

**FEDERAL UNIVERSITY OF ITAJUBÁ  
MECHANICAL ENGINEERING INSTITUTE**

# **Machine Learning-Based Fault Detection and Diagnosis in Electric Motors**

**Ronny Francis Ribeiro Junior**

Itajubá – MG  
2021

**FEDERAL UNIVERSITY OF ITAJUBÁ  
MECHANICAL ENGINEERING INSTITUTE**

# **Machine Learning-Based Fault Detection and Diagnosis in Electric Motors**

**Ronny Francis Ribeiro Junior**

A dissertation presented to the Post-graduate Program of Mechanical Engineering, from Mechanical Engineering Institute of the Federal University of Itajubá, as a requirement to obtain the title of Master in Mechanical Engineering

Advisor: Guilherme Ferreira Gomes

# Acknowledgements

I thank my parents, Ronny e Sandra, for all support and encouragement. Thanks for all the education and affection. I also thank all my family members for presence, encouragement and trust placed in me.

My dissertation advisor, Guilherme Ferreira Gomes, for guidance and teachings that resulted in this work. I also thank all professors and staff from UNIFEI for contributing to my academic education.

I would like to thank all my friends from PS Soluções for providing all the necessary equipment, in addition to helping with valuable knowledge. I also thank all my friends from Instituto Gnarus for the shared knowledge and for the moments I've had while working with you.

Finally, to God, who lights my way, putting people and opportunities in my life.

*“I have no special talent. I am only passionately curious.”*

*— Albert Einstein*

# Abstract

Fault diagnosis is critical to any maintenance industry, as early fault detection can prevent catastrophic failures as well as a waste of time and money. In view of these objectives, vibration analysis in the frequency domain is a mature technique. Although well established, traditional methods involve a high cost of time and people to identify failures, causing machine learning methods to grow in recent years. The Machine learning (ML) methods can be divided into two large learning groups: supervised and unsupervised, with the main difference between them being whether the dataset is labeled or not. This study presents a total of four different methods for fault detection and diagnosis. The frequency analysis of the vibration signal was the first approach employed. This analysis was chosen to validate the future results of the ML methods. The Gaussian Mixture model (GMM) was employed for the unsupervised technique. A GMM is a probabilistic model in which all data points are assumed to be generated by a finite number of Gaussian distributions with unknown parameters. For supervised learning, the Convolution neural network (CNN) was used. CNNs are feedforward networks that were inspired by biological pattern recognition processes. All methods were tested through a series of experiments with real electric motors. Results showed that all methods can detect and classify the motors in several induced operation conditions: healthy, unbalanced, mechanical looseness, misalignment, bent shaft, broken bar, and bearing fault condition. Although all approaches are able to identify the fault, each technique has benefits and limitations that make them better for certain types of applications, therefore, a comparison is also made between the methods.

**Keywords:** Vibration. Fault Diagnosis. Frequency Analysis. Machine Learning. Gaussian Mixture Model. Convolution Neural Network. Electric motor

# Resumo

O diagnóstico de falhas é fundamental para qualquer indústria de manutenção, a detecção precoce de falhas pode evitar falhas catastróficas, bem como perda de tempo e dinheiro. Tendo em vista esses objetivos, a análise de vibração através do domínio da frequência é uma técnica madura. Embora bem estabelecidos, os métodos tradicionais envolvem um alto custo de tempo e pessoas para identificar falhas, fazendo com que os métodos de aprendizado de máquina cresçam nos últimos anos. Os métodos de *Machine learning* (ML) podem ser divididos em dois grandes grupos de aprendizagem: supervisionado e não supervisionado, sendo a principal diferença entre eles é o conjunto de dados que está rotulado ou não. Este estudo apresenta um total de quatro métodos diferentes para detecção e diagnóstico de falhas. A análise da frequência do sinal de vibração foi a primeira abordagem empregada. foi escolhida para validar os resultados futuros dos métodos de ML. O *Gaussian Mixture Model* (GMM) foi empregado para a técnica não supervisionada. O GMM é um modelo probabilístico em que todos os pontos de dados são considerados gerados por um número finito de distribuições gaussianas com parâmetros desconhecidos. Para a aprendizagem supervisionada, foi utilizada a *Convolutional Neural Network* (CNN). CNNs são redes *feedforward* que foram inspiradas por processos de reconhecimento de padrões biológicos. Todos os métodos foram testados por meio de uma série de experimentos com motores elétricos reais. Os resultados mostraram que todos os métodos podem detectar e classificar os motores em várias condições de operação induzida: íntegra, desequilibrado, folga mecânica, desalinhamento, eixo empenado, barra quebrada e condição de falha do rolamento. Embora todas as abordagens sejam capazes de identificar a falha, cada técnica tem benefícios e limitações que as tornam melhores para certos tipos de aplicações, por isso, também é feita uma comparação entre os métodos.

**Palavras-chave:** Vibração. Diagnóstico de Falhas. Análise de Frequência. *Machine Learning*. *Gaussian Mixture Model*. *Convolution Neural Network*. Motor Elétrico

# List of Figures

Figure 1.1 – Machine learning methods. ....	2
Figure 2.1 - Broken bar pattern. Adapted from [34]. ....	7
Figure 2.2 - Experimental setup. ....	8
Figure 2.3 - Induced Fault conditions: (a) bent shaft, (b) broken bar, (c) misalignment, (d) mechanical looseness, (e) bearing fault and (f) Unbalanced. ....	10
Figure 2.4 - (a) IMI uniaxial accelerometer, threaded base for accelerometer fixation: (b) accelerometer 1, (c) accelerometer 2. ....	11
Figure 2.5 - Experimental methodology. ....	12
Figure 2.6 - Signals in time: (a) baseline, (b) unbalanced (1 bolt), (c) unbalanced (2 bolts), (d) mechanical looseness, (e) misalignment, (f) bent shaft, (g) broken bar, (h) bearing fault. ....	13
Figure 2.7 - Motor frequency spectrum with unbalance (1 bolt): (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — unbalanced load (1 bolt) and — baseline). ....	14
Figure 2.8 - Motor frequency spectrum with unbalance (2 bolts): (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — unbalanced load (2 bolts) and — baseline). ....	14
Figure 2.9 - Motor frequency spectrum with mechanical looseness: (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — mechanical looseness and — baseline). ....	14
Figure 2.10 - Motor frequency spectrum with misalignment: (a) Full spectrum (b) Zoom in fault characteristic region (legend: — misalignment and — baseline). ....	15
Figure 2.11 - Motor frequency spectrum with bent shaft: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — bent shaft and — baseline). ....	15
Figure 2.12 - Motor frequency spectrum with broken bar: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — broken bar and — baseline). ....	15
Figure 2.13 - Motor frequency spectrum with bearing fault: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — bearing fault and — baseline). ....	15
Figure 3.1 - General flowchart of the motor operation condition clustering using GMM. ....	22

Figure 3.2 - Vibrations signal for the six different fault conditions considering the vibration accelerometer #1. ....	23
Figure 3.3 - Vibrations signal for the six different fault conditions considering the vibration accelerometer #2. ....	24
Figure 3.4 - Scatter vibration signal for the six different fault conditions considering the vibration considering both the accelerometers #1 and #2. ....	26
Figure 3.5 - Gaussian mixture model results (RSM vs. Skewness) for the Bent and Normal conditions. ....	27
Figure 3.6 - Gaussian mixture model results (RSM vs. Skewness) for the Broken bar and Normal conditions. ....	28
Figure 3.7 - Gaussian mixture model results (RSM vs. Skewness) for the Misalignment and Normal conditions. ....	28
Figure 3.8 - Gaussian mixture model results (RSM vs. Skewness) for the Mechanical loss and Normal conditions. ....	29
Figure 3.9 - Gaussian mixture model results (RSM vs. Skewness) for the Bearing and Normal conditions. ....	29
Figure 3.10 - Gaussian mixture model results (RSM vs. Skewness) for the Unbalanced and Normal conditions. ....	30
Figure 3.11 - Mahalanobis distance for the different induced operational conditions... 31	31
Figure 3.12 - Mahalanobis distance for all the evaluated and induced operational conditions. ....	32
Figure 4.1 - Convolutional layer filter.....	36
Figure 4.2 - Max pooling function.....	36
Figure 4.3 - ANN: (a) Before dropout (b) After dropout. ....	37
Figure 4.4 - Data processing and transformation. ....	39
Figure 4.5 - STFT: (a) Normal, (b) Unbalanced, (c) Mechanical loss, (d) Misalignment, (e) Bent shaft, (f) Broken bar, (g) Bearing fault.....	40
Figure 4.6 - Convolution neural network model. ....	40
Figure 4.7 - (a): Accuracy under different learning rate, (b): Loss function under different learning rate (legend: — Test and — Train). ....	42
Figure 4.8 - (a): Accuracy under different batch-size, (b): Loss function under different batch-size (Legend: — Test and — Train).....	42
Figure 4.9 - Accuracy × Epochs (legend: — Test and — Train). ....	44
Figure 4.10 - Loss × Epochs (legend: — Test and — Train).....	44

Figure 4.11 - First CNN output layer for the different fault conditions. ....	46
Figure 4.12 - Prediction confusion matrix. ....	47
Figure 5.1 - Comparison between 1D and 2D CNNs filters. ....	50
Figure 5.2 - Proposed method flowchart. ....	53
Figure 5.3 - Accuracy score under different number of filters. ....	55
Figure 5.4 - Accuracy score under different learning rates. ....	56
Figure 5.5 - Accuracy score under different batch sizes. ....	56
Figure 5.6 - Classification report. ....	57
Figure 5.7 - Training and validation loss (legend: — Test and — Train).....	58
Figure 5.8 - Training and validation loss (legend: — Test and — Train).....	59
Figure 5.9 - (a) Training input layer, (b) Training output layer, (c) Test input layer, (d) Test output layer. ....	60
Figure 6.1 - Strengths and weaknesses of the methods. ....	63

# List of Tables

Table 2.1 - Marathon motor nameplate. ....	9
Table 4.1 - Training and test samples for each motor state. ....	39
Table 4.2 - Runs to select the best network parameters.....	42

# List of Abbreviations and acronyms

<i>ANN</i>	Artificial Neural Network
<i>ML</i>	Machine Learning
<i>DL</i>	Deep Learning
<i>AI</i>	Artificial intelligence
<i>GMM</i>	Gaussian Mixture Model
<i>CNN</i>	Convolutional Neural Network
<i>1D CNN</i>	One-dimensional Convolutional Neural Network
<i>2D CNN</i>	Two-dimensional Convolutional Neural Network
<i>STFT</i>	Short-time Fourier transform
<i>FFT</i>	Fast Fourier Transform
<i>HHT</i>	Hilbert–Huang transform
<i>BPFO</i>	Ball pass frequency in the outer race
<i>BPMI</i>	Ball pass frequency in the inner race
<i>FTF</i>	Fundamental train frequency
<i>BSF</i>	Ball spin frequency
<i>RSF</i>	Roller spin frequency
<i>EM</i>	Expectation-maximization
<i>SRA</i>	Square root of the amplitude
<i>RMS</i>	Root mean square
<i>SGD</i>	Stochastic gradient descent
<i>Relu</i>	Rectified Linear Unit
<i>Avg</i>	Average
<i>Conv</i>	Convolutional layer
<i>FC</i>	Fully connected
<i>TP</i>	True positive
<i>TN</i>	True negative
<i>FP</i>	False positive
<i>FN</i>	False negative
<i>t-sne</i>	t-distributed stochastic neighbor embedding
<i>CBM</i>	Condition-based monitoring



# Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Research Objective .....</b>	<b>3</b>
<b>1.2 Dissertation Outline.....</b>	<b>3</b>
<b>2. FAULT DETECTION AND DIAGNOSIS USING VIBRATION SIGNAL ANALYSIS IN FREQUENCY DOMAIN.....</b>	<b>5</b>
<b>2.1 Theoretical Background .....</b>	<b>5</b>
<b>2.2 Experimental Methodology .....</b>	<b>7</b>
2.2.1 Experimental setup .....	8
2.2.2 Vibration signal acquisition.....	11
2.2.3 Filter and window .....	12
<b>2.3 Results and Discussion .....</b>	<b>12</b>
2.3.1 Vibration signal for different fault condition in time domain .....	13
2.3.2 Vibration signal for different fault condition in frequency domain ...	14
<b>2.4 Chapter Conclusion.....</b>	<b>17</b>
<b>3. FAULT DETECTION AND DIAGNOSIS USING GAUSSIAN MIXTURE MODEL.....</b>	<b>19</b>
<b>3.1 Theoretical Background .....</b>	<b>19</b>
<b>3.2 Experimental Methodology .....</b>	<b>20</b>
3.2.1 Procedures of the Proposed Method.....	20
<b>3.3 Results and Discussion .....</b>	<b>22</b>
<b>3.4 Chapter Conclusion.....</b>	<b>33</b>

<b>4. FAULT DETECTION AND DIAGNOSIS USING CONVOLUTIONAL NEURAL NETWORK AND SHORT-TIME FOURIER TRANSFORM .....</b>	<b>34</b>
<b>4.1 Theoretical Background .....</b>	<b>34</b>
4.1.1 Input layer.....	35
4.1.2 Convolutional layer .....	35
4.1.3 Pooling layer.....	36
4.1.4 Fully connected layer and output layer.....	37
4.1.5 Reducing overfitting .....	37
<b>4.2 Experimental Methodology .....</b>	<b>38</b>
4.2.1 Data processing .....	38
4.2.2 CNN model.....	39
<b>4.3 Results and Discussion .....</b>	<b>40</b>
4.3.1 Selection of CNN hyperparameters .....	41
4.3.2 Feature visualization and fault identification .....	45
<b>4.4 Chapter Conclusion.....</b>	<b>47</b>
<b>5. FAULT DETECTION AND DIAGNOSIS USING 1D CONVOLUTIONAL NEURAL NETWORKS .....</b>	<b>49</b>
<b>5.1 Theoretical background .....</b>	<b>49</b>
5.1.1 Comparison between 1D and 2D CNNs.....	49
5.1.2 Multi-head CNNs .....	51
5.1.3 New Tricks of CNNs .....	51
<b>5.2 Experimental Methodology .....</b>	<b>51</b>
5.2.1 Procedures of the proposed method.....	51
<b>5.3 Results and Discussion .....</b>	<b>54</b>
5.3.1 Hyperparameter Tunning.....	54
5.3.2 Feature visualization and fault recognition .....	56
<b>5.4 Conclusion .....</b>	<b>60</b>

<b>6. GENERAL CONCLUSION .....</b>	<b>62</b>
<b>7. PUBLICATIONS.....</b>	<b>64</b>

# 1. Introduction

Electric motors are the most common rotating machines and play an important role in most industries in generating torque. Failures can be attributed to faults in individual dynamic components, such as ball bearings, for example, and/or faulty component installation or setup, in which malfunctioning can be caused by unbalance, misalignment, or loose fixtures, for example. Therefore, these machines must operate safely over time, ensuring productivity and avoiding losses due to unexpected downtime [1, 2]. Thus, identifying machine failures before they break is extremely important for the maintenance sector of any industry. [1,2,3].

Because of all these advantages, failure diagnosis has attracted many recent studies [4–9]. Great advances in science and technology make mechanical parts increasingly sophisticated, automated, intelligent, and interconnected. As a result, more than ever, a fault can result in significant losses at any location, increasing the need for monitoring [8]. Various strategies have been used to diagnose the failure of rotating machinery, like oil debris analysis, electrical characteristics, acoustic detection, vibration signal analysis, and temperature analysis [1,2,3].

In contrast to alternative approaches, vibration signal analysis is a lot more common and appropriate for the detection and identification of the most common motor failures [9]. Even machines in good condition generate vibration, and these vibrations are closely linked to periodic events in the operation of the machine, such as rotating shafts, gear teeth, rotating electric fields, etc. The frequency at which these events repeat generates sparks from the source. Therefore, many diagnostic techniques are based on frequency analysis [1].

A typical intelligent diagnostic method consists of four steps: data collection, feature extraction, feature selection, and feature classification. To thoroughly check the condition of the engine, a monitoring system is used to collect real-time data. Therefore, a large amount of data is retrieved after many hours of running the engine [10]. So, there is a need for diagnostic methods that can efficiently analyze large amounts of data and provide diagnostic results automatically [10].

In this way, machine learning (ML) is one of the main methods for dealing with data-based fault diagnosis. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. ML methods can be either supervised or unsupervised. The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.

We have two major subgroups in the realm of unsupervised methods: clustering and dimensionality reduction approaches. Clustering methods are used to identify groups of similar objects in a multivariate data sets and dimensionality reduction refers to techniques that reduce the number of input variables in a dataset. The application of the Gaussian Mixture Model to detect mechanical component problems stands out among clustering approaches [11-15].

Similarly, there are two broad classes in supervised methods: classification and regression techniques. Classification methods are used to predict class labels, whereas regression methods are used to generate the best-fitting model for the data. Convolutional Neural Networks stand out in classification issues, having been used in various studies [16-21].

Figure 1.1 shows the divisions and subdivisions of the machine learning algorithms discussed in this section.

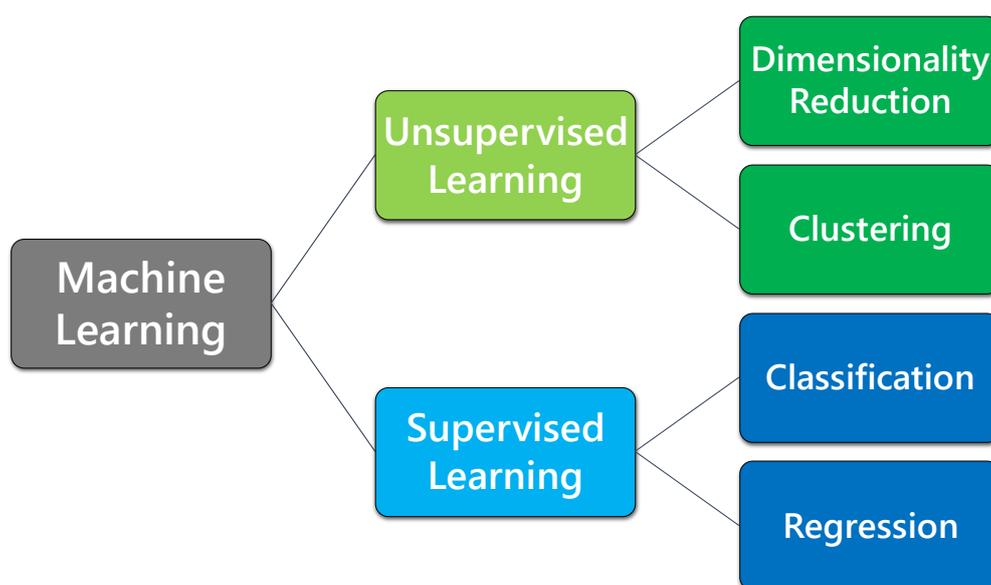


Figure 1.1 – Machine learning methods.

## 1.1 Research Objective

As mentioned in the introduction, electric motors are the main source of power for most industries. Therefore, it's vital to keep the maintenance of those machines up-to-date to ensure production and job security while they are up and running. The objective of this research is to develop and/or apply different signal processing and machine learning techniques to identify and classify faults in electric motors. Because this objective is broad, several studies and analyses can be done. Thus, this research applies a total of four different ways to identify and diagnose electric motor failures:

- Fault Detection and Diagnosis Using Vibration Signal Analysis in Frequency Domain
- Fault Detection and Diagnosis Using Gaussian Mixture Model
- Fault Detection and Diagnosis Using Convolutional Neural Network and Short-Time Fourier Transform
- Fault Detection and Diagnosis Using 1d Convolutional Neural Networks

## 1.2 Dissertation Outline

This dissertation is organized as follows:

- Chapter 2 presents the methodology to analyze the vibration signal of electric motors and diagnose the condition of the motor using time and frequency domain responses. For this purpose, the FFT was used. The results show that it is possible to identify and classify defects using this method.
- Chapter 3 presents a clustering method to identify and diagnose failures in electric motors using the acceleration signal. The Gaussian mixture model was used to accomplish this task. When combined with Mahalanobis distance, the approach has shown potential for identifying and diagnosing faults in electric motors.
- Chapter 4 presents a Convolution Neural Network (CNN) with a time-frequency feature map. Therefore, the short-time Fourier transform (STFT) was employed. The results reveal that this approach is exceptionally accurate for diagnosing and recognizing electrical motor faults.
- Chapter 5 presents a multi-head one-dimension Convolution Neural Network (1D CNN) to detect and diagnose faults. Since it is a multi-head network, the vibration signal from two accelerometers in different positions was used. The results show

that this method for identifying and classifying defects in electric motors is quick and accurate.

- Chapter 6 concludes the dissertation with an overview of the main conclusions, a comparison of all the methodologies presented, and suggestions for further research.

## **2. Fault Detection and Diagnosis Using Vibration Signal Analysis in Frequency Domain**

In this chapter, the development approach for the vibration signal analysis employed in this study is presented. A total of seven different operation conditions were considered (six faults), and a methodology to identify and classify them through the frequency domain is described. All faults have been carefully created to represent as closely as possible how these defects occur over time, and the frequency analysis is done using Fast Fourier Transform (FFT). Results showed that through the vibration analysis using the frequency domain response, it is possible to detect and classify the motors in several induced operation conditions.

The chapter is organized as follows: Section 2.1 provides the theoretical background for signal processing and fault detection based on the frequency spectrum. Section 2.2 presents the experimental methodology. Section 2.3 presents the results obtained through vibration analysis. Finally, in Section 2.4, the chapter conclusions are drawn.

### **2.1 Theoretical Background**

The characteristic components of the machine vibration signal help classify machines as being in a healthy condition or faulty. Vibration signals may appear random in the time domain, but in the frequency domain, they present a characteristic distribution. These signals are classified as stationary when their statistical properties do not change over time and non-stationary when their statistical properties change over time [1,22].

Therefore, classifying the signal type is critical for determining which processing technique can and should be employed. Therefore, the correct classification of the signal helps in the success of the analysis and diagnosis of the same.

In this work, all signals used are stationary, so the Fast Fourier Transform (FFT) will be used. If the signal were non-stationary, it would possibly be necessary to use other transforms such as Wavelet or Hilbert-Huang transform (HHT) [2,23,24,25].

Even if adequate, the FFT input signal is inherently truncated. This truncation is modeled as the multiplication of the signal with a rectangular window function. However, in the spectral domain, this type of windowing causes an effect called spectral leakage [26,27,28]. To mitigate this effect, another type of window was used, the Blackman window. The Blackman window is mathematically defined in Equation (2.1) [26,27,28].

$$\omega(n) = 0.42 - 0.50 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right), \quad n = 0, 1, 2, \dots, N \quad (2.1)$$

A model-based diagnosis needs a way to identify the engine in both its healthy condition and its faulty condition. Identify faults from vibration signals associated with items involving rotational movements such as gears, rotors, shafts, bearings, and couplings, depending on several factors such as: (i) rotation speed, (ii) noise and vibration level, (iii) location of the monitoring transducer, (iv) load characteristic [10]. In a sense, fault detection is the common practice of identifying faults in machines before they become severe enough to cause damage to yourself or the surrounding area. Previous studies have proven that vibration analysis can be used to detect engine failures [1,2,5,22,23,29-35]. This is possible because mechanical failures produce vibration patterns that are specific to faulty systems. Failures such as imbalance, mechanical looseness, misalignment, and bent shaft, for example, can be identified using the rotation frequency and its harmonics. Other defects such as gears, belts, and bearings are detected by the defect's characteristic frequency, which varies according to the geometric characteristics of the element and the operation of the machine.

Bearing failures have known defects [1,23,33,36]. The characteristic Equations for ball pass frequency in the outer race are shown in Equation 2.2. The fault in the inner race is shown in Equation 2.3. Also, the fundamental train frequency is highlighted in Equation 2.4 and finally the ball (roller) spin frequency is shown in Equation 2.5.

$$BPFO = 0.50n\pi f_r \left(1 - \frac{d}{D} \cos \phi\right) \quad (2.2)$$

$$BFPI = 0.50n\pi f_r \left(1 + \frac{d}{D} \cos \phi\right) \quad (2.3)$$

$$FTF = 0.50f_r \left(1 - \frac{d}{D} \cos \phi\right) \quad (2.4)$$

$$BSF(RSF) = \frac{D}{2d} \left(1 - \frac{d}{D} \cos \phi\right)^2 \quad (2.5)$$

where  $f_r$  is the rotation frequency,  $d$  stands for the ball diameter,  $D$  stands for the pitch diameter,  $n$  stands for the number of rolling elements, and  $\phi$  stands for the ball contact angle.

Broken bar failures are also known, following the pattern Figure 2.1[34], where  $f$  is the supply fundamental frequency and  $s$  is the motor slip.

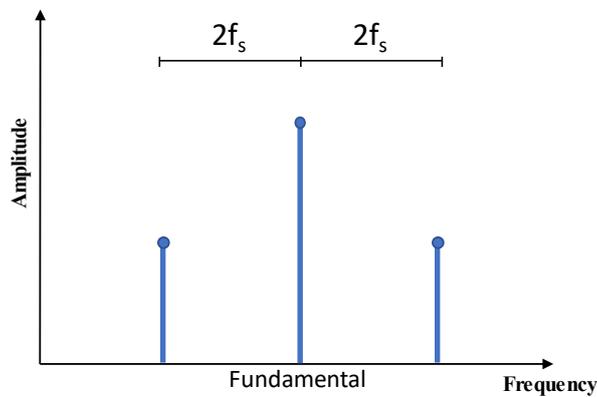


Figure 2.1 - Broken bar pattern. Adapted from [34].

## 2.2 Experimental Methodology

There are two approaches that can be used to create and investigate electric motor faults. The first involves the use of the motors until their failure, where the whole process is monitored and changes in time can be analyzed. Because it is a time-consuming process, failure is usually accelerated. In the second approach, defects are artificially

created, so it is possible to compare them with machines under normal operating conditions. In this study, the second approach is adopted, that is, considering induced failures.

### 2.2.1 Experimental setup

All tests were carried out at PS Soluções in Itajuba (Brazil). The test bench used (Figure 2.2) consisted of a 0.5 HP induction motor directly connected to the SpectraQuest machine failure simulator [37]. Accelerometer 1 measures vibration in the y direction and accelerometer 2 measures vibration in the x direction. Table 2.1 also shows the main characteristics of the motor at 60 Hz and 50 Hz.

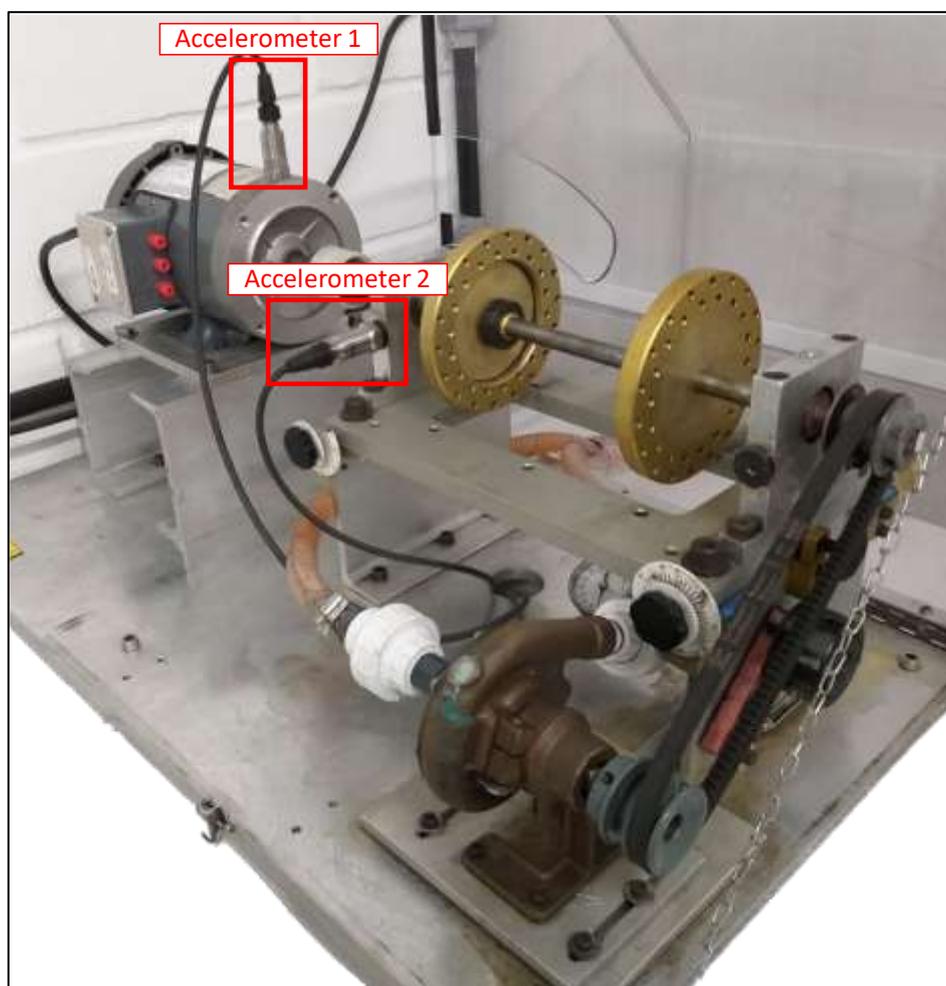


Figure 2.2 - Experimental setup.

Table 2.1 - Marathon motor nameplate.

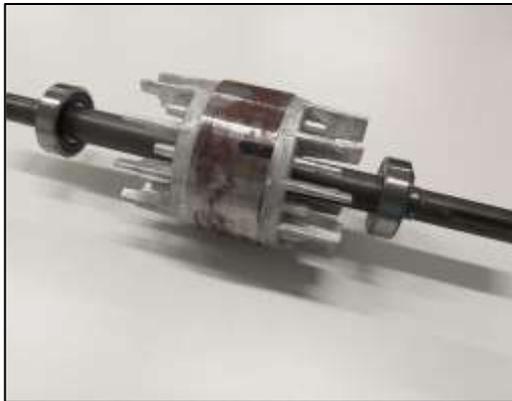
Parameter	60 Hz	50 Hz
Power (HP)	0.50	0.33
Rotation (RPM)	3450	2850
Voltage (V)	208-230/460	190/380
Amperage (A)	2.1-2.2/1.1	2.0/1.0
Service Factor	1.15	1.15

The experimental bench is composed of five electric motors with the possibility to evaluate up to seven operating conditions. These conditions involve normal, bent shaft, broken bar, misalignment, mechanical looseness, bearing fault and unbalanced.

- **Normal** condition is the electric motor in a healthy state, without any fault.
- **Bent shaft** refers to the motor whose shaft is slightly bent (approximately 4 mm), inducing a scratch between the rotor and the stator as shown in Figure 2.3(a). This fault mainly occurs due to overloads and coupling conditions.
- **Broken bar** refers to the motor whose squirrel cage rotor bar is broken. This fault is more common on bigger electric motors. As shown in Figure 2.3(b) this fault is simulated by drilling 6 holes with 2 mm diameter in the rotor.
- **Misalignment** refers to the motor whose shaft is misaligned. This fault occurs due to bad installation or overloads. Figure 2.3(c) shows how this defect is induced. On the bench, it is possible to generate up to 3 mm of misalignment
- **Mechanical looseness** refers to the motor whose screws are loosening. This fault commonly occurs due to bad installation but high vibration may cause this fault. Figure 2.3(d) shows which screws are loosened.
- **Bearing fault** refers to the motor whose bearing is damaged. This is the most common fault in electric motors. Figure 2.3(e) shows the bearing fault in the outer race. The bearing used is SKF-6203.
- **Unbalanced** refers to the motor whose load is unbalanced. This fault is very common in electric motors, being the reason for other faults. Figure 2.3(f) shows how this fault is induced.

These conditions cover the most common faults in electric motors. Several studies [1,5,34,36,38,39] verify that the chosen failures are in fact the most common faults in electric motors. In all, four runs of experiments were performed with all operating

conditions, collecting 28 signals from each accelerometer. For each run, the defects were dismantled and mounted in different motors, in order to better evaluate the defects. this bench will be used for all four techniques that were used in this work.



(a)



(b)



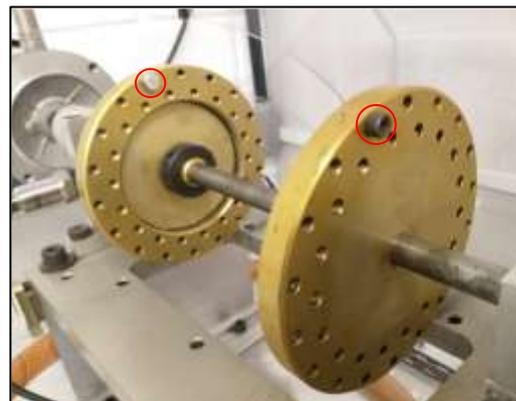
(c)



(d)



(e)



(f)

Figure 2.3 - Induced Fault conditions: (a) bent shaft, (b) broken bar, (c) misalignment, (d) mechanical looseness, (e) bearing fault and (f) Unbalanced.

## 2.2.2 Vibration signal acquisition

The IMI uniaxial accelerometer (Figure 2.4(a)) collects the two vibration signals. The accelerometer has a frequency range of  $\pm 3$  dB, measuring  $\pm 50$  g, the sensitivity of 100 mV/g and resonance at the frequency of 25 kHz. These characteristics make the sensor suitable for vibration measurement, as the frequencies of the collected signals are low and the engine mass is much higher than the mass of the accelerometer. The first accelerometer is attached to the motor via a threaded base (Figure 2.4(b)). The second is also fixed by a threaded base installed in the bearing housing (Figure 2.4(c)). All threaded bases were glued using Araldite. The acquisition system (Preditor®) has a sampling frequency of 11718.5 Hz and an acquisition window of 22.37 seconds, collecting 262144 sample points.



(a)



(b)



(c)

Figure 2.4 - (a) IMI uniaxial accelerometer, threaded base for accelerometer fixation: (b) accelerometer 1, (c) accelerometer 2.

### 2.2.3 Filter and window

Two techniques are applied in order to reduce noise from the signal, smooth the data, and avoid spectral leakage. The first is a low-pass filter at a sampling frequency divided by two. This choice is based on Nyquist-Shannon sampling theorem. The second is the Blackman-Harris window. Beyond what has been mentioned, this window also improves the spectrum contrast.

The entire procedure used for frequency analysis can be viewed globally through Figure 2.5. The experimental procedure is summarized by the acquisition of the vibration signal (acceleration) over a fixed time limit in electric motors operating in different conditions. For this analysis, only the accelerometer 1 signal will be considered. The entire acquisition was performed using the Predator® system, and then a Discrete Fast Fourier Transform was applied to this signal. From the frequency spectrum characteristic of each operating condition of the engine, it is possible to infer its alteration and modification.

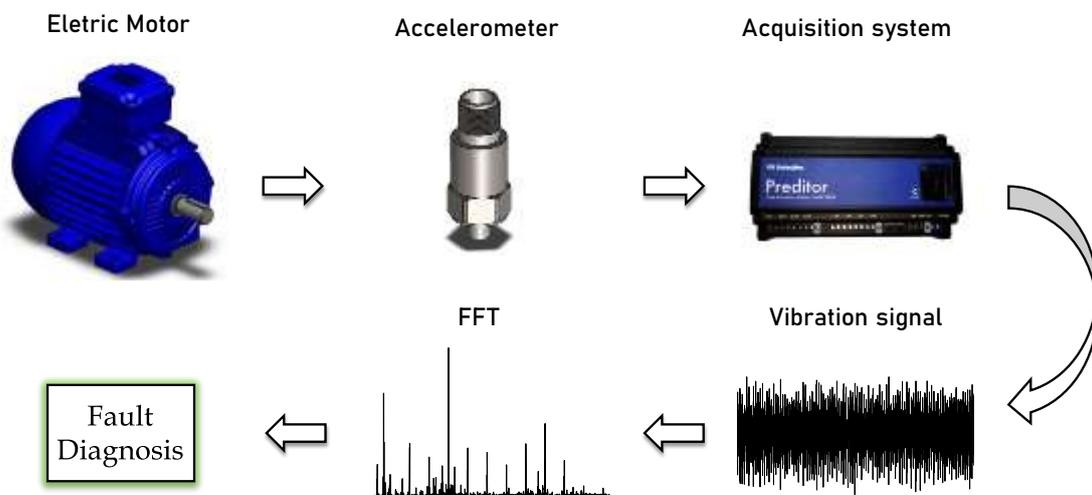


Figure 2.5 - Experimental methodology.

## 2.3 Results and Discussion

In this section, the results obtained through vibration analysis techniques will be discussed. The analysis will be made for all the induced faults, that is, unbalance, mechanical looseness, misalignment, bent shaft, broken bar, and bearing fault.

### 2.3.1 Vibration signal for different fault condition in time domain

Figure 2.6 shows the acceleration signal in the time domain of all the tested operating conditions. The vibration signal in time is very similar for all conditions, except for the bearing fault, which has a very high amplitude compared to other signals. However, only a high vibration amplitude is not conclusive in identifying the defect much less in diagnosing which defect is present. Therefore, with only the signal in time, it is not possible to extract information to detect and differentiate defects.

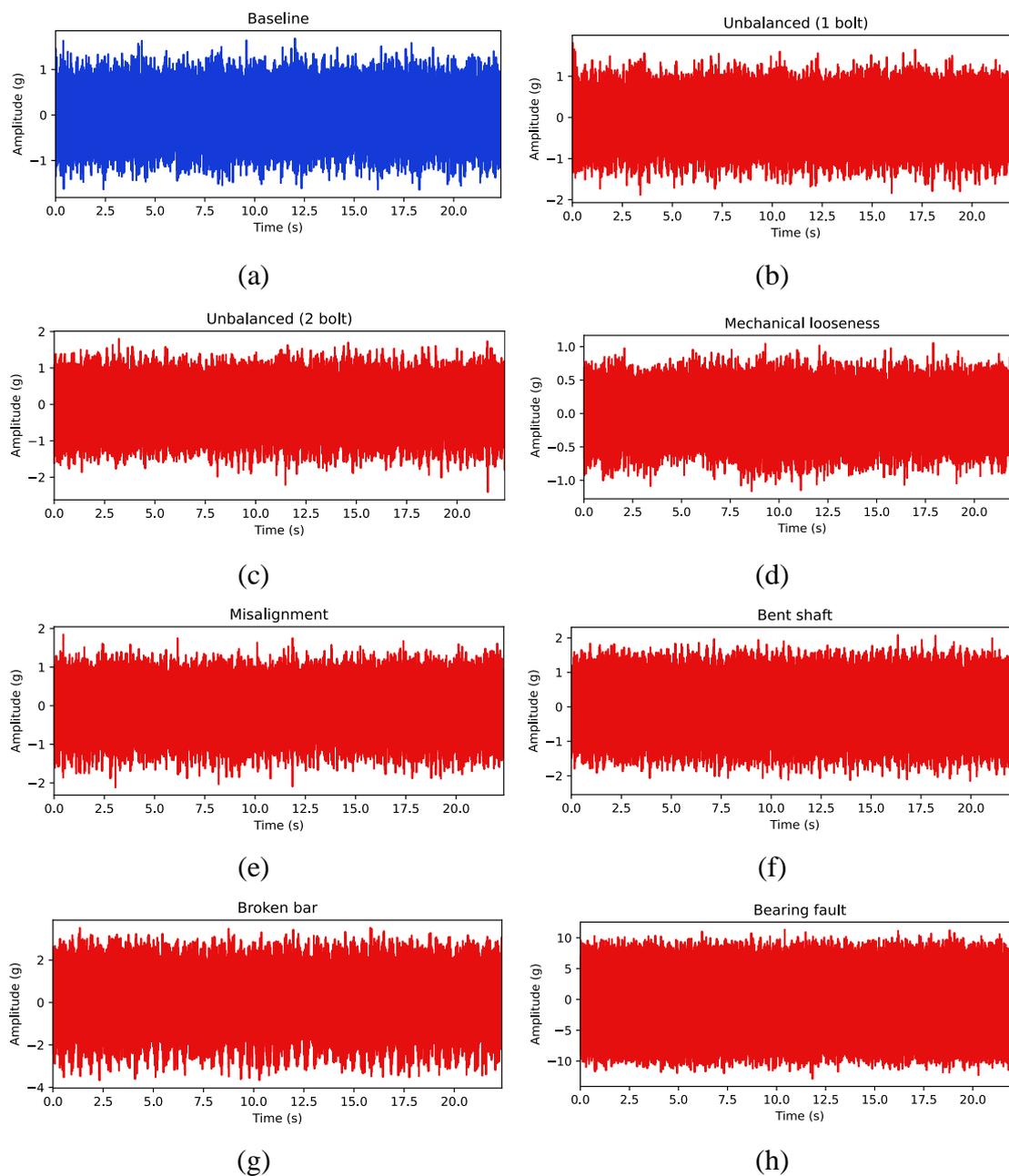


Figure 2.6 - Signals in time: (a) baseline, (b) unbalanced (1 bolt), (c) unbalanced (2 bolts), (d) mechanical looseness, (e) misalignment, (f) bent shaft, (g) broken bar, (h) bearing fault.

### 2.3.2 Vibration signal for different fault condition in frequency domain

To extract more information from the signal, the FFT was used. Figure 2.7 to Figure 2.13 shows the acceleration signal in the frequency domain of all tested operating conditions. In them, fault signals were plotted along with the healthy motor signal to make an easier comparison and identification of the defects. Through the frequency spectrum, it is possible to identify defects and differentiate them. Furthermore, as the severity of the defect increases, the amplitude also increases.

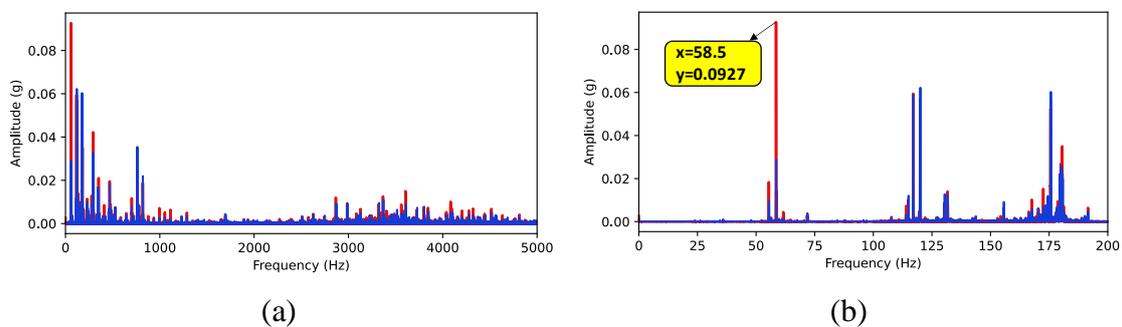


Figure 2.7 - Motor frequency spectrum with unbalance (1 bolt): (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — unbalanced load (1 bolt) and — baseline).

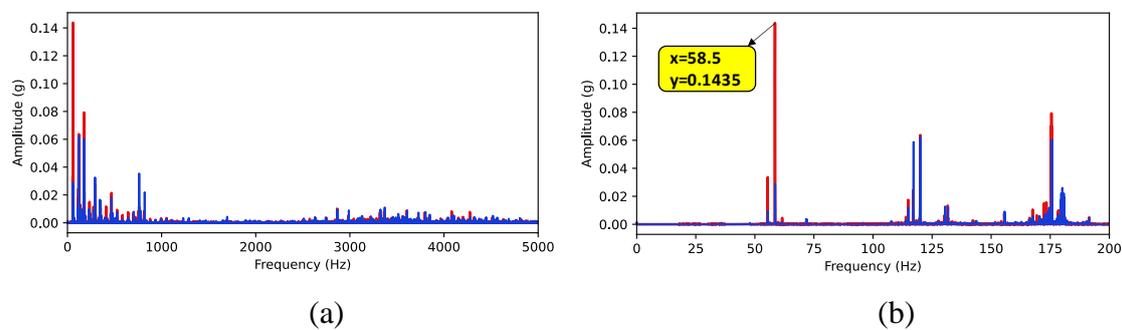


Figure 2.8 - Motor frequency spectrum with unbalance (2 bolts): (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — unbalanced load (2 bolts) and — baseline).

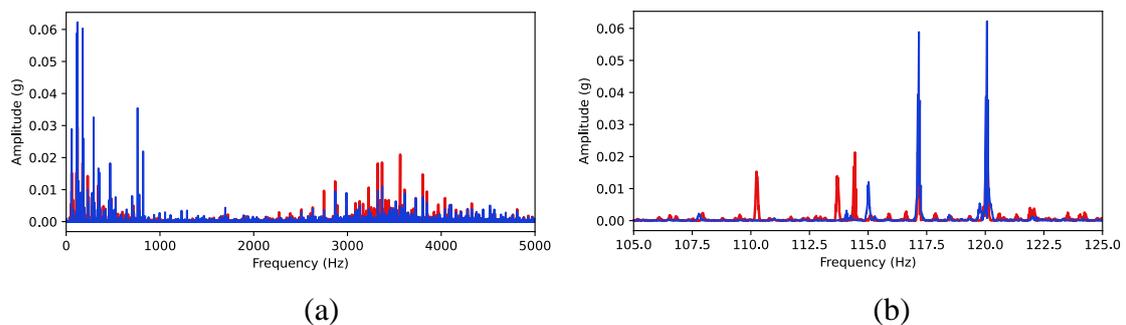


Figure 2.9 - Motor frequency spectrum with mechanical looseness: (a) Full spectrum, (b) Zoom in the fault characteristic region (legend: — mechanical looseness and — baseline).

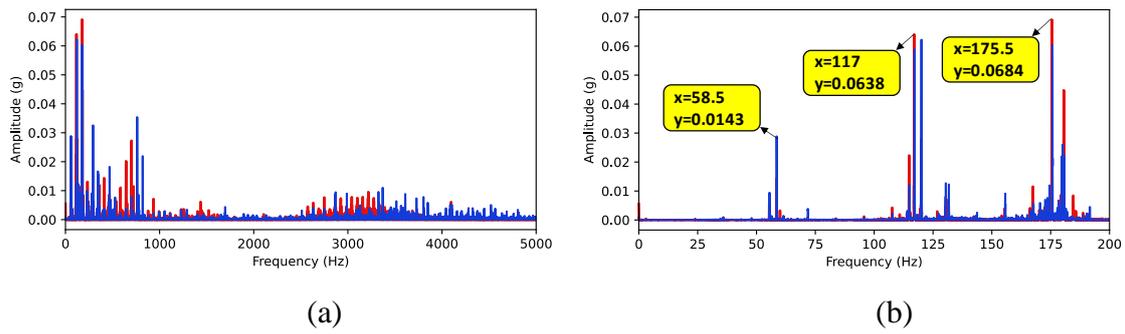


Figure 2.10 - Motor frequency spectrum with misalignment: (a) Full spectrum (b) Zoom in fault characteristic region (legend: — misalignment and — baseline).

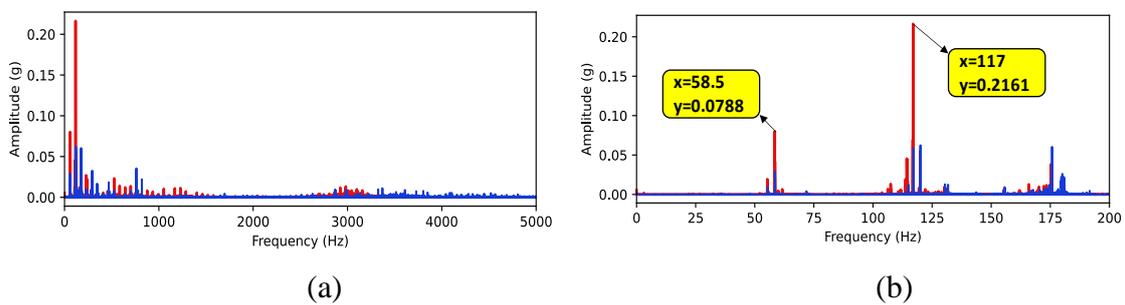


Figure 2.11 - Motor frequency spectrum with bent shaft: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — bent shaft and — baseline).

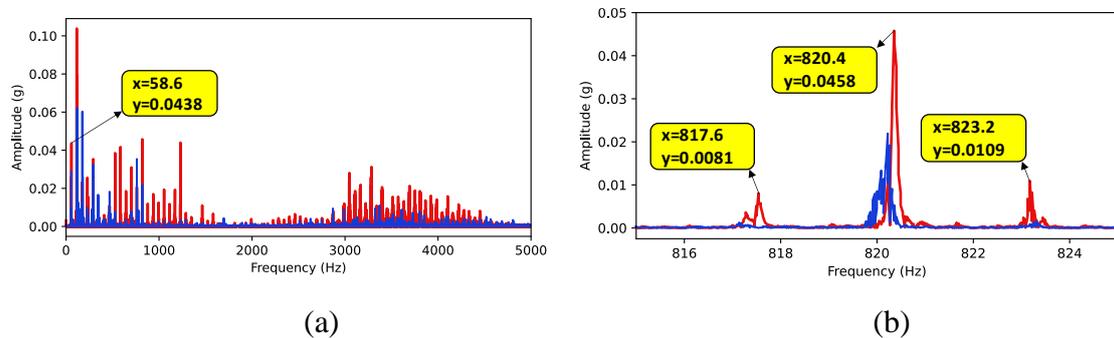


Figure 2.12 - Motor frequency spectrum with broken bar: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — broken bar and — baseline).

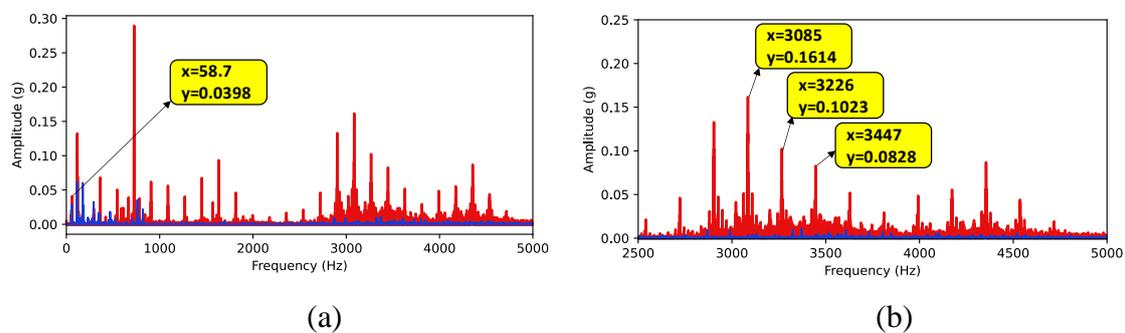


Figure 2.13 - Motor frequency spectrum with bearing fault: (a) Full spectrum (b) Zoom in the fault characteristic region (legend: — bearing fault and — baseline).

Analyzing the low-frequency defects, it is found that the unbalance defects (Figure 2.7 and Figure 2.8) are characterized by an increase only in the rotation frequency. Also, with the increase in severity, there is an increase in amplitude. The mechanical looseness (Figure 2.9) is characterized by the reduction of the amplitude of the rotating components and their harmonics with the appearance of other components along the spectrum, so a low amplitude of vibration is not always a healthy motor. The misalignment (Figure 2.10) is characterized by the increase in the amplitude of the 2nd and 3rd harmonics. As the amplitude increase was small, it is concluded that the misalignment is not severe for the motor. The bent shaft (Figure 2.11) has an increase in the rotation frequency and the 2nd harmonic. As the 2nd harmonic is the dominant frequency of the spectrum, it is known that the shaft has the greatest curvature close to the coupling.

The broken bar defect (Figure 2.12) is characterized by the appearance of components distant 2 times the motor slipping frequency. In the case studied, the defect is around the harmonic of 820.4 Hz. However, the rotation harmonic that the defect is found varies from motors to motors due to their constructive characteristics. Analyzing the points marked it is that the motor is at a rotation of 58.6 Hz, so the frequency of slipping is 1.4 Hz, then, using Equations (2.6) and (2.7) it is possible to verify that the marked points correspond to the broken bar defect.

$$2F_{s,1} = P_2 - P_1 \geq F_{p,1} = \frac{820.4 - 817.6}{2} = 1.40 \quad (2.6)$$

$$2F_{s,2} = P_3 - P_2 \geq F_{p,2} = \frac{823.2 - 820.4}{2} = 1.40 \quad (2.7)$$

Where: P1 = 1st point marked, P2 = 2nd point marked, P3 = 3rd point marked.

The bearing fault (Figure 2.13) is characterized by the motor rotation frequency multiplied by the characteristic frequency defect. In the case studied, the defect is located on the outer race, so it will be RPM  $\times$  BPFO (Ball Pass Frequency Outer) [1,2,13,14]. Consulting the manufacturer's catalog, it is found that the BPFO of the bearing is 3.07 Hz, so this value will be used as a guide to identify the defect. Analyzing the points marked and using Equation (2.2), it is checked that the marked points correspond to the fault defect on the outer track of the bearing.

$$BPFO_1 = \frac{P_2 - P_1}{\Omega} = \frac{3266 - 3085}{58.7} = 3.083 \quad (2.8)$$

$$BPFO_2 = \frac{P_3 - P_2}{\Omega} = \frac{3447 - 3266}{58.7} = 3.083 \quad (2.9)$$

where: P1 = 1st point marked, P2 = 2nd point marked, P3 = 3rd point marked,  $\Omega$  = Motor rotation frequency (Hz).

It is noticed that this pattern is easily identified throughout the high-frequency range. If the defect were in the spheres or anywhere else, the way to identify it would be the same, except that the frequency of the defect would change.

In short, to perform the analysis of the vibration signal in the frequency domain, the main thing to do is to identify the motor rotation frequency. Through the rotation frequency combined with other information such as harmonics, the frequency of the defect, and the motor slipping, it is possible to identify and classify the main defects that can occur in a motor. As these defects follow well-defined patterns, it is sufficient to know the frequency of rotation of the motor and the frequency of its components to replicate this study. several other studies also present similar results to those found in this chapter [1-3,29,30].

## 2.4 Chapter Conclusion

This chapter presented the application of vibration analysis to diagnose motor failures. The use and application of vibration analysis techniques were studied and verified through an experiment.

The use of FFT to identify motor failures has been effectively demonstrated. All the induced faults were identified, and it was possible to differentiate between them. It is possible to identify the motor rotation frequency, so all the analysis can be replicated on any electric motor without major problems.

Besides, the study is relevant to the industry. Through vibration analysis, it is possible to implement condition-based maintenance on the main assets of the industry, which generates a reduction in maintenance costs.

It is noteworthy that although very efficient and practical, vibration analysis encounters problems if the asset is difficult to access, as well as defects that can be confused with each other.

# 3. Fault Detection and Diagnosis

## Using Gaussian Mixture Model

In this chapter, a Gaussian mixture model to cluster, identify, and diagnose six different failures in electric motors using the motor vibration is proposed. Also, the Mahalanobis distance is used to measure how far the failure cluster is from the healthy cluster. The proposed method is verified through a series of experiments with the same bench used in Chapter 2. The results show that the proposed method is a potential method to identify failures in electric motors, especially bearing, unbalanced, and mechanical loss failures.

The chapter is organized as follows: Section 3.1 provides the theoretical background for the Gaussian Mixture Model. Section 3.2 presents the experimental methodology. Section 3.3 presents the results obtained by the Gaussian Mixture Model based on the vibration signals and Mahalanobis distance. Finally, in Section 3.4, the conclusions are drawn.

### 3.1 Theoretical Background

When analyzing the behavior of different phenomena, it is possible to verify that the information about to these phenomena usually presents a normal probability distribution or even presents a mixture of several normal distributions [15]. Thus, strategies such as GMM are used, as they consider several Gaussian probability distributions, contemplating a multidimensional probability density function [40]. Wang et al. (2020) highlight the use of this method for signal modeling and analysis, as it is able to approximate the density distribution in a simple way. In addition, this strategy is widely used in studies aimed at analyzing vibration and identifying damage [11–15, 40, 41].

In this sense, the GMM is used to estimate the probability distribution of samples, in which this estimate is given by the weighted sum of several Gaussian models [40]. Based on the number of Gaussian models (which represent a class), there is a probability that the class with the highest probability is used for decision making. The mathematical representation of the GMM for  $k$  component mixtures can be described according to

Equation (3.1), where  $l = 1, 2, \dots, k$  parameters of the model and that  $l$  is the mixture of weights that respects the restriction of the sum of the weights to be equal to 1.

$$p(\mathbf{x}|\Theta) = \sum_{l=1}^k \alpha_l p_l(x|\theta_l) \quad (3.1)$$

Considering that  $\mathbf{x}$  is a multidimensional vector ( $\mathbf{D}$ ) and that  $\theta_l = (\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ ,  $l = 1, 2, \dots, k$ , we have that the Gaussian distribution can be described according to Equation (3.2) [23], where  $\boldsymbol{\mu}_l$  represents the mean vector and  $\boldsymbol{\Sigma}_l$  represents the variance-covariance matrix.

$$g(\mathbf{x}|\Theta) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_l|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}_l^{-1} (\mathbf{x} - \boldsymbol{\mu}_l) \right\} \quad (3.2)$$

Based on this, it is necessary to use the maximum likelihood strategy to select the GMM parameters. Choosing this technique allows you to define parameters that aim to maximize the probability of the GMM. When dealing with parameters with non-linear structure, the Expectation-Maximization (EM) algorithm must be used to optimize the probability function [40]. From the selected model, there is an iterative process until the best probability result is reached.

## 3.2 Experimental Methodology

The experimental setup used is the same used in sections 2.2.1 and 2.2.2, that is, a bench with several electric motors with six known types of faults. Different from what was done in Section 2, a Gaussian Mixture Model based on the vibration signals is proposed for the diagnosis of faults in electric motors.

### 3.2.1 Procedures of the Proposed Method

The first step is to collect all motors vibration signals under different operating conditions. It was collected four signals for each operating condition, separating 75% for training and 25% for testing. After that, the signals are split into smaller sizes, and each size has 512 sample points. Finally, RMS and Skewness statistics were used to extract features from vibration signals to enhance the results of the classification [58]. Other statistics such as Kurtosis, SRA, Peak-to-Peak, Crest factor, and Shape factor were used, but the best pair to identify faults based on the vibration signal was RMS and Skewness [58].

The main parameter of GMM is the covariance matrix, from which it is possible to choose four different types:

- Full: each component has its own general covariance matrix
- Tied: all components share the same general covariance matrix
- Diag: each component has its own diagonal covariance matrix
- Spherical: each component has its own single variance

As each data set can have a parameter that fits best, all 4 types of covariance will be tested and then the best one will be chosen. Another technique used is starting the algorithm in a supervised way, that is, using the mean of each distribution as the initial weight. This can increase the performance of the method. As the model was started in a supervised manner, the number of mixture components will be set to two. The number of iterations was set to 10,000 with a convergence threshold of  $10^{-6}$ . Finally, the Mahalanobis distance is calculated to assess how far one cluster is from the other. All features used in the proposed method were developed in the Python language using the Scikit-learn library [42]. Figure 3.1 shows the general flowchart of the methodology used in this research.

(Intentionally left blank)

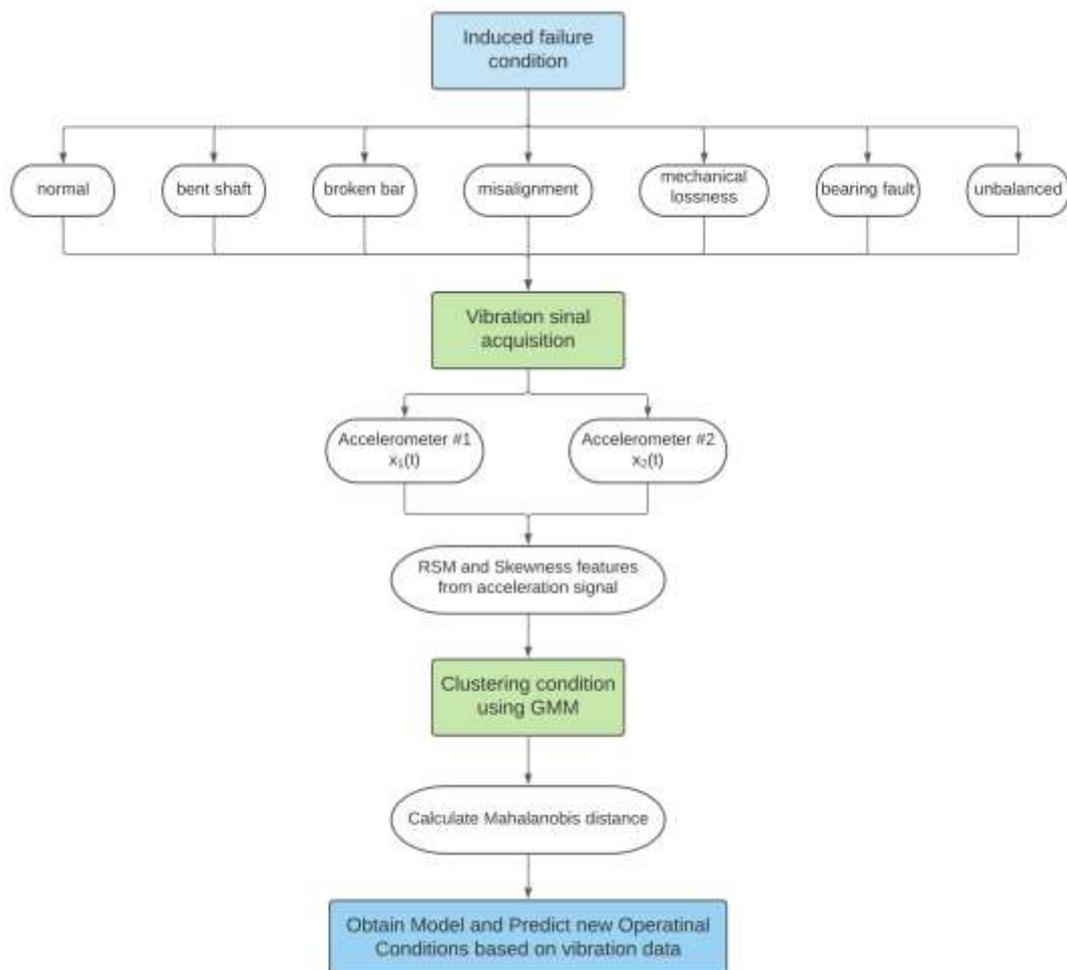


Figure 3.1 - General flowchart of the motor operation condition clustering using GMM.

### 3.3 Results and Discussion

In this section, will be discussed and displayed the main and most relevant results obtained in the analysis of the vibration signal in the seven different operating conditions (six faults) in electric motors. Figure 3.2 displays the time domain vibration signals for the six fault conditions compared to the normal signal for accelerometer 1. Similarly, Figure 3.3 displays the same results considering accelerometer 2. In a way, it is evident that the signal changes under fault conditions, each with a specific pattern. This sign is even more evident for accelerometer 1 and for operating conditions of bearing failure and lack of mechanical looseness. In addition, Figure 3.4 shows the scatter plot data for both the accelerometer signal and For the GMM results, only the accelerometer # 1 is used as input data.

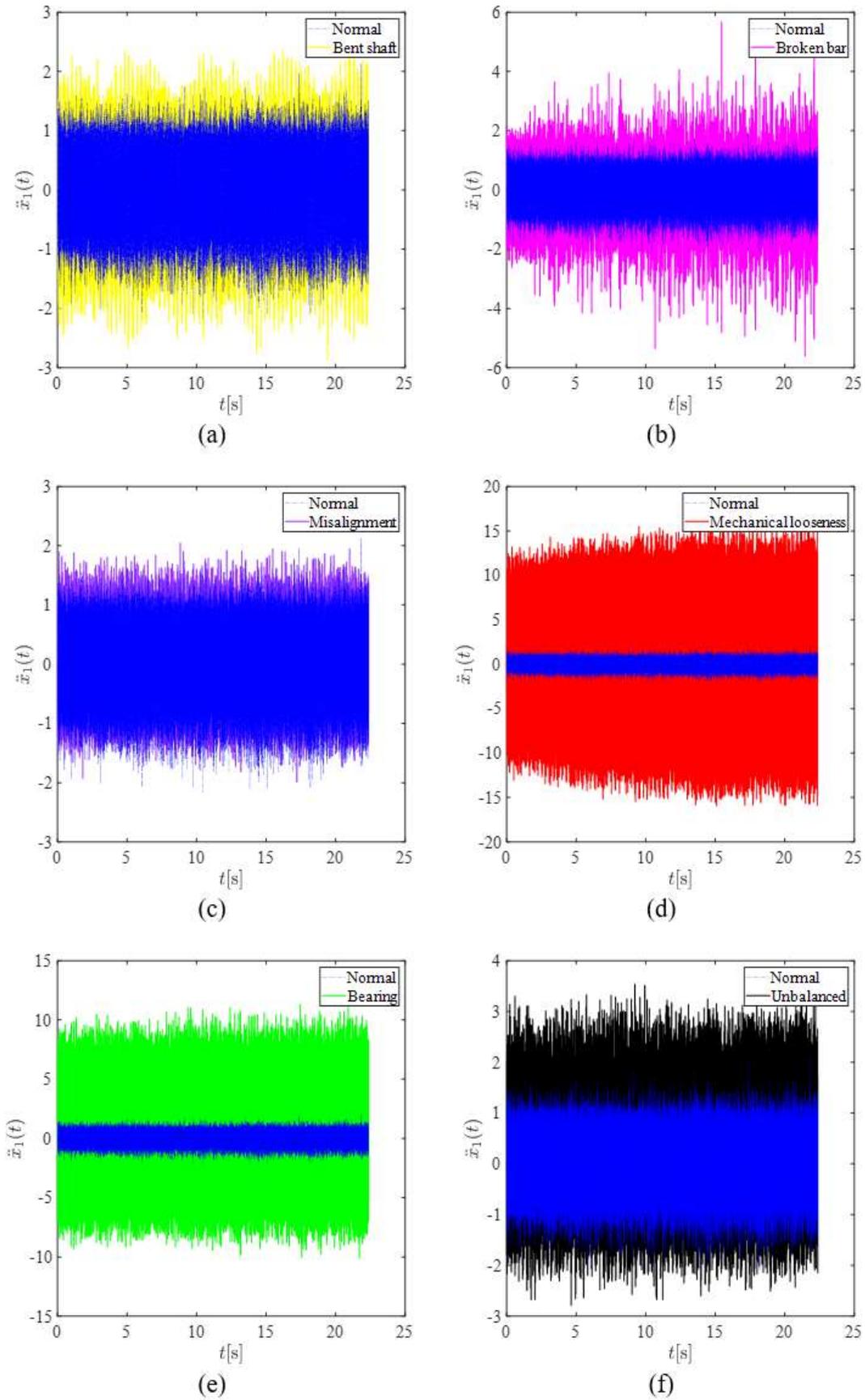


Figure 3.2 - Vibrations signal for the six different fault conditions considering the vibration accelerometer #1.

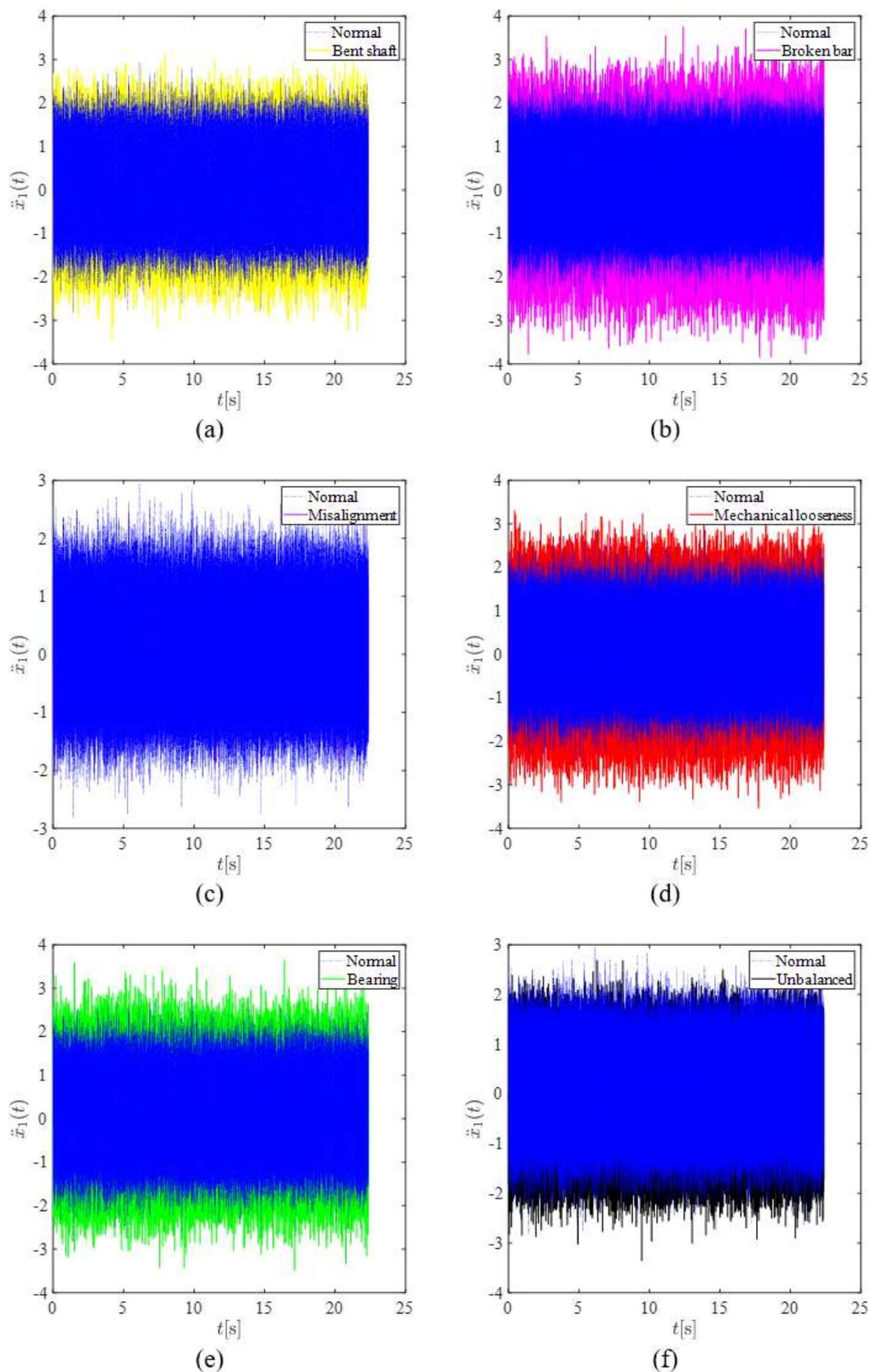


Figure 3.3 - Vibrations signal for the six different fault conditions considering the vibration accelerometer #2.

Visually, it can be seen that accelerometer 1 was more sensitive to changes in operating conditions. For the application of grouping using GMM, this signal will be used. A preprocessing step was performed in order to extract resources from this raw signal. The RSM and Skewness metrics were chosen due to their importance and ability to aggregate information [35]. As already mentioned, Figure 3.4 shows the acceleration data from the two accelerometers. Except for the mechanical loss operating condition (Figure 3.4(d)), where possible nonlinearities exist, a dot pattern is observed in all cases.

(Intentionally left blank)

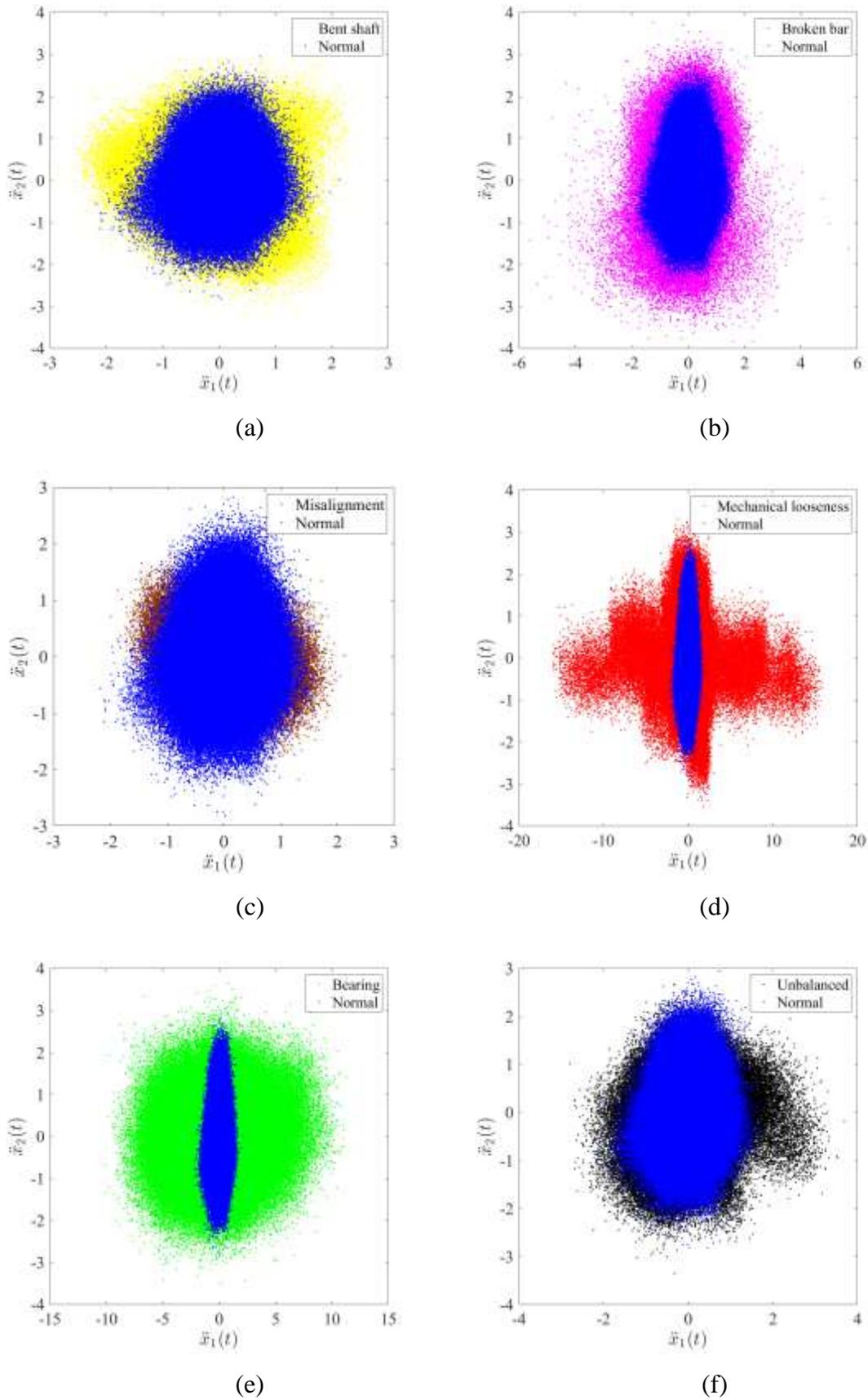


Figure 3.4 - Scatter vibration signal for the six different fault conditions considering the vibration considering both the accelerometers #1 and #2.

After an exploratory phase of vibration data in different operating conditions (induced failures), the second phase of machine learning was carried out. In this phase, an unsupervised learning algorithm was necessary. A Gaussian Mixture Model (GMM) was chosen because it is a flexible algorithm classified as "soft clustering." In other words, a GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

Figure 3.5 through Figure 3.10 show the results of training for the GMM algorithm under all operating conditions. The results display all parameters of the covariance matrix (full, tied, diag, and spherical).

Some clustering results exhibit more segregated (distant) clusters than others. Note that in the case of misalignment, the clusters are closer together. In the cases of bent and broken bars, the clusters are far enough apart yet share a boundary. In the other cases, the groups formed are more segregated. At this point, the use of GMM is justified because it is a soft algorithm.

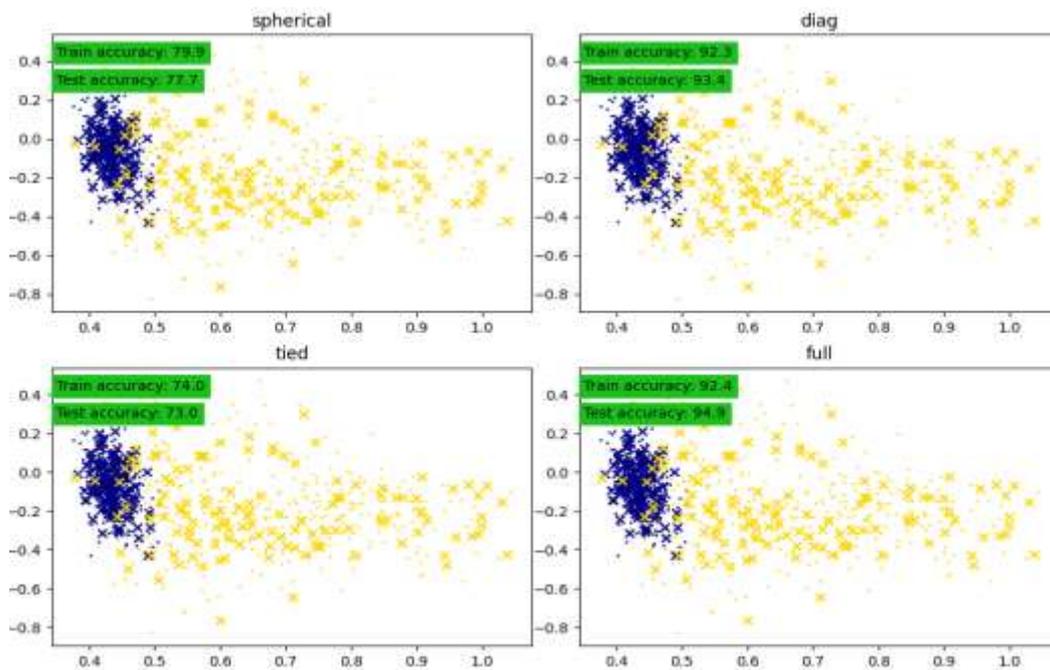


Figure 3.5 - Gaussian mixture model results (RSM vs. Skewness) for the Bent and Normal conditions.

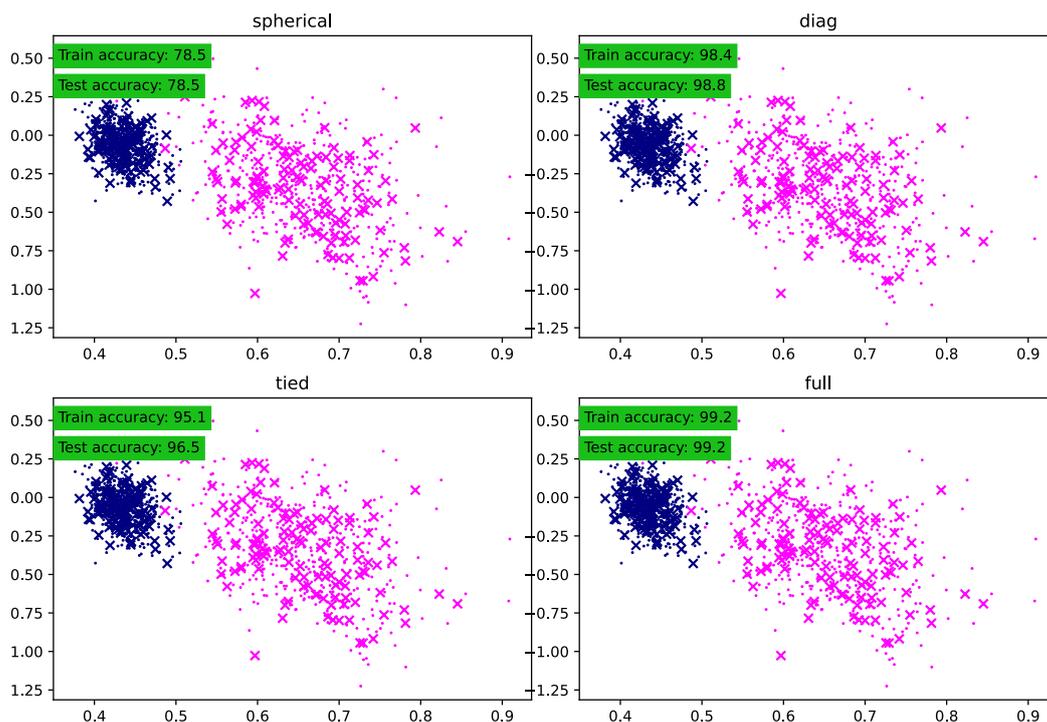


Figure 3.6 - Gaussian mixture model results (RSM vs. Skewness) for the Broken bar and Normal conditions.

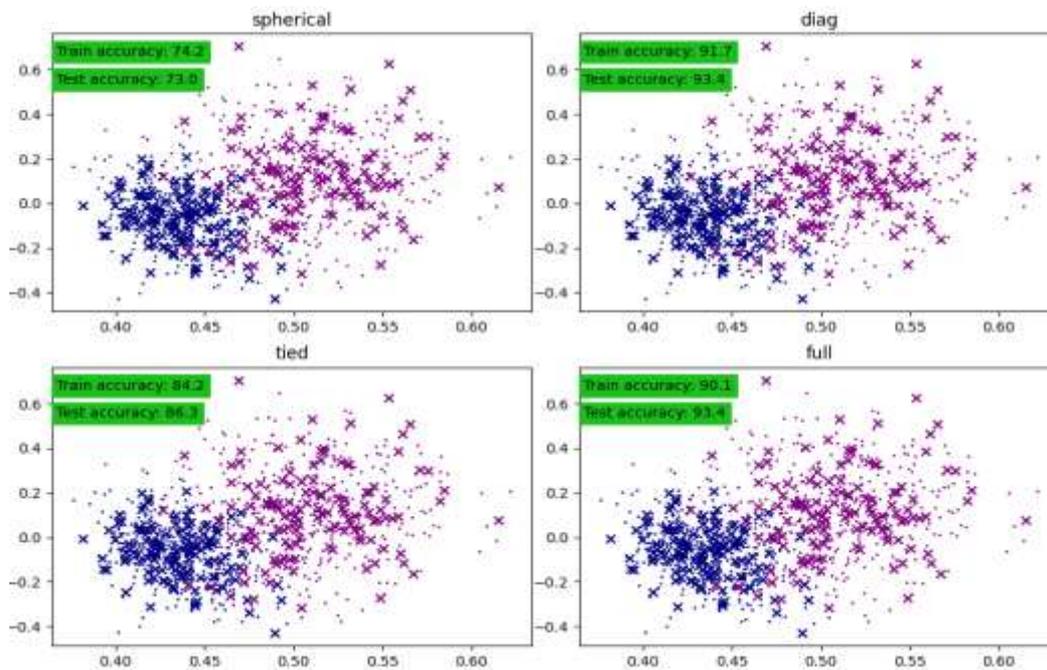


Figure 3.7 - Gaussian mixture model results (RSM vs. Skewness) for the Misalignment and Normal conditions.

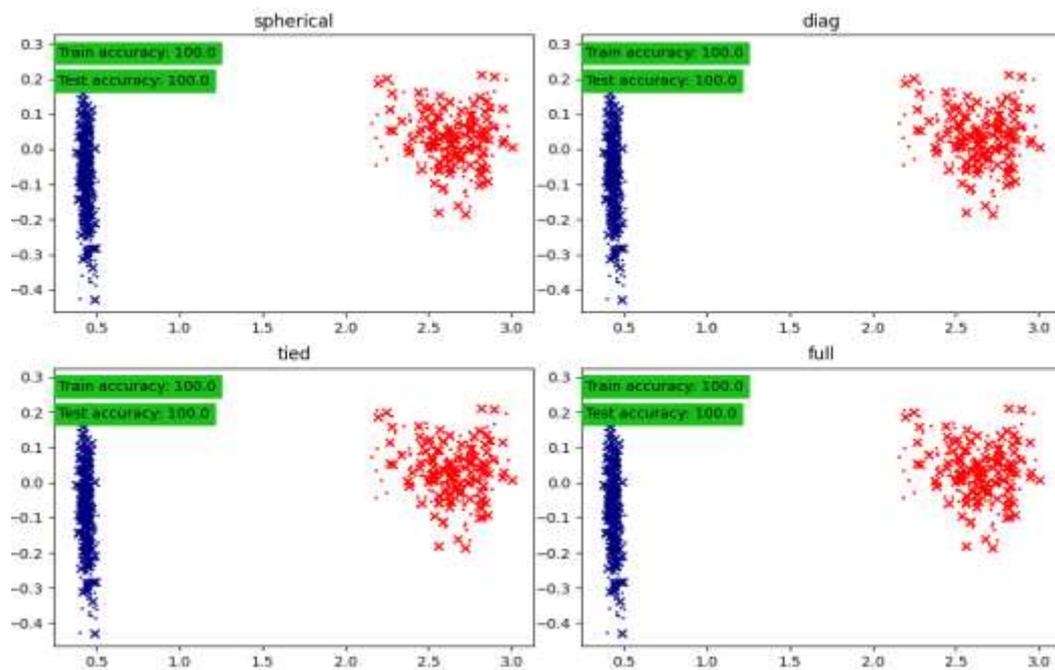


Figure 3.8 - Gaussian mixture model results (RSM vs. Skewness) for the Mechanical loss and Normal conditions.

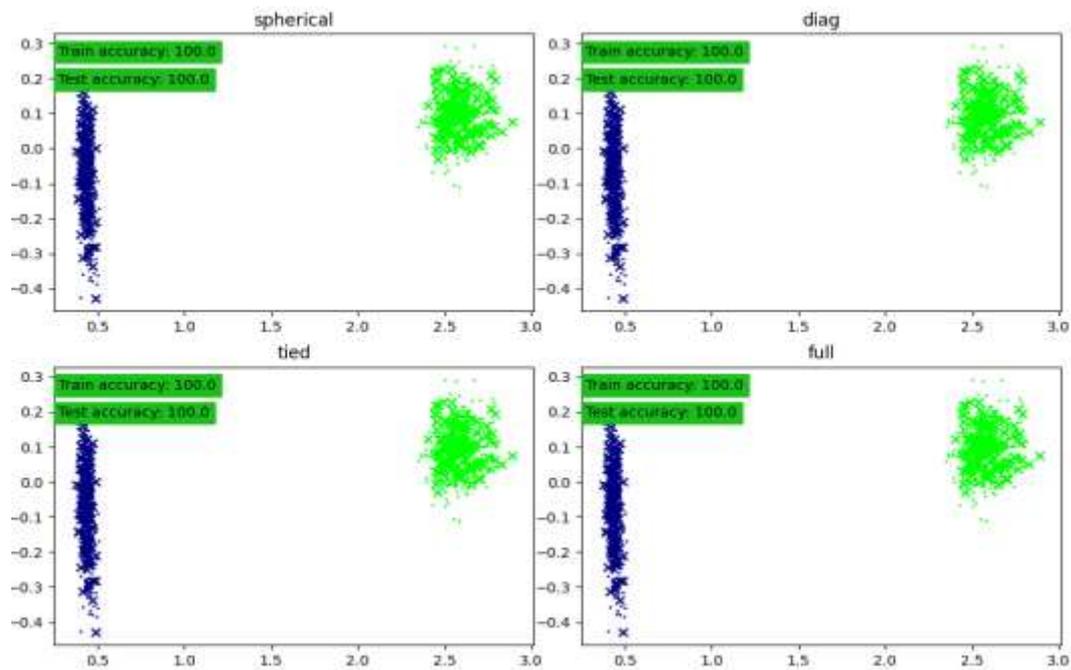


Figure 3.9 - Gaussian mixture model results (RSM vs. Skewness) for the Bearing and Normal conditions.

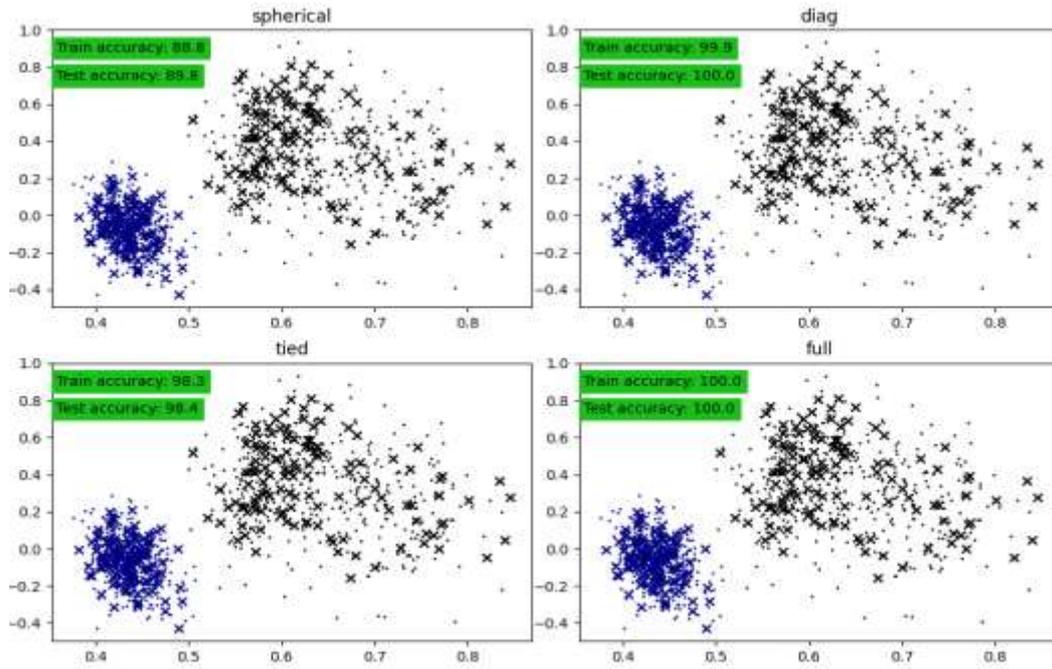


Figure 3.10 - Gaussian mixture model results (RSM vs. Skewness) for the Unbalanced and Normal conditions.

In order to assess how far apart the clusters are, the Mahalanobis distance metric was taken. For Gaussian distributed data, the distance of an observation  $x_i$  to the mode of the distribution can be computed using its Mahalanobis distance as shown in Equation 3.3.

$$d_{(m\Sigma)}(x_i)^2 = (x_i - m)^T \hat{\Sigma}^{-1}(x_i - m) \quad (3.3)$$

where  $\mu$  and  $\Sigma$  are the location and the covariance of the underlying Gaussian distributions. In practice,  $\mu$  and  $\Sigma$  are replaced by some estimates.

In addition, Figure 3.11 displays the Mahalanobis distances of the Gaussian distributions fitted to the normal operating condition. It was observed that the mechanical loss condition presented a particular pattern of distances. Figure 3.12 in turn groups all the distances. Note that mechanical loss and bearing cases are the operating conditions that are farthest from the normal operating case.

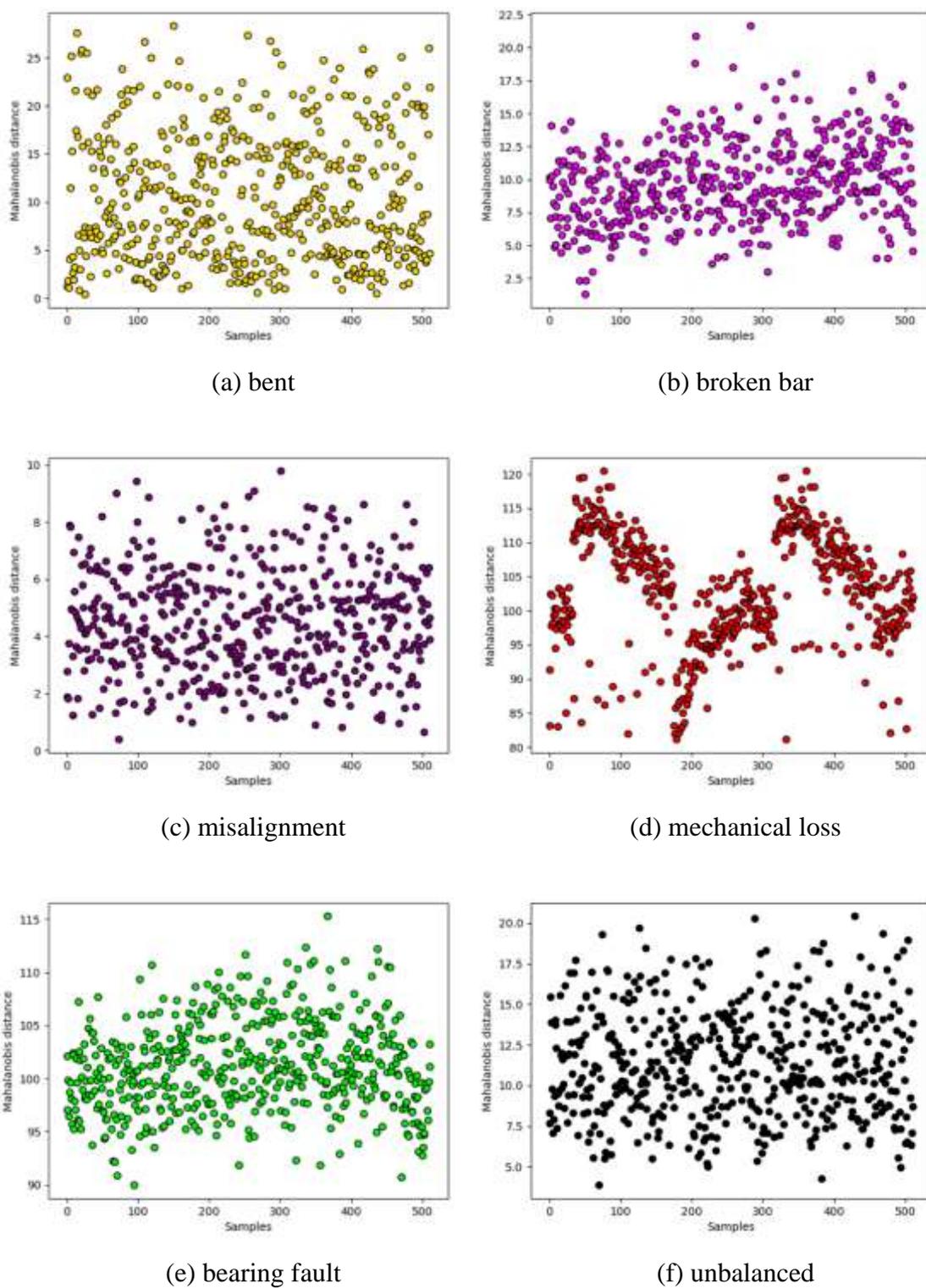


Figure 3.11 - Mahalanobis distance for the different induced operational conditions.

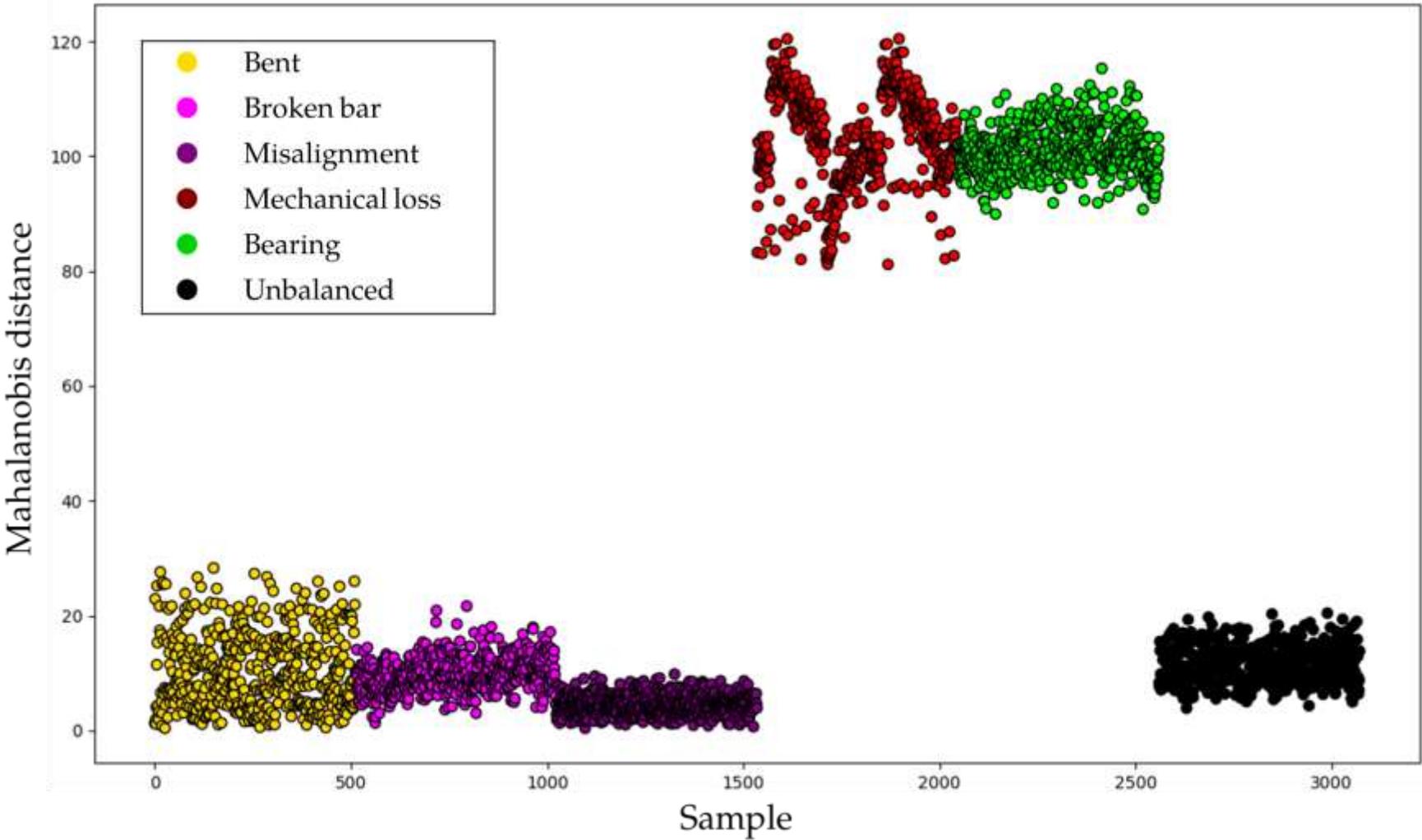


Figure 3.12 - Mahalanobis distance for all the evaluated and induced operational conditions.

### 3.4 Chapter Conclusion

This chapter presented a method based on vibration signal and Gaussian mixture model for fault identification on electric motors is designed. Vibration signals from electric motors are measured by two accelerometers in two different axes and then are used to extract RMS and Skewness features. In sequence, the GMM is applied and the Mahalanobis distance metric is used to validate how far one cluster is from another. Other studies have already applied GMM for fault detection [15,58], but most are limited to bearing faults only. As a novelty, this chapter applies the technique to other failures and compare these faults by applying the Mahalanobis distance

To validate the method, seven electric motors with different operations conditions (six faults) were used. Analyzing the results, it is evident that bearing, mechanical loss and unbalanced faults are easily identified by the method. Also, from Mahalanobis distance, mechanical loss presented a particular pattern of distances. The other defects were also differentiated, but not as significantly, so they still share at least some edge with the healthy condition cluster. This can make the model not behave well if used in other electric motors. Also, as it is an unsupervised learning algorithm, this method does not need human support to carry out the identification of defects. However, this advantage comes at the cost of a reduction in accuracy.

# 4. Fault Detection and Diagnosis Using Convolutional Neural Network and Short-Time Fourier Transform

In this chapter, a Convolution Neural Network (CNN) with Short-time Fourier Transform (STFT), a time-frequency feature map, is used to extract as much information as possible from vibration signals. A series of experiments are performed with seven different induced faults and operation conditions, the same as presented in chapter 2. A vibration signal was recorded to obtain the STFT response. Then, a CNN is trained to diagnose and predict the faults, considering the STFT as the only input. The results show the proposed method can diagnose the different faults accurately.

The chapter is organized as follows: Section 4.1 provides the theoretical background for Convolution Neural Network. Section 4.2 presents the Experimental methodology. Section 4.3 presents the results obtained by Convolution Neural Network with Short-time Fourier Transform. Finally, in Section 4.4, the conclusions are drawn.

## 4.1 Theoretical Background

CNN is a special ANN with a feed-forward structure. The CNN connection mode is a local connection that follows the sparse response characteristics of biological neurons [17]. Therefore, the dependence between the network model parameter size and the amount of training data can be greatly reduced. CNNs have important properties that make them powerful in 2D analysis. Examples include local receptive fields, weight sharing, and subsampling of spatial domains [43]. CNN consists of two parts. One is responsible for feature extraction and includes an input layer, a convolution layer, and a pooling layer [19]. The other is responsible for classification and includes the fully connected layer and the output layer [19]. CNNs use optimization algorithms and methods to modify neural network properties such as weights and learning rates to reduce

losses. Most CNNs use stochastic gradient descent methods such as SGD and ADAM [59.60].

### 4.1.1 Input layer

The input layer receives classified data or images. If the input data is an image, it is necessary to set the aspect ratio to  $2^n \times 2^n$ , such as  $64 \times 64$ ,  $128 \times 128$ , etc. This format is useful for filtering in the following layers [16]. This paper uses STFT as an input. In practice, the STFT computation procedure consists of splitting the long-period signal into shorter segments of equal length and then computing the Fourier transform for each shorter segment separately.

### 4.1.2 Convolutional layer

The convolutional layer contains a kernel (filter) that transforms the inputs of the previous layer to extract the local characteristics of the inputs (Figure 4.1). Filters use the same kernel to extract local features, so neurons in the same filter share weights. Recently, the Leaky ReLU activation function has the same characteristics as ReLU (Low compute cost and high-speed training) and can avoid the problem of neuron death (neuron without connection) [61]. The general form of the convolution operation is represented by Equation (4.1).

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (4.1)$$

where  $x_j^l$  denotes the  $j$ th feature map generated by the  $l$ th layer,  $x_i^{l-1}$  denotes the  $i$ th input feature map,  $k_{ij}^l$  denotes the  $j$ th kernel connected with  $i$ th input feature map,  $f(\ )$  is an activation function,  $b_j^l$  denotes the bias corresponding to  $j$ th kernel, and  $*$  denotes the 2D convolution operation.

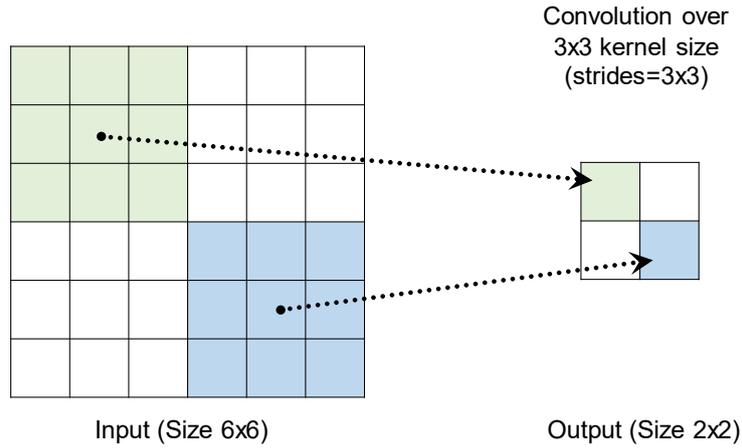


Figure 4.1 - Convolutional layer filter.

### 4.1.3 Pooling layer

Pooling layers are usually added after each convolutional layer. The function of the pooling operation is to reduce the spatial size, reduce computational complexity, and reduce the risk of overfitting. Commonly used pooling methods include average pooling, maximum pooling, and norm pooling. In this article, the max-pooling method is applied as shown in Figure 4.2. The pooling layer is described mathematically by Equation (4.2):

$$x_j^l = f \left[ \beta_j^l \mathbf{down} \left( x_j^{l-1} + b_j^l \right) \right] \quad (4.2)$$

where  $x_j^l$  denotes the  $j$ th feature map generated by the  $l$ th layer;  $\mathbf{down}(\cdot)$  represents a sub-sampling function;  $\beta_j^l$  denotes the sub-sampling function bias corresponding to  $j$ th kernel;  $x_i^{l-1}$  denotes the  $i$ th input feature map;  $b_j^l$  denotes the bias corresponding to  $j$ th kernel.

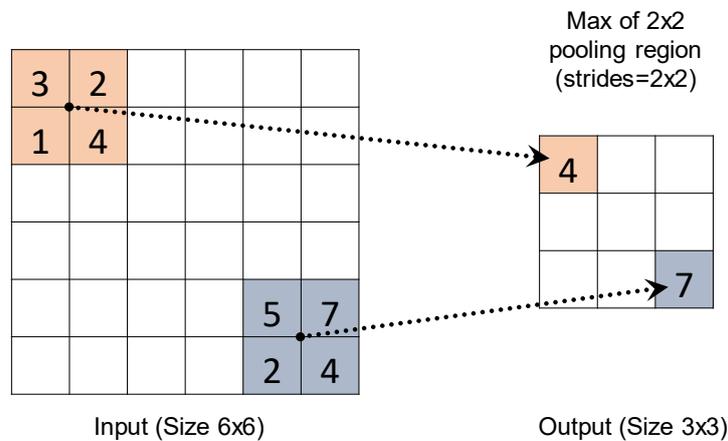


Figure 4.2 - Max pooling function.

#### 4.1.4 Fully connected layer and output layer

The fully connected layer simply consists of a feed-forward neural network. The fully connected layer forms the final layer of the network. The input of the fully connected layer is the output of the final group of convolutional layers and pooling layers. The last fully connected layer, also known as the output layer, in the classifier, the parameters represent the class score. In this article, the Softmax function is used in the output layer. The Softmax function takes a vector with any real value and reduces it to a vector of values from 0 to 1. The Softmax function is defined by Equation (4.3):

$$O = \begin{bmatrix} P(y=1|x;W_1,b_1) \\ \vdots \\ P(y=k|x;W_k,b_k) \end{bmatrix} = \frac{1}{\sum_{j=1}^k \exp(W_j x + b_j)} = \begin{bmatrix} \exp(W_j x + b_j) \\ \vdots \\ \exp(W_k x + b_k) \end{bmatrix} \quad (4.3)$$

where  $W_j$  and  $b_j$  are the weight matrix and bias, respectively;  $O$  is the result of the CNN.

#### 4.1.5 Reducing overfitting

Two techniques were used in this chapter to reduce overfitting and, consequently, CNN's generalization ability. The first is the use of a dropout layer. That is, some random nodes are omitted during the training process (Figure 4.3). When used after pooling layers, they can achieve the same effect as an increase in image noise [44,45].

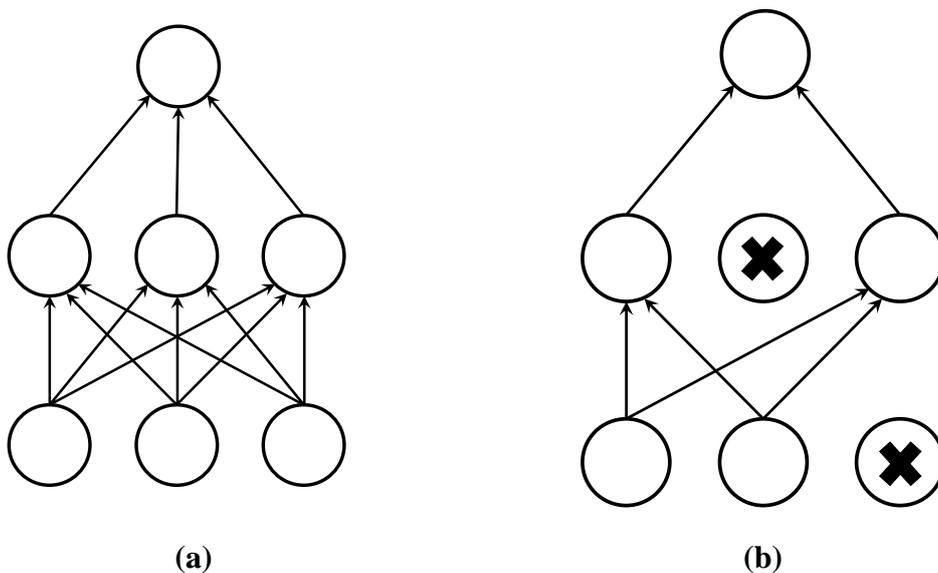


Figure 4.3 - ANN: (a) Before dropout (b) After dropout.

The second consists of normalizing the weights of the classes (layer weight regularizes). Regularizes allows class parameters to be penalized during optimization. These penalties are added to the network loss function. This penalty allows for less overfitting of the training model and better generalization [46, 47].

## 4.2 Experimental Methodology

The experimental setup used is the same used in sections 2.2.1 and 2.2.2, that is, a bench with seven different electric motors, six of which have known types of faults. In this method, only the vibration signal from the accelerometer 1 is used. Different from what was done in Section 2, a CNN with STFT is proposed for the diagnosis of faults in electric motors.

### 4.2.1 Data processing

In order to use CNN's features, the input should be an image. An image is a matrix represented by  $[m, n, k]$ , where  $m$  and  $n$  are the dimensions of the pixel matrix, and  $k$  is the tricolor (RGB) channel of the image. However, a better alternative to just converting a vibration signal into an image is converting the signal into a "time-frequency map". This is due to the fact that time-frequency analysis yields more information than time or frequency separated analysis. In this work, the STFT was chosen to perform the time-frequency analysis. To apply the STFT, the Scipy signal spectrogram was used. The parameters used were: (i) length of each segment of 1024, (ii) Blackman-Harris window, and (iii) overlap of 15%.

In this case, each signal was divided into 30 parts, and each part's STFT was performed (Figure 4.4). Finally, to reduce the number of calculations and facilitate the training of CNN, the image is compressed into the form of  $128 \times 128$  (Figure 4.4).

(Intentionally left blank)

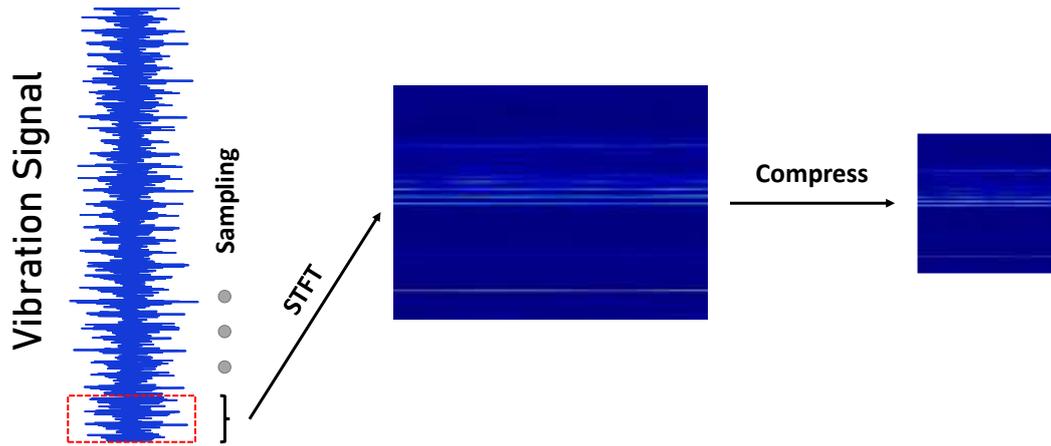


Figure 4.4 - Data processing and transformation.

### 4.2.2 CNN model

After data processing, it is possible to train CNN. Table 4.1 shows the total number of samples used for both training and testing. The samples are randomly selected for training and testing. Finally, Figure 4.5 shows an example of each motor state sample.

Table 4.1 - Training and test samples for each motor state.

Motor state	Training samples	Test samples	Label
Normal	120	30	1
Unbalanced	120	30	2
Mechanical loss	120	30	3
Misalignment	120	30	4
Bent shaft	120	30	5
Broken Bar	120	30	6
Bearing fault	120	30	7
<b>Overall</b>	<b>840</b>	<b>210</b>	-

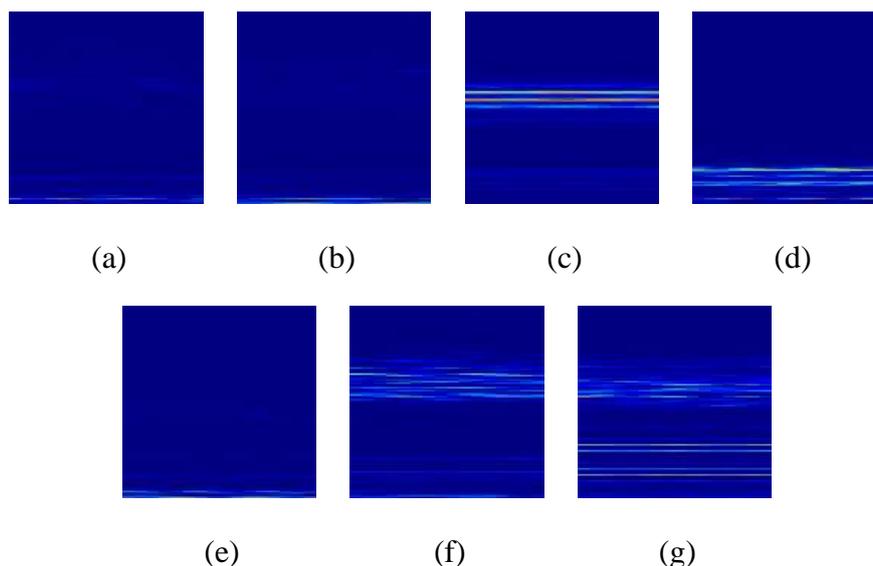


Figure 4.5 - STFT: (a) Normal, (b) Unbalanced, (c) Mechanical loss, (d) Misalignment, (e) Bent shaft, (f) Broken bar, (g) Bearing fault.

In the CNN model, as shown in Figure 4.6, there are 3 convolution layers with a kernel size =  $3 \times 3$ . It also has 3 pooling layers, with a pool size =  $2 \times 2$ . The activation function used is LeakyRelu, the optimizer used is ADAM and the classification layer uses the Softmax function. Besides that, there is a dropout of 0.33 before each pooling layer and the first fully connected layer. Also, the CNN has an L2 regularization with a weight of 0.001 in each convolution layer. The CNN model was developed in the Python language using the Keras API of TensorFlow [48].

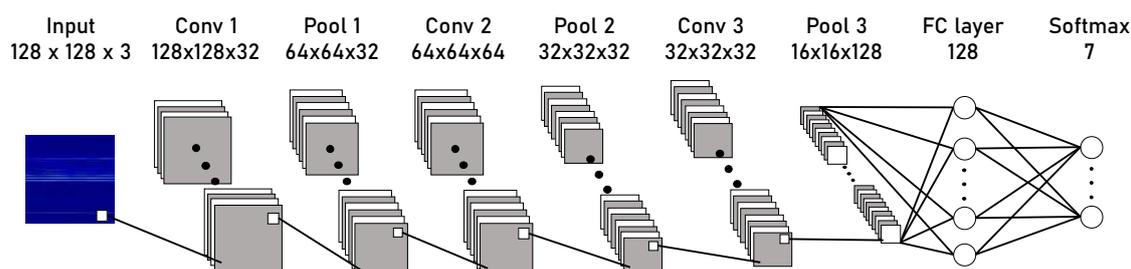


Figure 4.6 - Convolution neural network model.

### 4.3 Results and Discussion

In this section, the results obtained by CNN were discussed. First, an analysis will be conducted to choose the optimal CNN parameters by changing the learning rate, batch size, and epoch. Then, the feature visualization and fault prediction will be discussed.

### 4.3.1 Selection of CNN hyperparameters

Choosing the right settings for your CNN is critical to its success. The choice of the best parameters directly affects the learning process. As the parameters of each data change, you need to understand and review the data and choose the best one.

- **Learning rate:** Learning rate is a very important parameter that affects weight selection and error convergence. To increase the efficiency of a CNN, it is important to choose an appropriate learning rate for the data.
- **Batch size:** When training a CNN, due to large data size and hardware limitations, not all the data can be used for training. Therefore, the data is divided into small samples, called batch sizes. Choosing a compatible batch size makes your CNN network more reliable and accurate.
- **Epoch:** This is the parameter that determines the number of times the training algorithm operates on the training data set. The number of epochs should make it possible to obtain a stable network with a minimum of errors.

The first attempt was conducted, where the batch size was kept constant and the learning rate was changed in each run. The epoch number has been set to 10. Figure 4.7 shows the result of this test. Analyzing Figure 4.7, it is straightforward that if the learning rate is high or low, the training and test accuracy will be reduced, and the loss function will be penalized. So, intermediate values are more appropriate for this data.

On the second attempt, the learning rate was constant, and the batch size was changed in each run. The epoch number has been set to 10. Figure 4.8 shows the results of this test. Analyzing Figure 4.8, it is noticed that for lower values of batch size, the train and test accuracy are improved. Also, for lower values of batch size, the train and test loss function are minimized.

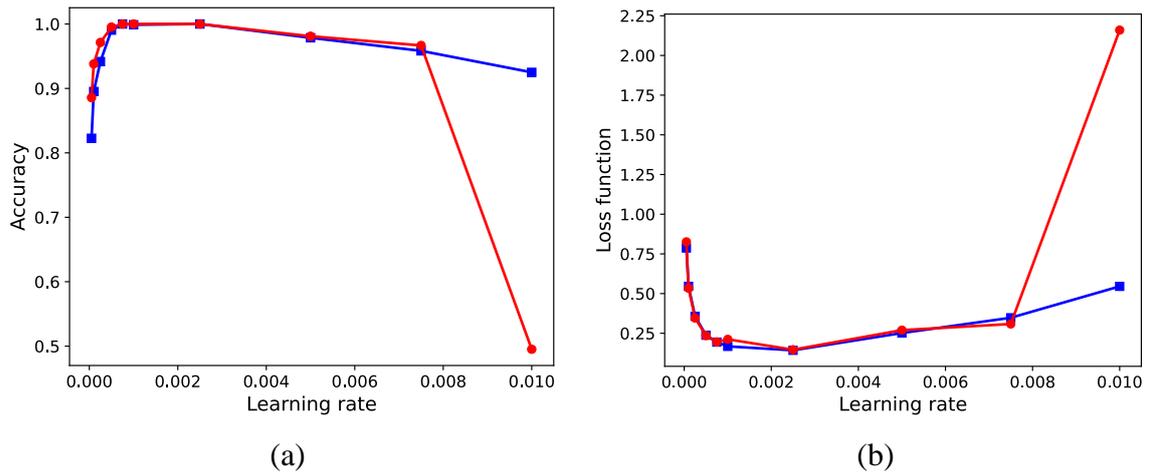


Figure 4.7 - (a): Accuracy under different learning rate, (b): Loss function under different learning rate (legend: — Test and — Train).

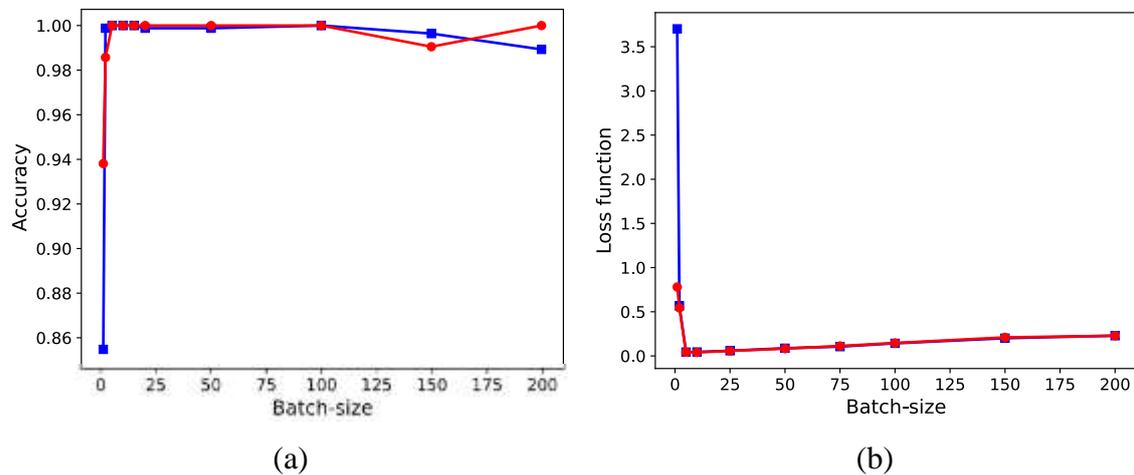


Figure 4.8 - (a): Accuracy under different batch-size, (b): Loss function under different batch-size (Legend: — Test and — Train).

Based on the analysis, a new test to determine the optimal parameters of CNN was performed. Because CNNs use stochastic algorithms, you should compare the average performance of the CNN over several runs to determine which parameter produces the most accurate and stable training. Table 4.2 shows the results of these new tests.

(Intentionally left blank)

Table 4.2 - Runs to select the best network parameters.

<b>Learning Rate</b>	<b>Batch Size</b>	<b>Mean Train-Loss</b>	<b>Mean Test-loss</b>	<b><math>\sigma</math> train-loss</b>	<b><math>\sigma</math> test-loss</b>	<b>Mean Train-acc</b>	<b>Mean Test-acc</b>
0.00075	5	0.0618	0.0610	0.0171	0.0123	0.9993	0.9990
0.00100	10	0.0671	0.0627	0.0254	0.0197	0.9993	0.9995
0.00250	25	0.3103	0.2880	0.1598	0.1262	0.9900	0.9895
0.00075	5	0.0665	0.0654	0.0032	0.0032	1.0000	0.9995
0.00100	10	0.0637	0.1174	0.1663	0.1174	0.9996	0.9990
0.00250	25	0.1504	0.0973	0.2493	0.1152	0.9920	0.9991
0.00075	5	0.1040	0.1131	0.0051	0.0122	0.9996	1.0000
0.00100	10	0.0864	0.0937	0.0031	0.0077	1.0000	1.0000
0.00250	25	0.0651	0.0650	0.0087	0.0072	1.0000	1.0000

Based on the data obtained, the optimum parameters chosen were a learning rate of 0.00250 and a batch-size of 25. Although in all combinations of parameters the accuracy is very similar, analyzing the loss function we have great differences between the configurations. The loss function is extremely important to know if the model is overfitting. With these parameters, a CNN with 100% accuracy is achieved both in training and testing. Analyzing the training and test loss for the chosen parameters, we see that both have low and similar values, which shows the stability of the network and also that it is not overfitting.

Therefore, it is necessary to evaluate the number of epochs. A test with 500 epochs was conducted. Figures 34 and 34 show the results. From Figure 4.9 and Figure 4.10, it can be seen that after 140 epochs, the network loses accuracy and becomes unstable. Analyzing the loss, it can be seen that there are regions where the loss grows, and at the same time, the accuracy goes down. By carrying out a comprehensive comparison, the number of epochs chosen was 10, ensuring not only the accuracy and loss but also saving training time.

Finally, the parameters chosen were a learning rate of 0.00250, batch size of 25, and 10 epochs.

(Intentionally left blank)

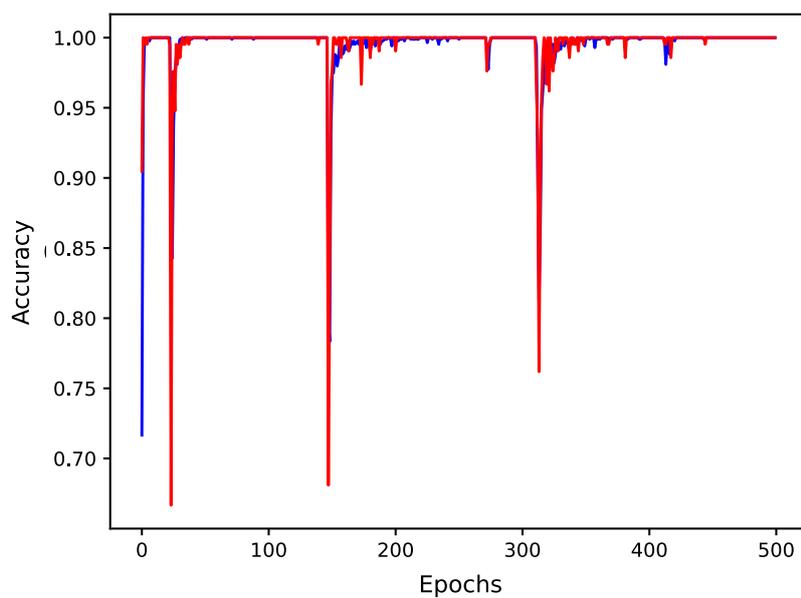


Figure 4.9 - Accuracy  $\times$  Epochs (legend: — Test and — Train).

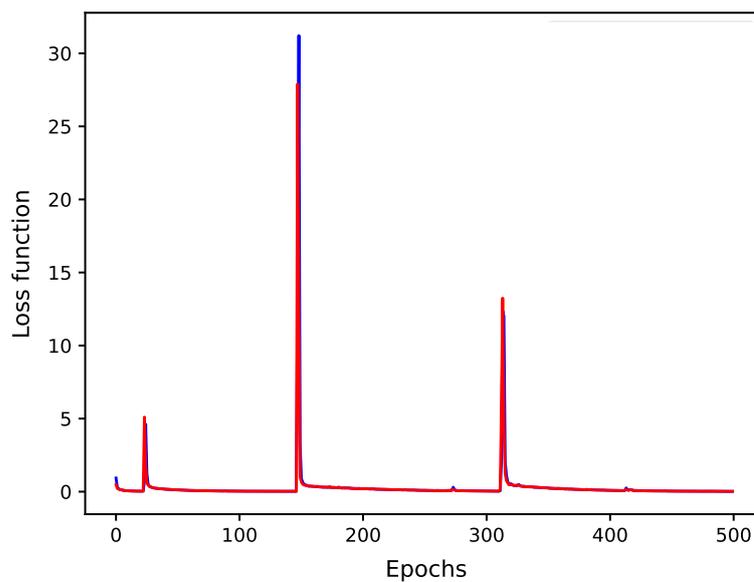


Figure 4.10 - Loss  $\times$  Epochs (legend: — Test and — Train).

### 4.3.2 Feature visualization and fault identification

To better understand the feature extraction process, a further analysis was conducted. Figure 4.11 shows the first layer output for all types of operating conditions. CNN's first layer gives an overview of how the network is extracting input features. Bright areas are the "activated" regions, meaning the filter detected some pattern. As already verified by Ribeiro Junior et al (2021), failures of mechanical components can be detected by frequency analysis. So, analyzing the outputs of CNN's first layer, it is noticed that the activated areas for each fault are different, representing a characteristic spectral pattern of the fault. For faults such as mechanical loss, broken bar and bearing faults, there is a higher concentration of high-frequency components. On the other hand, for unbalanced, misalignment and bent shaft defects, low frequency components are more evident. Also, there are a few "disabled" filters on the first layer. This shows CNN's great capacity for learning, since in the first layer it is already possible to extract the necessary information for the task.

The prediction confusion matrix, used to display CNN results, is shown in Figure 4.12. The CNN achieves an accuracy of 100%, that is, all defects and also the normal operating condition can be identified and classified using the proposed CNN. This shows that STFT combined with CNN is a powerful method to identify motor faults and achieve a fast and reliable diagnosis.

(Intentionally left blank)

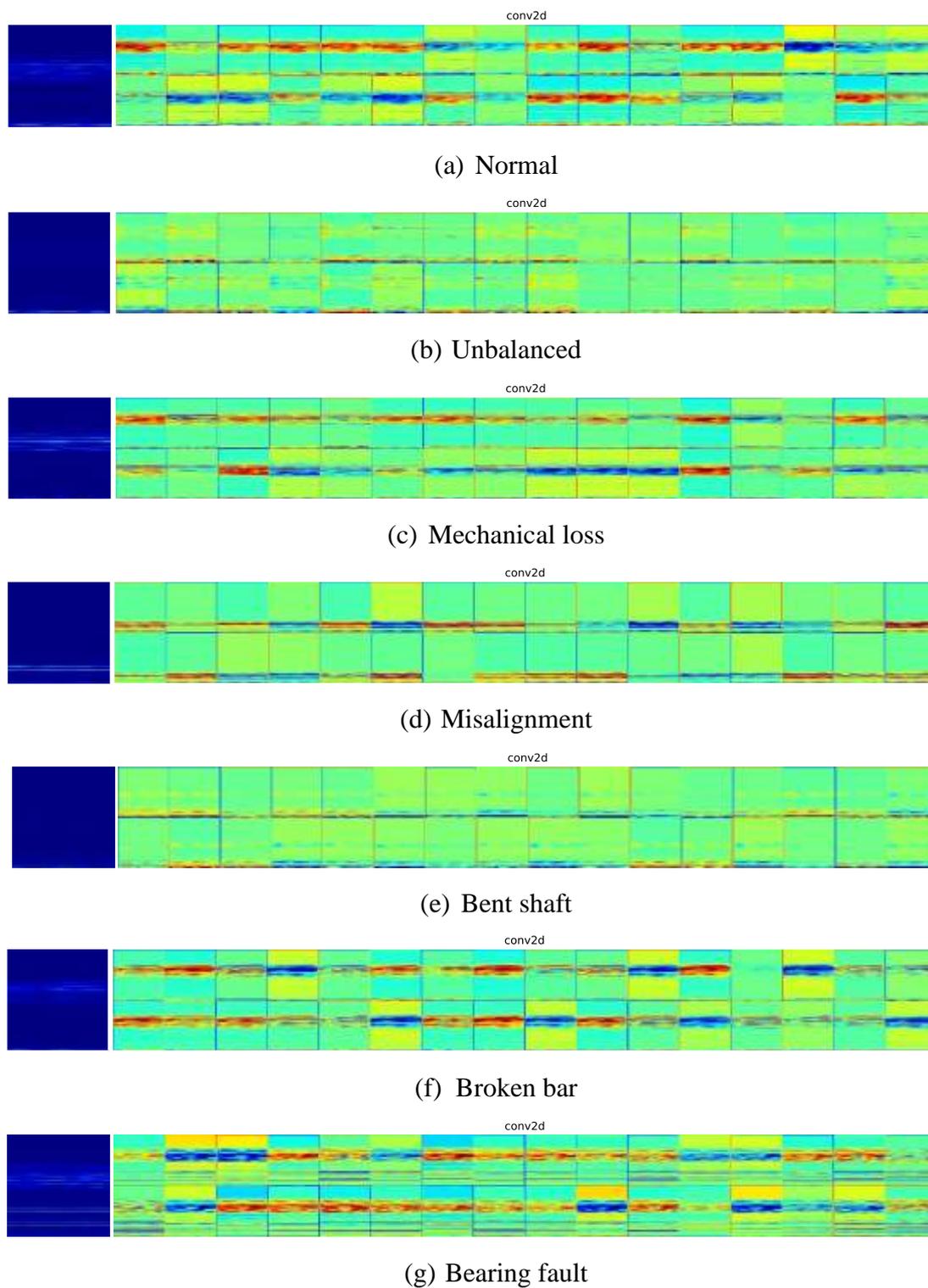


Figure 4.11 - First CNN output layer for the different fault conditions.

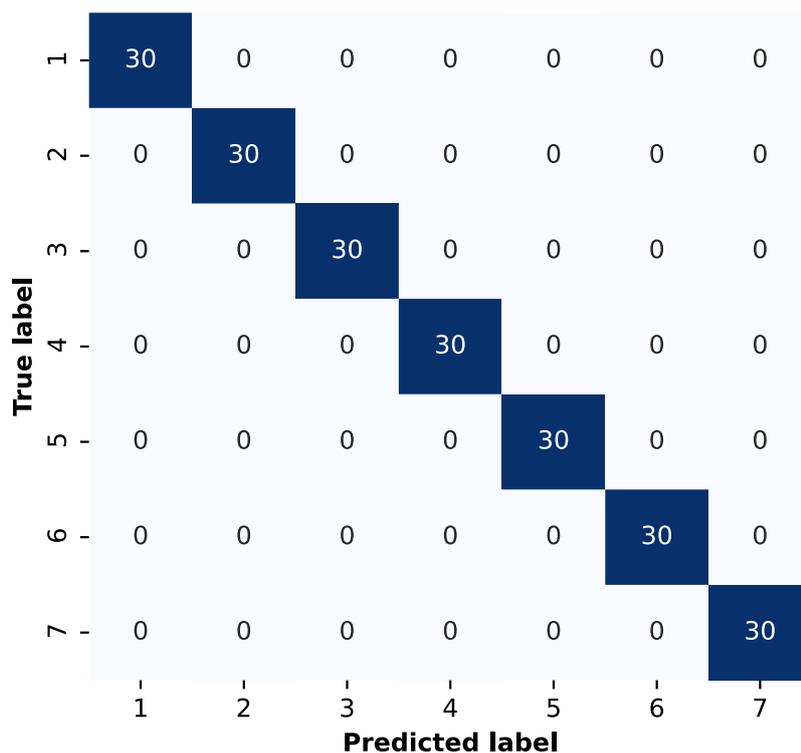


Figure 4.12 - Prediction confusion matrix.

## 4.4 Chapter Conclusion

In this chapter, an accurate and robust deep learning method for fault diagnosis in electric motors using STFT and CNN is presented. The CNN can use the entire dataset as input and provide the diagnosis result as an output.

The proposed approaches are verified experimentally with a setup several motor with up to 7 different operation conditions. By selecting different training parameters, an optimized one could be obtained to achieve a training and testing accuracy of 100%. Further, analyzing the loss function, the CNN is not overfitting. Because of these results, the proposed method can adaptively extract features and effectively diagnose faults in electric motors. To the best of the authors' knowledge, there are very few studies in the literature investigating the use of CNN in different fault conditions in rotating machines. Most of the studies focus on only detecting bearing failures [16-20]. Therefore, this work identified more failures than just bearing faults.

In addition, the proposed method in this chapter uses a fixed network architecture. However, it is an open challenge to determine the optimal parameters of the entire CNN,

especially if the architecture is deeper than the architecture adopted or if some other fault completely different is considered.

# 5. Fault Detection and Diagnosis Using 1D Convolutional Neural Networks

In this chapter, a multi-head one-dimensional Convolution Neural Network (1D CNN) to detect and diagnose six different types of faults in an electric motor using two accelerometers measuring in two different directions is proposed. This architecture was chosen due to the fact that each head can deal with each sensor individually, increasing feature extraction. A series of experiments are performed with seven different induced faults and operation conditions, the same as presented in chapter 2. The results show that the proposed architecture is very accurate and fast for multi-sensor fault detection using vibration time series.

The chapter is organized as follows: Section 5.1 provides the theoretical background on multi-head 1D Convolution Neural Network and some improvements on the CNNs architecture. Section 5.2 presents the Experimental methodology. Section 5.3 presents the results of a multi-head 1D convolution neural network based on a vibration signal. Finally, in Section 5.4, the conclusions are drawn.

## 5.1 Theoretical background

CNN is a special ANN with a feed-forward structure, with its characteristics better explained in Chapter 4, which has been successfully applied in many different fields such as image processing [43,49] and speech recognition [50,51], and signal analysis is a new application of CNN. The difference from the previous chapter is that this CNN will not need an image because the architecture of the new network will be for one-dimensional data. Although different, the concepts and architectures are very similar, with some differences that will be presented in the next sections.

### 5.1.1 Comparison between 1D and 2D CNNs

1D and 2D CNNs share almost the same architecture, with the exception of the filter sliding mechanism [52]. In 1D CNN, the filter slides in vertical order (height) to

extract the features, and the height determines the number of sample points for the convolutional operation. In contrast, the 2D CNN filter slides the entire matrix horizontally and vertically (height and width). The height and width of the 2D filter determine the range of convolution operations for each step [52]. Figure 5.1 compares 1D and 2D CNN.

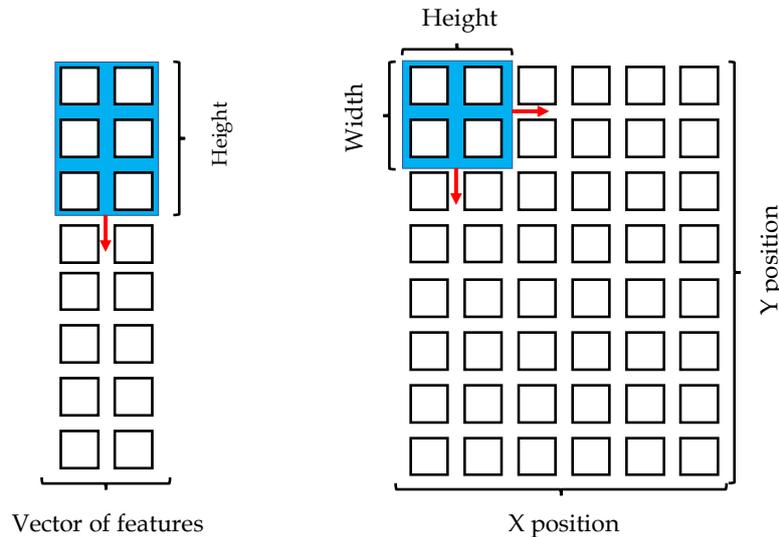


Figure 5.1 - Comparison between 1D and 2D CNNs filters.

As mentioned, 1D CNNs have advantages over their 2D counterparts and are preferred for certain applications, especially when processing 1D signals. This is due to the following reasons [53].

- There is a big difference in the complexity of 1D and 2D convolution calculations. The computational complexity is much lower under the same conditions (same configuration, same network, same hyperparameters).
- Most 1D CNN applications have a compression configuration consisting of one or two hidden CNN layers and few parameters in the network (usually less than 10,000). Although most 2D CNN applications have 1 million parameters (usually above 10 million) and many hidden layers, of course, neural networks have a lot to do with training and implementation.
- Due to its low computational requirements, 1D CNN is especially suitable for low-cost real-time applications on mobile and portable devices.
- As the 1D CNN works with the signal in time, it is not necessary to process the signal. In the case of a 2D CNN, it would be necessary to perform a preprocessing to transform the signal into something compatible with the 2D CNN.

### 5.1.2 Multi-head CNNs

In a multi-head convolutional neural network, each time series is handled independently, that is, each time will be handled by a convolutional head [54,55]. In many industrial situations, machines are installed with multiple sensors independent of each other, so there may be no correlation [54,55]. In this way, multiple sensors can be analyzed with the same CNN and extract the individual characteristics of each sensor. In addition, it can be choosing a sliding window that performs better in the different signals collected.

### 5.1.3 New Tricks of CNNs

Otherwise from what was done on the previous CNN, in this architecture new tips were used. The first is to use Batch Normalization. That is a technique that normalizes the input of a layer by keeping the mean close to zero and the standard deviation of the output close to 1. This technique is designed to reduce the shift of internal covariance and accelerate the training of deep neural networks [36].

The second is using Early Stop, an algorithm to stop training when a monitored metric stops improving. Assuming the common goal of training is to minimize the loss, the algorithm will check at the end of every epoch whether the loss is no longer decreasing after the N epochs are chosen. After N epochs if the loss does not decrease, the algorithm restores the weights of the best epoch.

## 5.2 Experimental Methodology

The experimental setup used is the same used in sections 2.2.1 and 2.2.2, that is, a bench with seven different electric motors, six of which have known types of faults. Different from what was done in section 2, a multi-head 1D CNN based on the vibration signals is proposed for the diagnosis of faults in electric motors.

### 5.2.1 Procedures of the proposed method

The multi-head 1D CNN algorithm proposed uses the fused data set of vibration signals in 2 channels as the input and then intelligently classifies these signals to identify six different faults.

Figure 5.2 shows a detailed flowchart of the proposed method. The first step is the acquisition of all vibration signals under different operation conditions. We collected four signals for each operation condition in each accelerometer, separating 75% of the signals for training and 25% for testing. After that, the signals are split into sample windows, and each sample window has 256 sample points. After the split, the data can be imported into the 1D CNN.

In each head, the 1D CNN has batch normalization, two convolutional layers, and two pooling layers; one fully connected layer and one SoftMax layer. The convolutional filter has 16 and the kernel size is 5. The pool filter is 16 and the pool size is 2. The node number is 256 at the fully connected layers, and lastly, there are seven outputs at the SoftMax layer, corresponding to the seven operation conditions in the experiment.

The network will be trained until, after 10 epochs, there is no decrease in the test loss. The visualization results and classification report are employed to verify the performance of the proposed method. All features used in the proposed method were developed in the Python language using the Keras API of TensorFlow [42] and Scikit-learn [48].

(Intentionally left blank)

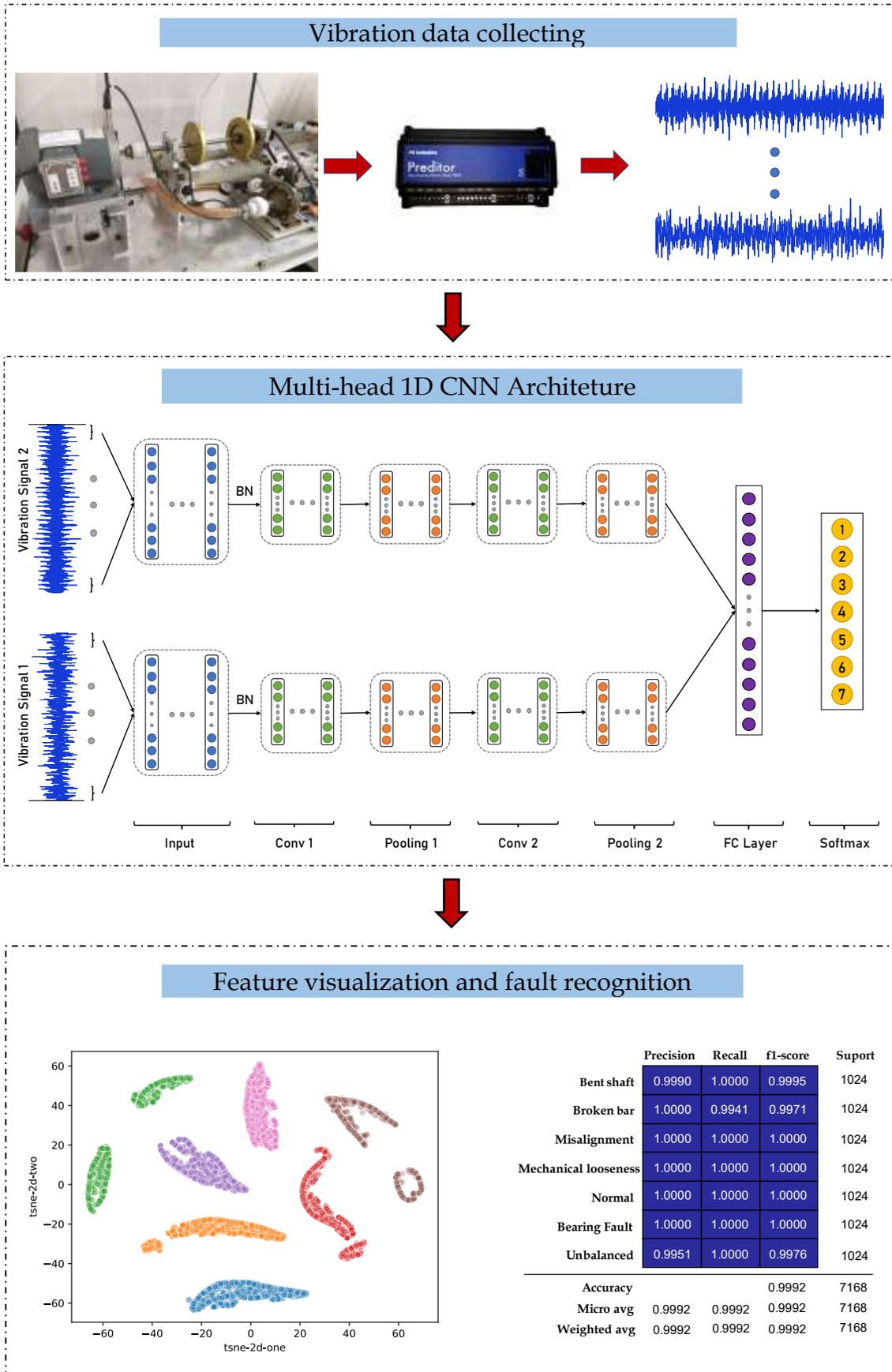


Figure 5.2 - Proposed method flowchart.

## 5.3 Results and Discussion

In this section, the results obtained by the proposed method were discussed. However, before that, an analysis will be performed to select the best 1D CNN hyperparameters. The accuracy of the proposed method is rated by the loss function, that is, the gap between the predicted value and the real value. As mentioned before, we use the early stop algorithm to aid in the selection of the best epoch of training (lower loss function). This way, the epoch parameter will not need to be optimized and analyzed. Finally, the results of 1D CNN are presented.

### 5.3.1 Hyperparameter Tuning

Tuning hyperparameters involves choosing the best set of hyperparameters for a learning algorithm. Hyperparameters are valuable parameters used to control the learning process. Finding all the optimal hyperparameters is an arduous process, so usually some main parameters are chosen to perform the tuning. In this paper, the hyperparameters selected for tuning are the number of filters, learning rate, and batch size.

Thus, a first attempt was conducted, where the learning rate was set at 0.01 and the batch size was 64 (algorithm default values), and the number of filters was changed in each run. In all, 10 runs for each filter value were made. Figure 5.3 shows the results in boxplot form. It was observed the presence of outliers for the number of filters of 8 and 32, in addition to a high variability for 64 and 128, characteristics of unstable networks. Therefore, based on the evidence, the number of filters was set to 16 because of the high score percentage and network stability. Another analysis that can be done is that the greater the number of filters, the longer it takes for the network to perform an iteration. In the second step, the number of filters was fixed at 16, the batch size was equal to 64, and the learning rate was changed in each run.

As with the first test, 10 runs for each learning rate value were made. Figure 5.4 shows the results in boxplot form. It is worth noting that there are some outliers for values of 0.0001, 0.001, and 0.01. In addition, for a value of 0.05, there is great variability. After analyzing the results, the learning rate was set to 0.0005 for having the highest score combined with low variability. Another analysis that can be done is that the value of the learning rate does not affect the CNN training time.

Concluding the tuning process, the number of filters was set to 16, the learning rate to 0.005, and the batch size was changed in each run. The same as the other tests were done, that is, 10 runs for each batch size value. Figure 5.5 shows the results in boxplot form. It was observed that the presence of outliers for the batch sizes of 64 and 128, besides a high variability for all batch sizes, except for the batch size of 32, was observed. Therefore, the batch size was set to 32 because of the highest score and low variability. Unlike the number of filters, the smaller the batch size, the longer it takes to perform an iteration. Based on the results, the optimal hyperparameters chosen were a number of filters of 16, a learning rate of 0.0005, and a batch size of 32.

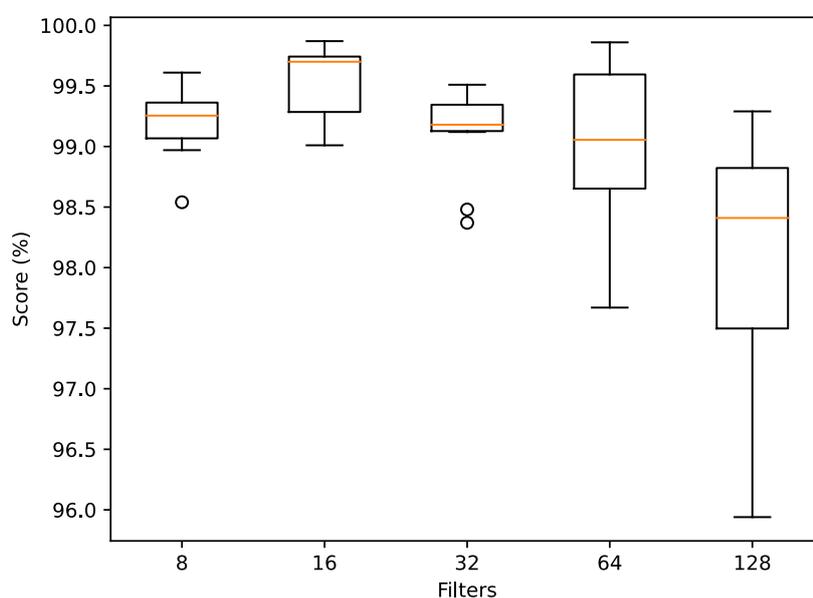


Figure 5.3 - Accuracy score under different number of filters.

(Intentionally left blank)

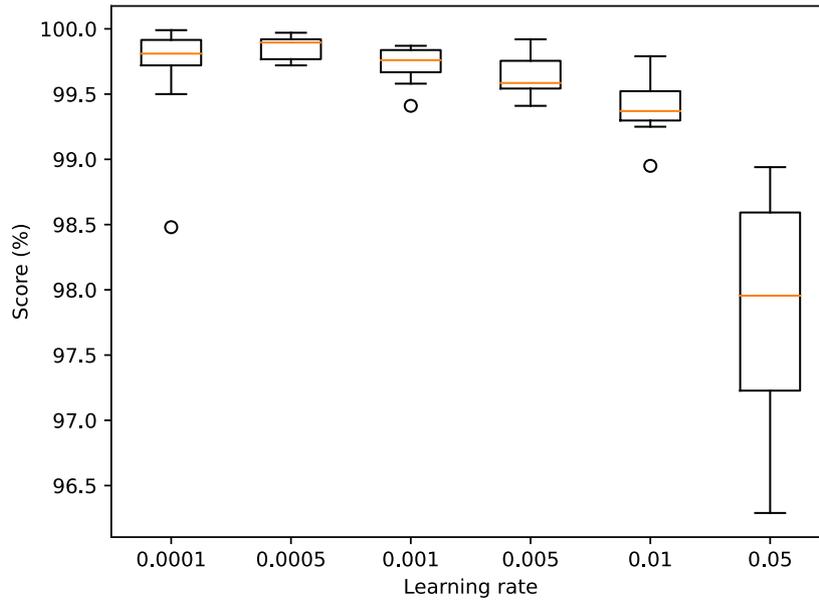


Figure 5.4 - Accuracy score under different learning rates.

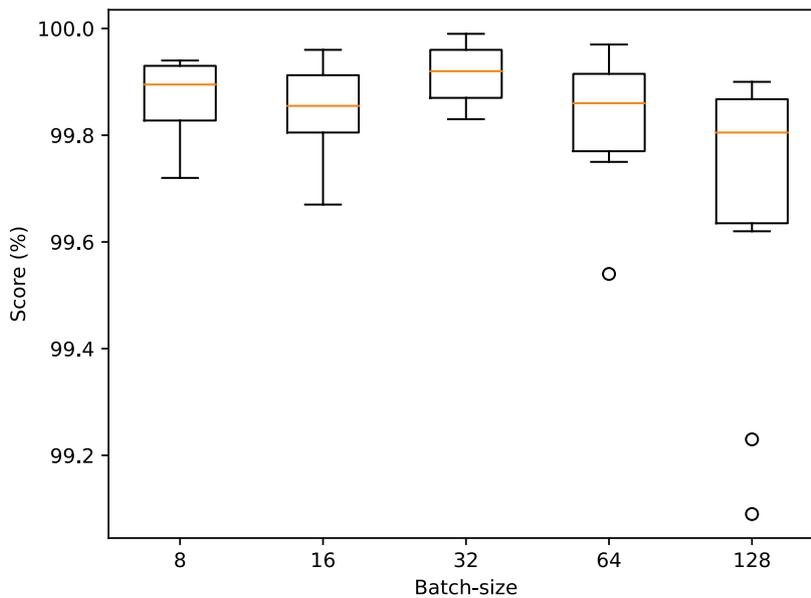


Figure 5.5 - Accuracy score under different batch sizes.

### 5.3.2 Feature visualization and fault recognition

The classification report was created to provide more detail on the CNN results, as seen in Figure 5.6. The classification report shows four measures: Accuracy, Precision, Recall and f1 score, described by Equations (5.1) to (5.4)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$f_1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.4)$$

Where *TP* indicates true positive, *TN* indicates true negative, *FP* indicates false positive and *FN* indicates false negative.

A f1 score is often more realistic than precision. This is especially true if the distribution of the classes is not uniform. Accuracy works better if the cost of a false negative is equal to the cost of a false positive. If there is a significant difference between the false positives and the false negative cost, both Precision and Recall should be verified.

	Precision	Recall	f1-score	Suport
Bent shaft	0.9990	1.0000	0.9995	1024
Broken bar	1.0000	0.9941	0.9971	1024
Misalignment	1.0000	1.0000	1.0000	1024
Mechanical looseness	1.0000	1.0000	1.0000	1024
Normal	1.0000	1.0000	1.0000	1024
Bearing Fault	1.0000	1.0000	1.0000	1024
Unbalanced	0.9951	1.0000	0.9976	1024
<b>Accuracy</b>			0.9992	7168
<b>Micro avg</b>	0.9992	0.9992	0.9992	7168
<b>Weighted avg</b>	0.9992	0.9992	0.9992	7168

Figure 5.6 - Classification report.

The recognition accuracy of the operation conditions is 99.92%. The misalignment, mechanical looseness, normal and bearing fault conditions reached 100%

accuracy, but the bent shaft, broken bar, and unbalanced had a minor error. However, it does not compromise the entire network classification, as this error is very small. This shows that a multi-head 1D CNN is a powerful method to identify electric motor faults with a fast and reliable diagnosis. Another advantage of 1D CNN compared to other types of networks is its low computational cost, making it very useful for real-time applications.

Network performance can be measured by both loss and accuracy. Figure 5.7 shows the evolution of the loss during each epoch of the training and test. The loss is measured using the Categorical cross-entropy loss function. Categorical cross-entropy is a loss function that is used in multi-class classification tasks. These are tasks where a sample can only belong to one out of many possible categories, and the model must decide which one and can be calculated as shown in Equation (5.5) [56].

$$Loss = - \sum_{i=1}^N y_i \log y_i^{\mu} \quad (5.5)$$

where  $\hat{y}_i$  is the  $i$ -th scalar value in the model output,  $y_i$  is the corresponding target value, and  $N$  is the output size in the model output.

Figure 5.8 shows the evolution of accuracy during each epoch of the training and testing. The proposed method rapidly converges and builds an accurate model. Also, from both loss and accuracy, the performance of the early stop can be seen.

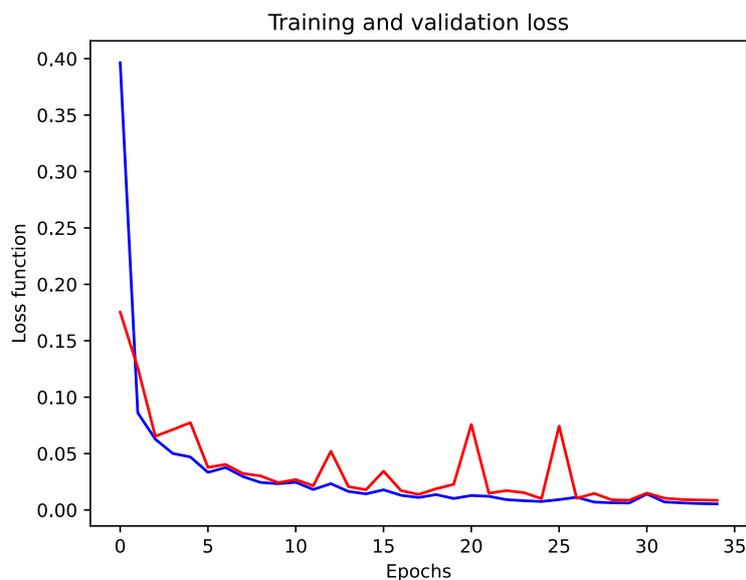


Figure 5.7 - Training and validation loss (legend: — Test and — Train)

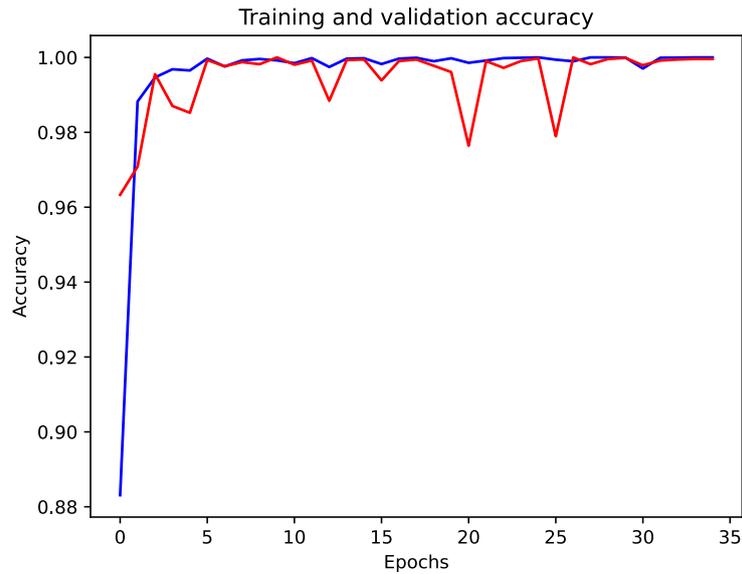


Figure 5.8 - Training and validation loss (legend: — Test and — Train)

To further comprehend the process of feature extraction, the t-distributed stochastic neighbor embedding (t-SNE) [57] was used to visualize the learning characteristics of the CNN. t-SNE is a technique to view high-dimensional data. It transforms similarities between data points into general probabilities and attempts to minimize Kullback-Leibler divergences between low-dimensional integrals and general probabilities of high-dimensional data. t-SNE provides ideas or insights into how data is organized in large-dimensional space. Therefore, the output of the algorithm gives visibility to high-dimensional data by projecting it into a 2-dimensional space. From Figure 5.9, we can see that at the entrance to the network, the different types of faults were all grouped, making them impossible to identify. At the end of the network, faults of the same type were grouped together, and the different fault types were separated. This shows that the proposed method has a powerful feature extraction ability.

(Intentionally left blank)

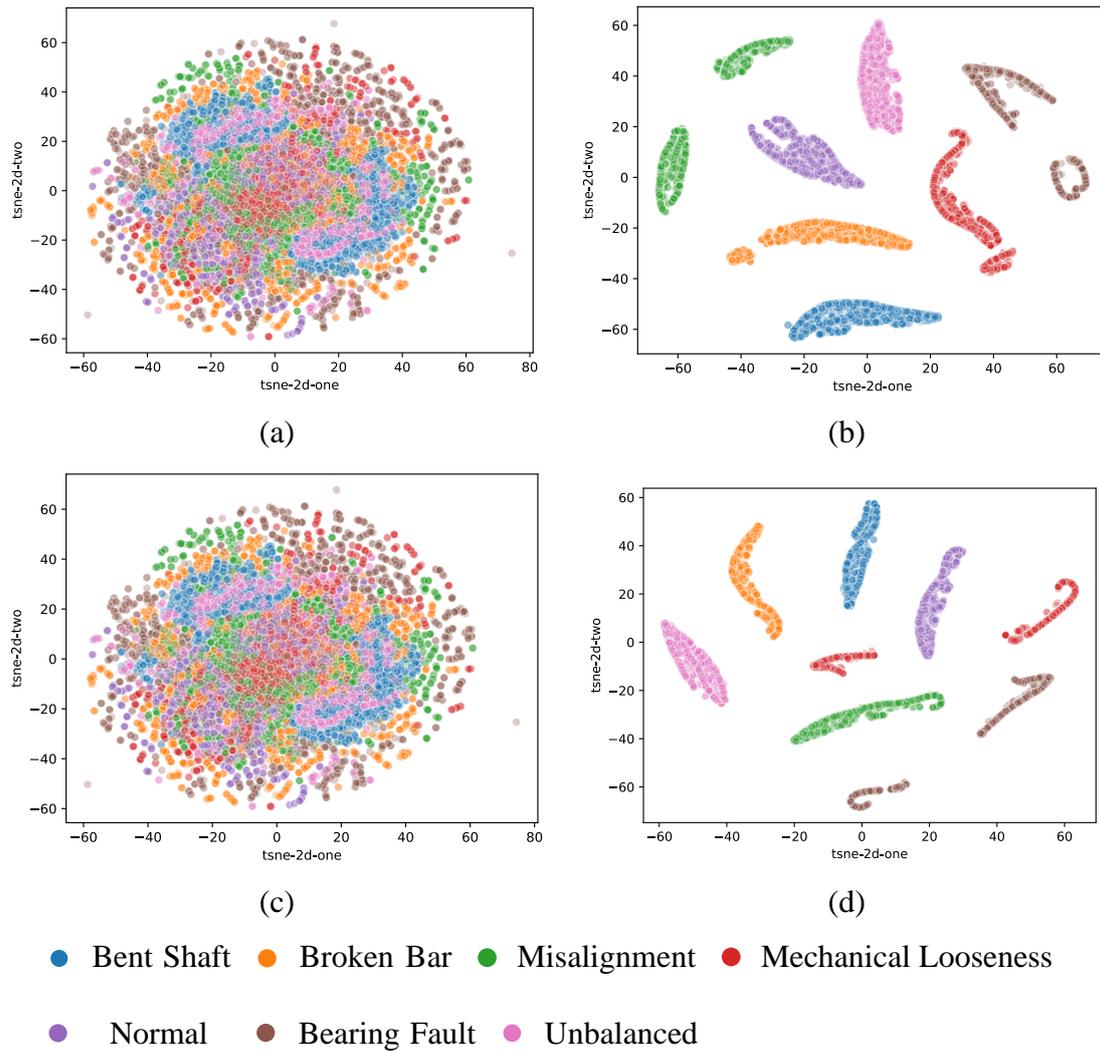


Figure 5.9 - (a) Training input layer, (b) Training output layer, (c) Test input layer, (d) Test output layer.

## 5.4 Conclusion

In this chapter, a method based on vibration signals and multi-head 1D-CNN for fault diagnosis on electric motors is designed. The vibration signals of electric motors are measured in two directions and then fed simultaneously to the multi-head 1D-CNN to train it. The method is verified experimentally with a setup of seven different conditions. We achieve a network with 99.92% accuracy by tuning the hyperparameter. Also, the t-SNE method is used to visualize the CNN learning process. Another positive point is that 1D networks are extremely fast to train and test (compared to conventional 2DCNN), making them very useful for real-time applications. These results show that the proposed method can successfully extract features and diagnose faults in an electric motor. In this

---

way, it is possible to follow in real-time the condition of the motors and do condition-based monitoring (CBM). When compared to other studies that use 1D CNN to detect faults in electric motors [53,54,62-66], the majority of the studies only detect bearing failures; thus, this work goes further and identifies more failures than just bearing faults.

## 6. General Conclusion

This study consisted of identifying and diagnosing failures in electric motors. Because electric motors are present in the majority of companies, detecting failures before they cause an unscheduled shutdown is critical for every company's maintenance department. With this in mind, four distinct strategies were created and tested during the dissertation to detect and diagnose faults in electric motors.

The first technique applied was the frequency analysis of the vibration signal. Because it is a well-established technique in the literature, the initial attempt was made to validate that those failures could be identified using the vibration signal. Although effective, it is a technique that requires a lot of professional knowledge and is time-consuming.

After confirming that the vibration signal contains defect information, it was considered how to solve the problems of the conventional fault detection technique. As a solution, Machine learning techniques were used. Machine learning handles the issues of time and professional knowledge better than any other method. However, generalization and data acquisition become more critical in those methods.

The convolution neural network is one of the greatest ML solutions when the data is labeled. 2D CNN provides great precision but requires preprocessing, whereas 1D CNN does not require preprocessing and, hence, is faster, but loses some precision. Despite their benefits and drawbacks, both strategies were successfully implemented.

If the data is not labeled, the Gaussian Mixture Model handles the data problem better because no prior knowledge of the system is required. However, it is the method with the lowest efficiency method when compared to others. In a new system, however, it may be the only suitable strategy.

Summarizing the discussions, Figure 6.1 shows the strengths and weaknesses of the four methods presented.



Figure 6.1 - Strengths and weaknesses of the methods.

## 6.1 Future works

Based on the results obtained in this work, and in addition to what has already been suggested in each chapter conclusion, the author suggests some topics to be studied in order to continue this research as follows:

- Combine other methods for detecting faults. Techniques such as motor current signature analysis (MCSA) and lubricating oil analysis to aid vibration analysis.
- Include more experimental tests, like composite faults or more faults types.
- Include different types of sensors in order to train the 1D CNN.
- Investigate the use of new machine learning techniques.

## 7. Publications

The publications resulting out of this dissertation are listed below:

- RIBEIRO JUNIOR, Ronny Francis; AREIAS, Isac Antônio dos Santos; GOMES, Guilherme Ferreira. Fault detection and diagnosis using vibration signal analysis in frequency domain for electric motors considering different real fault types. *Sensor Review*, v. 41, n. 3, p. 311-319, 2021.
- RIBEIRO JUNIOR, Ronny Francis; de ALMEIDA, Fabricio Alves; JORGE, Ariosto Bretanha; PEREIRA, João Luiz Junho; FRANCISCO, Matheus Brendon; GOMES On the use of gaussian mixture model for monitoring and fault diagnosis in dynamic components of electric motors. Submitted to publication.
- RIBEIRO JUNIOR, Ronny Francis; AREIAS, Isac Antônio dos Santos; CAMPOS, Mateus Mendes; TEIXEIRA, Carlos Eduardo; da SILVA, Luiz Eduardo Borges; GOMES, Guilherme Ferreira. Fault detection and diagnosis in mechanical vibrating systems using convolution neural network and short-time fourier transform. Submitted to publication.
- RIBEIRO JUNIOR, Ronny Francis; AREIAS, Isac Antônio dos Santos; CAMPOS, Mateus Mendes; Teixeira, Carlos Eduardo; da Silva, Luiz Eduardo Borges; GOMES, Guilherme Ferreira. Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals. Submitted to publication.

## References

- [1] RANDAL, R.B. *Vibration-based Condition Monitoring: Industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.
- [2] AL-BADOUR, F.; SUNAR, M.; CHEDED, L. *Vibration analysis of rotating machinery using time–frequency analysis and wavelet techniques*. *Mechanical Systems and Signal Processing*, v. 25, n. 6, p. 2083-2101, 2011.
- [3] RENWICK, John T.; BABSON, Paul E. *Vibration analysis---a proven technique as a predictive maintenance tool*. *IEEE Transactions on Industry Applications*, n. 2, p. 324-332, 1985.
- [4] DINIZ, Camila Aparecida et al. *Optimization of the layers of composite materials from neural networks with Tsai–Wu failure criterion*. *Journal of Failure Analysis and Prevention*, v. 19, n. 3, p. 709-715, 2019.
- [5] RIBEIRO JUNIOR, R.F., AREIAS, I.A.d.S. and GOMES, G.F. (2021), "Fault detection and diagnosis using vibration signal analysis in frequency domain for electric motors considering different real fault types", *Sensor Review*, Vol. 41 No. 3, pp. 311-319.
- [6] SHIFAT, Tanvir Alam; HUR, Jang Wook. *An effective stator fault diagnosis framework of BLDC motor based on vibration and current signals*. *IEEE Access*, v. 8, p. 106968-106981, 2020.
- [7] PANIGRAHY, Parth Sarathi; CHATTOPADHYAY, Paramita. *Tri-axial vibration based collective feature analysis for decent fault classification of VFD fed induction motor*. *Measurement*, v. 168, p. 108460, 2021.
- [8] ZHAO, Chunhui; GAO, Furong. *Fault subspace selection approach combined with analysis of relative changes for reconstruction modeling and multifault diagnosis*. *IEEE Transactions on Control Systems Technology*, v. 24, n. 3, p. 928-939, 2015.
- [9] GAO, Zhiwei; CECATI, Carlo; DING, Steven X. *A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches*. *IEEE Transactions on Industrial Electronics*, v. 62, n. 6, p. 3757-3767, 2015.

- [10] WANG, Li-Hua et al. Motor fault diagnosis based on short-time Fourier transform and convolutional neural network. *Chinese Journal of Mechanical Engineering*, v. 30, n. 6, p. 1357-1368, 2017.
- [11] HEYNS, T., HEYNS, P. S., & De Villiers, J. P. Combining synchronous averaging with a Gaussian mixture model novelty detection scheme for vibration-based condition monitoring of a gearbox. *Mechanical Systems and Signal Processing*, 32, 200-215, 2012.
- [12] AVENDAÑO-VALENCIA, Luis David; FASSOIS, Spilios D. Damage/fault diagnosis in an operating wind turbine under uncertainty via a vibration response Gaussian mixture random coefficient model-based framework. *Mechanical Systems and Signal Processing*, v. 91, p. 326-353, 2017.
- [13] WANG, Ziqi; SONG, Junho. Equivalent linearization method using Gaussian mixture (GM-ELM) for nonlinear random vibration analysis. *Structural Safety*, v. 64, p. 9-19, 2017.
- [14] PERERA, Lokukaluge P.; MO, Brage. Data analysis on marine engine operating regions in relation to ship navigation. *Ocean Engineering*, v. 128, p. 163-172, 2016.
- [15] HONG, Youngsun et al. Early fault diagnosis and classification of ball bearing using enhanced kurtogram and Gaussian mixture model. *IEEE Transactions on Instrumentation and Measurement*, v. 68, n. 12, p. 4746-4755, 2019.
- [16] WEN, Long et al. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics*, v. 65, n. 7, p. 5990-5998, 2017.
- [17] HUANG, Wenyi et al. An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis. *Neurocomputing*, v. 359, p. 77-92, 2019.
- [18] WANG, Huaqing et al. A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals. *Computers in Industry*, v. 105, p. 182-190, 2019.
- [19] LEE, Ki Bum; CHEON, Sejune; KIM, Chang Ouk. A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, v. 30, n. 2, p. 135-142, 2017.

- [20] CHEN, Xiaohan; ZHANG, Beike; GAO, Dong. Bearing fault diagnosis base on multi-scale CNN and LSTM model. *Journal of Intelligent Manufacturing*, v. 32, p. 971-987, 2021.
- [21] FUAN, Wang et al. An adaptive deep convolutional neural network for rolling bearing fault diagnosis. *Measurement Science and Technology*, v. 28, n. 9, p. 095005, 2017.
- [22] BETTA, Giovanni et al. A DSP-based FFT-analyzer for the fault diagnosis of rotating machine based on vibration analysis. *IEEE Transactions on Instrumentation and Measurement*, v. 51, n. 6, p. 1316-1322, 2002.
- [23] RAI, V. K.; MOHANTY, A. R. Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform. *Mechanical Systems and Signal Processing*, v. 21, n. 6, p. 2607-2615, 2007.
- [24] SABBAGHIAN-BIDGOLI, F.; POSHTAN, J. Fault Detection of Broken Rotor Bar Using an Improved form of Hilbert–Huang Transform. *Fluctuation and Noise Letters*, v. 17, n. 02, p. 1850012, 2018.
- [25] CHANDRA, N. Harish; SEKHAR, A. S. Fault detection in rotor bearing systems using time frequency techniques. *Mechanical Systems and Signal Processing*, v. 72, p. 105-133, 2016.
- [26] OPPENHEIM, A.V., and R.W. SCHAFER. *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice-Hall, pp. 468-471, 1999.
- [27] LAI, Edmund. *Practical digital signal processing*. Elsevier, 2003.
- [28] SMITH, Steven. *Digital signal processing: a practical guide for engineers and scientists*. Elsevier, 2013.
- [29] JAYASWAL, P.; WADHWANI, A. K.; MULCHANDANI, K. B. Machine fault signature analysis. *International Journal of Rotating Machinery*, 2008.
- [30] PATIL, S. S.; GAIKWAD, J. A. Vibration analysis of electrical rotating machines using FFT: A method of predictive maintenance. In: 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE, 2013.
- [31] PLANTE, Tristan et al. Rotating machine fault detection using principal component analysis of vibration signal. In: 2016 IEEE AUTOTESTCON. IEEE, 2016.
- [32] BIANCHINI, Claudio et al. Fault detection of linear bearings in brushless AC linear motors by vibration analysis. *IEEE Transactions on Industrial Electronics*, v. 58, n. 5, p. 1684-1694, 2010.

- [33] BALDASSARRE, Leonardo et al. Modeling of rotor bow during hot restart in centrifugal compressors. In: Proceedings of the 39th turbomachinery symposium. Texas A&M University. Turbomachinery Laboratories, 2010.
- [34] BONALDI, E. L., DE OLIVEIRA, L. E. D. L., DA SILVA, J. G. B., LAMBERT-TORRES, G., DA SILVA, L. E. B. Predictive maintenance by electrical signature analysis to induction motors. In: Induction Motors-Modelling and Control. IntechOpen, 2012.
- [35] RIBEIRO JUNIOR, R. F.; DE ALMEIDA, F. A.; GOMES, G. F. Fault classification in three-phase motors based on vibration signal analysis and artificial neural networks. NEURAL COMPUTING & APPLICATIONS, 2020.
- [36] ZAREI, Jafar; TAJEDDINI, Mohammad Amin; KARIMI, Hamid Reza. Vibration analysis for bearing fault detection and classification using an intelligent filter. Mechatronics, v. 24, n. 2, p. 151-157, 2014.
- [37] Spectra Quest, Inc., "Machinery Fault Simulator", <http://spectraquest.com/machinery-fault-simulator/details/mfs>, 2021.
- [38] CHOUDHARY, Anurag et al. Condition monitoring and fault diagnosis of induction motors: A review. Archives of Computational Methods in Engineering, v. 26, n. 4, p. 1221-1238, 2019.
- [39] ABRAMAN, A. situação da Manutenção no Brasil. In: Anais do 26º Congresso Brasileiro de Manutenção. Curitiba: Abraman, 2011.
- [40] WANG, Kaikai et al. An Automatic Recognition Method of Microseismic Signals Based on S Transformation and Improved Gaussian Mixture Model. Advances in Civil Engineering, v. 2020, 2020.
- [41] WANG, Guo Feng; LI, Yu Bo; LUO, Zhi Gao. Fault classification of rolling bearing based on reconstructed phase space and Gaussian mixture model. Journal of Sound and Vibration, v. 323, n. 3-5, p. 1077-1089, 2009.
- [42] PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, v. 12, p. 2825-2830, 2011.
- [43] HOANG, Duy-Tang; KANG, Hee-Jun. Rolling element bearing fault diagnosis using convolutional neural network and vibration image. Cognitive Systems Research, v. 53, p. 42-50, 2019.
- [44] SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, v. 15, n. 1, p. 1929-1958, 2014.

- [45] DAHL, George E.; SAINATH, Tara N.; HINTON, Geoffrey E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013. p. 8609-8613.
- [46] BÜHLMANN, Peter; VAN DE GEER, Sara (2011). "Statistics for High-Dimensional Data ". Springer Series in Statistics.
- [47] KROGH, Anders; HERTZ, John A. A simple weight decay can improve generalization. In: Advances in neural information processing systems. 1992. p. 950-957.
- [48] CHOLLET, François et al. Keras: Deep learning library for theano and tensorflow. <https://keras.io>, 2021.
- [49] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, v. 25, p. 1097-1105, 2012.
- [50] ABDEL-HAMID, Ossama et al. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, v. 22, n. 10, p. 1533-1545, 2014.
- [51] HUANG, Zhengwei et al. Speech emotion recognition using CNN. In: Proceedings of the 22nd ACM international conference on Multimedia. 2014. p. 801-804.
- [52] CHAO, Qun et al. Cavitation intensity recognition for high-speed axial piston pumps using 1-D convolutional neural networks with multi-channel inputs of vibration signals. Alexandria Engineering Journal, v. 59, n. 6, p. 4463-4473, 2020.
- [53] KIRANYAZ, Serkan et al. 1D convolutional neural networks and applications: A survey. Mechanical systems and signal processing, v. 151, p. 107398, 2021.
- [54] ABDELJABER, Osama et al. Fault detection and severity identification of ball bearings by online condition monitoring. IEEE Transactions on Industrial Electronics, v. 66, n. 10, p. 8136-8147, 2018.
- [55] CANIZO, Mikel et al. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. Neurocomputing, v. 363, p. 246-260, 2019.
- [56] MURUGAN, Pushparaja. Implementation of deep convolutional neural network in multi-class categorical image classification. arXiv preprint arXiv:1801.01397, 2018.

- [57] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [58] PANIĆ, Branislav; KLEMENC, Jernej; NAGODE, Marko. Gaussian Mixture Model Based Classification Revisited: Application to the Bearing Fault Classification. *Strojnski Vestnik/Journal of Mechanical Engineering*, v. 66, n. 4, 2020.
- [59] BOTTOU, Léon. Stochastic gradient descent tricks. In: *Neural networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 2012. p. 421-436.
- [60] KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [61] DUBEY, Arun Kumar; JAIN, Vanita. Comparative study of convolution neural network’s relu and leaky-relu activation functions. In: *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*. Springer, Singapore, 2019. p. 873-880.
- [62] WANG, Xin; MAO, Dongxing; LI, Xiaodong. Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network. *Measurement*, v. 173, p. 108518, 2021. <https://doi.org/10.1016/j.measurement.2020.108518>.
- [63] INCE, Turker et al. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics*, v. 63, n. 11, p. 7067-7075, 2016. 10.1109/TIE.2016.2582729.
- [64] KOLAR, Davor et al. Fault diagnosis of rotary machines using deep convolutional neural network with wide three axis vibration signal input. *Sensors*, v. 20, n. 14, p. 4017, 2020. <https://doi.org/10.3390/s20144017>.
- [65] EREN, Levent; INCE, Turker; KIRANYAZ, Serkan. A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems*, v. 91, n. 2, p. 179-189, 2019. <https://doi.org/10.1007/s11265-018-1378-3>
- [66] HUANG, Wenyi et al. An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis. *Neurocomputing*, v. 359, p. 77-92, 2019. <https://doi.org/10.1016/j.neucom.2019.05.052>