, both with and without including bottling speed as a feature. The results, shown in Fig. 24, indicate that incorporating bottling speed in the design matrices has a negligible effect on accuracy. This aligns with expectations, as bottling speed is inherently embedded in the other features, rendering it non-essential for accurately identifying the health state.

4.4 Learning curves

As it was described in Section 2, the vibration signal obtained in each experiment is split in many 5-second segments to increase the number of examples available to train the ML models. To check if the resulting number of segments are enough, the corresponding learning curves are depicted in Fig. 25, where each plot represents the evolution of the classification accuracy as the number of training examples is increased.

**Fig. 23** Learning curves. The accuracy of the random forest classifier using different set of features is plotted as a function of the number of examples used to train the model

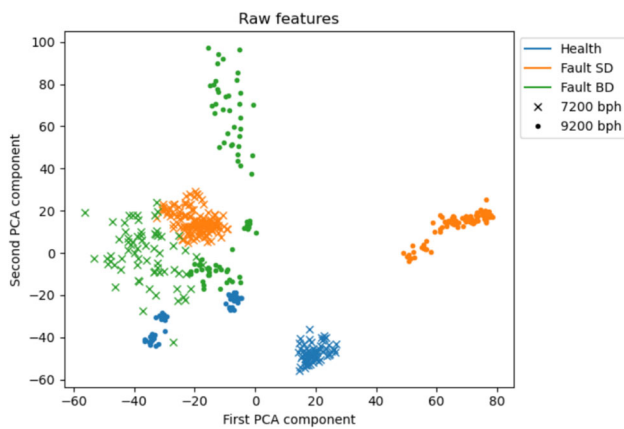


Fig. 24 Classification accuracy as a function of noise level added to the original dataset, evaluated across different feature types, with and without bottleneck speed as an additional feature

It can be concluded that the random forest classifier using generic features on a dataset with no added noise requires just 50 examples to achieve the maximum accuracy, while the use of the specialized features requires about 175, and the use of raw features do not get the maximum accuracy for the available training samples (the blue line does not stabilize for the range of available training instances). In the case of adding a noise level $\sigma_N = 1$, the classifier using generic features requires 125 instances, that based on specialized features needs about 200, while the use of raw features appears to require more instances that those available in the training dataset. By performing more experiments and/or increasing their duration, the classifiers based on raw features would probably increase its accuracy even over the specialized features.

4.5 Length of segments

The number of examples to train the models could be easily increased by reducing the length of each segment. The intuition says that a very short segment may not be represen-

tative enough of its health condition and, therefore, classifiers trained on that segments will show lower performance.

On the other hand, choosing very long segments yields to a very scarce training dataset which certainly will affect the classifier performance as it was shown in the previous Section. Additionally, as the testing dataset will also be very scarce, the value of the accuracy will have a large variance.

Then, a balance must be sought about the length of the segments: nor too short under-representative signals, nor too long but scarce instances. To explore this balance, the classification accuracy has been computed using different length of segments. The results are depicted in Fig. 26 for the three types of features using the original dataset (no noise), and a perturbed datasets with a noise level of $\sigma_N = 1$. It can be seen that short segments (leftmost part of the plots) show comparatively low accuracy, while long segments (rightmost part of the plots) also have low accuracy with values very spread (large variance due to the scarcity of testing instances). A trade-off can be reached for segments of 5-seconds length, as it is indicated in the figure, justifying the election made in Section 2.4.

Splitting signals to increase dataset size is a scalable methodology suitable for handling very long signals. The optimal segment length, however, is application-specific and should be tailored to each case. For example, in [47], the authors work with one-year-long signals, which they segment into one-day intervals. From a computational standpoint, splitting is highly efficient, as feature extraction becomes less costly with shorter segments. Overall, the computational impact of splitting is minimal.

4.6 Explaining the Overperformance of Generic Features

As demonstrated in the results section, the generic features proposed in this research consistently outperform the specialized features, which are primarily based on kurtosis. This outcome remains consistent even when alternative sets of

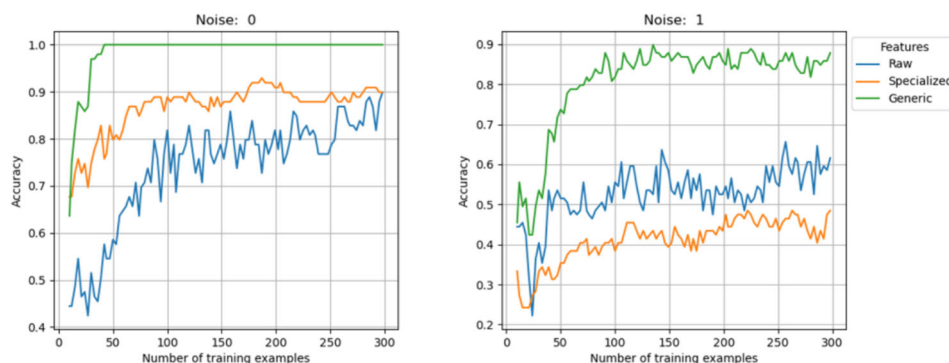


Fig. 25 Learning curves. The accuracy of the random forest classifier using different set of features is plotted as a function of the number of examples used to train the model (for two noise levels)

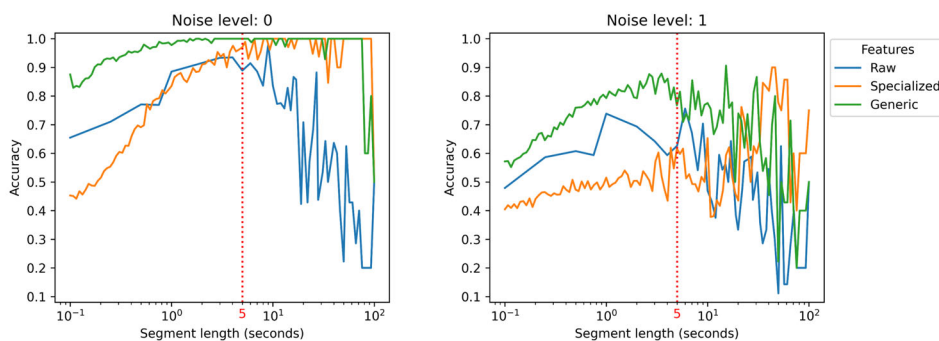


Fig. 26 Classification accuracy for different segment lengths. The 5-second length selected for this research is depicted as a red dotted line

generic features are employed, as shown in Section 4.2. Here, we will delve into the possible reasons behind these findings.

To begin, let’s examine the random forest classifier’s accuracy in identifying the health state condition when only a single feature—either specialized or generic—is used. The results, shown in Fig. 27, reveal that the carefully crafted kurtosis-based specialized feature achieves accuracy well above the mean. However, it performs at a level similar to several generic features, such as T_9 , T_{12} , and F_2 . This suggests that, for our application, the lengthy design and time-consuming extraction process of the kurtosis-based feature does not yield a significant advantage over certain off-the-shelf generic features.

Next, consider the correlation of the kurtosis-based specialized feature with other features, as measured by the absolute value of Pearson’s correlation coefficient. In Fig. 28, the blue line represents the kurtosis-based feature’s correlation with other features, while the orange line represents the root mean square (RMS)—a companion specialized feature. RMS offers a distinct perspective of the vibration signal, demonstrated by its low correlation with kurtosis, which can enhance feature diversity. Interestingly, other generic features also exhibit low correlation with kurtosis while achieving better performance in single-feature classification

(e.g., T_9 , F_2 , and F_3), making them potentially more valuable complements to the kurtosis-based specialized feature.

Finally, we explore the cumulative accuracy achieved by classifiers as more features are added, using the Forward Feature Selection (FFS) algorithm [48]. In the first step, T_{12} is chosen, as it yields the highest accuracy in Fig. 27. Adding F_4 further improves performance. This process is iteratively repeated until all features are considered, as shown in Fig. 29 (blue line). For comparison, we also conducted the same process starting with the kurtosis-based specialized feature, followed by the remaining features in sequence; this result is depicted by the orange line. In both cases, accuracy stabilizes around 15 features, suggesting that the superior results of the generic features are not attributable to any single optimal feature. Instead, they reflect the cumulative contribution of various features, each providing complementary insights into the vibration signals.

A hunting analogy illustrates this concept: to catch elusive prey, a shotgun firing multiple pellets—each with slightly diverging trajectories—is often more effective than a single, high-precision bullet. This analogy underscores the advantage of using a broad set of generic features over a limited set of specialized ones, a principle that appears to be domain-independent. Similar benefits have been observed in

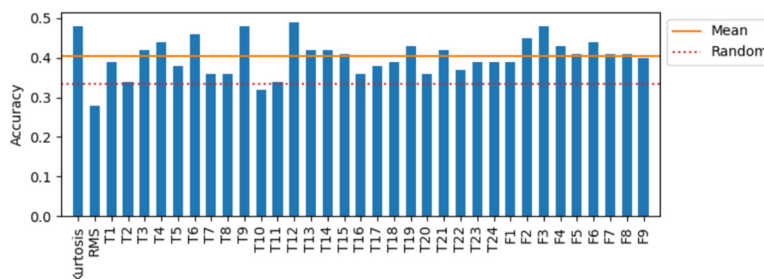


Fig. 27 Classifier accuracy using single features

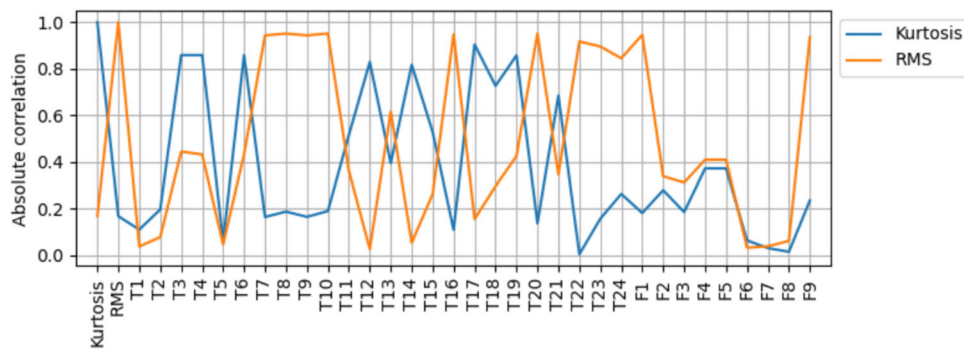


Fig. 28 Absolute Pearson's correlation of kurtosis (blue) and RMS (orange) with other features

industrial applications like battery health monitoring [47] and various other classification contexts [49, 50].

4.7 Computational cost

In previous Sections, the three types of features have been analyzed from the perspective of their performance, as they are measured by the classification accuracy. However, the computing effort required by each method should also be considered. This effort can be broken down into three categories: extraction, training, and testing.

The extraction time considers the processes required to convert a 5 second vibration signal to a vector of raw, specialized, or generic features. In the case of raw features, no

process is required, as the 64,000 values of the vibration (sampling rate of 12.8 kHz) already are the raw features.

In the other hand, to extract the specialized features it is required to filter the signal, to get its envelope and then compute its kurtosis and RMS values (see Section 2.3). This process must be repeated for a grid of parameters of the filter, combining pairs of values for central frequency f_c and bandwidth b_f . The specialized features have been extracted using a grid of 50×50 pairs of values, which is a very time-consuming process.

Finally, the extraction of generic features implies making some calculations on the raw values to obtain the time-domain features, computing the spectrum of the signal using a Fast Fourier Transform (FFT), and eventually deriving the

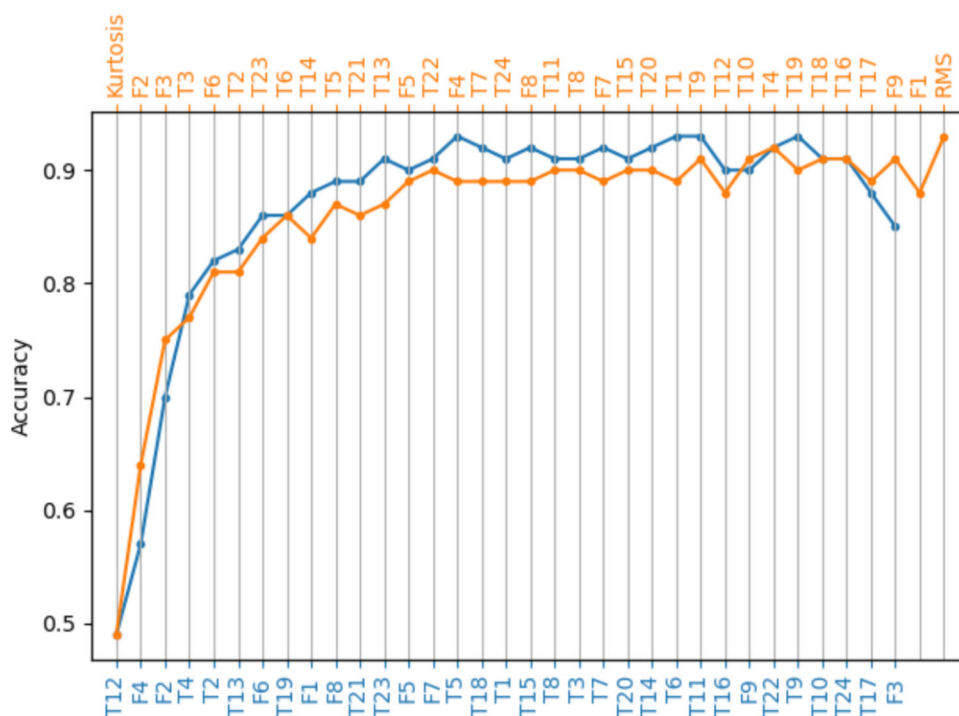


Fig. 29 Classifier accuracy with incremental feature selection using Forward Feature Selection: generic features (blue) vs. kurtosis-based plus remaining features (orange)

frequency-domain features from the spectrum values (see Section 2.4).

Once the design matrix has been constructed, its training portion is used to obtain the parameters defining the classifier. And finally, the time required to classify the instances in the testing dataset is analyzed. Usually, longer feature vectors drive to more computing costs on training and testing. Extraction, training, and testing costs have been measured using a Python 3 script running in a desktop computer powered by an Intel®i7-11700 processor at 2.10 GHz, under a Windows-11 operating system. The resulting computing times are shown (in logarithmic scale) in Fig. 30

Using raw features means no extraction effort but the corresponding longer feature vectors require significant training and testing times. Alternatively, specialized features demand very heavy computing for extraction (highly dependent on the grid size used to determine the optimum filter), but they have the shortest training and testing times due to the low dimensionality of the feature vectors. Finally, generic features are relatively easy to extract while its use for training and testing demands slightly higher computing resources than the specialized features.

These results demonstrate that both raw and generic features are suitable for real-time environments, as the extraction and prediction (testing) times are well below the segment duration (real-time training is not required). However, the extraction time for specialized features on a high-end laptop approaches the segment duration, presenting a significant challenge for real-time implementation on low-cost edge devices.

These values can be redrawn to replicate with actual data, the sketch depicted in Fig. 2. For this purpose, the classification effort will be assimilated to the sum of training and testing computing times. Results are shown in Fig. 31.a where the specific values are the center of the ellipses corresponding

to each type of feature. Additionally, the computational effort (the sum of extraction, training, and testing times) associated to each features type can be compared to the classification accuracy obtained for that type, as it is depicted in Fig. 31.b. Raw and specialized features have a similar classification performance at a comparable computational costs, while generic features clearly overperforms both of them requiring a much lower effort.

4.8 Implementation and scalability considerations

The integration of AI-based health monitoring with existing maintenance workflows in bottling plants presents opportunities to transition from preventive to predictive maintenance strategies. Unlike traditional preventive maintenance, which operates on scheduled intervals, predictive maintenance leverages real-time anomaly detection to address emerging issues proactively. By identifying subtle signs of wear or malfunction in gripping pliers, the proposed AI system allows maintenance teams to plan interventions more strategically, aligning with production goals of minimizing unscheduled downtime and enhancing equipment reliability.

However, deploying this solution at scale involves several practical challenges. One of the primary considerations is the installation and positioning of sensors on all relevant equipment, which could be logistically demanding and may necessitate wireless solutions to reduce operational disruptions. Furthermore, the high data throughput required for real-time monitoring could strain existing network infrastructure, especially if additional critical assets are to be monitored. Approaches such as edge computing or selective monitoring of high-priority assets may mitigate data handling challenges, facilitating smoother implementation at scale.

Compatibility with existing IT infrastructure is another critical aspect, as data integration between new AI systems

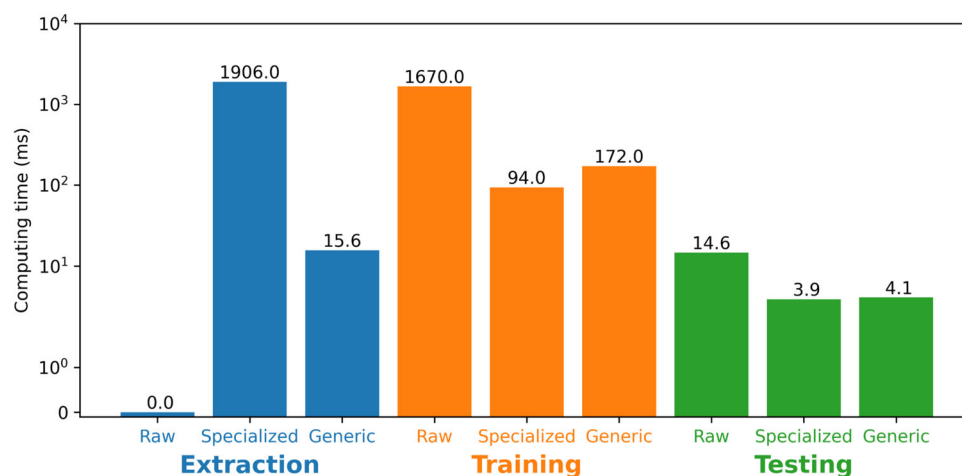


Fig. 30 Computing times for different processes using raw, specialized, and generic features: feature extraction (left), model training (center), and classifier testing (right)

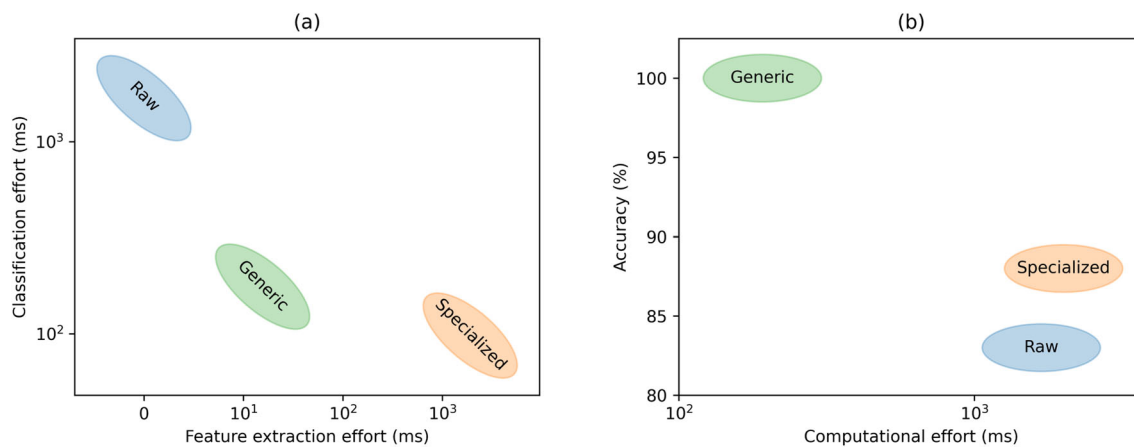


Fig. 31 Summary of computational effort: left (a), trade-off between classification and feature extraction efforts for the actual data of this research; right (b), classification accuracy and computational effort

and legacy maintenance frameworks can be technically complex. To address this, middleware solutions or APIs designed for seamless data interoperability can enhance integration, ensuring that the AI-driven system communicates effectively with established control systems. Additionally, data security is paramount, as real-time data processing and AI-driven maintenance introduce cybersecurity vulnerabilities. Industry-standard security practices, such as data encryption and role-based access control, are recommended to safeguard sensitive operational information and maintain system integrity.

From an operational standpoint, workforce considerations are essential for successful adoption. Implementing an intuitive user interface and providing training for interpreting AI-driven insights could improve usability for maintenance technicians unfamiliar with AI systems. Moreover, establishing a feedback loop with plant operators can further refine the model, allowing the system to adapt to the specific conditions and operational rhythms of individual plants.

The long-term implications of deploying this AI health monitoring system extend to both economic and operational benefits. By enhancing equipment uptime and reducing reactive maintenance, the system has the potential to yield a significant return on investment (ROI) over time. Future research should focus on validating the system's effectiveness across various agri-food production environments and exploring its application to other critical equipment. Studying the system's performance under diverse operational conditions would provide a more comprehensive understanding of its broader industrial applicability and impact.

5 Conclusions

The main finding of this research is that, for the analyzed industrial application, the use of a set of generic features

clearly overperforms the other two alternatives: either raw unprocessed features or highly treated over-specialized knowledge-rooted attributes. Its classification accuracy is significantly higher while its computational cost is much lower.

Although specialized features have achieved excellent results in detecting fault conditions, they got mediocre results to identify the type of fault. This problem has been overcome using generic features that have shown a remarkable performance in discriminating the causes of faulty conditions.

Generic features have also shown a more robust behavior than raw or specialized features in synthetic datasets where the original design matrices were corrupted by an increasing level of random noise.

On the other hand, the size of the dataset required for classifiers using generic features is remarkably smaller than those needed for specialized or, even more, for raw features. The size of the dataset available for this research (about 40 minutes of vibration signals) has revealed to be enough for the two first cases, but insufficient for the raw features case.

The research has also revealed that, splitting the original vibration signals in smaller segments, increases the number of training examples, which can help to achieve a better classification accuracy. An optimal segment length has been deduced, as too short segments do not properly represent their health states and too large ones yield few training instances.

To enhance the applicability of our method, future studies could explore extending this approach to other types of machinery commonly used in the agri-food industry, such as conveyors, mixers, or pumps. These machines also generate vibration signals, and a similar classification framework could potentially improve maintenance practices across diverse equipment types.

To further improve dataset size, a logical next step would be to conduct data collection campaigns across multiple facilities and under varied operational conditions. This would

enable a more comprehensive dataset that captures a wider range of potential equipment states and wear patterns. Additionally, integrating advanced signal processing techniques, such as wavelet transforms or cepstral analysis, could help extend generic feature extraction to other domains, improving the method's robustness and accuracy.

These steps could contribute to a more generalized, scalable AI-driven solution for equipment health monitoring in the agri-food sector.

Acknowledgements This work has been supported by the GOYA - Antonio Unanue in Engineering in the Agrifood Industry Chair of the University of Seville (Spain) and the VII Own Research and Transfer Plan 2023 of the University of Seville (Projects 2023/00000378 and 2023/00000390).

Author Contributions All authors contributed to the conception and design of the study. Funding acquisition was supported by Amalia Luque and experimental equipment, material preparation and data collection were done by Mirko Mazzoleni, Antonio Ferramosca and Fabio Previdi. Analysis and writing of the first manuscript draft were performed by Daniel Campos and Amalia Luque, and it was reviewed by Alejandro Carrasco. All authors provided comments on previous versions of the manuscript. All authors have read and approved the final manuscript.

Funding Funding for open access publishing: Universidad de Sevilla /CBUA.

Data Availability Restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available.

Declarations

Competing Interests All authors disclosed no relevant interests.

Ethical and Informed Consent for Data Used Written informed consent for publication of this paper was obtained.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Campos D, Carrasco A, Mazzoleni M, Ferramosca A, Luque A (2024) Screening of machine learning techniques on predictive maintenance: a scoping review. *DYNA* 99:159–165. <https://doi.org/10.6036/10950>
- Silva RF, Souza GFM (2022) Modeling a maintenance management framework for asset management based on iso 55000 series guidelines. *J Qual Maint Eng* 28(4):915–937. <https://doi.org/10.1108/JQME-08-2020-0082/FULL/HTML>
- Arunraj NS, Maiti J (2007) Risk-based maintenance—techniques and applications. *J Hazard Mater* 142(3):653–661. <https://doi.org/10.1016/j.jhazmat.2006.06.069>
- Custeau K (2020) Asset performance management 4.0 and beyond with risk-based maintenance. *Technews Industry Guide: Maintenance, Reliability & Asset Optimisation*
- Errandonea I, Alvarado U, Beltrán S, Arrizabalaga S (2022) A maturity model proposal for industrial maintenance and its application to the railway sector. *Appl Sci* 12(16):8229. <https://doi.org/10.3390/app12168229>
- Sarazin A, Bascans J, Sciau J-B, Song J, Supiot B, Montarnal A, Lorca X, Truptil S (2021) Expert system dedicated to condition-based maintenance based on a knowledge graph approach: Application to an aeronautic system. *Expert Syst Appl* 186:115767
- Jieyang P, Kimmig A, Dongkun W, Niu Z, Zhi F, Jiahai W, Liu X, Ovtcharova J (2022) A systematic review of data-driven approaches to fault diagnosis and early warning. *J Intell Manuf.* <https://doi.org/10.1007/S10845-022-02020-0>
- Alamo T, Luque A, Ramirez DR, Tempo R (2012) Randomized control design through probabilistic validation. In: *Proceedings of the American control conference*, pp 839–844. <https://doi.org/10.1109/ACC.2012.6314838>
- Randomized methods for design of uncertain systems (2015) Alamo T, Tempo R, Luque A, Ramirez D.R. Sample complexity and sequential algorithms. *Automatica* 52:160–172. <https://doi.org/10.1016/j.automatica.2014.11.004>
- Zhang W, Yang D, Wang H (2019) Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Syst J* 13(3):2213–2227
- Zhou H, Wang S, Miao Z, He C, Liu S (2019) Review of the application of deep learning in fault diagnosis. In: *2019 Chinese control conference (CCC)*, IEEE, pp 4951–4955
- Subramanian J, Simon R (2013) Overfitting in prediction models—is it a problem only in high dimensions? *Contemp Clin Trials* 36(2):636–641
- Turan EM, Jäschke J (2021) Classification of undesirable events in oil well operation. In: *2021 23rd International conference on process control (PC)*, IEEE, pp 157–162
- Kim H-G (2004) Moreau N, Sikora T: Audio classification based on mpeg-7 spectral basis representations. *IEEE Trans Circuits Syst Video Technol* 14(5):716–725. <https://doi.org/10.1109/TCSVT.2004.826766>
- Luque A, Gómez-Bellido J, Carrasco A, Barbancho J (2018) Optimal representation of anuran call spectrum in environmental monitoring systems using wireless sensor networks. *Sensors* 18(6):1803. <https://doi.org/10.3390/s18061803>
- Kucevic D, Tepe B, Englberger S, Parlikar A, Mühlbauer M, Bohlen O, Jossen A, Hesse H (2020) Standard battery energy storage system profiles: Analysis of various applications for stationary energy storage systems using a holistic simulation framework. *J Energy Storage* 28:101077. <https://doi.org/10.1016/j.est.2019.101077>
- Randall RB, Antoni J (2011) Rolling element bearing diagnostics—a tutorial. *Mech Syst Signal Process* 25(2):485–520. <https://doi.org/10.1016/J.YMSSP.2010.07.017>
- de Souza EF, Bittencourt TN, Ribeiro D, Carvalho H (2022) Feasibility of applying mel-frequency cepstral coefficients in a drive-by damage detection methodology for high-speed railway bridges. *Sustainability* 14(20):13290. <https://doi.org/10.3390/su142013290>
- Alamelu Manghai TM, Jegadeeshwaran R (2019) Vibration based brake health monitoring using wavelet features: A machine learning approach. *JVC/Journal of Vibration and Control*

- 25(18):2534–2550. https://doi.org/10.1177/1077546319859704/ASSET/IMAGES/LARGE/10.1177_1077546319859704-FIG17.JPEG
20. Das I, Arif MT, Oo AMT, Subhani M (2021) An improved hilbert-huang transform for vibration-based damage detection of utility timber poles. *Appl Sci* 11(7):2974. <https://doi.org/10.3390/APP11072974>
 21. Barbosh M, Singh P, Sadhu A (2020) Empirical mode decomposition and its variants: a review with applications in structural health monitoring. *Smart Mater Struct* 29(9):093001. <https://doi.org/10.1088/1361-665X/ABA539>
 22. Gupta M, Gao J, Aggarwal CC, Han J (2014) Outlier detection for temporal data: A survey. *IEEE Trans Knowl Data Eng* 26(9):2250–2267. <https://doi.org/10.1109/TKDE.2013.184>
 23. Zhang C, Mousavi AA, Masri SF, Gholipour G, Yan K, Li X (2022) Vibration feature extraction using signal processing techniques for structural health monitoring: A review. *Mech Syst Signal Process* 177:109175. <https://doi.org/10.1016/J.YMSSP.2022.109175>
 24. Sharma G, Umopathy K, Krishnan S (2020) Trends in audio signal feature extraction methods. *Appl Acoust* 158:107020. <https://doi.org/10.1016/J.APACOUST.2019.107020>
 25. Krishnan S, Athavale Y (2018) Trends in biomedical signal feature extraction. *Biomed Signal Process Control* 43:41–63. <https://doi.org/10.1016/J.BSPC.2018.02.008>
 26. Inyama GK, Oke SA (2021) Maintenance downtime evaluation in a process bottling plant. *International Journal of Quality and Reliability Management* 38(1):229–248. <https://doi.org/10.1108/IJQRM-12-2018-0340/FULL/PDF>
 27. Zennaro I, Battini D, Sgarbossa F, Persona A, Marchi R (2018) Micro downtime: Data collection, analysis and impact on oee in bottling lines the san benedetto case study. *International Journal of Quality and Reliability Management* 35(4):965–995. <https://doi.org/10.1108/IJQRM-11-2016-0202/FULL/PDF>
 28. Brock A, Williams I (2020) Life cycle assessment and beverage packaging. *Detritus* 13:47–61. <https://doi.org/10.31025/2611-4135/2020.14025>
 29. Notarnicola B, Sala S, Anton A, McLaren SJ, Saouter E, Sonesson U (2017) The role of life cycle assessment in supporting sustainable agri-food systems: A review of the challenges. *J Clean Prod* 140:399–409. <https://doi.org/10.1016/j.jclepro.2016.06.071>
 30. Nestle M (2019) Food politics. In: *Food politics*. University of California Press., <https://doi.org/10.1525/9780520955066>
 31. Luque A, Peralta ME, Heras A, Córdoba A (2017) State of the industry 4.0 in the andalusian food sector. *Procedia Manufacturing* 13:1199–1205. <https://doi.org/10.1016/J.PROMFG.2017.09.195>
 32. Voigt T, Flad S, Struss P (2015) Model-based fault localization in bottling plants. *Adv Eng Inform* 29(1):101–114. <https://doi.org/10.1016/J.AEI.2014.09.007>
 33. Valceschini N, Mazzoleni M, Pitturelli L, Previdi F (2022) Experimental fault detection of input gripping pliers in bottling plants. *IFAC-PapersOnLine* 55(6):778–783. <https://doi.org/10.1016/j.ifacol.2022.07.221>
 34. Cerrada M, Sánchez R-V, Li C, Pacheco F, Cabrera D, Oliveira JV, Vásquez RE (2018) A review on data-driven fault severity assessment in rolling bearings. *Mech Syst Signal Process* 99:169–196. <https://doi.org/10.1016/j.ymssp.2017.06.012>
 35. Liang K, Zhao M, Lin J, Ding C, Jiao J, Zhang Z (2020) A novel indicator to improve fast kurtogram for the health monitoring of rolling bearing. *IEEE Sens J* 20(20):12252–12261. <https://doi.org/10.1109/JSEN.2020.2999107>
 36. Altin C, Er O (2016) Comparison of different time and frequency domain feature extraction methods on elbow gesture's emg. *Eur J Interdiscip Stud* 2(3):25–34. <https://doi.org/10.26417/ejis.v2i3.p35-44>
 37. Peng B, Bi Y, Xue B, Zhang M, Wan S (2021) Multi-view feature construction using genetic programming for rolling bearing fault diagnosis [application notes]. *IEEE Comput Intell Mag* 16(3):79–94. <https://doi.org/10.1109/MCI.2021.3084495>
 38. Wang X, Zheng Y, Zhao Z, Wang J (2015) Bearing fault diagnosis based on statistical locally linear embedding. *Sensors* 15(7):16225–16247. <https://doi.org/10.3390/s150716225>
 39. Parmar A, Katariya R, Patel V (2019) A review on random forest: An ensemble classifier. In: *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI)*, vol. 26, pp. 758–763. https://doi.org/10.1007/978-3-030-03146-6_86/FIGURES/1
 40. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in python. *J Mach Learn Res* 12(85):2825–2830
 41. Luque A, Carrasco A, Martín A, Heras A (2019) The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recogn* 91:216–231. <https://doi.org/10.1016/j.patcog.2019.02.023>
 42. Luque A, Carrasco A, Martín A, Lama JR (2019) Exploring symmetry of binary classification performance metrics. *Symmetry* 11(1):47. <https://doi.org/10.3390/SYM11010047>
 43. Gewers FL, Ferreira GR, Arruda HFD, Silva FN, Comin CH, Amancio DR, Costa LdF (2021) Principal component analysis: A natural approach to data exploration. *ACM Comput Surv (CSUR)* 54(4):1–34. <https://doi.org/10.1145/3447755>
 44. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
 45. Nayana B, Geethanjali P (2017) Analysis of statistical time-domain features effectiveness in identification of bearing faults from vibration signal. *IEEE Sens J* 17(17):5618–5625
 46. Yan X, Jia M (2018) A novel optimized svm classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing. *Neurocomputing* 313:47–64
 47. Luque J, Tepe B, Larios D, León C, Hesse H (2023) Machine learning estimation of battery efficiency and related key performance indicators in smart energy systems. *Energies* 16(14):5548
 48. Ghojogh B, Samad MN, Mashhadi SA, Kapoor T, Ali W, Karray F, Crowley M (2019) Feature selection and feature extraction in pattern analysis: A literature review. [arXiv:1905.02845](https://arxiv.org/abs/1905.02845)
 49. Azab A, Khasawneh M, Alrabaae S, Choo K-KR, Sarsour M (2024) Network traffic classification: Techniques, datasets, and challenges. *Digit Commun Netw* 10(3):676–692
 50. Tosin MC, Cene VH, Balbinot A (2020) Statistical feature and channel selection for upper limb classification using semg signal processing. *Res Biomed Eng* 36:411–427

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Amalia Luque was born in Seville Spain, in 1982. She received her Industrial Engineering degree in 2007, Master's in Automation, Robotics and Telematics in 2010 and an Industrial Engineering Doctoral Degree in 2014. She has been involved in teaching related to Project Engineering at the University of Seville since 2015, where she is currently a Professor. Her main areas of research are control, business intelligence, data mining, feature extraction and artificial intelligence.



Daniel Campos Olivares was born in Madrid, Spain, in 1995. He received the Bachelor's degree in Software Engineering from the University of Extremadura in 2018 and the Master's degree in Computer Engineering from the University of Seville in 2021. He is currently a Ph.D. candidate at the University of Seville. Since 2022, he has been working as a Software Architect at SAP. His research and professional interests include artificial intelligence,

predictive maintenance, generative artificial intelligence and knowledge graphs.



Mirko Mazzoleni was born in Ponte San Pietro, Italy, in 1989. He received the M.Sc. degree (summa cum laude) in computer engineering and the Ph.D. degree in engineering and applied sciences (control systems) from the University of Bergamo, Italy, in 2014 and 2018, respectively. He is currently an Assistant Professor with the University of Bergamo. His main research interests include system identification, machine learning, and fault diagnosis.



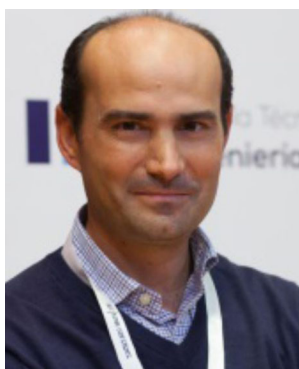
Antonio Ferramosca (M'20) was born in Maglie (LE), Italy, in 1982. He is an Associate Professor at the Department of Management, Information and Production Engineering of the University of Bergamo (Italy). He received his Bachelor's and Master's degrees in Computer Science Engineering, both from the University of Pavia (Italy), in 2004 and 2006 respectively, and the Ph.D. degree in Engineering, from the University of Seville (Spain) in 2011. He was a Postdoctoral

Fellow and then Research Associate at the Argentine National Scientific and Technical Research Council (CONICET), (from 2012 to 2020). He was a visiting researcher at the Department of Chemical and Biological Engineering at University of Wisconsin-Madison (USA) in 2009, and a visiting professor at the Department of Automation and Systems Engineering at the University of Seville (Spain) in 2015. He serves as Associate Editor for the Optimal Control Applications and Methods journal, and for the Journal of Control, Automation and Electrical Systems. He is the author and co-author of more than 100 publications. His research interests are in the field of control theory for constrained linear and nonlinear systems, including Model Predictive Control, learning-based control, Artificial Pancreas, and control applications to industrial and biological systems.



Fabio Previdi (M'10) was born in Milan, Italy, in 1968. He received the M.Sc. degree in electronic engineering (control systems) and the Ph.D. degree in computer science and control systems engineering from the Politecnico di Milano, Milan, Italy, in 1993 and 1999, respectively. From 1999 to 2000, he was a Research Assistant with the Centre for Systems and Control, Department of Mechanical Engineering, University of Glasgow, Glasgow, U.K. From 2000 to 2002, he was a Research

Assistant with the Dipartimento di Elettronica e Informazione, Politecnico di Milano. From 2002 to 2006, he was an Assistant Professor with the Engineering Faculty, Università degli Studi di Bergamo, Dalmine, Italy, where he became an Associate Professor (2006) and a Full Professor (2017). His research interests include fault diagnosis, system identification, control of biomedical systems, and application of control systems methods in physic.



Alejandro Carrasco received his Computer Engineering degree in 1998 and his Ph.D. in Computer Engineering in 2003 from the University of Seville (Spain). Since 1997, he has worked for several companies in the area of software engineering and computer networks, has founded a new technology-based company (NTBF) and he has actively participated and directed several R&D projects. Currently Dr. Carrasco is a Lecturer and Researcher at the Department of Electronic

Technology at the University of Seville, and his research activities are focused on data mining, distributed services and artificial intelligence applied to industrial computing and cybersecurity.

Authors and Affiliations

Amalia Luque^{1,2}  · Daniel Campos Olivares³ · Mirko Mazzoleni² · Antonio Ferramosca² · Fabio Previdi² · Alejandro Carrasco³

✉ Amalia Luque
amalia luque@us.es

Daniel Campos Olivares
dacamposol@gmail.com

Mirko Mazzoleni
mirko.mazzoleni@unibg.it

Antonio Ferramosca
antonio.ferramosca@unibg.it

Fabio Previdi
fabio.previdi@unibg.it

Alejandro Carrasco
acarrasco@us.es

¹ Departamento de Ingeniería del Diseño, Escuela Politécnica Superior, Universidad de Sevilla, Virgen de África 7, Sevilla 41011, Spain

² Department of Management, Information and Production Engineering, University of Bergamo, Via G. Marconi 5, Dalmine 24044, Bergamo, Italy

³ Department of Electronic Technology, School of Computer Engineering, Universidad de Sevilla, Avenida Reina Mercedes, Sevilla 41011, Spain