Journal of
Manufacturing and Materials Processing

MDPI

Article

End-to-End Methodology for Predictive Maintenance Based on Fingerprint Routines and Anomaly Detection for Machine Tool Rotary Components

Amaia Arregi *, Aitor Barrutia and Iñigo Bediaga

Ideko—Basque Research and Technology Alliance (BRTA), Arriaga Kalea, 2, 20870 Elgoibar, Gipuzkoa, Spain; a.barrutia@ideko.es (A.B.); ibediaga@ideko.es (I.B.)
* Correspondence: aarregui@ideko.es; Tel.: +34-943748000

Abstract: This work introduces an end-to-end methodology, from data gathering to fault notification, for the predictive maintenance of rotary components of machine tools. This is done through fingerprint routines; that is, processes that are executed periodically under the same no-load conditions to obtain a snapshot of the machine condition. High-frequency vibration data gathered during these routines combined with knowledge about the machine structure and its components are used to obtain failure-specific features. These features are then introduced to an anomaly and paradigm shifts detection algorithm. The method is evaluated through three distinct scenarios. First, we use synthetically generated data to test its ability to detect controlled variations and edge cases. Second, we use with publicly available data obtained from bearing run-to-failure tests under normal load conditions on a specially designed test rig. Finally, the methodology is validated using real-world data collected from a spindle bearing installed in a machine tool. The novelty of this work lies in performing anomaly detection using failure-specific features derived from fingerprint routines, ensuring stability over time and enabling precise identification of machine conditions with minimal data requirements.

Keywords: machine tools; predictive maintenance; fingerprint routines; anomaly detection; concept drift; Industry 4.0; smart manufacturing

Academic Editor: Enkhsaikhan Boldsaikhan

Received: 3 December 2024
Revised: 20 December 2024
Accepted: 24 December 2024
Published: 3 January 2025

Citation: Arregi, A.; Barrutia, A.; Bediaga, I. End-to-End Methodology for Predictive Maintenance Based on Fingerprint Routines and Anomaly Detection for Machine Tool Rotary Components. *J. Manuf. Mater. Process.* **2025**, *9*, 12. https://doi.org/10.3390/jmmp9010012

1. Introduction

Predictive Maintenance (PdM) is one of the most trending topics in Industry 4.0. In the last years, many survey papers have focused on the topic from a data-driven point of view [1–4]. While most of the principles in PdM date back to the 1990s, recent advances in IoT, cloud computing, data analytics, and machine learning have enabled its wide adoption [5]. Moreover, incorporating artificial intelligence into a facility operations program can reduce labor hours and analysis time while also offering unbiased recommendations for predictive maintenance and investments [6].

The manufacturing sector faces various challenges in maintaining equipment and machinery. Current data-driven PdM solutions, while promising, present several disadvantages that hinder their implementation. Firstly, they require a large amount of historical data to function effectively [7–9]. In addition, lack of labeled data is a typical problem, which impedes the development of classification models [10]. Due to this, significant numbers of data-driven PdM solutions are based on anomaly detection algorithms [11,12], and, often, these approaches use Deep Learning (DL) methods [7–9,13]. Nevertheless, many of these methods rely on static thresholds that do not adapt to changing equipment

conditions, leading to false alarms or the failure to detect imminent issues. Changes in the conditions also impact the obtained data, resulting in variations in the monitored data, which means that the distribution of points changes over time. In other words, the statistical properties of the extracted features may change arbitrarily throughout the components' life. This phenomenon is known as concept drift [14], and it refers to techniques and methods that identify and quantify this event by detecting change points or intervals. Indeed, the authors in [12] encourage the use of dynamic and self-adjusting thresholds; thus, methods that detect these changepoints are preferable. Therefore, it is essential for a truly reliable and robust PdM solution to include a system that is capable of detecting potential concept drifts in the data and resetting the anomaly detection method accordingly. In this regard, the use of DL techniques hampers the interpretability of the results. Moreover, implementing such solutions in a real-world environment is complex, requiring intensive integration with existing systems and constant adaptation to the specifics of the operational environment [15]. Using changepoint detection (CPD) techniques for concept drift allows for the identification of abrupt shifts in data distributions, enabling timely adaptation of anomaly detection frameworks. These methods provide a statistical foundation to detect significant changes, ensuring that the systems remain accurate and responsive to evolving patterns in dynamic environments. This approach is particularly effective when concept changes are sudden and noticeable. Many techniques for changepoint detection have been studied in the literature, but among them, those based on search methods such as window-based, bottom-up, and binary segmentation stand out for being intuitive and easy to implement [16]. These methods are particularly popular due to their simplicity and effectiveness in detecting changes in data distributions.

In this work, we define an end-to-end methodology for PdM of rotary components based on data gathered from predefined processes, named Fingerprint Routines (FRs). An end-to-end approach ensures the methodology encompasses all stages, from data acquisition to anomaly detection, providing a complete and autonomous solution. We combine the use of these routines with the knowledge of the kinematic chain of the rotary components of the machines, exemplified by rolling bearings. By integrating knowledge about the machine structure, this new method allows for computing features regarding very specific faults. Moreover, we also develop an anomaly and concept drift detection algorithm based on dynamic thresholds. Detecting these changes is crucial for two reasons: first, to automatically restart the algorithm once faults are corrected; and second, because faults may not always appear as point anomalies, but rather as a gradual drift. Thus, we automatically detect anomalies in the evolution of several features extracted from their failure-related frequencies. The following are the contributions of our work:

- The anomaly detection algorithm is initialized using very little data.
- The thresholds are dynamically adjusted based on the evolution of the data over time.
- The algorithm also detects changepoints and automatically resets the threshold definitions.
- The failure-specific features that are extracted from the FRs support the interpretability of the results, and the causality of the anomalies can be analyzed.
- All tasks are performed in an online manner, ensuring real-time adaptability and responsiveness.

The anomaly detection algorithm is applied to all the extracted features, no matter the number of them, and can be configured to send notifications or activate alarms. It also states how to integrate external information and gives the pipeline to extract the relevant information. This implies a step towards productization, especially when multiple machines are monitored simultaneously. In order to prove its effectiveness, we have evaluated our methodology using different approaches. First, we assessed the algorithm's ability to detect changepoints, using artificially generated data with labeled changes, regardless of

the type of features extracted from the data. Next, we compared the performance of our algorithm against several classical CPD techniques using real vibration data. Finally, we evaluated the complete framework using a limited amount of real production data.

The rest of this article is structured as follows: first, the theoretical background for feature extraction is thoroughly explained. Then, the methodology for data processing, online anomaly detection, and the CPD approach are described. Afterwards, the results of the methodology evaluations are presented and discussed. Last, conclusions and future work are outlined.

2. Theoretical Background for Feature Extraction

In this section, we introduce some prior knowledge necessary for the extraction of specific failure features.

2.1. Fingerprint Routines

FRs are predefined test cycles in no-load conditions [17]. In this work, FRs focus on the machine's rotary components. The FRs must meet the following requirements:

- Spin the rotary component without holding any tool or performing any machining operation,
- For a predefined time interval, and
- At a constant, predefined rotational speed.

It is essential to ensure that during each FR execution, no workpieces or tools are loaded, and the machine is in the same operational state. This guarantees that any variation in the collected data can be attributed to the state of the rotary component and not to changes in operating or loading conditions.

Since FR executions require the machine to be out of active production, the optimal strategy is to schedule these runs regularly, at times that do not interfere with production. The frequency of FR executions depends on multiple factors, such as machine availability, and the temporal and/or economic costs of halting production. Striking a proper balance between FR regularity and production impact is key to maximizing the effectiveness of predictive maintenance without compromising operational efficiency.

2.2. Vibration

Vibration is a mechanical phenomenon in which oscillations occur around a point of equilibrium. Usually, vibration is captured by accelerometers, so it is measured in m/s^2. Failures can be detected using different processing techniques on the acceleration signal. The adequacy of each technique depends on the stage of the failure.

Regarding the early-stage failures, these are observed at high frequencies, and envelope-based demodulation techniques are applied to the acceleration signal. This is done to extract repetitive impact patterns from the signal, resulting in a demodulation spectrum where the frequency components correspond to the bearing fault frequencies [18]. This technique has already been used to study incipient failures [19,20]. In the case of more advanced-stage failures, the position displacement with respect to the point of equilibrium is more significant. Via the integration property of Fourier's transform [21], if

$$x(t) \overset{\text{FT}}{\longleftrightarrow} X(\omega) \tag{1}$$

then

$$\int_{-\infty}^{t} x(s)ds \overset{\text{FT}}{\longleftrightarrow} \frac{X(\omega)}{i\omega} \text{ if } X(0) = 0 \tag{2}$$

where $A \overset{\text{FT}}{\longleftrightarrow} B$ means that B is the Fourier transform of A. Since the position equals integrating twice the acceleration, the amplitude of the frequency ω of the position is divided by ω^2. This means that for medium (5–1000 Hz) and high (1000–10,000 Hz) frequencies, the amplitudes of the signal are mitigated. Lastly, the medium frequency range is where bearing faults typically become apparent once they have fully developed. Thus, by studying the velocity instead of the displacement, the frequencies are only divided by ω, so the medium frequencies are not so mitigated and faults are well observed. While the theoretical foundation of FFT applies to continuous time series, in practice, we work with discretized signals. In our work, numerical methods are applied to approximate the integral over the sequence of discrete values, ensuring accurate feature extraction tailored to the nature of the data.

2.3. Characteristic Fault Frequencies in Bearings

When a bearing is faulty or rolls on a faulty raceway, it impacts the raceway, generating a detectable vibration [22]. Most commonly, failures come from wear in their inner race, their outer race, or their balls, including cracks, pits, spalls on the rolling surface, and race spalls [20]. This causes high-frequency resonances in the machine structure to be excited. In this work, we focus on four types of failures: Ball Pass Frequency for Outer race (BPFO), Ball Pass Frequency for Inner race (BPFI), Ball Spin Frequency (BSF), and Fundamental Train Frequency (FTF). These frequencies are calculated as follows:

$$\begin{aligned} BPFO &= f_r \cdot \frac{n}{2} \cdot \left(1 - \frac{B_D}{P_D} \cos \alpha\right) \\ BPFI &= f_r \cdot \frac{n}{2} \cdot \left(1 + \frac{B_D}{P_D} \cos \alpha\right) \\ BSF &= f_r \cdot \frac{P_D}{2 \cdot B_D} \cdot \left[\left(1 - \frac{B_D}{P_D} \cos \alpha\right)^2\right] \\ FTF &= f_r \cdot \frac{1}{2} \cdot \left(1 - \frac{B_D}{P_D} \cos \alpha\right), \end{aligned} \tag{3}$$

where n is the number of balls, f_r is the rotation frequency of the shaft to which they are attached, α the contact angle, B_D the ball diameter, and P_D the pitch diameter. While the remaining factors are intrinsic properties of the bearing, f_r depends on the operating conditions of the machine. In this work, we label the fundamental fault frequencies (FFFs) of a bearing as BPFO, BPFI, BSF, and FTF providing a rotation speed of 60 RPM; that is, $f_r = 1$ Hz.

An example is shown in Figure 1. The amplitude corresponding to a fault is highlighted, as well as the values of the first ten multiples of its associated frequency. Notice how these values are also the peaks of the spectrum. This example corresponds to the bearing SKF [23] NN 3030 K/SPW33, whose FTF fault frequency is 0.455. The spectrum was captured for 2000 RPM rotation. Then, if the FTF fault is taking place, an amplitude peak is expected to happen at frequency $(2000/60) \cdot 0.455 \approx 15.166$. Also remember that, with a vibration in a frequency, harmonic vibration usually takes place in frequencies that are multiples of the fundamental fault frequency.

Knowing the bearing, its fault frequency information can be found, as many bearing manufacturers provide them. For instance, the information on the NN 3030 K/SPW33 bearing is given by SKF [23]. By selecting the calculation, checking bearing frequencies, and introducing the 60 RPM, the fault frequencies are computed.

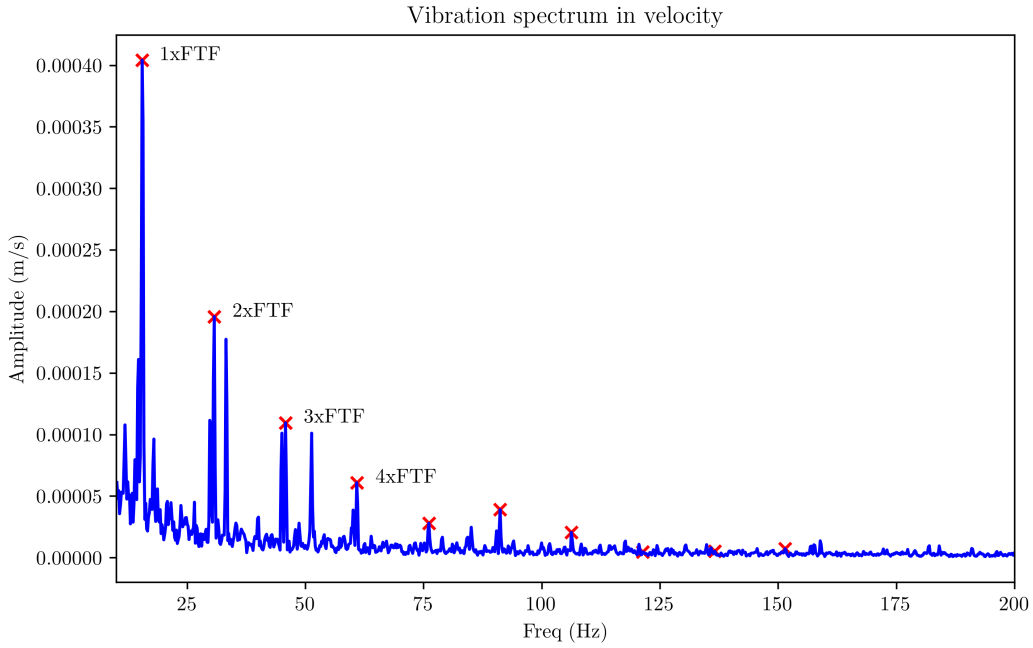

Figure 1. Example of a frequency spectrum obtained from the vibration data of a rolling bearing. The red crosses correspond to the amplitude peaks of the frequency associated with the FTF failure and its harmonics.

2.4. Kinematic Chains

Machine tools contain a number of components, including motors, axles, and spindles. Their movement is generated by motors and transmitted to other components through kinematic chains of variying complexity, using elements such as gears and belts. On the one hand, a simple example of this kind of chain is a spindle whose rotation is transmitted from a motor by a belt. On the other hand, a more complicated chain is a motor transmitting the rotation to a spindle by a combination of axles interacting with each other through gears. In this work we assume that the kinematic chain is linear and the transmission along the chain is performed by a ratio, considering that the vibration monitoring sensor is installed on the last element of the kinematic chain.

Here are the definitions of some specific concepts related to kinematic chains:

- The main speed (MS) of a kinematic chain is the speed rotation by which the rest of the speed rotations are computed. Take, for instance, a system where motor A transmits its rotation to spindle B with a 1.5 rate. Then, if the motor speed is 1000 RPM, the spindle speed is 1500 RPM. The MS of the system would be 1000 RPM, and the other speed is computed from it.
- The main frequency (MF) is defined as the MS in Hz; that is, MS/60 if MS is given in RPM.
- A transmission component (TC) is a machine component that has a rotating speed and might transmit it to another TC.
- The relative speed (RS) of a TC is the speed that the TC rotates as a multiplier of the MS.
- A TC is the child of a parent (another TC) if its RS is computed as a multiple of the RS of its parent.
- For any parent–child tuple, the transmission rate (TR) from parent to child is the ratio between the RS of the parent and the child.
- If a TC is not the child of any other TC, then it is called the main component (MC), and its speed is computed as a multiple of the MS (generally with an RS of 1).

- A potentially faulty element (PFE) is a component that spins with a TC. Each PFE is associated with one, and only one, TC. Their transmission rate with respect to its parent TC is usually 1.
- A fault is a property associated with a PFE. Each fault is defined by its name and its relative fault frequency (RFF) with respect to the rotation speed of the PFE (which, in the case of bearings, it is the FFF). The names of the faults, in the case of bearings, will be the ones introduced in the correspondent section (BPFI, BPFO, etc.), and their RFFs will be the relative frequencies with respect to their rotation that are excited when those faults take place.
- The specific fault frequency (SFF) of a fault type for a specific PFE is the frequency that will be excited in case a fault happens. It is computed as the RFF of the fault times the RS of the bearing times the MF.

An illustrative example of a kinematic chain is shown in Figure 2. In this example, there are four TCs: motor, axle 1, axle 2, and spindle. Each TC is the parent of the next TC. The RS of the motor is 1; that is, its transmission rate with respect to the MS, since it is the MC. For the rest of the TCs, its RS is the product of its TR and the TRs of all the TRs in the chain going backwards until the MC. That is, The RS of Axle 2 would be $1.6 \cdot 1.5 \cdot 1 = 2.4$. The RFF for the BPFI failure for Bearing 1 is computed as $1 \cdot 1.5 \cdot 1.6 \cdot 2.1 \cdot 1 \cdot 12.34 = 62.1936$. Then, if the rotation speed of the motor was 300 RPM, then the SFF for the BPFI failure for Bearing 1 would be $300\ \text{RPM}/60 \cdot 62.1936 = 310.968$ Hz. Therefore, a BPFI fault in Bearing 1 would imply a peak for the 310.968 Hz frequency (and its multiples).

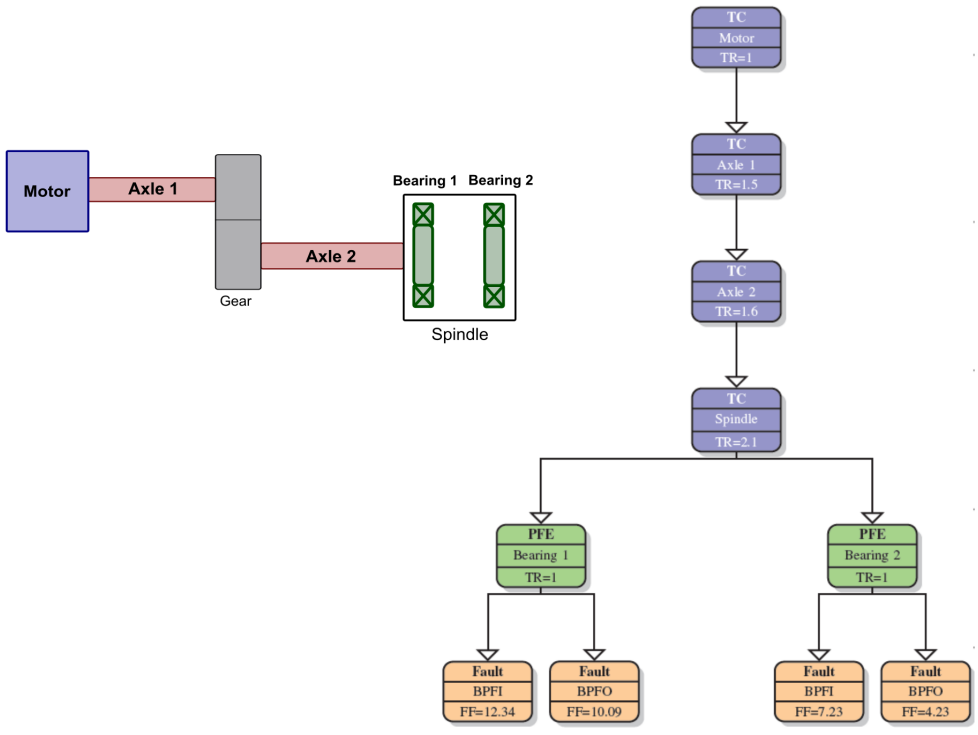

Figure 2. Example diagram of a kinematic chain.

It is important to note that resonances in the kinematic chain can significantly impact the analysis. If the elements of the chain (motors, shafts, etc.) share specific frequencies, the amplitudes observed in the FFTs at those frequencies may become significantly elevated. This phenomenon can lead to confusion when interpreting the peaks, as they might not directly indicate faults but rather inherent resonances within the system. However, this work does not address this issue, leaving it as a consideration for future studies.

3. Methodology

The methodology consists of an end-to-end pipeline with four stages, which are executed sequentially:

1. FR execution: The FR is executed.
2. Data gathering: The accelerometer data are stored.
3. Data processing: Data undergo a series of transformation steps.
4. Anomaly detection: Detection and notification of relevant events.

The data processing stage contains all the steps, from receiving the data and transforming it into features suitable for anomaly detection to integrating knowledge about the kinematic chain. Additionally, the anomaly detection stage introduces the algorithm that will be used to deal with the problem using very short time series. These two stages are thoroughly explained in the following sections.

3.1. Data Processing

In this stage, the raw data are processed in order to extract failure-related features. We divide the data processing stage into three sub-stages.

3.1.1. Preprocessing

As mentioned in Section 2.2, some transformations are applied to the raw vibration signal before the FFT is calculated. In order to perform the vibration analysis in velocity, it is necessary to compute the integral of the acceleration signal. In addition, envelope-based techniques are applied to the raw signal to compute the FFT in demodulation.

3.1.2. Feature Extraction Considering Kinematic Chains

In order to analyze the faults using FRs, we need to obtain the RFF for each of the faults of the components in the kinematic chain. For example, in the kinematic chain defined in Figure 2, the RFFs of the faults are computed by multiplying all the factors for each line from each fault until the first TC. Thus, the kinematic chains need to be well defined and stored so that the information can be extracted on demand for pipeline automation.

The way to structure this information is using a relational database [24], with three types of elements. The types of elements are the following:

- TCs: Each TC is defined by its TR and its parent (if they have one).
- PFEs: Each PFE is defined by its TR and the TC they are attached to.
- Faults: Each fault is defined by its FFF and the PFE they are related to.

TC chains can be indefinitely long, so the model contains a field of a table that refers to another field of the same table. An example of the data model for a kinematic chain of a machine is given in Figure 3. In TCs, the field `Parent_TC` contains its parent TC, and it must be consistent with the names in `TC_name`. If a TC is the MC, then `Parent_TC` will be null.

The steps to retrieve all the relevant information about the chain from the database are the following:

1. Iterate through all the faults in Faults. For each of them:
2. Multiply its FFF by the TR of the PFE it is attached to.
3. Multiply the result by the TR of the TC the PFE is attached to.
4. If the `Parent_TC` of that TC is null, stop. If it is not, look for the table row for which `TC_name` is equal to the `Parent_TC` of the current TC. Multiply our previous product for the TR of this new TC. Repeat this step until `Parent_TC` is null. The result, for each fault, is their RFF.

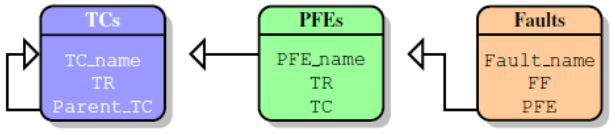

Figure 3. Data model schema.

3.1.3. Severity Computations

Once we obtain the RFF of the faults and MS, the SFF can be computed. As mentioned earlier, if a fault occurs, then vibration peaks will be observed in the SFF and its harmonics, as seen in Figure 1. Peaks are observed for velocity or demodulation, depending on the fault being apparent or incipient.

To measure the significance of the vibration data, we introduce the concept of severity. We define the following set:

$$\mathcal{B} = \bigcup_{k=1}^{10} [\min(k \cdot \beta - 0.5, 0.98 \cdot k \cdot \beta), \max(k \cdot \beta + 0.5, 1.02 \cdot k \cdot \beta)]. \tag{4}$$

where β is the SFF for a fault. This set is the union of intervals around the first ten multiples of β, with a width of $\pm 2\%$ that is a multiple of β, and with a lower width bound of ± 0.5 Hz. Figure 4 show these frequencies represented by the green bands.

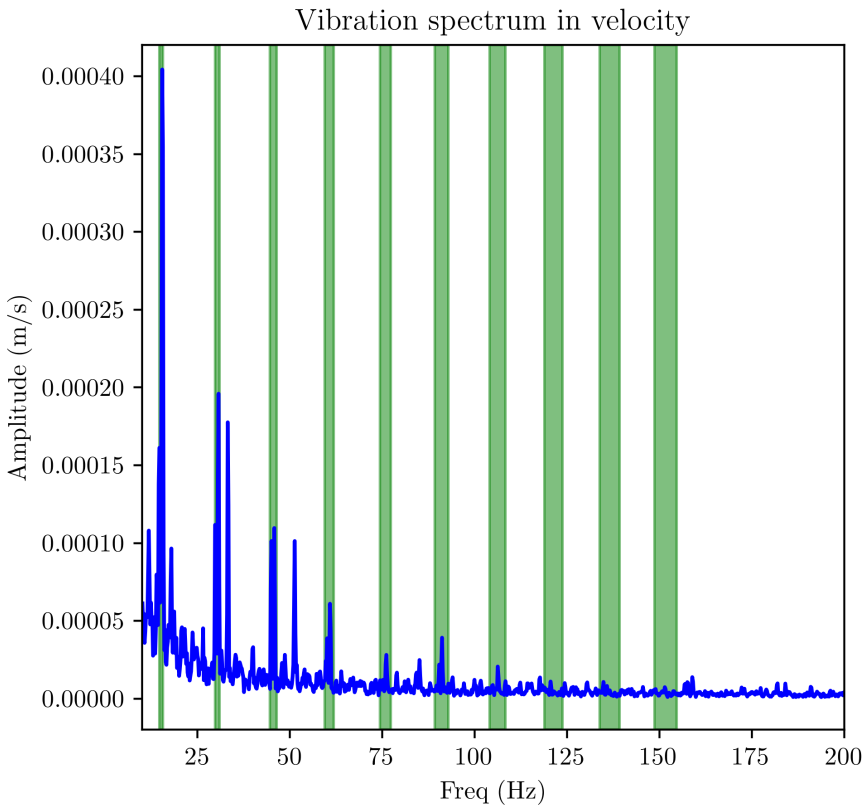

Figure 4. Example of a FFT spectrum in velocity, obtained from the vibration data of a rolling bearing. The green bands represent the set $\mathcal{B}$, i.e., the union of frequency intervals used for the severity computation.

We denote

$$\{(f_i, a_i) | i = 1, \ldots, N\} \tag{5}$$

where f_i and a_i are the i-th frequency and amplitude values of the FFT, respectively. Then, we can define the severity as follows:

$$S = \left(\sum_{f_i \in \mathcal{B}} a_i^2 \right)^{1/2}, \tag{6}$$

that is, the square root of the quadratic sum of amplitudes correspondent to frequencies that belong to $\mathcal{B}$.

3.2. Anomaly Detection and Changepoint Detection

An algorithm for anomaly detection is introduced based on previous works [25]. Nevertheless, the algorithm is adapted to the particularities of the context, considering the length of the series and the evolution of the data.

First, the proposed algorithm deals the problem of anomaly detection for very short time series. For example, if a FR is executed every two weeks, we collect 28 instances a year, a very small sample size for an anomaly detection algorithm, and yet this is too much time for the manufacturer to start obtaining insights about the critical components of the machines. The proposed algorithm evaluates each new value of the series (from a new FR execution), and it determines if the observation is anomalous with respect to the previous ones. In addition, the first n values of the series are used to build the normality scenario, so they are not evaluated. Notice that n must be a very small number, since we want to detect anomalies as soon as possible. Otherwise, waiting too much for evaluating the features would allow faults to go undetected.

Second, scenario changes can occur within the series, meaning that the distribution of points may shift over time. In other words, the statistical properties of the extracted features may change arbitrarily over time. This phenomenon is known as concept drift [14]. This concept refers to techniques and methods that identify and quantify this phenomena by detecting change points or intervals. There are different types of concept drifts, but in this work, we are focusing on sudden, gradual, and incremental drifts. Detecting these changes is crucial for two reasons: first, to automatically restart the algorithm once faults are corrected; and second, because faults may not always appear as point anomalies, but rather as a gradual drift. Existing algorithms are not designed for time series as short as ours, so we developed a custom algorithm.

The algorithm needs to:

- detect anomalies,
- detect concept drifts,

and derived from these

- to determine when to notify.

This point is significant, as it is crucial to report only events that are truly relevant. In a real production scenario, we might have thousands of features to analyze, so it is important to keep the number of notifications low by avoiding all the notifications that do not reveal a fault (false positive).

Before defining the algorithm, we introduce some parameters:

- cs: Critical severity.
- dt: Density threshold.
- bs: Buffer size.
- N_{min}: Minimal amount of points in the buffer before considering anomalies or changes.
- af: Average factor.
- csa: Critical severity for average factor.

Moreover, the following terms are also used in the algorithm, where their calculation is performed iteratively and conditionally:

- $x_1, x_2, x_3, \ldots$: Incoming and ordered values of the series.
- m_i: Accumulated mean of the series at the i-th point.
- $m_{2,i}$: Accumulated mean of the squared series at the i-th point.
- s_i: Accumulated standard deviation of the series at the i-th point: $\sqrt{m_{2,i} - m_i^2}$.
- N_i: Number of values used for the mean and cumulative value at the i-th point.
- a_i: Anomaly score of the series at the i-th point.
- B_i: Buffer at the i-th point.
- aB_i: Anomaly buffer at the i-th point.

In Algorithm 1, the computations of the terms m_i, $m_{2,i}$, and s_i for each step and scenario are defined. The buffer B_i is defined as a set of values x_j with $j \leq i$ that is preserved for later use. It is defined as a FIFO (First In First Out) object with a maximal length of bs. If

$$B_i = \{x_{i_1}, x_{i_2} \ldots, x_{i_k}\} \tag{7}$$

with $i_1 < i_2 < \ldots < i_k$, then we define the operator $append$ as follows:

$$append(B_i, x) = \begin{cases} \{x_{i_1}, x_{i_2} \ldots, x_{i_k}, x\} & if\ |B_i| < bs \\ \{x_{i_2} \ldots, x_{i_k}, x\} & if\ |B_i| = bs \end{cases}. \tag{8}$$

That is, if B_i has size bs, then to append a new element implies removing the oldest. The anomaly buffer is defined similarly, using the anomaly scores instead.

Once the previous parameters are defined, we can describe the anomaly detection algorithm. Taking the sequence $x_1, \ldots, x_M$, the goal is to compute an anomaly threshold based on the cumulative mean and standard deviation to identify when this threshold is exceeded. Additionally, the algorithm tracks when the cumulative mean of the most recent values significantly deviates from the cumulative mean of the entire sequence. Anomalies and changes are not considered until the buffers contain at least N_{min} points, establishing a normality scenario. After adding the values of N_{min} to the buffer, the values of m_i and $m_{2,i}$ remain constant until a concept drift occurs. This approach ensures that gradual changes in the values (such as a slow but steady increase) are eventually detected.

The algorithm is presented in Algorithm 1. It indicates how new values are added to the algorithm, and it uses the function *eval_change*, presented in Algorithm 2. The next step is to evaluate whether change happens by $c_i := eval\_change$.

- If there is not a minimal number of values in the buffer ($N_{i-1} < N_{min}$), then no change is evaluated.
- If $c_i =$ `NoChange`, it means that no change of scenario happens.
- If $c_i =$ `ChangeAnom`, then a change of scenario happens via the accumulation of anomalous values. This happens when the added value is anomalous, and the proportion of anomalous values in the buffer exceeds the parameters dt.
- If $c_i =$ `ChangeAvg`, then a change of scenario happens through deviation from the mean value m_i. This happens when the mean of the values in the buffer that are not anomalous under the *csa* threshold deviate from the previous mean value m_{i-1}, in proportion, more than *csa*; that is, when the last values are not necessarily anomalous, but on average, they deviate too much from what was established as the average scenario.

Then, the value is added to the buffers, and the cumulative variables (m_i, $m_{2,i}$, s_i, N_i) are updated.

Algorithm 1 Anomaly and change detection algorithm

$N_1 := 1$; $m_1 := x_1$; $m_{2,1} := x_1^2$; $B_1 := \{x_1\}$
for i in 2, ..., M **do**
  if $N_{i-1} < N_{min}$ **then**
    $m_i := \frac{m_{i-1} \cdot (N_i - 1) + x_i}{N_i}$, $m_{2,i} := \frac{m_{2,i-1} \cdot (N_i - 1) + x_i^2}{N_i}$
    $s_i = \sqrt{m_{2,i} - m_i^2}$
    $N_i := N_{i-1} + 1$
    $B_i := append(B_{i-1}, x_i)$
    $aB_i := append(aB_{i-1}, 0)$
  else
    $c_i := eval\_change(B_{i-1}, aB_{i-1}, m_{i-1}, m_{2,i-1}, x_i)$
    if $c_i =$ `NoChange` **then**
      $a_i := \frac{x_i - m_{i-1}}{s_{i_1}}$
      $m_i = m_{i-1}$, $m_{2,i} = m_{2,i-1}$
      $N_i := N_{i-1}$
      $B_i := append(B_{i-1}, x_i)$
      $aB_i := append(aB_{i-1}, a_i)$
    else if $c_i =$ `ChangeAnom` **then**
      $B_{aux} := append(B_{i-1}, x_i)$
      $aB_{aux} := append(aB_{i-1}, a_i)$
      $aB_i := \{0 | a_{i_k} \in aB_{aux}, |a_{i_k}| > cs\}$
      $B_i := \{x_{i_k} | x_{i_k} \in B_{aux}, |a_{i_k}| > cs\}$
      $N_i := |B_i|$
      $m_i = \frac{1}{N_i} \sum_{x \in B_i} x$, $m_{2,i} = \frac{1}{N_i} \sum_{x \in B_i} x^2$
    else if $c_i =$ `ChangeAvg` **then**
      $B_{aux} := append(B_{i-1}, x_i)$
      $aB_{aux} := append(aB_{i-1}, a_i)$
      $B_i := \{x_{i_k} | x_{i_k} \in B_{aux}, |a_{i_k}| < csa\}$
      $aB_i := \{a_{i_k} | a_{i_k} \in aB_{aux}, |a_{i_k}| < csa\}$
      $N_i := |B_i|$
      $m_i = \frac{1}{N_i} \sum_{x \in B_i} x$
      $m_{2,i} = \frac{1}{N_i} \sum_{x \in B_i} x^2$
    end if
    $s_i = \sqrt{m_{2,i} - m_i^2}$
  end if
end for

This algorithm is inspired by streaming algorithms [26], where cumulative statistics are computed without storing all the previous values. The main advantage of this kind of algorithm is that the anomaly score has a statistical significance that is independent to the magnitude of the values in the series. In contrast, the previous values of the series are forgotten, but this is not suitable for scenario changes such as those discussed in this work. Therefore, the buffer object is introduced.

The first n values of the series are not evaluated, since the algorithm needs some time to establish a normality scenario. Since the executions of FRs are expected to be sparse in time, the parameters need to be set so they offer reliable notifications after not too long.

The criteria for notifications are the following:

- The execution is anomalous and above the cumulative mean ($a_i > cs$). The anomaly score is signed, so it is positive if the incoming value is above the cumulative mean and negative if it is below.
- A changepoint is detected based on drift in cumulative mean ($c_i =$ `ChangeAvg`) and the mean value after if it is greater than the mean value before ($m_{i-1} < m_i$).

Algorithm 2 *eval_change*$(B_{i-1}, aB_{i-1}, m_{i-1}, m_{2,i-1}, x_i)$

$a_i := \frac{x_i - m_{i-1}}{s_{i_1}}$
$B_{aux} := append(B_{i-1}, x_i)$
$aB_{aux} := append(aB_{i-1}, a_i)$
$change :=$ `NoChange`
if $a_i > cs$ **then**
  {Evaluate change via anomaly accumulation.}
  $aB_{anom} = \{a | a \in aB_{aux}, |a| > cs\}$
  if $\frac{aB_{anom}}{|aB_{aux}|} \geq dt$ **then**
    $change :=$ `ChangeAnom`
  end if
else
  {Evaluate change via deviation in mean value.}
  $B_{avg} = \{x_{i_k} | x_{i_k} \in B_{aux}, |a_{i_k}| < csa\}$
  $p_{avg} = \frac{1}{|B_{avg}|} \sum_{x \in B_{avg}} x$
  if $|p_{avg} - m_{i-1}| / m_{i-1} > af$ **then**
    $change :=$ `ChangeAvg`
  end if
end if
return *change*

4. Results and Discussion

We evaluate our methodology in three distinct scenarios to ensure its robustness and applicability. The first scenario involves artificially generated data, used exclusively to test the methodology's ability to detect concept drifts, regardless of the type of feature extracted from the data. In the second scenario, we use real experimental data to assess performance on standardized benchmarks. Finally, the third scenario is focused on real-world industrial data to validate the methodology's effectiveness in practical applications.

4.1. Artificially Generated Dataset

To thoroughly assess the performance of our methodology's ability to detect concept drifts, we create an artificially generated dataset. This dataset is designed to include controlled concept drifts, representing shifts in the underlying data distribution over time. By incorporating these artificial changes, we establish a well-labeled ground truth that allows us to evaluate how accurately our methodology detects and adapts to these transitions. This setup not only provides a controlled environment to test our approach but also helps us measure its precision and responsiveness in detecting scenario changes under varying conditions.

The synthetically generated sequence consists of 2000 samples, divided into five segments, each with distinct statistical properties. The first segment follows a standard normal distribution ($\mu = 0$, $\sigma = 1$), while the second segment has a shifted mean ($\mu = 3$, $\sigma = 0.7$). The third segment follows a normal distribution with a mean of $\mu = -2$ and a smaller standard deviation ($\sigma = 0.4$). The fourth segment has a mean of $\mu = -2$ with a larger standard deviation ($\sigma = 1.2$). The fifth segment exhibits an upward trend, modeled as the cumulative sum of normal noise ($\sigma = 0.2$), with additional Gaussian noise ($\mu = 0$, $\sigma = 0.5$) to simulate real-world variability. Concept drifts are introduced at the boundaries between segments, simulating abrupt changes in data distributions and patterns. This setup provides a controlled environment for benchmarking our algorithm. The generated sequence is illustrated in Figure 5, where the concept drifts are clearly marked at the segment boundaries.

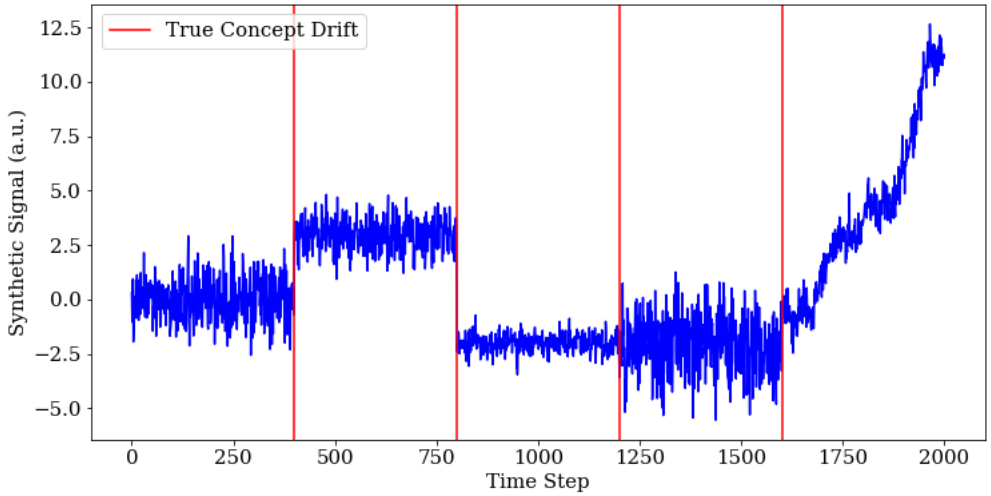

Figure 5. Synthetic signal with annotated concept drift points (red lines). The signal is presented in arbitrary units (a.u.).

Our methodology is applied using a specific set of parameters, which are detailed in Table 1. The results obtained from applying the method are graphically illustrated in Figure 6.

Table 1. Parameters for anomaly detection algorithm using synthetically generated dataset.

cs	dt	bs	N_{min}	af	csa
2	0.5	10	8	0.2	4

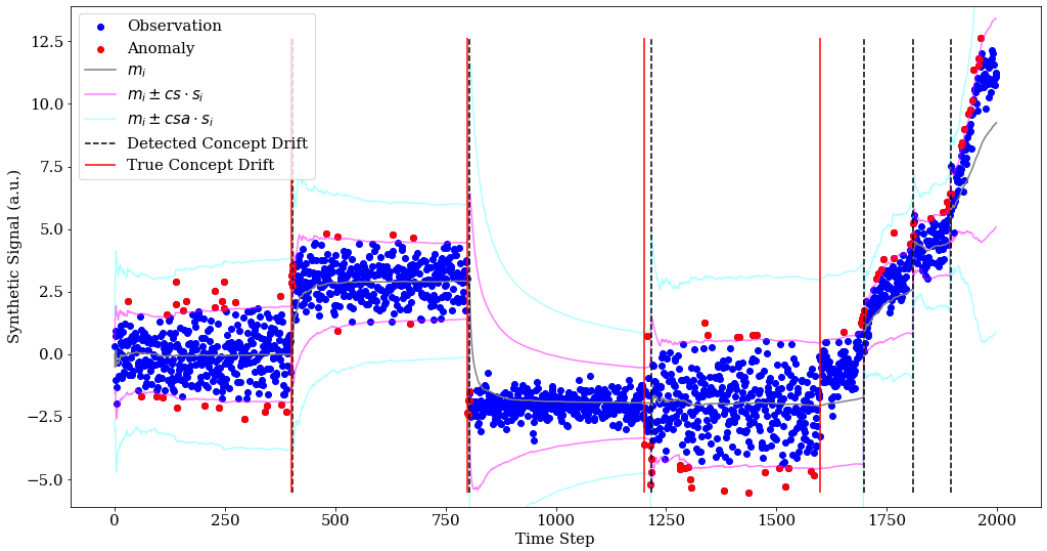

Figure 6. Results obtained using the proposed methodology on the synthetically generated dataset.

Our method effectively and quickly detects changes in the mean and/or increases in the standard deviation. In this regard, our algorithm identifies three true positives. However, in the case of time series with a marked trend, the algorithm does not reset its parameters as quickly as in other cases. It fails to detect the last true positive while subsequently identifying three false positives. Nevertheless, this behavior is actually preferable, as failing to adapt in such cases could potentially mask an underlying deterioration or change in the system. The trade-off between adaptation speed and accuracy in trend-

dominated series highlights the robustness of the method in identifying significant changes without overfitting to non-relevant patterns.

4.2. Real Experimental Data

This subsection focuses on evaluating the performance of our methodology using a publicly available dataset. The dataset consists of real vibration data obtained from an experimental study conducted by NSF I/UCR Center for Intelligent Maintenance Systems (IMS) with support from Rexnord Corp. in Milwaukee, WI, where four bearings were installed on a shaft and run-to-failure. The experiments were conducted with a constant rotational speed of 2000 RPM. PCB 353B33 High-Sensitivity Quartz ICP accelerometers were installed in the test rig, one per bearing. This study includes data from three different tests, each designed to take the bearings to their breaking point. Each test is composed of several files containing the vibration signals obtained by the accelerometers. Readers interested in further details of the experimental setup are referred to [27].

In the present work, we only consider the data obtained from the first bearing of the second test. This test comprises 984 files, each containing accelerometer signals recorded during the operation of the bearings. Specifically, the bearing analyzed in this study exhibited an outer race fault. It is worth noting that anomalies within the dataset are not explicitly labeled; however, information is available for each test regarding the specific failures experienced by the bearings. This means we know how the life of each component ends, but the onset of the degradation phase remains unknown. For the analysis, we extract the severity related to the BPFO failure. A graphical illustration of its evolution throughout the bearings' life is shown in Figure 7.

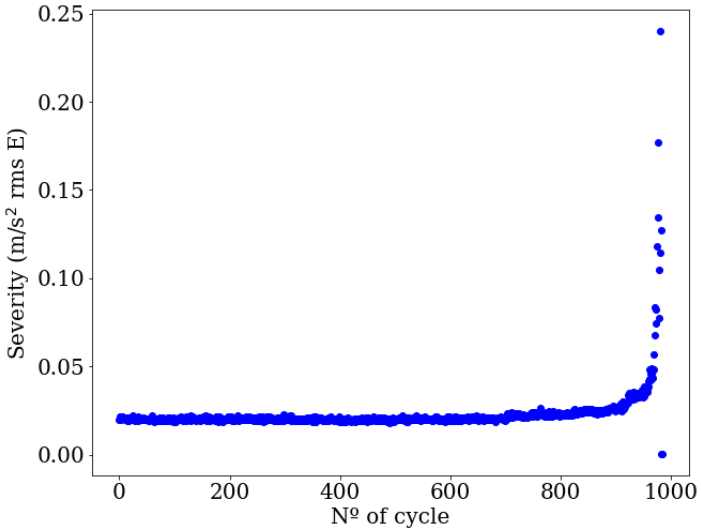

Figure 7. Evolution of the severity values related to the BPFO failure.

Using our technique, both anomalies and concept drifts are effectively detected. To assess the performance of scenario change detection, we compare our method with classical CPD techniques, which are executed offline. The selected methods are binary segmentation, windowing, bottom-up, and kernel-based approaches [16]. The specific parameters used for each of these techniques are detailed in Table 2, allowing for a fair and transparent comparison with our methodology. A key distinction of our methodology is that it operates online, making it suitable for real-time applications. We also emphasize the importance of detecting scenario changes promptly while avoiding an excessive number of detections that could overwhelm the end user. A high frequency of scenario change alerts may compromise the perceived reliability of the system, making it critical to strike a balance between detection speed and the clarity of actionable insights.

Table 2. Parameters for changepoint detection algorithms.

Window Based	Bottom-Up	Binary Segmentation	Kernel Based
model = 'l2' min_size = 6 window_size = 4	model = 'l2' min_size = 6	model = 'l2' min_size = 6	kernel = 'rbf' min_size = 6 gamma = 100

We present the results starting from cycle 600, as no method detects scenario changes before this point. The discussion of the results is divided into two sections: cycles 600 to 950 (Figure 8), and cycles beyond 950 (Figure 9), to better highlight the details. It is important to note that, due to the lack of labels, accuracy rates cannot be computed.

In the first section (cycles 600 to 950), the first visually noticeable change occurs around cycle 700. This change is detected by all methods except the window-based approach. The kernel-based method detects an excessive number of changes, while the window-based method fails to identify any. The other two methods produce similar segmentations; however, our approach identifies one additional change. This extra change corresponds to a slight, consecutive increase in feature values, triggering a model update. In the second section (cycles beyond 950), the kernel-based method continues to detect an excessive number of scenario changes. The other methods, including ours, identify approximately two changes, which is consistent with the strong trend present in the data. This behavior is desirable, as it avoids unnecessary updates while still capturing the main changes. Considering that our method operates online while the others are executed offline, it clearly stands out in comparison. Despite the inherent challenges of real-time processing, our approach achieves highly satisfactory results, effectively balancing the timely detection of scenario changes with the avoidance of excessive updates. This highlights the robustness and practicality of our method in dynamic environments, where online performance is critical.

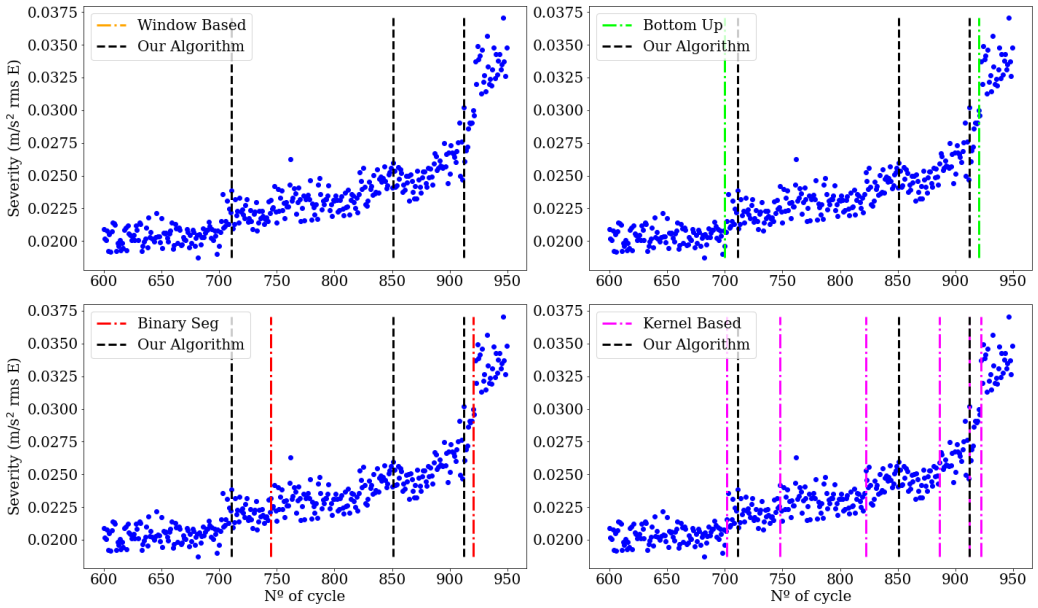

Figure 8. Performance comparison between changepoint detection algorithms and our method for cycles 600 to 950.

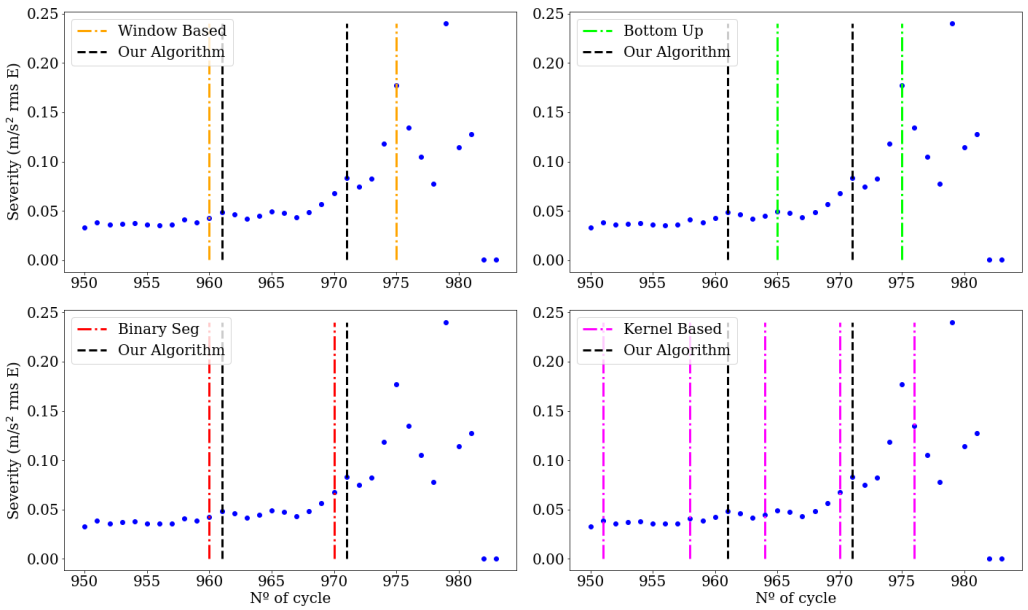

Figure 9. Performance comparison between changepoint detection algorithms and our method for cycles 950 to 984.

4.3. Real Use Case

This section analyzes the performance of our methodology in a real-world use case, focusing on data collected from a machine tool under operational conditions. The use case involves the spindle of a centerless grinding machine equipped with an NN 3016 KTN/SP bearing, manufactured by SKF. Throughout the time frame of the dataset, the bearing initially operates in good condition. However, as more FRs are executed, signs of deterioration start to emerge. Specifically, the bearing begins to exhibit issues with the outer ring, causing the spindle to resonate at its BPFO frequency. Figure 10 illustrates the damage to the outer ring. Eventually, an intervention is performed after the failure occurs, and the machine resumes operation under a new normal, where further deterioration of the bearing is controlled. To the best of our knowledge, no previous work has addressed this issue by integrating information about the kinematic chain. Additionally, the limited amount of available data makes the application of DL techniques impractical. Furthermore, obtaining real industrial data where the rotating component's condition is clearly detectable is a challenging task. These factors complicate the comparison with other approaches, failure types, and similar studies.

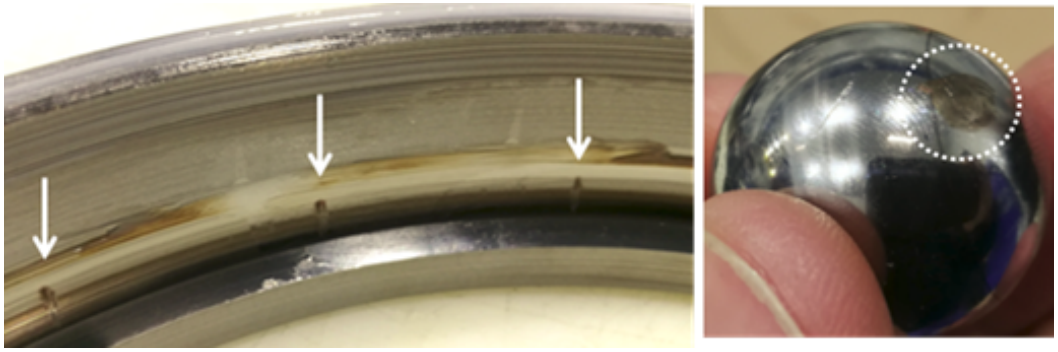

Figure 10. Spalls on the bearing outer ring, highlighted with white arrows on the left, and wear in one of the bearing balls, marked with a dotted frame on the right.

4.3.1. FR Definition and Data Gathering

Data monitoring is done using two triaxial accelerometers placed in the front and rear part of the spindle, as stated in [28]. The calibration of the entire system is performed by measuring a vibration generator. This vibration generator is a small, handy, completely self-contained vibration reference source. It is intended for rapid checking of vibration measurement, monitoring, and recording systems using piezoelectric accelerometers, as well as other types of vibration transducers, having a maximum weight of 210 g. The accelerometers operate at a sampling rate of 25 kHz. On the other hand, the FR consists of rotating a grinding spindle at 600 RPM for 12 s. The FRs are executed every two weeks, provided it does not interfere with the production, in which case the execution could be rescheduled.

The data gathering process is automated. Provided the machine is equipped with the suitable IoT infrastructure, the captured data are written into a file. These data are stored in structured files, which are indexed by the placement of the accelerometers, containing for each placement a time series of time–acceleration tuples. These files are structured by the location of the accelerometers. Each file type contains a series of time–acceleration tuples. Figure 11 shows the raw signal from the accelerometer placed in the front of the spindle.

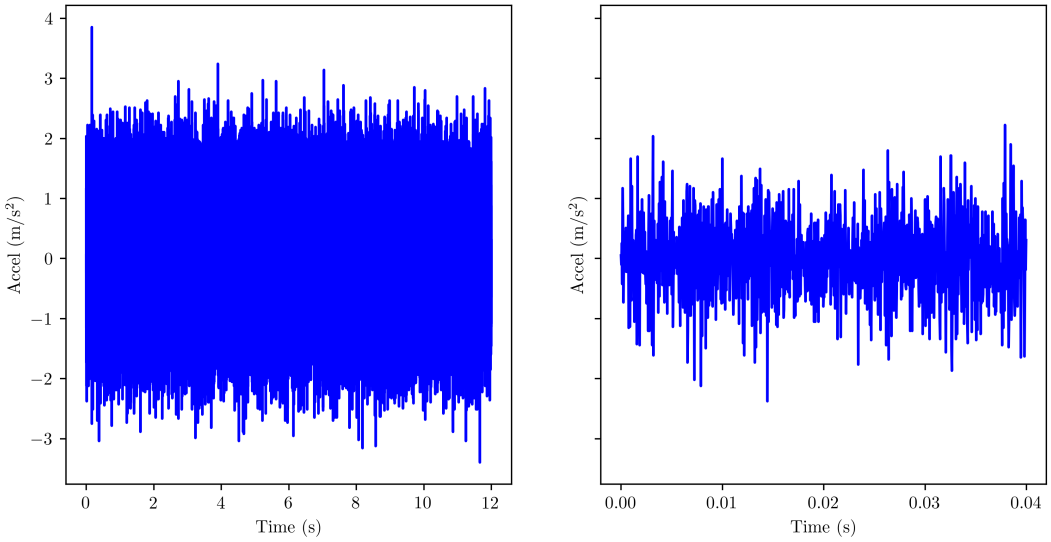

Figure 11. (**left**) Raw acceleration plot during a FR; (**right**) raw acceleration plot during a FR (detail).

4.3.2. FFT Signal Transformations

For each accelerometer, two FFT spectra are computed. Figure 12 shows the spectra for the FR shown in Figure 11. Low frequencies are not significant for detecting peaks, since they are masked by the non-stationary part of the series. The rotation speed of this particular FR is 600 RPM; that is, the 10 Hz frequency. Thus, amplitude peaks appear at 10 Hz, which is natural, since in the absence of further anomalies, it should be the most dominant frequency. Therefore, only the amplitudes of the frequencies above 10 Hz are considered.

4.3.3. Visual Exploration

The effectiveness of the solution is demonstrated through an example in which the bearing mentioned above begins to deteriorate. The process of obtaining the relevant frequency is straightforward, as the spindle's rotation is transmitted directly from the motor in a 1:1 ratio. The BPFO frequency of the bearing is 11.783 times its rotational

frequency. Given that the spindle rotates at 600 RPM during the FR, the SFF for the BPFO fault is calculated as follows:

$$\frac{600 \text{ RPM}}{60} \cdot 11.783 = 117.83 \text{ Hz}. \quad (9)$$

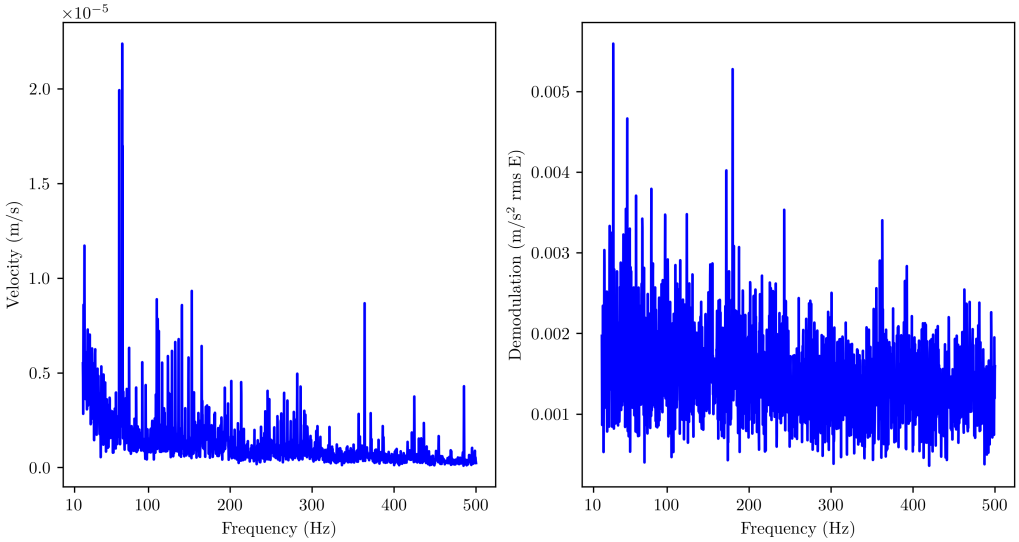

Figure 12. FFTs in velocity (**left**) and demodulation (**right**). The data used to calculate these FFTs are the same as in Figure 11.

Figure 13 shows how the demodulation spectra evolve over time. The bands related to the SFF of the BPFO of that bearing are highlighted in green, and their values remain steady until, at some point, they start to increase.

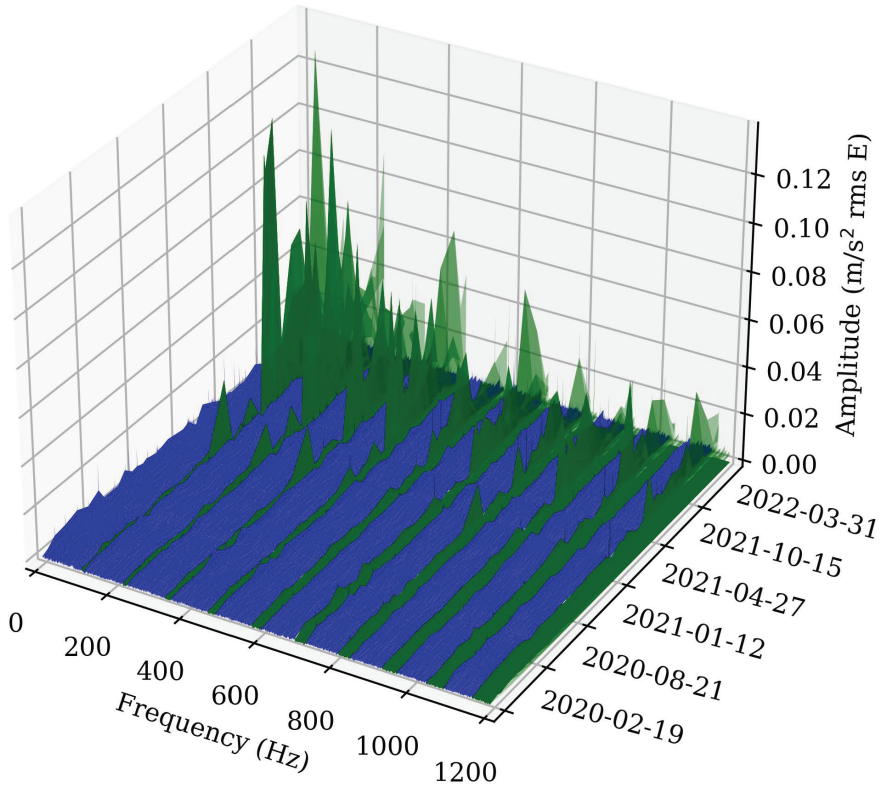

Figure 13. Evolution of the demodulation spectrum over the time of the NN 3016 KTN/SP bearing. In green, the bands related to the SFF for the BPFO. The amplitudes of the spectrum become higher progressively, indicating that the bearing is deteriorating.

4.3.4. Algorithm Tuning and Anomaly Detection

The parameters of the anomaly detection algorithm must be configured to reliably detect anomalies as early as possible. These values have been determined heuristically and are summarized in Table 3.

Table 3. Parameters for anomaly detection algorithm.

cs	dt	bs	N_{min}	af	csa
1	0.5	6	4	0.2	4

The parameters cs and csa depend on the desired sensibility. No specific rule of thumb was followed, except for being satisfied with the level of notifications obtained. The parameter $af = 0.2$ tells that if the average of the last values is more than 20% higher than the global average, a change in scenario will be notified; $N_{min} = 4$ means no notifications before four executions of the FRs. A FR is performed every two weeks, so it took approximately two months before the new values could be evaluated using the anomaly detection algorithm. Machine components are generally expected to have a much longer life, so waiting during that time without making any inferences using the proposed algorithm is considered an acceptable risk.

The evolution of the severity values over time is shown in Figure 14. In the beginning, no notifications are issued, as no anomalous events are observed. The abnormally low values correspond to problems with the measurement system. Then, in the end of 2021, abnormally high severity values start to occur. This was because of an incipient problem in the outer ring; thus, it was observed in the demodulation signal. A notification is sent to manufacturers at the time that anomalies occur. As can be seen, the values continue increasing, and more notifications and changes take place. Due to these alarms, the manufacturers were able to perform maintenance works. Therefore, this algorithm enabled the prevention of more significant damage to the rolling bearing. After detecting a few anomalies, the algorithm automatically considers a change of scenario. After the detection of some anomalies, the algorithm establishes a new level of normality, so no new notifications are issued. In reality, the start of the new normality scenario corresponds to the moment when maintenance work was performed. This automation spares the maintenance personnel from having to manually restart the algorithm.

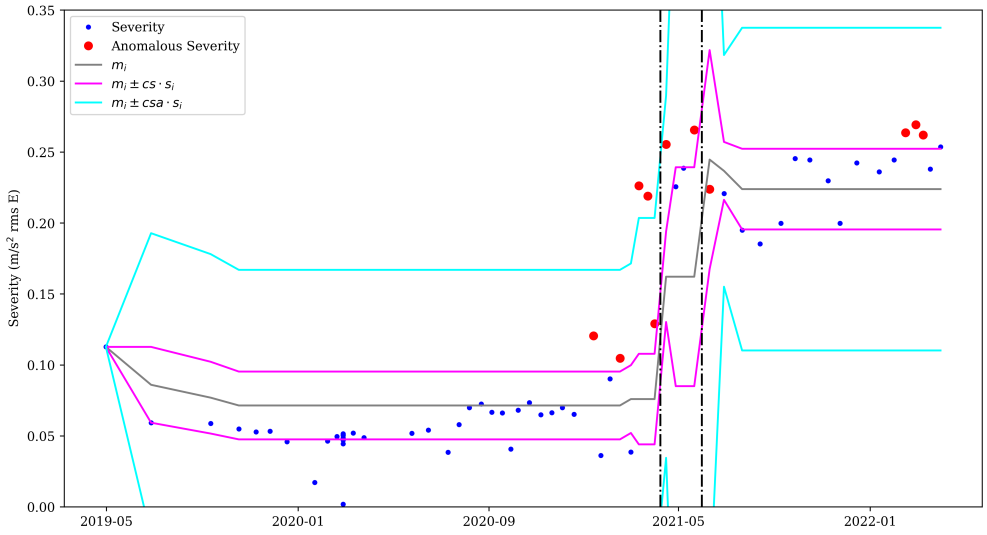

Figure 14. Re-stabilization of anomaly detection algorithm. The green dotted lines mark changes in scenario.

5. Conclusions

This work introduces an end-to-end methodology, from data gathering to fault notification, for the predictive maintenance of rotary components of machine tools. This is achieved through FRs; that is, processes that are executed periodically under the same no-load conditions to obtain a snapshot of the machine condition. It addresses common challenges such as data variability, limited historical datasets, and dynamic operational conditions. By integrating domain-specific knowledge and dynamic anomaly detection, it bridges the gap between theoretical approaches and real-world industrial needs. The proposed methodology proves to be highly versatile, demonstrating its effectiveness in various scenarios for detecting both concept drifts and anomalies. Moreover, when combined with the extraction of key features related to the machine's health status, it becomes an invaluable tool for monitoring and diagnosing industrial systems.

This work opens new lines of work:

- In this work, we assume the kinematic chains to be linear. Using data extraction to solve cases with more complex chains is an open line of work.
- Within the field of production engineering, the proposed methodology could be applied using data gathered from repetitive manufacturing processes [29].
- It is essential to have more public datasets available in order to conduct effective benchmarking studies.
- Developing advanced techniques that integrate anomaly detection and concept drift detection is crucial.

Author Contributions: A.A.: conceptualization, methodology, software, visualization, writing—original draft. A.B.: conceptualization, supervision, writing—review and editing. I.B.: conceptualization, supervision, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the company DANOBATGROUP S.COOP., in the context of the DDMS project (Data-Driven Maintenance Service). This project was granted the SMART-EUREKA label and funded by the INNNOGLOBAL call by the Spanish Ministry of Science through its Centro para el Desarrollo Tecnológico Industrial (CDTI), grant number INNO-20182039.

Data Availability Statement: The datasets used in this work originate from different sources, which are detailed in each corresponding section of the manuscript. The dataset presented in Section 4.1 is artificially generated to illustrate the proposed methodology under controlled conditions. The dataset used in Section 4.2 is publicly available and provided by the NASA Prognostics Data Repository, ensuring reproducibility [27]. Finally, the dataset in Section 4.3 is based on private industrial data collected from real-world operations, used to validate the methodology in a practical setting.

Conflicts of Interest: 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.

References

1. Zonta, T.; da Costa, C.A.; da Rosa Righi, R.; de Lima, M.J.; da Trindade, E.S.; Li, G.P. Predictive maintenance in the Industry 4.0: A systematic literature review. *Comput. Ind. Eng.* **2020**, *150*, 106889. [CrossRef]
2. Vollert, S.; Atzmueller, M.; Theissler, A. Interpretable Machine Learning: A brief survey from the predictive maintenance perspective. In Proceedings of the 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 7–10 September 2021; pp. 1–8. [CrossRef]
3. Serradilla, O.; Zugasti, E.; Rodriguez, J.; Zurutuza, U. Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. *Appl. Intell.* **2022**, *52*, 10934–10964. [CrossRef]
4. Wen, Y.; Rahman, M.F.; Xu, H.; Tseng, T.L.B. Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. *Measurement* **2022**, *187*, 110276. [CrossRef]

5. Corporation, M. Introduction to Predictive Maintenance in Manufacturing—Azure Architecture Center. Available online: https://learn.microsoft.com/en-us/azure/architecture/ (accessed on 23 December 2024).
6. Scaife, A.D. Improve predictive maintenance through the application of artificial intelligence: A systematic review. *Results Eng.* **2023**, *21*, 101645. [CrossRef]
7. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 387–395. [CrossRef]
8. Munir, M.; Siddiqui, S.A.; Dengel, A.; Ahmed, S. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* **2018**, *7*, 1991–2005. [CrossRef]
9. Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; Chawla, N.V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1409–1416. [CrossRef]
10. Moens, P.; Vanden Hautte, S.; De Paepe, D.; Steenwinckel, B.; Verstichel, S.; Vandekerckhove, S.; Ongenae, F.; Van Hoecke, S. Event-Driven Dashboarding and Feedback for Improved Event Detection in Predictive Maintenance Applications. *Appl. Sci.* **2021**, *11*, 10371. [CrossRef]
11. da Silva Arantes, J.; da Silva Arantes, M.; Fröhlich, H.B.; Siret, L.; Bonnard, R. A novel unsupervised method for anomaly detection in time series based on statistical features for industrial predictive maintenance. *Int. J. Data Sci. Anal.* **2021**, *12*, 383–404. [CrossRef]
12. Giannoulidis, A.; Gounaris, A.; Nikolaidis, N.; Naskos, A.; Caljouw, D. Investigating thresholding techniques in a real predictive maintenance scenario. *ACM Sigkdd Explor. Newsl.* **2022**, *24*, 86–95. [CrossRef]
13. Rögnvaldsson, T.; Nowaczyk, S.; Byttner, S.; Prytz, R.; Svensson, M. Self-monitoring for maintenance of vehicle fleets. *Data Min. Knowl. Discov.* **2018**, *32*, 344–384. [CrossRef]
14. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. [CrossRef]
15. Giada, C.V.; Rossella, P. Barriers to Predictive Maintenance implementation in the Italian machinery industry. *IFAC-PapersOnLine* **2021**, *54*, 1266–1271. [CrossRef]
16. Truong, C.; Oudre, L.; Vayatis, N. Selective review of offline change point detection methods. *Signal Process.* **2020**, *167*, 107299. [CrossRef]
17. Fogliazza, G.; Arvedi, C.; Spoto, C.; Trappa, L.; Garghetti, F.; Grasso, M.; Colosimo, B.M. Fingerprint analysis for machine tool health condition monitoring. *IFAC-PapersOnLine* **2021**, *54*, 1212–1217. [CrossRef]
18. D'Amato, E.; Rissone, P. Using the envelope method to monitor rolling bearings. In Proceedings of the 1st International Machinery Monitoring and Diagnostic Conference, Las Vegas, NV, USA, 11–14 September 1998; pp. 560–566.
19. Kallappa, P.; Byington, C.S.; Kalgren, P.W.; DeChristopher, M.; Amin, S. High Frequency Incipient Fault Detection for Engine Bearing Components. In Proceedings of the Turbo Expo: Power for Land, Sea, and Air, Reno, NV, USA, 6–9 June 2005; Volume 47276: Turbo Expo 2005, pp. 413–427. [CrossRef]
20. Bediaga, I.; Mendizabal, X.; Arnaiz, A.; Munoa, J. Ball bearing damage detection using traditional signal processing algorithms. *IEEE Instrum. Meas. Mag.* **2013**, *16*, 20–25. [CrossRef]
21. Bringham, E.O. *Fast Fourier Transform and Its Applications*; Prentice Hall: Hoboken, NJ, USA, 1988.
22. Pacheco-Chérrez, J.; Fortoul-Díaz, J.A.; Cortés-Santacruz, F.; Aloso-Valerdi, L.M.; Ibarra-Zarate, D.I. Bearing fault detection with vibration and acoustic signals: Comparison among different machine leaning classification methods. *Eng. Fail. Anal.* **2022**, *139*, 106515. [CrossRef]
23. SKF, A. NN 3030 K/SPW33—Cylindrical Roller Bearings, Double Row, Super-Precision | SKF. 2022. Available online: https://www.skf.com/sg/industries/machine-tools/products-and-solutions/super-precision-double-row-cylindrical-bearings (accessed on 23 December 2024).
24. Hernandez, M.J. *Database Design for Mere Mortals: A Hands-on Guide to Relational Database Design*; Addision-Wesley: Reading, MA, USA, 1997.
25. Angelov, P.P. Anomaly detection based on eccentricity analysis. In Proceedings of the 2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS), Orlando, FL, USA, 9–12 December 2014; pp. 1–8.
26. Gama, J. *Knowledge Discovery from Data Streams*; Chapman and Hall/CRC: London, UK, 2010.
27. Qiu, H.; Lee, J.; Lin, J.; Yu, G. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J. Sound Vib.* **2006**, *289*, 1066–1090. [CrossRef]

28. *ISO 17243-1:2014*; ISO Central Secretary. Machine Tool Spindles—Evaluation of Machine Tool Spindle Vibrations by Measurements on Spindle Housing—Part 1: Spindles with Rolling Element Bearings and Integral Drives Operating at Speeds Between 600 min^{-1} and 30000 min^{-1}. Technical Report; International Organization for Standardization: Geneva, Switzerland, 2014.
29. Arregi, A.; Inza, I.; Bediaga, I. Vibration analysis for rotatory elements wear detection in paper mill machine. In Proceedings of the International Workshop on Applied Research, Technology Transfer and Knowledge Exchange in Software and Data Sciences, Vienna, Austria, 22–24 August 2022.