UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

DIAGNÓSTICO DE RODAMIENTOS DEFECTUOSOS BASADO EN ANÁLISIS DE VIBRACIONES CON APLICACIONES EN EL MONITOREO DEL ESTADO DE AEROGENERADORES. UNA CLASIFICACIÓN ADICIONAL BASADA EN MACHINE LEARNING Y DEEP LEARNING

Trabajo Fin de Máster

Máster Universitario en Ingeniería Industrial (Acceso desde Grado I. Mecánica)

AUTOR/A: Cascales Fulgencio, David

Tutor/a: Quiles Cucarella, Eduardo

CURSO ACADÉMICO: 2021/2022

Máster Universitario en Ingeniería Industrial (Acceso desde Grado I. Mecánica)

AUTOR/A: Cascales Fulgencio, David

Tutor/a: Quiles Cucarella, Eduardo

CURSO ACADÉMICO: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

INDUSTRIAL ENGINEERING MASTER THESIS

VIBRATION ANALYSIS-BASED DIAGNOSIS OF FAULTY ROLLING ELEMENT BEARINGS WITH APPLICATIONS IN CONDITION MONITORING OF WIND TURBINES. A FURTHER MACHINE LEARNING AND DEEP LEARNING-BASED CLASSIFICATION

AUTHOR: David Cascales Fulgencio

SUPERVISOR: Eduardo Quiles Cucarella

Academic year: 2021-22

ACKNOWLEDGEMENTS

The author would like to thank his thesis supervisor, Professor Eduardo Quiles Cucarella, for guiding and advising him through this project. He would also like to acknowledge Professor Wade A. Smith, from the University of New South Wales (UNSW), for providing him with the mathematical tools necessary to diagnose faults in defective Rolling Element Bearings (REBs) correctly and for solving the doubts that arose at a theoretical and practical level.

RESUMEN

El monitoreo del estado de aerogeneradores es un área del mantenimiento predictivo que engloba las herramientas y técnicas necesarias para la supervisión del ciclo de vida de los diferentes elementos mecánicos que conforman las turbinas eólicas. Uno de los componentes de mayor interés por su exposición a elevados esfuerzos son los rodamientos, tendiendo a desgastarse con facilidad, apareciendo fallos en partes como la pista exterior, la pista interior además de los propios elementos rodantes. Las señales de vibraciones captadas con un acelerómetro contienen información esencial sobre estos fallos y el espectro de la envolvente ha demostrado ser una herramienta robusta para su diagnóstico en un estado de desarrollo temprano. En este proyecto, se han combinado diversas técnicas de preprocesamiento de señales (Discrete/Random Separation (DRS), Self-Adaptive Noise Cancellation (SANC) & Cepstrum Pre-Whitening (CPW)) con el análisis del espectro de la envolvente, para diagnosticar estos defectos en las señales de rodamientos de la famosa base de datos de la Case Western Reserve University (CWRU). Además, se han extraído y analizado diversas características, que han servido para que conocidos algoritmos de Machine Learning (ML) lleven a cabo una clasificación de dichas señales, encontrando correlaciones entre las características y los diferentes estados de salud de los rodamientos. Finalmente, se ha diseñado una Red Neuronal Convolucional (CNN) para implementar la clasificación de las señales de rodamientos desde el enfoque del Deep Learning (DL).

Palabras Clave: Monitoreo del Estado de Aerogeneradores; Rodamientos; Análisis de Vibraciones; Aprendizaje Automático; Aprendizaje Profundo.

RESUM

El monitoratge de l'estat d'aerogeneradors és una àrea del manteniment predictiu que engloba les eines i tècniques necessàries per a la supervisió del cicle de vida dels diferents elements mecànics que conformen les turbines eòliques. Un dels components de major interés per la seua exposició a elevats esforços són els rodaments, tendint a desgastar-se amb facilitat, apareixent fallades en parts com la pista exterior, la pista interior a més dels propis elements rodants. Els senyals de vibracions captades amb un acceleròmetre contenen informació essencial sobre aquestes fallades i l'espectre de l'envolupant ha demostrat ser una eina robusta per al seu diagnòstic en un estat de desenvolupament primerenc. En aquest projecte, s'han combinat diverses tècniques de preprocessament de senyals (Discrete/Random Separation (DRS), Self-Adaptive Noise Cancellation (SANC) & Cepstrum Pre-Whitening (CPW)) amb l'anàlisi de l'espectre de l'envolupant, per a diagnosticar aquests defectes en els senyals de rodaments de la famosa base de dades de la Case Western Reserve University (CWRU). A més, s'han extret i analitzat diverses característiques, que han servit perquè coneguts algorismes de Machine Learning (ML) duguen a terme una classificació d'aquests senyals, trobant correlacions entre les característiques i els diferents estats de salut dels rodaments. Finalment, s'ha dissenyat una Xarxa Neuronal Convolucional (CNN) per a implementar la classificació dels senyals de rodaments des de l'enfocament del Deep Learning (DL).

Paraules Clau: Monitoratge de l'Estat d'Aerogeneradors; Rodaments; Anàlisi de Vibracions; Aprenentatge Automàtic; Aprenentatge Profund.

ABSTRACT

Condition monitoring of wind turbines is an area of predictive maintenance that encompasses the tools and techniques necessary to supervise the life cycle of the different mechanical elements that build these machines. Rolling Element Bearings (REBs) are of the utmost vital parts since they tend to outwear easily due to their exposure to high loads. Different faults may appear in parts such as the outer race, inner race, and rolling elements. Vibration signals captured with an accelerometer carry essential information about these faults. The envelope spectrum has proven to be a robust tool for their diagnosis at an early stage of development. In this project, several signal preprocessing techniques (Discrete/Random Separation (DRS), Self-Adaptive Noise Cancellation (SANC) & Cepstrum Pre-Whitening (CPW)) have been combined with the envelope spectrum analysis to diagnose these faults in REBs signals from the famous Case Western Reserve University (CWRU) database. In addition, several features have been extracted and analysed, which have been used by widely renowned Machine Learning (ML) algorithms to classify the signals, finding correlations between such features and the different REBs' health states. Finally, a Convolutional Neural Network (CNN) has been designed to implement the classification of REBs signals from a Deep Learning (DL) approach.

Key Words: Condition Monitoring of Wind Turbines; Rolling Element Bearings; Vibration Analysis; Machine Learning; Deep Learning.

General Table of Contents

Document I REPORT

Document II BUDGET

Universitat Politècnica de València

Escola Tècnica Superior D'Enginyeria Industrial

Vibration analysis-based diagnosis of faulty rolling element bearings with applications in condition monitoring of wind turbines. A further machine learning and deep learning-based classification

Report

Author:

David Cascales Fulgencio

Supervisor:

Eduardo Quiles Cucarella

July 6, 2022

Document I

REPORT

Table of Contents

Acronyms

A

AGI Artificial General Intelligence.
AI Artificial Intelligence.
ANN Artificial Neural Network.
ANOVA Analysis Of Variance.

B

BPFI Ball Pass Frequency of the Inner Race.
BPFO Ball Pass Frequency of the Outer Race.
BSF Ball Spin Frequency.

C

CNN Convolutional Neural Network.
CPW Cepstrum Pre-Whitening.
CWRU Case Western Reserve University.

D

DL Deep Learning.
DoF Degrees of Freedom.
DRS Discrete/Random Separation.
DT Decision Tree.

F

FFNN Feed-Forward Neural Network.
FFT Fast Fourier Transform.

G

GD Gradient Descent.

K

k-NN k-Nearest Neighbors.

M

ML Machine Learning.

N

NAI Narrow Artificial Intelligence.
NB Naive Bayes.

O

OT Order Tracking.

R

REB Rolling Element Bearing.
RL Reinforcement Learning.
RMS Root Mean Square.

S

SANC Self-Adaptive Noise Cancellation.
SGD Stochastic Gradient Descent.
SGDM Stochastic Gradient Descent with Momentum.
SK Spectral Kurtosis.
SL Supervised Learning.
SNR Signal-to-Noise Ratio.
SVM Support Vector Machine.

U

UL Unsupervised Learning.
UNSW University of New South Wales.

1 Introduction

1.1 Motivation

Over the estimated 4470 M years of our planet's existence [Rudge et al., 2010], it has undergone numerous cyclical changes in its climate that have determined, positively or negatively, the development of various forms of life. However, from the 1970s onwards, the term "global warming" was coined, understood as an earth's climate forced alteration caused by humankind, far from a natural process to which the planet is subjected [Broecker, 1975]. This came about as the scientific community definitively accepted that the greenhouse effect, while necessary for life on earth, has been involved in most of the climatic changes it has experienced. Human emissions of various gases and aerosols into the atmosphere have contributed to the greenhouse effect, causing progressive and forced global warming, with negative consequences for life. From 1750 to 2022, atmospheric CO_2 concentrations have increased by 50% [NASA, 2022]. This is roughly the equivalent of 20k years of change concentrated in just 272 years.

To tackle global warming, eradicate poverty and ensure prosperity for humanity, on the 25th of September 2015, following the broadest and most participatory consultation process in the United Nations' history, its General Assembly adopted the 2030 Agenda for Sustainable Development, which sets the normative basis for meeting a series of goals by 2030 [UN, 2015]. It represents a consensus between governments, the private sector, academia and civil society representatives. Among the 17 goals set by this strategy is goal 7: "Ensure access to affordable, reliable, sustainable and modern energy". Within the previous goal's targets are 7A: "[...] enhance international cooperation to facilitate access to clean energy research and technology, including renewable energy [...]" and 7B: "[...] expand infrastructure and upgrade technology for supplying modern and sustainable energy services [...]". One of the clean energies that have gained the most momentum in recent years is wind energy, whose global installed capacity grew by 53% in 2020 to 743 GW, with more than 93 GW of newly installed capacity [GWEC, 2021].

Wind energy is generated by wind turbines that convert the wind's kinetic energy into electrical energy. The geometry of a wind turbine blade is designed to spin the rotor hub when positioned at a certain angle of attack with the wind. The torque is transmitted through a mechanical system consisting of shafts, couplings, Rolling Element Bearings (REBs) and gearboxes to an asynchronous machine specifically designed for this system, called a doubly-fed asynchronous machine. In this asynchronous machine, the stator windings are connected to the grid, and the rotor windings are connected to a variable frequency drive via slip rings. This variable frequency drive controls the rotor and grid currents, adjusting the active and reactive power fed into the grid from the stator independently of the rotor's rotational speed. Hence, the injection's frequency into the grid is always constant [Rodríguez Pozueta, 2016].

This thesis focuses on diagnosing mechanical faults in REBs by studying the envelope spectrum of vibration signals captured with an accelerometer in the first place and using Machine Learning (ML) and Deep Learning (DL) algorithms in the second place. Its motivation is framed within United Nations' 2030 Agenda for Sustainable Development's goal 7 and the two targets named above.

1.2 Objectives

This thesis aims to learn and apply the theoretical novelties introduced in recent years concerning the vibration-based diagnosis of faulty REBs, such as the benchmark method proposed in [Smith & Randall, 2015]. Such knowledge is of great use to the energy industry and, in particular, to the predictive maintenance of wind turbines, given that REBs are drive train elements that deteriorate easily due to their exposure to high loads. The Case Western Reserve University (CWRU) bearing database [CWRU, nd], which includes vibration signals from REBs with faults in the inner race, outer race and balls, will be used to perform the diagnosis.

In addition, the signals sampled at 12k Hz will be classified using ML algorithms such as Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbors (k-NN) and Naive Bayes (NB). For this purpose, several features will be extracted from each signal. These will be statistically analysed to determine their importance and then used to train the different models. Finally, a Convolutional Neural Network (CNN) will be designed to classify the signals from a DL point of view. For this purpose, 1-D time-domain signals will be converted into 2-D signal images, where each pixel of an image represents an original signal's feature. The CNN will find correlations between the features to classify the images according to the health state corresponding to the original REB. The models trained in this thesis can be used to diagnose the health status of never-seen-before REBs signals automatically.

2 Vibration Analysis. Theoretical Background

The torque is transmitted from the rotor hub to the doubly-fed asynchronous machine through a drive train in a wind turbine. From a mechanical vibration point of view, each drive train component excites the entire system at its corresponding characteristic frequency. When an accelerometer is placed at a certain point in the machine, the signal obtained is a sum of the convolutions of the excitations of the different components and the transfer path of these excitations to the measuring element, modelled mathematically as a specific impulse response function. A complete explanation of the above can be found in section 1.4.1 of Barszcz's book [Barszcz, 2019]. For those who need to remember the concept of convolution, refer to section 3.2.6 of Randall's book [Randall, 2010].

REBs are components whose function is to transmit the loads coming from the supported shaft. They consist of an outer race, an inner race, the rolling elements and a cage to maintain their relative position. When a spall appears on the surface of one of these parts due to fatigue, the rolling elements periodically impact it. These impacts excite the system in the form of periodic impulses at the corresponding fault's characteristic frequency, a function of the shaft rotation frequency, the REB geometry, the number of rolling elements and the load angle. The theoretical faults' characteristic frequencies that can occur on the surfaces of the different REB parts can be found in section 1.4.7 of Barszcz's book [Barszcz, 2019]. By computing the Fast Fourier Transform (FFT) (see section 3.2 of [Randall, 2010]) of a time-domain signal captured by an accelerometer, the harmonics of such excitation will be observed in a specific bandwidth, where the impulse's resonant frequency will be the fault's characteristic frequency. Usually, the fault information will be masked in a complex signal such as that of a wind turbine drive train, with multiple excitations and random noise sources.

The envelope spectrum is a widely used tool for diagnosing REB faults in an early stage of development and limited size. Information about the different excitation sources is extracted from the spacing between impulse responses and not from excited frequencies. Therefore, the envelope spectrum consists of a series of impulse responses in the frequency domain spaced at the characteristic frequency of the different excitation sources. The process for obtaining the envelope spectrum of a time-domain signal is described in section 2.4 of Barszcz's book [Barszcz, 2019]. When obtaining the envelope spectrum of a time-domain signal for diagnosing REB faults, it is essential to demodulate the signal in the bandwidth where the fault information is present. For this purpose, a tool called the Kurtogram can be used.

As mentioned above, the rolling elements' impacts on a spall excite the mechanical system in the form of periodic impulses at the corresponding fault's characteristic frequency. These impulses take the form of peaks in a time signal. Spectral Kurtosis (SK) is a remarkably sensitive indicator of a signal's peakedness. SK is a statistical method to detect non-Gaussian components in a signal, i.e. very impulsive components. Randall and Antoni demonstrated the usefulness of SK in detecting faults in rotating machines [Antoni & Randall, 2004]. The practical application of SK is the Kurtogram proposed by Antoni and Randall in the same paper, a 2-D colour map representing the SK at different levels and bandwidths of a signal. The Kurtogram returns the signal's most impulsive bandwidth and its centre frequency. Later, Antoni proposed an optimised version of the Kurtogram, called the Fast Kurtogram [Antoni, 2006]. This version reduces the number of variants for which the filter parameters are calculated without affecting the result's accuracy by applying the filter bank approach. The main drawback of SK is that it is susceptible to non-Gaussian random components external to the mechanical system, such as noise or interference of various sorts. A very elegant explanation of SK can be found in section 5.2 of Barszcz's book [Barszcz, 2019].

Occasionally, the fault information is so masked in the vibration signal of an accelerometer that the impulse responses spaced at the fault's characteristic frequency are not visible in the raw time signal's envelope spectrum. This usually occurs when the signal is corrupted by noise sources or random interference. On these occasions, it is advisable to pre-process the signal before proceeding to the study of its envelope spectrum. For this purpose, there are numerous techniques, such as Self-Adaptive Noise Cancellation (SANC) or Discrete/Random Separation (DRS), whose function is to separate the deterministic components from the random ones in a vibration signal. SANC is a self-adaptive filter that separates these components based on their different correlation lengths. The self-adaptive filter is fed with a reference signal, a delayed version of the primary one. DRS also achieves the separation of discrete and random components of a signal based on the transfer function between the primary signal and a delayed version of itself. It is more efficient than SANC because it works in the frequency domain, taking advantage of the FFT. The full explanation of these two techniques can be found in [Randall et al., 2011].

Additionally, Cepstrum Pre-Whitening (CPW) [Sawalhi & Randall, 2011] can be applied to a raw time signal to enhance the presence of REB's fault's impulse responses in the envelope spectrum and remove both unwanted components' harmonics and side-bands. For simplicity, REB fault signals have been treated as periodic. However, they are not strictly periodic: they are second-order cyclostationary. This means that their second-order statistic, the variance, is periodic but not the impulses themselves. The repetitive impulses of REB failures are not precisely periodic due to the rolling elements' slightly random location in the cage-free space and the non-exact cage's rotational speed due to slippage. This is why the REB fault information is not altered by applying CPW, as the cyclostationary components of a signal do not represent significant peaks in the cepstral domain. A detailed explanation of CPW can be found in [Borghesani et al., 2012].

3 Faulty Rolling Element Bearings Diagnostics

This section will display the three methods proposed in [Smith & Randall, 2015] to diagnose the different types of REB failures from the CWRU database.

3.1 Experimental Set-Up

The experiment layout can be found in [CWRU, nd]. The different geometries and faults' characteristic frequencies are detailed in Tables 9 to 12 in Appendix B for each REB. In summary, the test rig consisted of a two-hp electric motor, a torque transducer/encoder and a dynamometer. Faults were implanted into the different REBs' parts with an electro-discharge machine. Outer race faults were seeded at different positions relative to the REB load zone since they are stationary, having the fault's position an essential effect on the motor/REB system's vibration response. Vibration signals were captured with accelerometers. During each test, acceleration was measured at the drive end REB housing, the fan end REB housing, and the motor support base plate. Data was captured at 12k and 48k Hz sampling rates using a 16-channel data acquisition card.

3.2 Method 1: Raw Time Signals' Envelope Spectrum

This method analyses the envelope spectrum of the raw time signals' total bandwidth. The justification given by Smith and Randall for using this method is that in many of the signals, the different faults' information is not masked by other excitation sources. This project has applied this method to three 12k Hz sampled signals from the **6205-2RS JEM SKF** REB (drive end REB): IR021_0_DE, OR007@3_0_BA & OR007@6_0_FE and two 12k Hz sampled signals from the **6203-2RS JEM SKF** REB (fan end REB): IR014_0_FE & IR021_3_FE. The signals' breakdown can be found in Table 13 in Appendix B. MATLAB® Script 1 presents the code for the application of Method 1 to the IR021_0_DE signal. This code is identical for all the signals, taking into account the variables that may change depending on the case, such as the fault's characteristic frequency, the sampling frequency or the shaft rotation frequency. In each case, the envelope spectrum has been computed in the frequency band between 0 and the Nyquist frequency. This has been done to obtain the entire signal bandwidth's envelope spectrum without overlapping.

The results can be seen in Figures 1 and 2, representing the drive end and fan end REB signals, respectively. The impulse responses spaced at the fault's characteristic frequencies are visible in each signal's envelope spectrum. Also distinguishable, in each case, are the impulse responses spaced at the shaft rotation frequency, another source of excitation that has a significant presence in the signals' envelope spectra. In Figure 1, it is relevant to compare signals OR007@3_0_BA & OR007@6_0_FE. Both belong to the drive end REB and have a defect on the outer race of 0.007". Additionally, both were tested at a shaft rotational speed of approximately 1797 rpm.

MATLAB® Script 1: Method 1 applied to IR021_0_DE signal.

```
load('IR021_0.mat')
fs = 12000; %Units in Hz.
fshaft = X209RPM/60; %Units in Hz.
BPFI = 5.4152 * fshaft; %Units in Hz.
ps = 1/fs; %Units in s.
total_time_r = length(X209_DE_time) * ps; %Changes the signal length to seconds.
time_vector_r = linspace (0, total_time_r, length(X209_DE_time)); %Vector representing ...
    the measurement's total time in seconds.
plot (time_vector_r, X209_DE_time)
xlabel('Time, (s)')
ylabel('Acceleration (g)')
legend('Raw time signal')
xlim([0 5])
[es,f,env,t] = envspectrum(X209_DE_time,[fs], 'Method','demod','Band',[0 5999]); ...
    %Performs the input signal's squared envelope spectrum.
plot(f,es)
ncomb = 50; %Set numer of cursors to 50.
helperPlotCombs(ncomb, [fshaft BPFI]) %Plot cursors at fshaft, BPFI.
xlabel('Frequency (Hz)')
ylabel('Peak Amplitude')
legend('Envelope spectrum', 'fshaft', 'BPFI')
xlim([0 900])
```

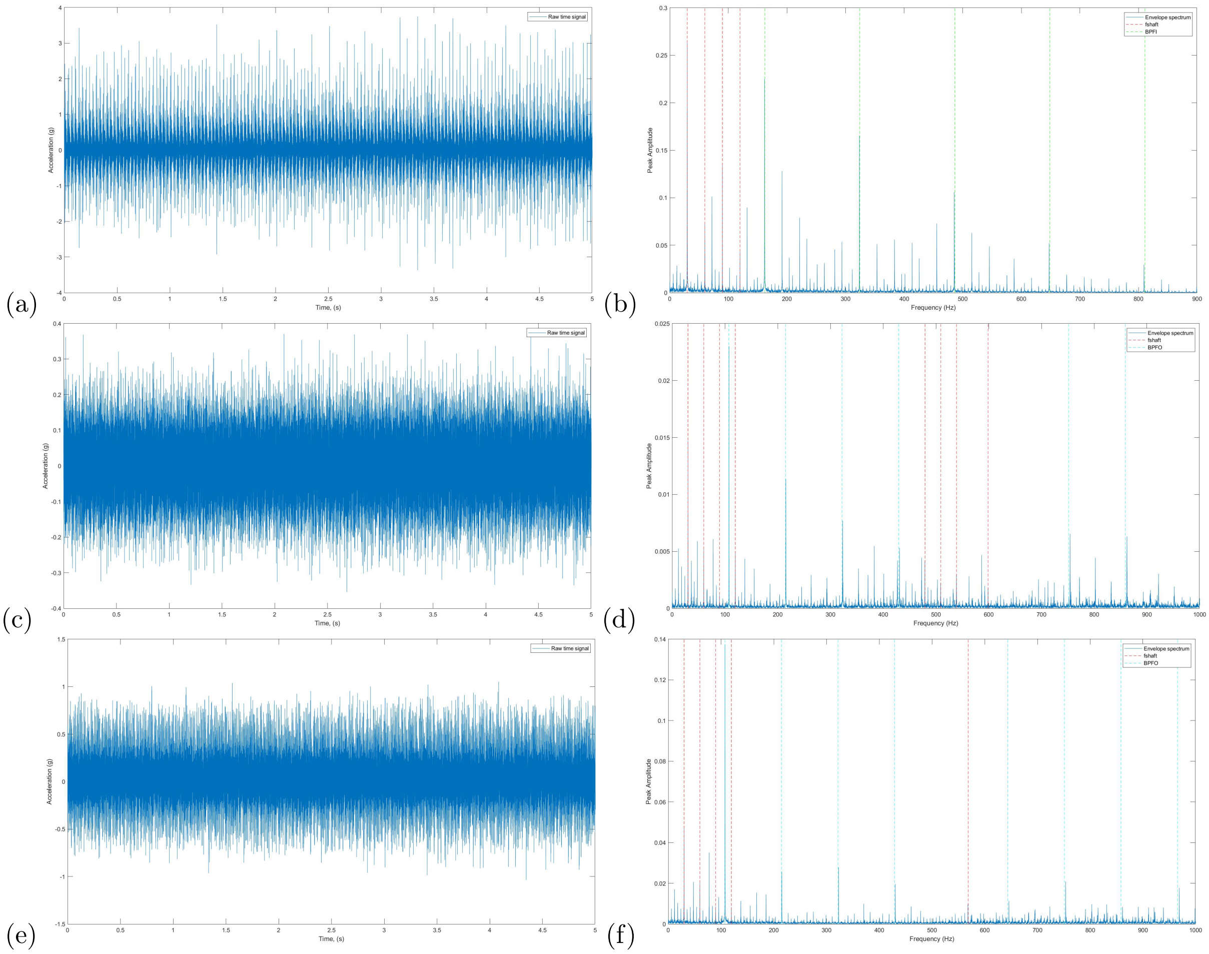


Figure 1: *Raw time-domain signals (a, c, e) and their corresponding envelope spectra (b, d, f) of IR021_0_DE (a & b), OR007@3_0_BA (c & d) and OR007@6_0_FE (e & f) signals.*

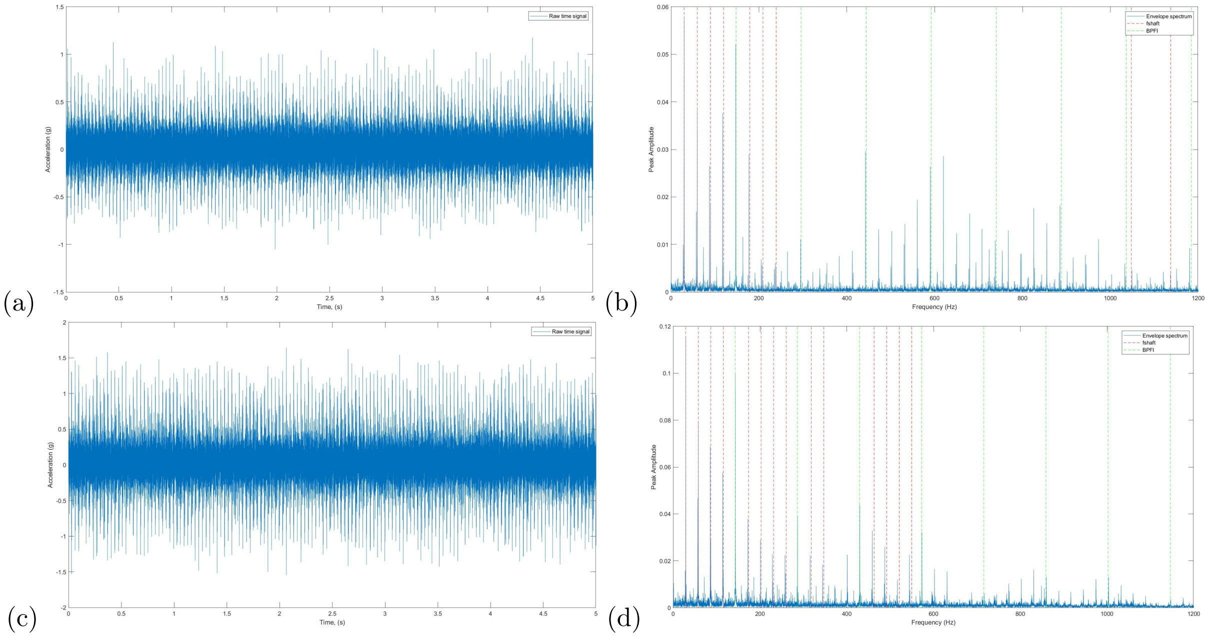


Figure 2: *Raw time-domain signals (a, c) and their corresponding envelope spectra (b, d) of IR014_0_FE (a & b) and IR021_3_FE (c & d) signals.*

Nonetheless, the defects are located in different positions relative to the REB load zone: in the first case, orthogonal to the load zone and in the second case, centred in the load zone. It can be seen how the envelope spectra of both signals are very similar, with the first four impulse responses spaced at the fault's characteristic frequency being noticeable. However, the impulse responses' maximum amplitude when the fault is centred in the REB load zone is approximately six times larger than when the fault is orthogonal to the load zone ($0.14 > 0.023$). This is because the rolling elements impact the fault with more pressure when it is located at the maximum load vertical than if it forms a 90° angle with the load zone. In Figure 2, both signals pertain to the fan end REB. The shaft rotational speeds were approximately 1797 and 1730 rpm, respectively. The faults were located on the REB's inner race with a diameter of 0.014" and 0.021" each. It can be seen from the respective envelope spectra that the impulse responses' maximum amplitude when the fault has a diameter of 0.021" is approximately twice as large as when the fault has a diameter of 0.014" ($0.1 > 0.05$). It can be concluded that the fault's size also significantly affects the maximum impulse response in the signal envelope spectrum.

3.3 Method 2: Cepstrum Pre-Whitening

This method will analyse the envelope spectra of temporal signals to which CPW [Sawalhi & Randall, 2011] has been employed to enhance the impulse responses of rolling element impacts with faults. This project has applied this method to one 12k Hz sampled signal from the **6205-2RS JEM SKF** REB (drive end REB): IR021_0_FE and three 12k Hz sampled signals from the **6203-2RS JEM SKF** REB (fan end REB): IR007_1_BA, IR014_3_BA & OR014@6_0_DE. The signals' breakdown can be found in Table 13 in Appendix B.

MATLAB® Script 2: Method 2 applied to IR007_1_BA signal.

```
x = real(ifft(fft(X279_BA_time) ./ abs(fft(X279_BA_time)))); %Performs the input ...
    signal's Cepstrum Pre-Whitening.
plot (time_vector_r, x)
xlabel('Time, (s)')
ylabel('Acceleration (g)')
legend('Time signal from Method 2 (Pre-Whitening)' )
xlim([0 5])
[es,f,env,t] = envspectrum(x,[fs],'Method','demod','Band',[0 5999]); %Performs the ...
    input signal's squared envelope spectrum.
plot(f,es)
```

```
helperPlotCombs(ncomb, [fshaft BPFI]) %Plot cursors at fshaft, BPFI.
xlabel('Frequency (Hz)')
ylabel('Peak Amplitude')
legend('Envelope spectrum from Method 2', 'fshaft', 'BPFI')
xlim([0 1446])
```

MATLAB® Script 2 presents the code for the application of Method 2 to the IR007_1_BA signal. Figure 3 shows the raw time signals and their corresponding envelope spectra over the entire bandwidth (Method 1). The only diagnosable signals are IR021_0_FE and OR014@6_0_DE. Figure 4 shows the temporal signals after applying CPW and their corresponding full-bandwidth envelope spectra (Method 2). It can be seen, firstly, that the peak amplitudes of the signals' envelope spectra, unlike in Figure 3, are all of the same order of magnitude ($\cdot 10^{-4}$).

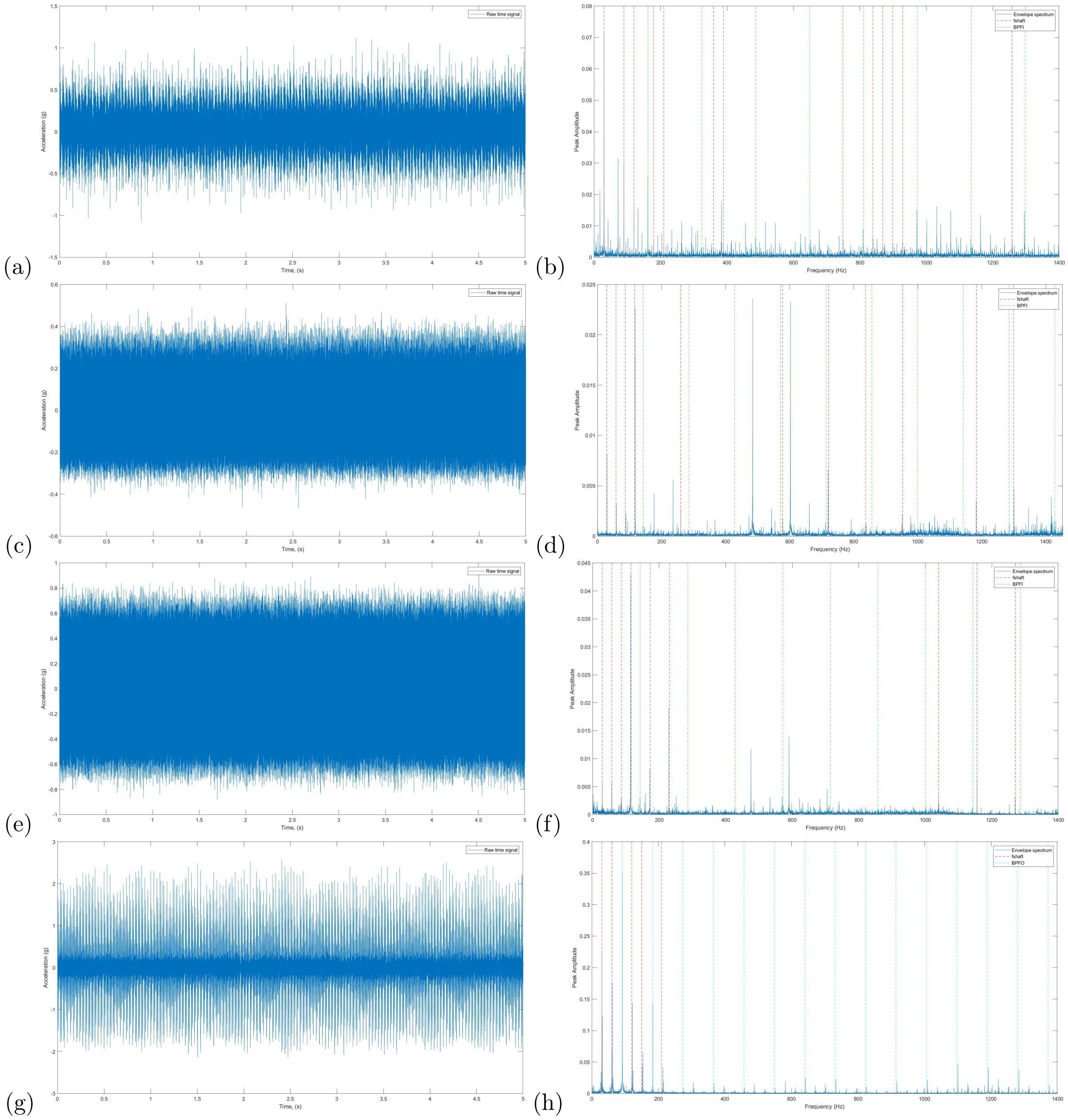


Figure 3: *Raw time-domain signals (a, c, e, g) and their corresponding envelope spectra (b, d, f, h) of IR021_0_FE (a & b), IR007_1_BA (c & d), IR014_3_BA (e & f) and OR014@6_0_DE (g & h) signals.*

As Randall and Smith point out in their paper, CPW equalises all frequency components to the same dimension. Secondly, it can be seen how the different faults' impulse responses have been enhanced in all cases. Signals IR021_0_FE and OR014@6_0_DE were already diagnosable by applying Method 1. However, Method 2 has provided a more straightforward analysis, with a more significant number of impulse responses visible. Signals IR007_1_BA and IR014_3_BA did not offer impulse responses typical of inner race faults using Method 1. In the respective envelope spectra of each of these signals, after Method 2, a significant impulse response spaced at the fault's characteristic frequency from the origin can be seen, enabling a correct diagnosis. In the previous section, when comparing the signals in Figure 2, it was concluded that a larger fault diameter resulted in a larger impulse responses' amplitude in the envelope spectrum.

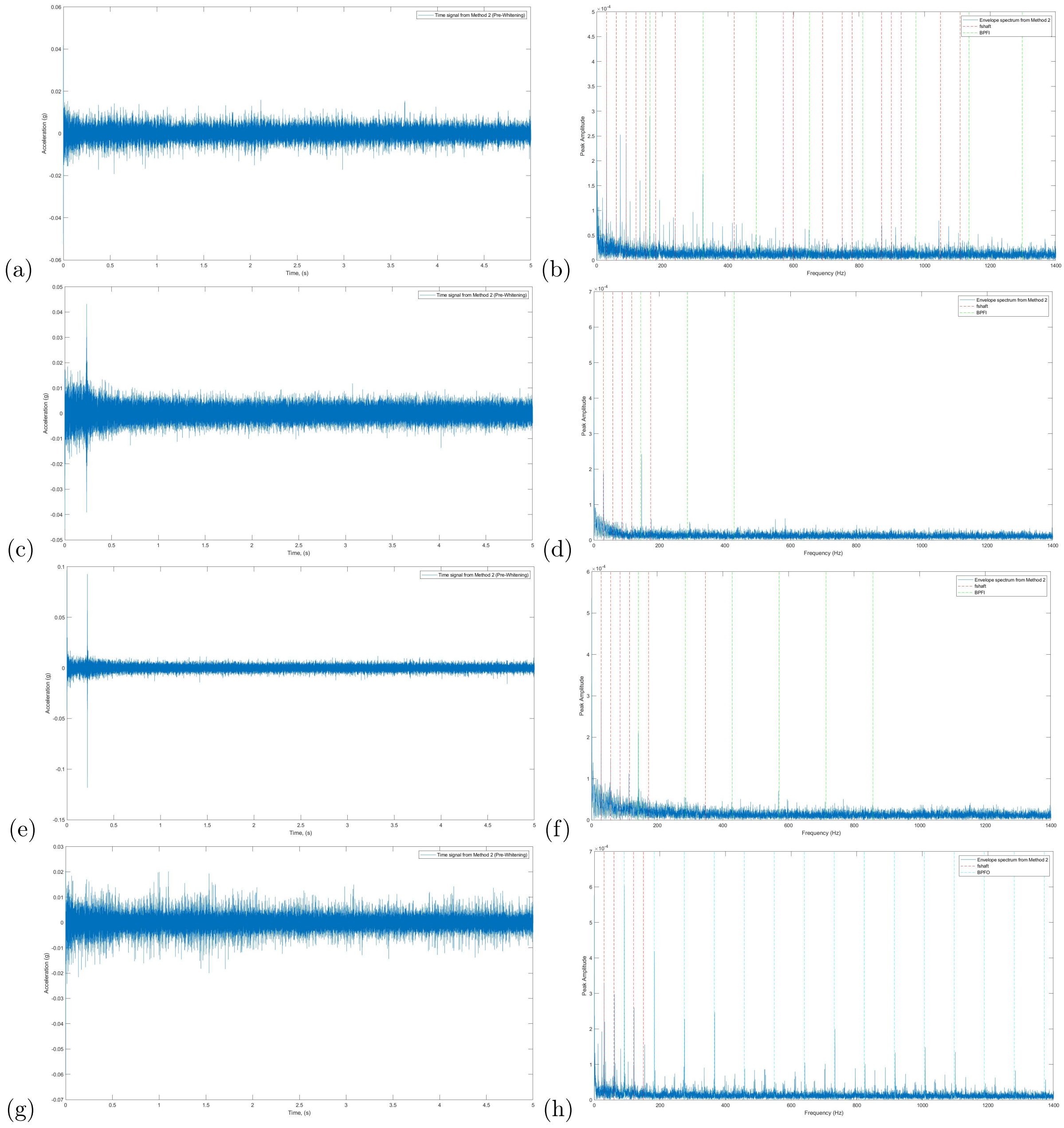


Figure 4: *Time-domain signals after CPW (a, c, e, g) and their corresponding envelope spectra (b, d, f, h) of IR021_0_FE (a & b), IR007_1_BA (c & d), IR014_3_BA (e & f) and OR014@6_0_DE (g & h) signals.*

In Figure 4, the same cannot be said for the IR007_1_BA and IR014_3_BA signals because CPW sets all the frequency components to the same magnitude, making it impossible to draw conclusions on the impulse responses' amplitude in this case.

3.4 Method 3: Benchmark Method

This section will present the Benchmark Method [Smith & Randall, 2015] implemented in four examples. This method involves applying Discrete/Random Separation (DRS) to the raw time signal to separate the deterministic components from the random ones. Subsequently, the processed signal's Kurtogram is studied, and the signal is filtered in the most impulsive bandwidth. Finally, the filtered signal's envelope spectrum is observed in search of impulse responses spaced at the fault's characteristic frequency to be diagnosed. The necessary settings for the DRS, the filter length N and the delay Δ were set by Randall and Smith in their paper to values of $N = 16384$ and $\Delta = 500$ (in number of samples) for signals obtained at a sampling frequency of 12k Hz. Another technique, Self-Adaptive Noise Cancellation (SANC), will be used to separate the deterministic from the random components in the signals. The objective is to compare the two techniques, although it is known beforehand that DRS is more efficient than SANC because it works in the frequency domain and benefits from the Fast Fourier Transform (FFT). The filter length N and the delay Δ will be kept at the same values as for DRS. The adaptation ratio μ will be set to a value of 0.05. Antoni's Fast Kurtogram [Antoni, 2006] will be used to analyse the processed signals' Spectral Kurtosis (SK).

Method 3 has been applied to four signals sampled at 12k Hz from the **SKF 6203-2RS JEM** REB (fan end REB): IR007_0_FE, IR014_3_FE, OR014@3_1_DE & OR021@3_2_BA. The signals' breakdown can be found in Table 13 in Appendix B. The code for applying this method to any temporal signal can be found in MATLAB® Script 3. The results are shown in Figures 5-9. Figure 5 shows the raw time signals and their corresponding envelope spectra. Figure 6 contains the time signals after processing with DRS and their corresponding envelope spectra. Figure 8 shows the time signals after filtering with SANC and their corresponding envelope spectra. Figures 7 and 9 contain the Kurtograms of the corresponding time signals processed with DRS and SANC, respectively.

MATLAB® Script 3: Method 3 applied to OR014@3_1_DE signal.

```matlab
[G,y,e,g]=DRS(X309_DE_time,500,16384,8192); %Performs DRS to the input signal.
plot (time_vector_r, e)
xlabel('Time, (s)')
ylabel('Acceleration (g)')
legend('Time signal from Method 3 (DRS)')
xlim([0 5])
nlevel = floor(log2(length(e)) - 7); %Define maximum number of levels for the kurtogram.
c = Fast_kurtogram(e,nlevel,fs); %Performs the input signal's Fast Kurtogram.
[es,f,env,t] = envspectrum(e,[fs],'Method','demod','Band',[0 5999]); %Performs the ...
    input signal's squared envelope spectrum.
plot(f,es)
helperPlotCombs(ncomb, [fshaft BPFO]) %Plot cursors at fshaft and BPFO.
xlabel('Frequency (Hz)')
ylabel('Peak Amplitude')
legend('Envelope spectrum from Method 3 (DRS)', 'fshaft', 'BPFO')
xlim([0 1400])
out = sanc(X309_DE_time,16384,0.05,500); %Performs SANC to the input signal.
plot (time_vector_r, out.filteredSignal)
xlabel('Time, (s)')
ylabel('Acceleration (g)')
legend('Time signal from Method 3 (SANC)')
xlim([0 5])
nlevel2 = floor(log2(length(out.filteredSignal)) - 7); %Define maximum number of levels ...
    for the kurtogram.
c = Fast_kurtogram(out.filteredSignal,nlevel2,fs); %Performs the input signal's fast ...
    kurtogram.
[es,f,env,t] = envspectrum(out.filteredSignal,[fs], 'Method','demod','Band', ...
    [5812.5-(375/2) 5999]); %Performs the input signal's squared envelope spectrum.
plot(f,es)
helperPlotCombs(ncomb, [fshaft BPFO]) %Plot cursors at fshaft, BPFO.
xlabel('Frequency (Hz)')
```

```
ylabel('Peak Amplitude')
legend('Envelope spectrum from Method 3 (SANC)', 'fshaft', 'BPFO')
xlim([0 1000])
```

IR007_0_FE has been an example of a signal poorly corrupted by random noise components. Its Signal-to-Noise Ratio (SNR) is -8.9125 dBc (decibels relative to the carrier), which means that the relative signal's strength is lower than the carrier signal's strength. The envelope spectrum in Figure 5 shows the impulse responses spaced at the fault's characteristic frequency, which allows a direct diagnosis without signal processing. After applying DRS to the signal and studying its Kurtogram, in Figure 7, it can be seen that, in general, the signal presents SK levels between 0 and 15 in all its bandwidth. Only a tiny bandwidth of 23.4375 Hz presents a SK of 46 in the low-frequency spectrum (centre frequency of 35.1563 Hz).

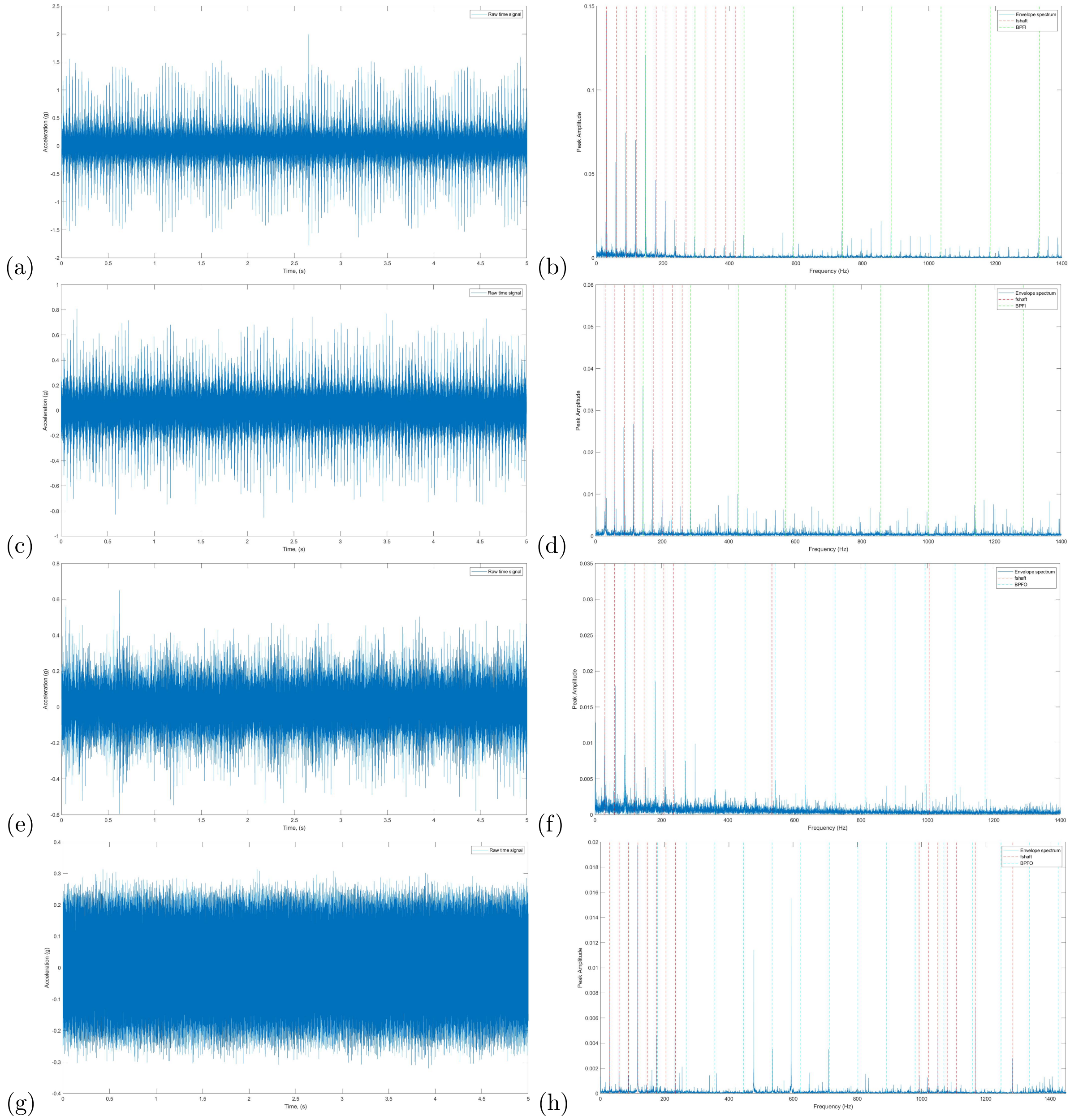


Figure 5: *Raw time-domain signals (a, c, e, g) and their corresponding envelope spectra (b, d, f, h) of IR007_0_FE (a & b), IR014_3_FE (c & d), OR014@3_1_DE (e & f) and OR021@3_2_BA (g & h) signals.*

It was found that this impulsivity was not caused by a REB inner race fault but by another source. As mentioned above, the Kurtogram does not return the bandwidth where the fault information is located but the most impulsive bandwidth, being sensitive to different sources of impulsivity. The time-domain signal's envelope spectrum processed with DRS was computed over the whole signal bandwidth, between 0 and the Nyquist frequency. The result is a precise diagnosis similar to that of Method 1. After filtering the signal with SANC and studying its Kurtogram in Figure 9, it can be said that, in general, the SK levels are much lower than those obtained from the signal processed with DRS. The most impulsive bandwidth is also very narrow in the low-frequency spectrum (SK of 8.2 at a centre frequency of 5.8594 Hz and a bandwidth of 11.7188 Hz). This source of impulsivity is the same as the one obtained in the previous case and is deterministic, as it has not been eliminated by either technique, although the fault in the inner race has not caused it.

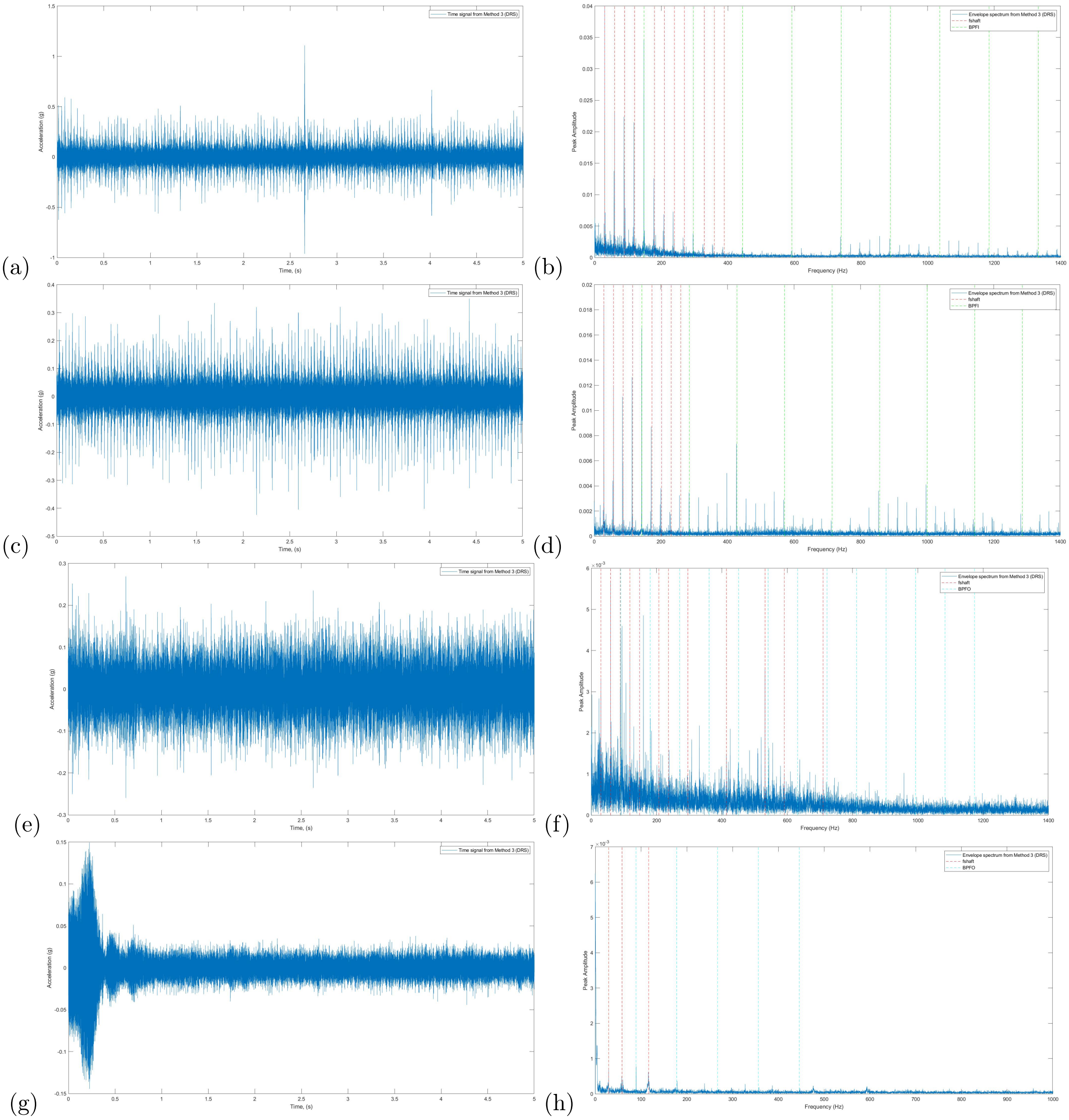


Figure 6: *Time-domain signals after DRS (a, c, e, g) and their corresponding envelope spectra (b, d, f, h) of IR007_0_FE (a & b), IR014_3_FE (c & d), OR014@3_1_DE (e & f) and OR021@3_2_BA (g & h) signals.*

After computing the envelope spectrum over the whole signal's bandwidth processed with SANC, the impulse responses spaced at the fault's characteristic frequency can be observed, which are already familiar from the two previous cases.

The signal IR014_3_FE has presented similar results to the previous one, with a SNR value of -10.4 dBc. The diagnosis has been perfectly satisfactory applying Method 1. After processing the raw time signal with DRS and analysing its Kurtogram, it is observed that the SK of almost all the signal bandwidth is between 0 and 10, with a narrow bandwidth of 11.7188 Hz presenting higher impulsivity (SK of 24.4) at a central frequency of 5.8594 Hz. As it is known that the REB's inner race fault does not cause these sources of impulsivity at low frequencies, the signal envelope spectrum has been analysed over the whole bandwidth, obtaining a sharper diagnosis than with Method 1, with more impulse responses spaced at the fault's characteristic frequency being identifiable. After filtering the raw signal with SANC and analysing its Kurtogram, a bandwidth of 750 Hz at a centre frequency of 5625 Hz was found to be much more impulsive than the rest of the signal, with a SK of 46.5. After computing the envelope spectrum in that particular bandwidth, the impulses spaced at the fault's characteristic frequency can be observed much more enhanced than in the previous cases, with the fault information present in that bandwidth.

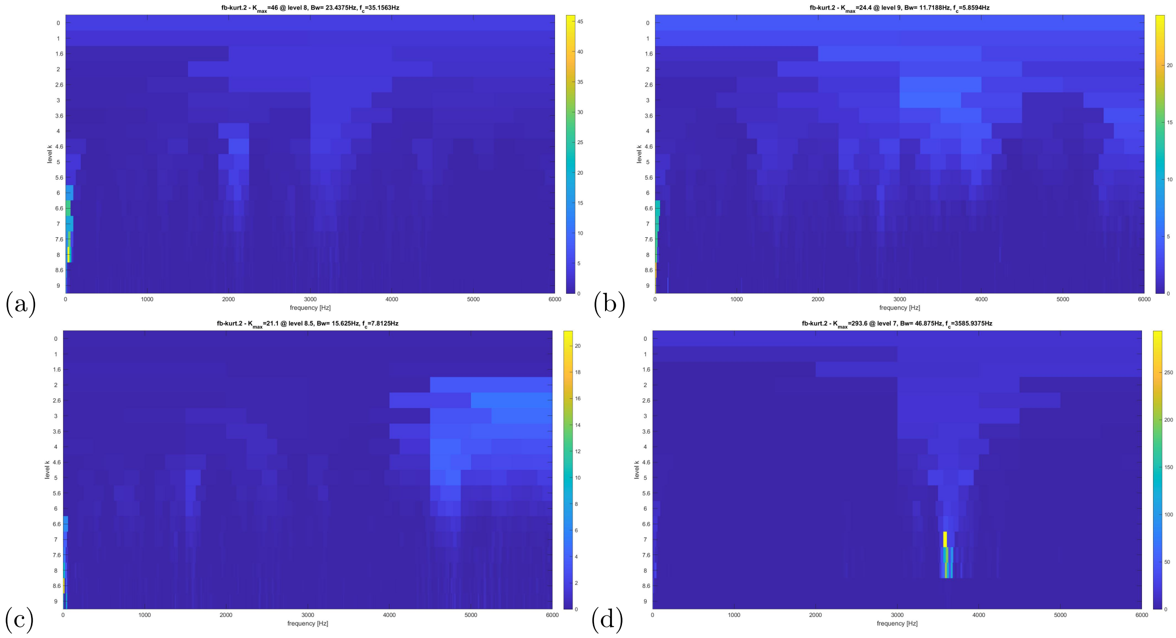


Figure 7: *Kurtogram after DRS of IR007_0_FE (a), IR014_3_FE (b), OR014@3_1_DE (c) and OR021@3_2_BA (d) signals.*

The signal OR014@3_1_DE, with an SNR of -15.1959 dBc, was also diagnosable applying Method 1. After applying Method 3 and studying the signal's Kurtogram, it was decided to calculate its envelope spectrum again over its entire bandwidth, obtaining an equally satisfactory diagnosis, although somewhat less clear than with Method 1. After applying SANC, once again, the processed signal's Kurtogram offered a more impulsive bandwidth than the rest (Spectral Kurtosis of 28.2 at a centre frequency of 5812.5 Hz and a bandwidth of 375 Hz), which turned out to be the bandwidth where the fault information is located given the unequivocal diagnosis obtained from the signal's envelope spectrum.

The signal OR021@3_2_BA, with an SNR of 11.1884 dBc, i.e. with a noise intensity higher than the carrier signal intensity, did not give a satisfactory diagnosis after applying Method 1. After processing the signal with DRS and observing its Kurtogram, the signal's SK was found to be between 0 and 100. However, a narrow bandwidth of 46.875 Hz at a centre frequency of 3585.9375 Hz presented a SK of 293.6. As this impulsivity source differed from those observed for the previous signals, it was decided to compute the processed signal's envelope spectrum in this bandwidth. Unfortunately, this impulsivity source was not caused by the fault in the REB's outer race, as the diagnosis was not satisfactory. After computing the processed signal's envelope spectrum over its total bandwidth, a very uninformative envelope spectrum is obtained, with impulse responses spaced at the fault's characteristic frequency and the shaft rotation frequency of little significance. However, the diagnosis has been considered successful since it is the first time that typical fault impulse responses have been observed in this signal. After filtering the signal with SANC and observing its Kurtogram, it is concluded that it is much less impulsive than the previous one, with a SK between 0 and 6 throughout its bandwidth. A narrow bandwidth of 11.7188 Hz at a centre frequency of 3626.9531 Hz with a SK of 11.2 is also identified. Again, this bandwidth is not the one where the fault information is located. After obtaining the signal's envelope spectrum over its entire bandwidth, the diagnosis was unsatisfactory, being poorer than that obtained from Method 1. Although not shown in the results, the best diagnosis for this signal was obtained by applying Method 2.

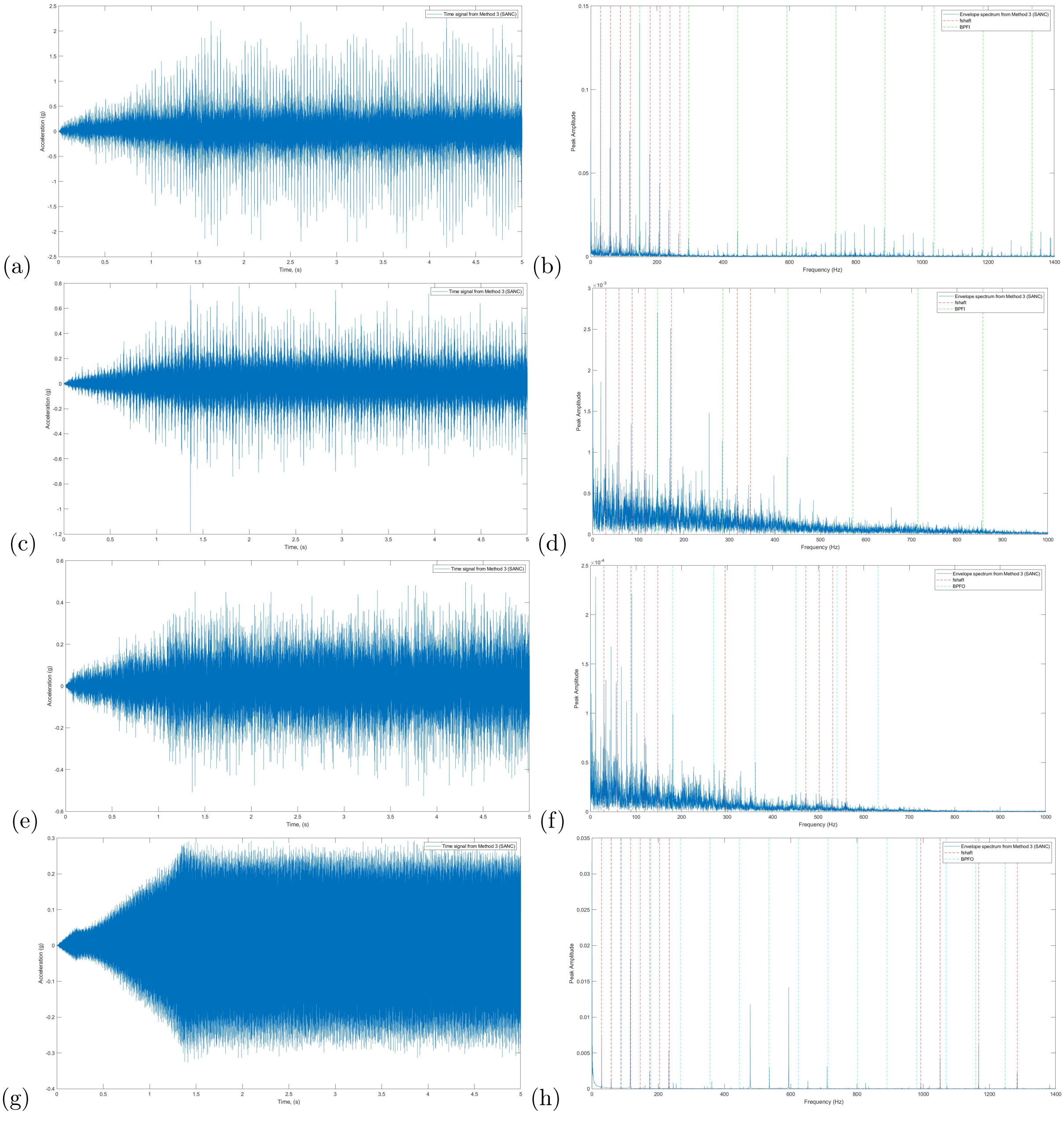


Figure 8: *Time-domain signals after SANC (a, c, e, g) and their corresponding envelope spectra (b, d, f, h) of IR007_0_FE (a & b), IR014_3_FE (c & d), OR014@3_1_DE (e & f) and OR021@3_2_BA (g & h) signals.*

In general, it was observed that signals whose noise intensity was lower than the carrier intensity were easier to diagnose than those that were not. In global terms, SANC required a more exhaustive study of the Kurtogram to identify the bandwidth where the fault was located, obtaining more precise diagnoses if the envelope spectrum was obtained over that bandwidth than in the entire signal bandwidth. In contrast, DRS gave the best results when the envelope spectrum was calculated across the processed signal's total bandwidth. Both methods have proven effective in separating signal's deterministic and random components, although SANC is more computationally time-consuming than DRS. This is due, as mentioned above, to the greater effectiveness of DRS because it works in the frequency domain.

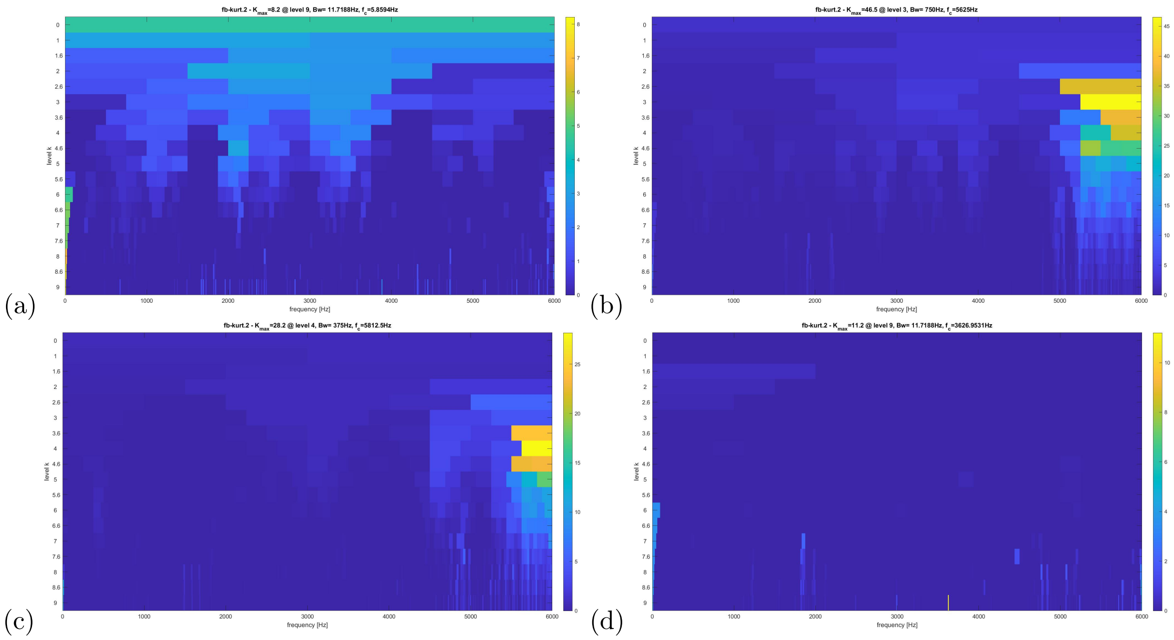


Figure 9: *Kurtogram after SANC of IR007_0_FE (a), IR014_3_FE (b), OR014@3_1_DE (c) and OR021@3_2_BA (d) signals.*

In this section, the three methods proposed in [Smith & Randall, 2015] have been applied to some examples from the CWRU database to demonstrate their effectiveness in diagnosing REB failures. Method 2 has proved to be the most effective, followed by Method 3. In this thesis, it has been decided not to present examples of REBs with defective rolling elements, as Randall and Smith demonstrated in their paper that these were the most complex cases to diagnose. This issue will recur in later sections, where it will be shown that REB signals with faulty rolling elements will be the most difficult to classify by ML and DL algorithms.

The three methods presented in this project are of superlative importance for predictive maintenance of wind turbines, although the variation of a wind turbine's operating conditions is an issue that cannot be overlooked. The CWRU data records were obtained at a constant shaft speed. This has significant advantages when analysing them. Although many techniques have been developed over the years to control the rotational speed of a wind turbine, the input variable of a wind turbine system is still the wind speed, which is essentially not constant and influences the vibrations of several machine's parts. Preprocessing steps such as Order Tracking (OT) (in Section 3.6.1 of [Randall, 2010]) would be needed to suppress the speed variation by eliminating the frequency modulation effects associated with variable speed.

4 Artificial Intelligence. Theoretical Background

Researchers have defined Artificial Intelligence (AI) differently over the last 70 years. Simmons and Chappell described AI as "the behaviour of a machine which, if a human behaves in the same way, is considered intelligent" in [Simmons & Chapell, 1988]. Rich proposed an alternative definition in [Rich, 1983]: "artificial intelligence is the study of making computers do things which, at the moment, humans do better". It is undoubtedly difficult to construct such a definition since it is directly linked to the concept of intelligence (understood as a human quality). Nonetheless, an accurate definition of AI might combine the two and many others over the years. "Artificial intelligence is a sub-discipline in the field of computing which seeks the creation of machines that can imitate intelligent behaviours". This definition has been widely used by scientific popularizers and professionals in the field, although it is unknown who suggested it in the first place.

The imitations referred to in the previous definition could be, for instance, analyzing patterns, driving, voice recognition or also defeating top players in a specific table game. There are many ways in which a machine can imitate intelligent behaviour. Unfortunately, what these machines will never do is something they are not programmed for. That quote seems both logical and pointless at the same time. However, there is a hidden meaning behind it: would machines ever be able to learn by themselves?

There are two types of AI: Narrow Artificial Intelligence (NAI) & Artificial General Intelligence (AGI) [Goralski & Tan, 2019]. We refer to NAI as those systems that can fulfil a minimal amount of tasks, such as playing chess, self-driving cars or image recognition. On the other hand, AGI is considered a more robust form of AI and includes those capable of being applied to various problems and domains. Although it remains theoretical to this date, it is idealized as "being able to perform as effectively as an extremely gifted human in all intellectual tasks" [Goralski & Tan, 2019]. Elaborating further on the use of the verb "to imitate" in the previous definition: AIs are programmed to perform a specific task as if they seemed intelligent. Nevertheless, if there is a quality that defines us as intelligent agents, it is learning.

Machine Learning (ML) is an Artificial Intelligence's branch, which seeks to provide machines with learning capacity [IBM, 2020], understood as the generalisation of knowledge from a set of experiences. There are three types of learning paradigms: Supervised Learning (SL), Unsupervised Learning (UL) & Reinforcement Learning (RL). We refer to SL as a type of learning based on discovering the relationship between some input and output variables. Learning arises from teaching these algorithms the result wanted to be obtained for a particular value. After showing it many examples, if the conditions are met, the algorithm will give a correct result even when it is shown values that it has not seen before. One extraordinary example is the algorithm created by Chris Danforth and Andrew Reece, which can detect if someone is suffering from depression, with 70% accuracy, just by analysing the pictures published on their Instagram account [Reece & Danforth, 2017]. Logistic regression, classification and back-propagation neural networks are some examples of using SL algorithms.

UL is the paradigm that produces knowledge only from the data provided as inputs, without explaining to the system what result is wanted to be obtained, i.e. without providing error signals to evaluate the potential solution [Sathya & Abraham, 2013]. Clustering, dimensional reduction, association, anomaly detection, etc., use this algorithm [Neupane & Seok, 2020]. The great advantage of UL versus SL is that it does not require the cost of processing and labelling the input data. The algorithm must find logical patterns by itself from the input data to arrive at a solution. RL involves training ML models to learn the first-class actions possible to achieve their goals, i.e. to make the best decisions in an uncertain, potentially complex environment. For the artificial intelligence to do what the programmer wants, it is rewarded or punished for the actions it performs. Its goal is to maximize overall reward. A learning paradigms' features summary is presented in Table 1.

Artificial Neural Networks (ANNs) are the techniques that have gained the most momentum in the last years. The most attractive fact about ANNs is that they learn hierarchically, i.e. the information is level-learned. The first levels are those where the basic concepts are learned, for example, what is a screw, a mirror and a tyre. This information is used in the following levels to learn abstract concepts, such as a car, motorbike, or truck. The only limit to the number of levels is the computational cost. The increment in the number of levels and their complexity is why these algorithms are known as Deep Learning (DL) algorithms.

A theoretical approach to AI is mandatory since it will be put into practice in the next section using the MATLAB® apps *Diagnostic feature designer™* & *Classification learner™* and the CWRU data sets. A Convolutional Neural Network (CNN) will also be designed to perform the CWRU's data record's DL-based classification. The three main types of learning paradigms (SL, UL and RL) are explained in detail in Section 1.2 of Burkov's book [Burkov, 2019]. The fundamental learning algorithms are presented in Chapter 3 of Burkov's book and Sections 3.4, 6.9 & 8.2 of Mitchell's book [Mitchell, 1997]. The most frequently used optimization algorithm, called Gradient Descent (GD), which finds the best parameters for the learning algorithms when the optimization criterion is differentiable, is covered in Section 4.2 of Burkov's book. Section 5.4 of Burkov's book defines the issues of underfitting and overfitting. The basics of ANNs and Deep Learning (DL) are described in Chapter 6 of Burkov's book.

Learning Paradigm / **Criteria**	**Supervised Learning**	**Unsupervised Learning**	**Reinforcement Learning**
Definition	Learns by using labelled data	Trained using unlabelled data without a guidance	Works on interacting with the environment
Type of data	Labelled data	Unlabelled data	No-predefined data
Type of problem	Regression and Classification	Association and Clustering	Exploitation or Exploration
Supervision	Extra supervision	No supervision	No supervision
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of actions
Application	Risk Evaluation, Forecast Sales	System Recommendation, Anomaly Detection	Self Driving Cars, Gaming, Healthcare

Table 1: *Differences between Supervised, Unsupervised and Reinforcement Learning.*

5 Faulty & Healthy Rolling Element Bearings Classification

In this section, the CWRU data sets diagnosed in Section 3 are going to be classified using some of the state-of-the-art learning algorithms: Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbors (k-NN) and Naive Bayes (NB). In order to achieve the classification, different features must be extracted from each signal. Furthermore, a Deep Learning-based classification will be accomplished using a Convolutional Neural Network (CNN).

5.1 Machine Learning-Based Classification

The Machine Learning-based classification problem will be solved using the MATLAB® app *Classification Learner™*, which trains models to classify data. Two sets of features will be extracted from each signal to accomplish the classification. The first group are time-domain features, such as basic statistics (mean, standard deviation, RMS and shape factor), high order statistics (kurtosis and skewness) and impulsive metrics (peak value, impulse factor, crest factor and clearance factor). Each feature belonging to this group will be calculated for each signal using the MATLAB® app *Diagnostic Feature Designer™* which, using a graphical interface, allows the development of diagnostic features and the evaluation of potential condition indicators.

The second group are frequency-domain features, such as the impulse response in the envelope spectrum, whose amplitude is the maximum of all impulse responses' amplitudes spaced at each fault's characteristic frequency for every signal (BPFI Amplitude, BPFO Amplitude and BSF Amplitude) and a logarithmic ratio between each one:

$$Log\left(\frac{BPFI\ Amplitude}{BPFO\ Amplitude}\right)$$

$$Log\left(\frac{BSF\ Amplitude}{BPFO\ Amplitude}\right)$$

$$Log\left(\frac{BPFI\ Amplitude}{BSF\ Amplitude}\right)$$

As demonstrated in Section 3, when a signal is demodulated correctly, a series of impulse responses will appear in the envelope spectrum spaced at the fault's characteristic frequency. When a fault affects a specific REB's part, the impacts with the fault will excite a particular frequency band in the form of repetitive impulses. If this band (including the transfer path from the place of impacts to the sensor) is selected for demodulation, the impulse responses in the envelope spectrum spaced at the actual fault's characteristic frequency will have a greater amplitude than those spaced at other fault's characteristic frequencies. The logarithmic ratio between the BPFI Amplitude and the BPFO Amplitude has been demonstrated to be a valuable feature for REBs' classification in [MathWorks®, 2013], hence included in this project along with the other logarithmic ratios. This set of features has been calculated for every signal using the MATLAB® function depicted in Listing 6 in Appendix A. This function calculates each raw time signal's Kurtogram and stores the centre frequency and the most impulsive bandwidth in every case. It then calculates the envelope spectrum in that bandwidth and measures its amplitude at the points spaced at the characteristic frequency of each type of fault. It then stores the maximum value of the impulse responses' amplitudes of each type of fault in three cell arrays and computes the logarithmic ratios between the features. Finally, it provides a table with all the data as a numeric array to be processed by the MATLAB® app *Classification Learner™*.

5.1.1 Feature Extraction and Analysis

The most important step when approaching a ML classification problem is selecting the best feature combination to achieve the most accurate results possible. This is of the utmost importance when the data set's size is limited. The faulty REBs vibration signals wanted to be classified belong to the drive and fan end REBs tables [CWRU, nd], accompanied by healthy REBs vibration signals. These vibration signals have in common that they were collected at a sample rate of 12 kHz. Thus, four groups are derived from the data set: healthy, inner race fault, outer race fault and ball fault. If vibration signals collected at 48 kHz had wanted to be included, they would have formed four different groups, taking into account the difference between sample rates. For simplification purposes, they have not been included in the data set. As mentioned in Section 3.1, each file contains fan end (FE), drive end (DE) and base (BA) accelerometers data. There are overall 307 vibration signals, 8 corresponding to healthy REBs, 76 to inner race fault REBs, 76 to ball fault REBs and 147 to outer race fault REBs.

Health condition	**Total data set**	**Class labels**
Normal	8	0
Inner race fault	76	1
Ball fault	76	2
Outer race fault	147	3

Table 2: *12k-sampled REBs signals. Health conditions and class labels.*

The time-domain features previously mentioned (along with their formulae) are summarized in Table 3, where x_i represents a data point within the signal and N, its length. These features are worth including since they have been used extensively in multiple pieces of research, such as the works of Magar et al. [Magar et al., 2021] and Samanta & Al-Balushi [Samanta & Al-Balushi, 2003], among others, delivering outstanding results. Typically, an in-depth analysis of the advantages of such features would be required to justify their use in a classification problem adequately. In this project, however, multi-class ranking methods such as one-way Analysis Of Variance (ANOVA) and the Kruskal-Wallis test will rank all features by their importance in a classification problem. Both ranking methods will be covered in this section. Besides, the frequency-domain features are of the utmost importance since signals from REBs are considered second-order cyclostationary. The envelope spectrum is a very efficient diagnostic tool for the faults mentioned above, as the information about the faults is extracted from the spacing between pulses and not from the excited frequencies.

The featurization's results are shown in Tables 14 & 15 in Appendix B, where a small portion of the 307 member data set is shown in each table together with their corresponding features. As already stated, the time-domain features have been calculated using the MATLAB® app *Diagnostic Feature Designer™*, whereas the frequency-domain features using the MATLAB® function shown in Listing 6 in Appendix A. The most natural questions to be asked at this point are: how are these features meaningful to the classification problem? Is any of them better than the rest? To answer these questions, the MATLAB® app *Diagnostic Feature Designer™* offers several classification ranking techniques. The ranking techniques for data sets containing more than two classes are the one-way Analysis Of Variance (ANOVA) and the Kruskal-Wallis test. Comparing the means of three or more unrelated groups within a dependent variable (feature), the one-way ANOVA determines whether there are statistically significant differences between them. A non-parametric alternative to the one-way ANOVA test is the Kruskal-Wallis test. Unlike one-way ANOVA, where means are compared, the Kruskal- Wallis test contrasts the samples' distribution to determine whether they belong to the same population. Both tests make assumptions that the data of each group in the dependent variable must meet. The Kruskal-Wallis test is known to be used if the one-way ANOVA's assumptions are not met for every class within the dependent variable since they are more potent than those of the Kruskal-Wallis test. Next, both tests will be briefly covered.

Number	**Feature**	**Formula**
1	Mean	$\frac{1}{N}\sum_{i=1}^{N} x_i$
2	Peak value	$max(\lvert x_i \rvert)$
3	RMS	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
4	Standard deviation	$\sqrt{\frac{\sum_{i=1}^{N}(x_i-\overline{x})^2}{N-1}}$
5	Clearance factor	$\frac{max(\lvert x_i \rvert)}{\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i \rvert}\right)^2}$
6	Kurtosis	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2\right)^2}$
7	Skewness	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2}\right)^3}$
8	Impulse factor	$\frac{max(\lvert x_i \rvert)}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i \rvert}$
9	Crest factor	$\frac{max(\lvert x_i \rvert)}{RMS}$
10	Shape factor	$\frac{RMS}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i \rvert}$

Table 3: *Time-domain features.*

One-Way ANOVA [NIST/SEMATECH, 2012]

One-way ANOVA tests the null hypothesis that all group means are equal ($H_0 : \mu_1 = \mu_2 = ... = \mu_j$) against the alternative hypothesis that at least one group mean is different from the others ($H_1 : \mu_i \neq \mu_j$ for at least one i and j) in a one-way layout. To perform one-way ANOVA, the total variation is calculated:

within-group variation and across-group variation. If the ratio of across-group variation to within-group variation (F) is larger than the critical value of the F-distribution with $(J-1,\ J(I-1))$ Degrees of Freedom (DoF) and a significance level of α [SOCR (b), 2002], the null hypothesis is rejected. Since the null hypothesis states that all group means are equal, large values of F rank better if what is wanted to be considered is the degree of difference between the groups to evaluate if the dependent variable is suitable for a classification problem. For the one-way ANOVA results to be reliable, the residuals of the $i-th$ observations belonging to the $j-th$ groups need to meet the following assumptions:

1. They have to be normally distributed in each group being compared in the one-way ANOVA. In practice, the dependent variable is tested to be normally distributed in each group rather than the residuals, since the results are the same,
2. There is homogeneity of variances (homoscedasticity) between each group of residuals. Again, the population variances in each group are tested to be equal, rather than the residuals' variances in each group,
3. Independence. The residuals (or rather the observations) need to be independent.

One-way ANOVA is known to be a robust test against the normality assumption, especially for large data sets. When the homoscedasticity condition has been violated, the Welsch-correlated ANOVA test has proved reliable. It has been stated that the most critical assumption to fail is the lack of independence between observations.

Kruskal-Wallis Test [NIST/SEMATECH, 2012]

In the Kruskal-Wallis procedure, is tested the null hypothesis that J groups from potentially different populations actually derive from a similar population, at least regarding their central tendencies or medians. As an alternative hypothesis, not all groups are derived from the same population. When the J groups are drawn from the same population, i.e. if the null hypothesis is true, then the test statistic, H, used in the Kruskal-Wallis procedure, is distributed approximately as a chi-square statistic with $J-1$ Degrees of Freedom (DoF). If H is larger than critical value of the chi-square distribution with $J-1$ Degrees of Freedom (DoF) and a significance level of α [SOCR (a), 2002], the null hypothesis is rejected. It can be concluded, since the null hypothesis states that the J groups derive from the same population, that large values of H rank better if what is wanted to be considered is the degree of difference between the groups to evaluate if the dependent variable is suitable for a classification problem. For the results of the Kruskal-Wallis test to be reliable, the observations within each group need to meet the following assumptions:

1. They do not have to be normally distributed in each group. However, the observations within each group have to belong to the same continuous distribution,
2. There is homogeneity of variances (homoscedasticity) between each group of observations,
3. Independence of observations.

The use of the Kruskal-Wallis test is recommended when the populations to be compared are clearly asymmetric, it is fulfilled that they are all in the same direction and that the variance is homogeneous.

To assess if the data within every group belonging to each feature is suitable for the results of the one-way ANOVA or the Kruskal-Wallis test to be reliable, several hypothesis tests [MathWorks®, 2021] are going to be applied using the MATLAB® Statistics and Machine Learning Toolbox™.

Anderson-Darling Test [NIST/SEMATECH, 2012]

Using the MATLAB® function *adtest(**x**)™*, the Anderson-Darling test returns a decision for the null hypothesis that the data in vector **x** is from a population with a normal distribution. The alternative hypothesis is that **x** is not from a population with a normal distribution. The result is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise.

Bartlett & Levene Test [NIST/SEMATECH, 2012]

The Homoscedasticity condition has been investigated using the MATLAB® function *vartestn(**X**)™*, which encompasses various techniques to test the homogeneity of variances between different groups of data. Two of these techniques are the Bartlett test and the Levene test. These tests return a decision for the null hypothesis that the columns of data matrix **X** have the same variance. The alternative hypothesis is that not all columns of data matrix **X** have the same variance. The Bartlett test should be used when data sets are from populations with a normal distribution. The Levene test should be used otherwise.

The result is the Bartlett test statics T, or the Levene test statics W. According to the Bartlett test, the variances are judged to be unequal if $T >$ the critical value of the chi-square distribution with $J-1$ Degrees of Freedom (DoF) and a significance level of α [SOCR (a), 2002]. The Levene test rejects the null hypothesis that the variances are equal if $W >$ the upper critical value of the F-distribution with $(J-1,\ J(I-1))$ Degrees of Freedom (DoF) and a significance level of α [SOCR (b), 2002].

For each test, the p-value (a scalar value in the range [0,1]) is given. It represents the probability of observing a test statistic as extreme as, or more extreme than the observed value under the null hypothesis. If the p-value is less than a chosen significance level, the null hypothesis is rejected. A 5% significance level has been chosen.

Kolmogorov-Smirnov Test [NIST/SEMATECH, 2012]

The equality of continuous distributions has been investigated using the MATLAB® function *kstest2(*$\boldsymbol{x}_1$*,* $\boldsymbol{x}_2$*)*™. The Kolmogorov-Smirnov test returns a decision for the null hypothesis that the data in vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are from the same continuous distribution. The alternative hypothesis is that $\mathbf{x}_1$ and $\mathbf{x}_2$ are from different continuous distributions. The result is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise.

The hypothesis tests results are summarized in Table 16 in Appendix B. As it can be observed, every feature within the data set violates at least the requirements of normality and equality of continuous distributions. Besides, only one feature fulfils the homoscedasticity condition. The independence of observations condition is known to be fulfilled for the whole data set since the features have been calculated from the raw time signals, which were independently obtained as described in Section 3.1, from different REBs having been placed in different parts of the set-up, in addition to the accelerometers.

It has been stated that the one-way ANOVA is a robust test against the violation of the normality assumption, especially for large data sets. The Kruskal-Wallis test is recommended only when the populations to be compared are clearly asymmetric. Since both are the only statistic tests offered by the MATLAB® app *Diagnostic Feature Designer*™ to be used as feature ranking for a multi-class Machine Learning-based classification problem, they are both going to be applied to the data set, being already aware that the results will probably not be reliable.

Prior to ranking any feature, let us observe various scatter plots of both data sets. Significant differences between the features' distributions for each group are aimed to be found. Figures 10 & 11 depict 3D scatter plots from the time- and frequency-domain features' data sets. As it can be drawn from these figures, the frequency-domain features seem to be the ones that take the most segregated values for each group, especially the logarithmic ratios between the maximum amplitudes of the envelope spectrum around the pulses spaced at the faults' characteristic frequency. The time-domain features seem to take less segregated values between the groups. Therefore, a lower accuracy from the ML algorithms is expected to be obtained from these features. Moreover, data belonging to the faulty balls class (2) is more spread than the other groups' data in the cases where a significant difference between the latter can be observed. It can be concluded that REBs signals with faulty balls will be the most difficult for the algorithms to classify.

Considering all of the above, six case studies will be defined.

1. Case A) The Machine Learning (ML) algorithms will compute the classification of REBs signals using the time-domain features of Table 3,
2. Case B) The ML algorithms will compute the classification of REBs signals using the frequency-domain features,
3. Case C) The ML algorithms will compute the classification of REBs signals using both the time- and frequency-domain features,

4. Case D) The ML algorithms will compute the classification of REBs signals using the time-domain features of Table 3. The 2^{nd} class, corresponding to REBs with faulty balls, will not be included,

5. Case E) The ML algorithms will compute the classification of REBs signals using the frequency-domain features. The 2^{nd} class, corresponding to REBs with faulty balls, will not be included,

6. Case F) The ML algorithms will compute the classification of REBs signals using both the time- and frequency-domain features. The 2^{nd} class, corresponding to REBs with faulty balls, will not be included.

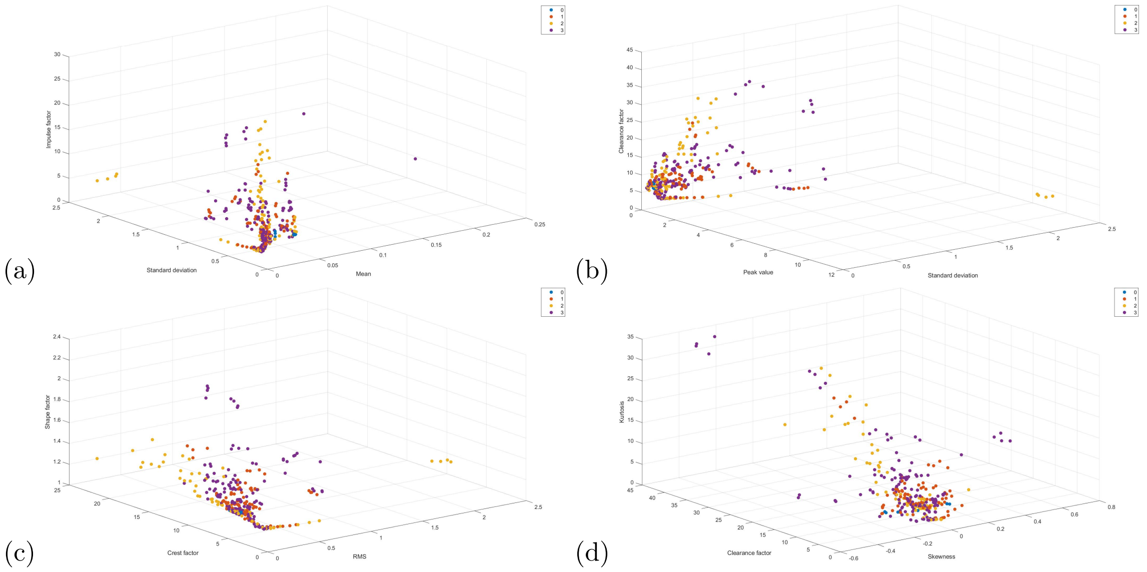


Figure 10: *Scatter plots of time-domain features. (a) Mean, Standard deviation & Impulse factor. (b) Standard deviation, Peak value & Clearance factor. (c) RMS, Crest factor & Shape factor. (d) Skewness, Clearance factor & Kurtosis.*

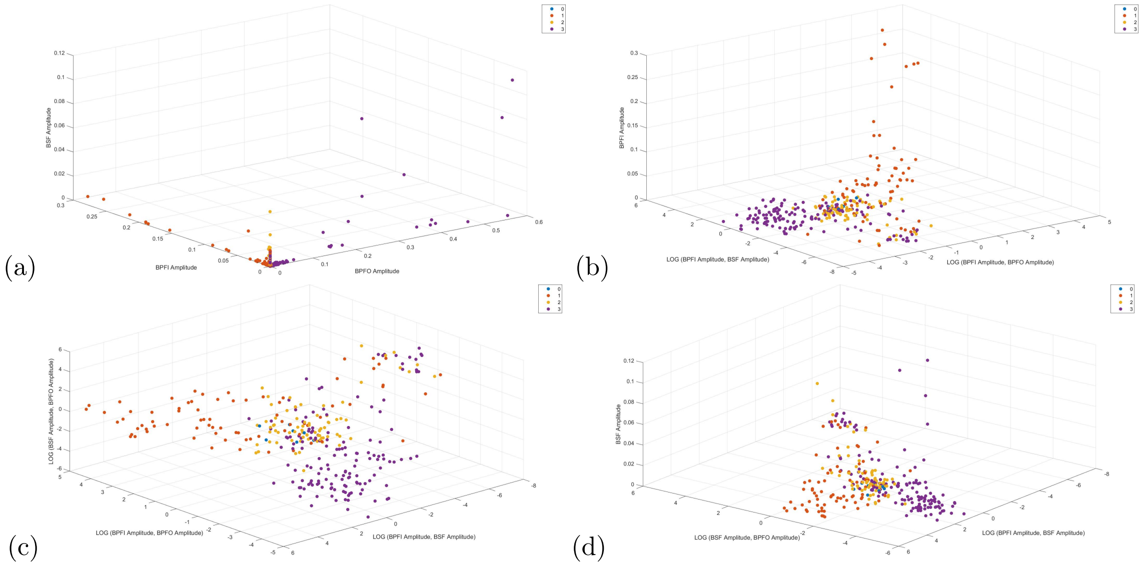


Figure 11: *Scatter plots of frequency-domain features. (a) BPFO Amplitude, BPFI Amplitude & BSF Amplitude. (b) Log(BPFI Amplitude, BPFO Amplitude), Log(BPFI Amplitude, BSF Amplitude) & BPFI Amplitude. (c) Log(BPFI Amplitude, BSF Amplitude), Log(BPFI Amplitude, BPFO Amplitude) & Log(BSF Amplitude, BPFO Amplitude). (d) Log(BPFI Amplitude, BSF Amplitude), Log(BSF Amplitude, BPFO Amplitude) & BSF Amplitude.*

The features will be ranked next, according to one-way ANOVA & the Kruskal-Wallis test, for every case. Since it has been proved that the results of these tests will not be rigorous, given the fact that the data do not fulfil their requirements (Table 16), the procedure to investigate which features will deliver the highest accuracy for every ML algorithm will be as follows: for every case, each ML algorithm will be computed with a feature vector that will be of dimension 1, firstly, increasing its dimensionality until it reaches its maximum, following the order of importance established by each statistical test. Therefore, two tables will be produced, each with the feature vector's dimensionality being increased in a different order. For each case and algorithm, taking into account the results of each pair of tables, those features offering a significant improvement in the accuracy of each algorithm concerning the best result previously obtained will be chosen. This way, the best results can be achieved with the fewest features possible, understanding which features are valid and which are not for every algorithm.

The results of the features' ranking are collected in Tables 17, 18, 19 & 20 in Appendix B. The statistical tests were run independently in all six cases. Cases C and F are a combination of cases A & B and D & E for every test. Cases C and F are summarized as bar graphs in Figure 12.

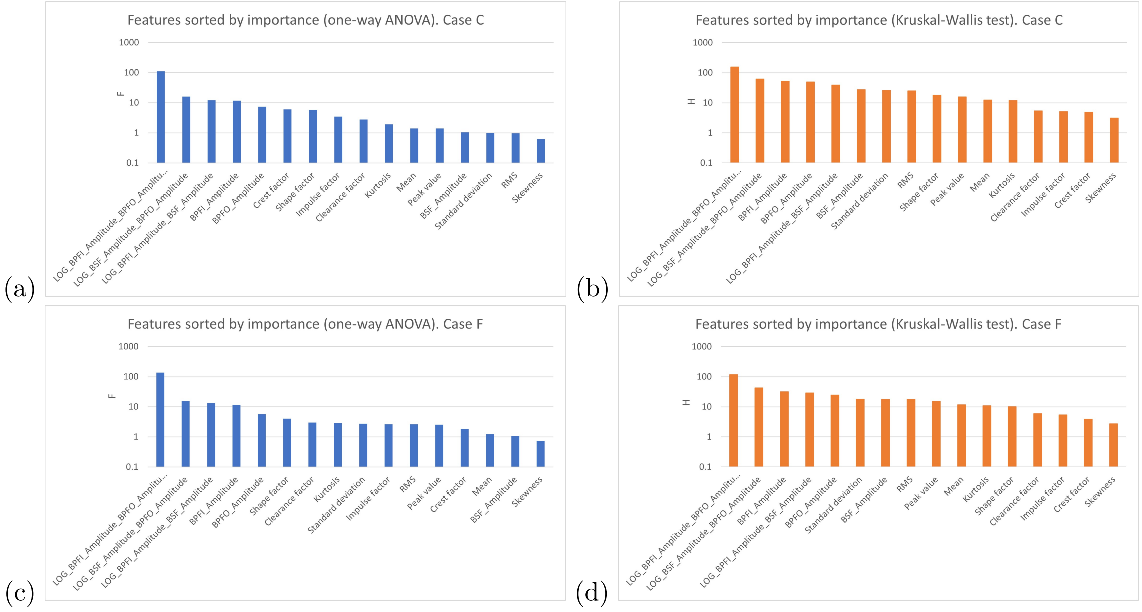


Figure 12: *Bar graphs of cases C & F for both the one-way ANOVA and the Kruskal-Wallis test.*

As it can be drawn from Figure 12, for every case, the frequency-domain features are ranked better than the time-domain features, especially the logarithmic ratios between the maximum amplitudes of the envelope spectrum around the pulses spaced at the faults' characteristic frequency. This matches the conclusions drawn from the scatter plots in Figures 10 & 11.

5.1.2 Classification Results

In the following section, the classification's outcomes of the CWRU's healthy and faulty REBs signals are going to be presented. The metric used to assess the different algorithms under the different cases is accuracy, defined as the proportion of correctly classified examples divided by the total number of examples within the data set. As mentioned in the previous section, in order to learn which features work better for every ML algorithm under each case, the procedure will be to compute each algorithm with a feature vector of dimension 1, firstly, increasing its dimensionality until it reaches its maximum for every case, following the order of importance established by each statistical test. Ultimately, the algorithms will be run for every case, selecting the features that significantly improve the accuracy over the best result previously acquired. Several line graphs will be obtained from the results, allowing the comparison between algorithms, cases and ranking tests.

Using the MATLAB® app *Classification Learner™*, models have been trained by applying five-fold cross-validation, using 80% of the data set for training and 20% for validation during each iteration. Five-fold cross-validation also protects against overfitting by estimating accuracy on each fold. The results are collected in Tables 21 & 23 in Appendix B, which correspond to Figures 13 & 14 and Tables 22 & 24 in Appendix B, corresponding to Figures 15 & 16.

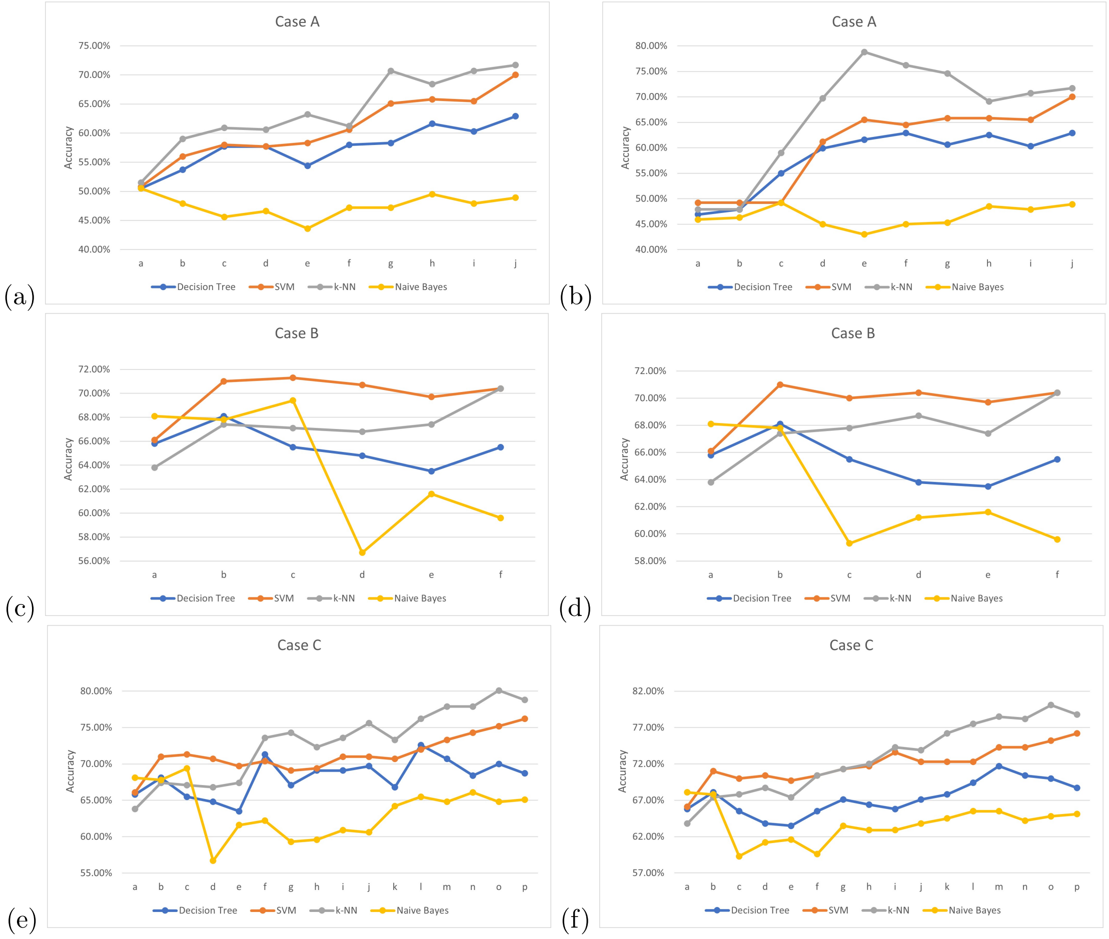
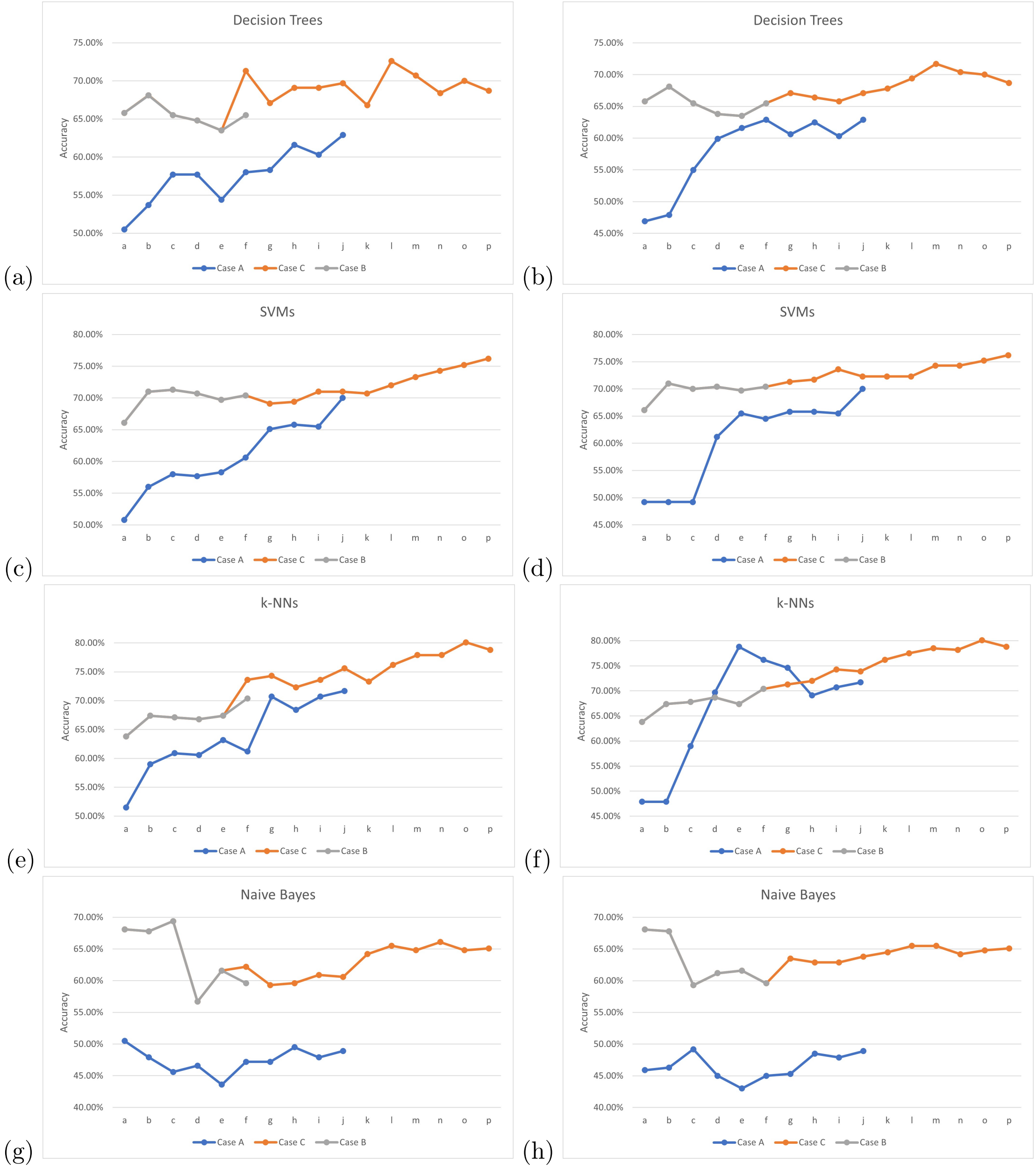


Figure 13: *Line graphs of cases A, B & C. Plots (a), (c) & (e) represent cases ordered by the one-way ANOVA ranking. Plots (b), (d) & (f) represent cases ordered by the Kruskal-Wallis test ranking.*

Figure 13 displays the algorithms' accuracy for cases A, B and C. The first thing to notice is that it generally improves when the feature vector's dimensionality increases. This is true, especially for the DT, SVM and k-NN models. In case B, however, the DT offers a lower accuracy, having been computed with six features rather than one. It can also be noticed that case B displays very similar results for both ranking methods. This is because the features' order, in this case, is very much alike for the one-way ANOVA and the Kruskal-Wallis test. On the other hand, the NB classifier rarely offers a better accuracy when more features are included in the feature vector. The exception can be found in case A, ordered by the Kruskal-Wallis test, improving the accuracy by 3% in the last classification over the first. Another characteristic regularly fulfilled in every case is the accuracy's instability for every algorithm as the feature vector's dimensionality is increased. This can be seen especially in case D, ordered by the one-way ANOVA, but also in cases A and B. Cases ordered by the Kruskal-Wallis test show a more steady development. The accuracy's instability is explained because some features do not contain important information for the classification of REBs and are better ranked than others because, as shown in the previous section, the data sets do not meet the requirements established by both ranking tests.

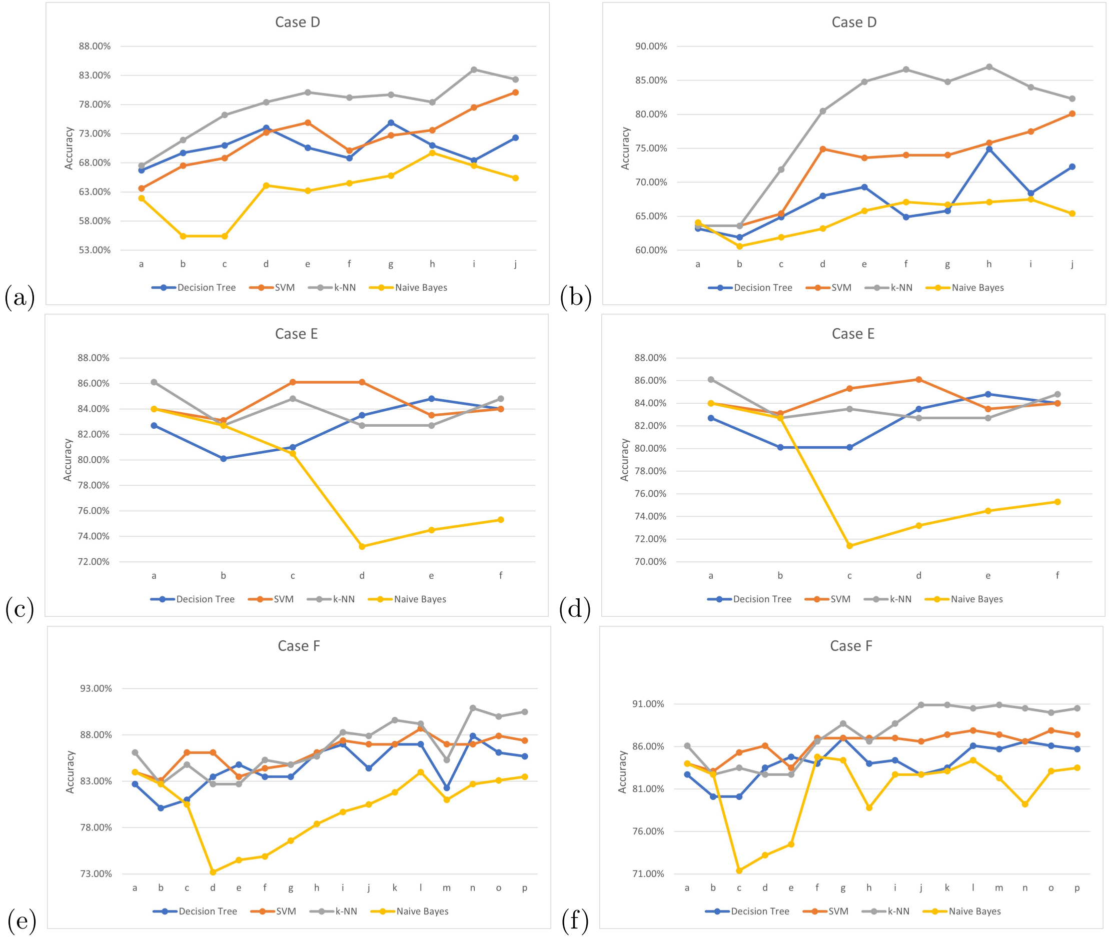
Besides, Figure 14 also shows the algorithms' accuracy for cases A, B and C, but in this case, the three cases are compared for every algorithm. It can be observed that the frequency-domain features, along with the time-domain ones, display a better accuracy than the time-domain features alone. This is one of the few facts that agree with the ranking of features since, as discussed in the previous section, the frequency-domain features are better positioned than the time-domain ones in all cases. An exception to this rule can be observed in the k-NN algorithm development for the case in which the features were ordered according to the Kruskal-Wallis test. Another remark is that the accuracy line of case B overlaps that of case C for every algorithm. This is also due to the ranking of features since case C ranking is a combination of cases A and B rankings, and the frequency-domain features generally have a higher score than the others. In quantitative terms, the k-NN algorithm displayed better results than the rest, followed by the SVM classifier and the DT. The NB classifier offered the poorest accuracy.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 14: *Line graphs of classification algorithms. Plots (a), (c), (e) & (g) represent cases ordered by the one-way ANOVA ranking. Plots (b), (d), (f) & (h) represent cases ordered by the Kruskal-Wallis test ranking.*

Figure 15 displays the algorithms' accuracy for cases D, E and F. Regarding the comments made for the previous cases, an upward trend in the accuracy is also observed. In these three cases, all algorithms' accuracy notably improves by eliminating the signals of REBs with faulty balls from the data set. Unlike the previous cases, the instability in the evolution of accuracy is much more evident. There is no longer a steady development in the accuracy of cases whose features have been ordered by the Kruskal-Wallis tests. Concerning the NB algorithm, it is also true in these three cases, in general, that its precision does not improve as the feature vector's dimensionality increases to its maximum. The exception is found again in case D, although there is no significant improvement.



(a) (b) (c) (d) (e) (f)

Figure 15: *Line graphs of cases D, E & F. Plots (a), (c) & (e) represent cases ordered by the one-way ANOVA ranking. Plots (b), (d) & (f) represent cases ordered by the Kruskal-Wallis test ranking.*

Figure 16 also shows the algorithms' accuracy for cases D, E and F, but the three cases are compared for every algorithm in this instance. It is necessary to comment again on the remarkable increase in the algorithms' accuracy for these three cases to the previous ones. Overall, the k-NN offered the highest accuracy, followed by the SVM and the DT. Finally, the NB classifier showed the poorest accuracy again. The rest of the observations made for the previous cases are also fulfilled in these three cases.

Tables 4 & 5 show the maximum accuracy obtained for every algorithm. Cases A, B, D and E are not included since it has been proved that the combination of frequency- and time-domain features yield better results than time-domain features alone. Moreover, the accuracy lines of cases B and E overlapped those of cases C and F because the frequency-domain features are better positioned in the ranking than the time-domain ones in all cases. In Table 4, it can be seen that, out of the six frequency-domain features, the logarithmic ratios between the maximum amplitudes of the envelope spectrum around the pulses spaced at the faults' characteristic frequency have been the most important for the classification of healthy and faulty REBs. For the time-domain features, the crest factor and the peak value have been demonstrated to be the most important, followed by the clearance factor, mean, standard deviation and RMS. The SVM and k-NN algorithms have required a higher feature vector's dimensionality than the rest to yield the best results. In the SVM algorithm case, the skewness and the BSF Amplitude, two features that have rarely worked well, have also been included.

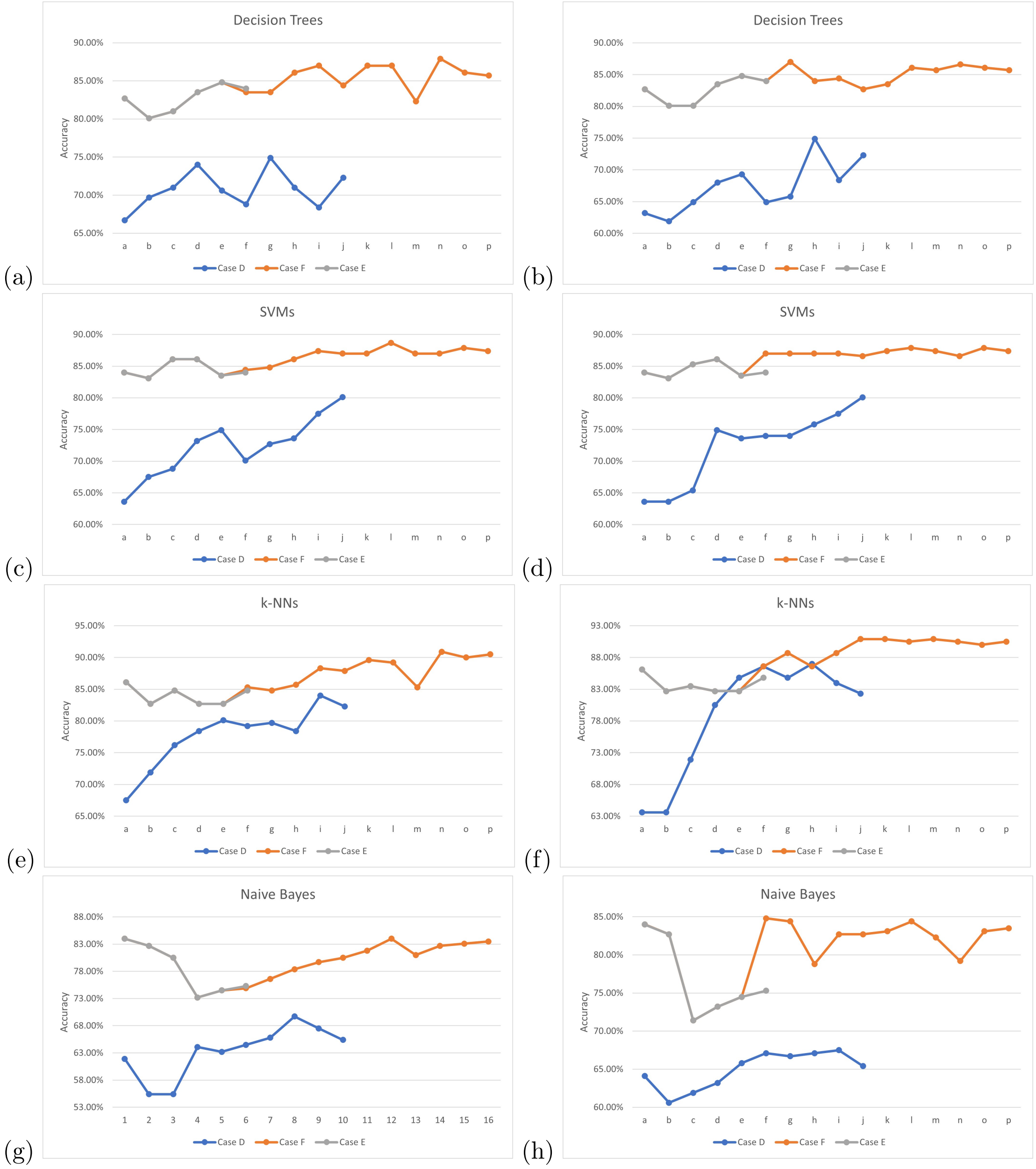


Figure 16: *Line graphs of classification algorithms. Plots (a), (c), (e) & (g) represent cases ordered by the one-way ANOVA ranking. Plots (b), (d), (f) & (h) represent cases ordered by the Kruskal-Wallis test ranking.*

In Table 5, it can be observed that the algorithms have generally needed fewer features to reach these results and offered a better accuracy. This makes sense taking into account the line plot of case F, whose instability was much higher than the one of case C, meaning that when signals of REBs with faulty balls are eliminated from the data set, fewer features help to reach higher accuracy. Among the frequency-domain features, the most important one has been demonstrated to be the logarithmic ratio between the BPFI Amplitude and the BPFO Amplitude. By all of the time-domain features, the most important ones have been demonstrated to be the standard deviation, kurtosis and mean. The SVM and k-NN algorithms have again needed more features to reach the best results, including the clearance factor, peak value, RMS and skewness.

Case C				
Features \ **Algorithms**	DT (72.30%)	SVM (77.50%)	k-NN (80.10%)	NB (69.70%)
LOG_BPFI_Amplitude_BPFO_Amplitude	x	x		x
LOG_BSF_Amplitude_BPFO_Amplitude	x		x	
LOG_BPFI_Amplitude_BSF_Amplitude		x	x	x
BPFI_Amplitude				
BPFO_Amplitude				
BSF_Amplitude		x		
Crest factor	x	x	x	
Shape factor			x	
Impulse factor				
Clearance factor		x	x	
Kurtosis				
Mean		x	x	
Peak value	x	x	x	
Standard deviation		x	x	
RMS		x	x	
Skewness		x		

Table 4: *Maximum accuracy of all algorithms for case C and their corresponding features.*

Case F				
Features \ **Algorithms**	DT (90.00%)	SVM (89.60%)	k-NN (92.20%)	NB (86.10%)
LOG_BPFI_Amplitude_BPFO_Amplitude	x	x	x	x
LOG_BSF_Amplitude_BPFO_Amplitude				
LOG_BPFI_Amplitude_BSF_Amplitude				
BPFI_Amplitude				
BPFO_Amplitude				
BSF_Amplitude				
Crest factor				
Shape factor				
Impulse factor				
Clearance factor			x	
Kurtosis	x	x	x	
Mean	x	x	x	
Peak value		x		x
Standard deviation	x	x	x	x
RMS			x	
Skewness			x	

Table 5: *Maximum accuracy of all algorithms for case F and their corresponding features.*

5.2 Deep Learning-Based Classification

In the following section, a Convolutional Neural Network (CNN) will be built to classify REB signals from a DL point of view. CNNs are a Feed-Forward Neural Network (FFNN) initially designed for image processing. Therefore, the CWRU's healthy and faulty REBs signals must be transformed into images. Every pixel within an image will represent a single feature. CNNs are known to deliver outstanding results when the training data set size is considerably large [Zhang et al., 2017]. In this section, its performance will be tested using a data set of a limited extent (307 REBs signals, as shown in Table 2).

5.2.1 Feature Extraction and Analysis

To transform a 1-D time-domain signal into a 2-D image to serve as an input to the CNN, N data points are split off from a signal $\mathbf{x}$. Each data point is aligned sequentially, shaping m rows of n points. A matrix is then built by placing the m rows one onto the following. To create square images of acceptable size while preserving as much information as possible about the defect, a 40k data points signal section has been transformed into a 200 x 200 pixels image. The REBs signals sampled at 12k Hz have approximately

120k data points, given that each reading took about 10 seconds. Suppose the test stand's shaft rotational speed is 1730 rpm. In that case, a single rolling element will impact approximately 29 times against the fault every second, giving a total of 87 impacts of each rolling element with the fault during these three seconds (this pattern is not valid if the defect is in the rolling element). In mathematical terms, the transformation described above looks like this:

$$\mathbf{I} = \begin{bmatrix} x(t) & \dots & x(t+n-1) \\ \vdots & \ddots & \vdots \\ x(t+(m-1)n) & \dots & x(t+mn-1) \end{bmatrix}$$

Where **I** denotes the signal image and $x(t)$ the vibration data point of time t. The above can be written in MATLAB® code as follows:

MATLAB® Script 4: 1-D time-domain signal transformation into a greyscale 2-D image.

```
load('IR007_0.mat')
X105_DE_time_section = X105_DE_time(1:40000);
X105_DE_time_maxtrix = reshape(X105_DE_time_section,[200,200]);
M = max(X105_DE_time_maxtrix(:));
m = min(X105_DE_time_maxtrix(:));
imshow(X105_DE_time_maxtrix,[m M])
```

Some signal images are depicted in Figure 17:

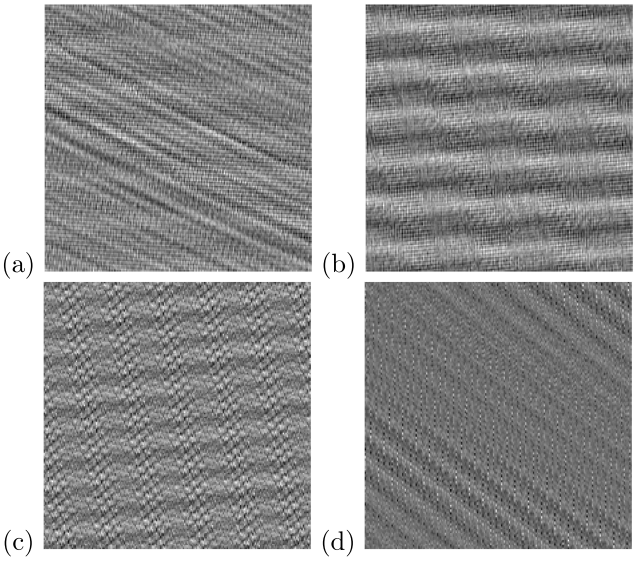


Figure 17: *Greyscale images of signals B007_0_BA (a), B014_1_BA (b), IR007_2_FE (c) & OR021@3_0_DE (d).*

5.2.2 Classification Results

The architecture of the proposed CNN model is shown in Listing 5. It comprises one input layer responsible for receiving external data, two hidden layers accountable for filtering the inputs, a fully connected layer responsible for the classification, and an output layer. Each hidden layer comprises a convolutional layer, a batch normalisation layer, an activation layer, and a max-pooling layer. The convolutional layer convolves the local input regions with filter kernels and then generates the output features by computing the activation function. The *ReLU* function is used as the activation layer to sort out the issue of vanishing gradients. The batch normalisation layer, placed between the convolutional layer and the activation layer, helps to reduce the CNN's sensitivity to network initialisation. It is the procedure to set the weights of an ANN to small random values that define the starting point for the optimization, stabilising the learning process and dramatically reducing the number of training epochs, which speeds up the training. Finally, a max-pooling layer is placed between the first activation layer and the second convolutional layer. It stores the maximum value of every local input throughout the filtering process with kernels, reducing the input size for the second convolutional layer. This is known to improve the CNN's accuracy. The fully connected layer comprises as many neurons as there are labels, four in this case. The *softmax* function is used as an activation layer for the latter.

Every ML algorithm aims to minimize a loss/cost function or maximize a likelihood function, depending on the existing model, to find its optimal values and achieve the most accurate prediction. In a CNN, every unit within a convolutional layer is a slight regression model trained by learning the filter kernel's parameters. The loss function, in this case, is defined by the task. For image classification, the cross-entropy loss function is used to compute the difference between the softmax output probability distribution and the label probability distribution. The Stochastic Gradient Descent with Momentum (SGDM) is used to minimise the cross-entropy loss function. It is an upgraded version of the Stochastic Gradient Descent (SGD) that accelerate gradients vectors in the right directions.

The hyper-parameters for the proposed CNN are shown in Table 6. Furthermore, the training options for the SGDM are set to 0.01 for the learning rate, 15 epochs, a mini-batch size of 35 and a frequency of network validation in 50 iterations. These parameters have been tuned by trial and error.

	Kernel		**Stride**		**Padding**			
	Height	Width	Vertical step size	Horizontal step size	Top	Bottom	Left	Right
Convolutional Layers	5	5	1	1	aut.	aut.	aut.	aut.
Pooling layers	2	2	2	2	0	0	0	0

Table 6: *Hyper-parameters of the proposed CNN.*

MATLAB® Script 5: Architecture of the proposed CNN.

```
load('imdsTrain_A.mat')
load('imdsValidation_A.mat')
imageSize = [200 200 3];
layer = [
    imageInputLayer(imageSize) % Input layer (image size).

    convolution2dLayer(5,5,'Padding','same') % First convolutional layer using a [5x5] ...
        kernel ([Height Width]). The stride's size is [1 1] by default ([a b]), where ...
        'a' is the vertical step size and 'b' is the horizontal step size. The software ...
        automatically calculates the padding's size at training time so that the output ...
        has the same size as the input.
    batchNormalizationLayer % A batch normalisation layer normalizes a mini-batch of ...
        data across all observations for each channel independently to speed up the ...
        convolutional neural network's training.
    reluLayer % Activation function ReLU to overcome the issue of vanishing gradients.

    maxPooling2dLayer(2,'Stride',2) % [2x2] pooling layer ([Height Width]), to reduce ...
        the first convolutional layer's output size. The stride's size is [2 2] ([a ...
        b]), where 'a' is the vertical step size and 'b' is the horizontal step size. ...
        The padding's size is set to zero by default.

    convolution2dLayer(5,5,'Padding','same') % Second convolutional layer.
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2) % Second pooling layer.

    fullyConnectedLayer(4) % Fully connected layer of 4 neurons (one per class).
    softmaxLayer % Softmax layer to apply a softmax function to the input.
    classificationLayer]; % Output layer.

option = trainingOptions('sgdm', ... % The network will be trained using the Stochastic ...
    Gradient Descent with Momentum (SDGM) method.
    'InitialLearnRate',0.01, ... % The learning rate determines the size of an update.
    'MaxEpochs',15, ... % Set the maximum number of epochs to 15.
    'MiniBatchSize',35, ... % Each epoch will be done in NTrain/35 iterations. Each ...
        iteration will include 35 training samples passed through the model.
    'Shuffle','every-epoch', ... % Shuffle the training data before each training epoch ...
        and shuffle the validation data before each network validation.
    'ValidationData',imdsValidation_A, ... % Define data to use for validation during ...
        training.
    'ValidationFrequency',50, ... % Frequency of network validation in 50 iterations.
    'Verbose',false, ... % Indicator not to display training progress information in ...
        the command window.
    'Plots','training-progress'); % Indicator to display training progress information ...
        in the plot.

net = trainNetwork(imdsTrain_A,layer,option)
```

The previous section concluded that the faulty balls cases were the most difficult for the ML algorithms to classify. The question is whether the CNN will also have the same difficulties finding correlations between cases with faulty balls as the ML algorithms. To answer this, two case studies are going to be defined.

1. Case A) The CNN will compute the classification of the REBs signals in Table 2. The data set will be split into two subsets: a training subset of 245 signal images, representing 80% of the overall data set, and a validation subset containing 62 signal images, representing 20% of the total data set,

2. Case B) The CNN will compute the classification of the REBs signals in Table 2, not including the 2$^{\text{nd}}$ class, corresponding to REBs with faulty balls. The data set will be split into two subsets: a training subset of 183 signal images, representing 80% of the overall data set, and a validation subset containing 48 signal images, representing 20% of the total data set.

Both subsets have been split manually in each case, including signal images corresponding to every fault type, fault depth and REB type to avoid underfitting. Signal images have been randomly divided within every subclass. The metric used to assess the CNN is accuracy. In addition, it will have to be taken into account that the number of neurons in the fully connected layer will change for cases A and B.

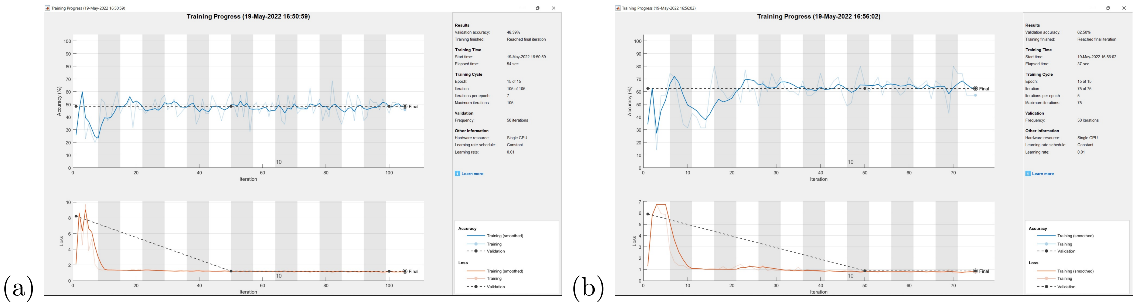
The classification's results are shown in Table 7 & Figure 18. As observed, low-grade results have been obtained compared to those of the best ML algorithms tested in the previous section. In fact, one of the main drawbacks of CNNs is their inferior performance when the data set is of limited size. Wei et al. carried out a DL-based classification of REBs from the CWRU bearing data centre in [Zhang et al., 2017], demonstrating that CNNs can deliver outstanding results when trained sufficiently, obtaining accuracies of over 99.7% on different trials. The authors performed data augmentation to the original data set, having obtained 30000 training data and 7500 testing data by splitting each signal into a series of 2400 data points sections and transforming these sections into 60 x 40 pixels images, as described in Section 5.2.1. They also considered ten different classes, including the different faults' sizes seeded into the REBs. Despite the poor results, it's still clear that the faulty balls cases are yet the most difficult to classify, even for a CNN. The difference in accuracy is about 14% between both cases.

	Accuracy
Case A	48.39%
Case B	62.50%

Table 7: *Results of the CNN for cases A and B.*

To improve the CNN's accuracy, the data set will be augmented by splitting each original signal image of 200 x 200 pixels into 16 images of 50 x 50 pixels, each image containing 2500 data points. The CNN's accuracy is going to be tested for both cases. The CNN's hyper-parameters will remain identical, considering the new image size and the different number of neurons in the fully connected layer for both cases. For case A, the data set will be split into two subsets: a training subset of 3920 signal images, representing 80% of the overall data set, and a validation subset containing 992 signal images, representing 20% of the total data set. For case B, the training subset will contain 2928 signal images, representing

80% of the overall data set, and the validation subset, 762 signal images, representing 20% of the total data set.



(a)

Training Progress (19-May-2022 16:50:59)

Results
Validation accuracy: 48.39%
Training finished: Reached final iteration
Training Time
Start time: 19-May-2022 16:50:59
Elapsed time: 54 sec
Training Cycle
Epoch: 15 of 15
Iteration: 105 of 105
Iterations per epoch: 7
Maximum iterations: 105
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

(b)

Training Progress (19-May-2022 16:56:02)

Results
Validation accuracy: 62.50%
Training finished: Reached final iteration
Training Time
Start time: 19-May-2022 16:56:02
Elapsed time: 37 sec
Training Cycle
Epoch: 15 of 15
Iteration: 75 of 75
Iterations per epoch: 5
Maximum iterations: 75
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

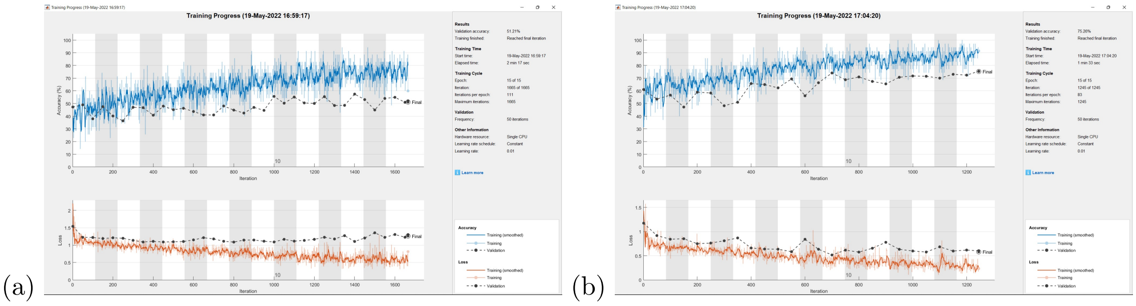
Figure 18: *Results of the CNN for cases A (plot a) and B (plot b).*

One question to be asked before introducing the results is: Why is the data set presented by Wei et al. in [Zhang et al., 2017] so much larger than the one obtained for this project if the signal images contain approximately the same number of data points? The answer is that in [Zhang et al., 2017], the authors obtained such images from the time-domain signals of healthy and faulty REBs directly, and in this project, the initially obtained images have been split instead. Note that the REBs signals sampled at 12k Hz have a size of the order of 120k data points, as stated in Section 5.2.1, so three times more images can be obtained in this way rather than directly splitting the images of 40k data points, the size of which is 1/3 of the original signals' length.

The new classification results are shown in Table 8 & Figure 19. It can be seen that the accuracy increments slightly for case A and somewhat more noticeably for case B. The overall iterations increase in both cases due to the augmentation in the data set, which directly impacts the convergence time. It should also be noted that for this second trial, the iterations' development is more irregular than for the first one, whose behaviour was smoother. Overall, it can be concluded that it is essential for CNNs to have a sufficiently large training data set to learn the correlations that determine if and where a REB has a failure. Furthermore, it has been well proven that REBs with faulty balls are much more challenging to diagnose than those presenting any other defect.

	Accuracy
Case A	51.21%
Case B	75.26%

Table 8: *Results of the CNN for cases A and B having applied data augmentation.*



(a)

Training Progress (19-May-2022 16:59:17)

Results
Validation accuracy: 51.21%
Training finished: Reached final iteration
Training Time
Start time: 19-May-2022 16:59:17
Elapsed time: 2 min 17 sec
Training Cycle
Epoch: 15 of 15
Iteration: 1665 of 1665
Iterations per epoch: 111
Maximum iterations: 1665
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

(b)

Training Progress (19-May-2022 17:04:20)

Results
Validation accuracy: 75.26%
Training finished: Reached final iteration
Training Time
Start time: 19-May-2022 17:04:20
Elapsed time: 1 min 33 sec
Training Cycle
Epoch: 15 of 15
Iteration: 1245 of 1245
Iterations per epoch: 83
Maximum iterations: 1245
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Figure 19: *Results of the CNN for cases A (plot a) and B (plot b) having applied data augmentation.*

6 Summary & Conclusions

6.1 Vibration Analysis-Based Diagnosis

The diagnosis of defective REBs from the CWRU database with vibration analysis techniques was conducted in Section 3. From the cases analysed with Method 1, it is concluded that the defect's relative position (when the fault is located on the outer race) to the REB's maximum vertical load has a significant influence on the maximum amplitude of the impulse responses spaced at the fault's characteristic frequency in the envelope spectrum. In addition, the fault's size also has a strong influence. Larger defect sizes had larger maximum amplitudes of the impulse responses spaced at the faults' characteristic frequency.

Method 2 proved to be very effective in enhancing the impulse responses typical of REB faults in the signals' envelope spectra. As mentioned in Section 2, REB fault signals are second-order cyclostationary and not strictly periodic. This means that their second-order statistic, the variance, is periodic but not the impulses themselves. This is why the fault information is not altered by filtering the raw time signals with CPW and is therefore enhanced.

Method 3, which consisted of applying DRS or SANC to the raw time signals to separate deterministic and random components, proved equally effective. It is concluded that after processing the signals with SANC, an exhaustive study of their Kurtogram is necessary to find the bandwidth where the fault information is present. On the other hand, after processing the signals with DRS, the fault information was usually present in the whole signal's bandwidth. In both cases, the Kurtograms showed impulsivity in bandwidths that were not caused by the different faults. It should be remembered that the Kurtogram does not indicate the bandwidth where the fault is necessarily present but the one with the highest impulsivity.

In general terms, Method 2 gave the best results, followed by Method 3. Finally, it should be noted that to apply these techniques to wind turbine signals to diagnose REB faults, non-constant rotational speed must be considered. In addition to those already mentioned, other techniques are necessary to suppress the frequency modulation effects associated with variable rotational speeds.

6.2 Machine Learning- and Deep Learning-Based Classification

In Section 5, various Machine Learning (ML) algorithms and a Convolutional Neural Network (CNN) have been trained to classify healthy and faulty REBs. The results are summarised in Figure 20. The different models have been trained using the CWRU bearing data centre signals. These models are prepared to receive features of never-before-seen REBs' signals to diagnose their state of health, with an accuracy corresponding to the validation result obtained for each model.

Different features have been extracted from the raw time signals to classify with ML algorithms. These were time- and frequency-domain features. Before training the different algorithms, these features have been subjected to statistical studies to determine if the classes are sufficiently statistically different for each feature and, therefore, which are more relevant for the classification. These tests' results and the different algorithms' accuracy have shown that the frequency-domain features, particularly the logarithmic ratios between the maximum amplitudes of the envelope spectrum around the pulses spaced at the faults' characteristic frequency, have been the most important ones. These, in addition to some of the most critical time-domain features, have been able to yield the highest accuracies in this project. Among the ML algorithms, the k-NN classifier was the best performer and the NB one the worst. In between are the SVM and DT classifiers, which also performed very well. The accuracy of these algorithms, and the features used in each are given in Tables 4 and 5. It can also be seen from the two tables that the signals corresponding to REBs with faulty balls have been the most difficult for the algorithms to classify. The algorithms have been trained to protect the models from overfitting by applying five-fold cross-validation, using 80% of the data set for training and 20% for validation during each iteration.

To carry out the classification from a DL point of view, a 40k data point signal section has been converted into a greyscale image for each one of the signals. Each 40k data points signal section represents 3 seconds of REBs' vibrations readings. CNNs were designed for image processing, in which each pixel represents a signal feature. The architecture of the proposed CNN and the chosen hyper-parameters have been defined in the previous section. CNNs are known to deliver outstanding results when trained with a sufficiently large data set. This fact has been proven true considering the results obtained from the limited data set. In order to improve the results, data augmentation has been applied by splitting each image into 16 images of a smaller size to see if the CNN can find more correlations between the different faults, slightly improving the results. Signals corresponding to REBs with faulty balls have also been the most difficult to classify in this case. Better accuracies could have been obtained by taking advantage of more data points from the original signals. Approximately three times as many images could have been obtained for each case. As well as for the ML case, the CNN has been trained with 80% of the images in the data set and validated with the remaining 20% of them in each case.

In general terms, excellent accuracies have been obtained from the ML algorithms. The CNN's accuracy can be improved by increasing the data set's size, as mentioned in Section 5.2.2. These models can be potentially helpful for automatically diagnosing the health status of REBs.

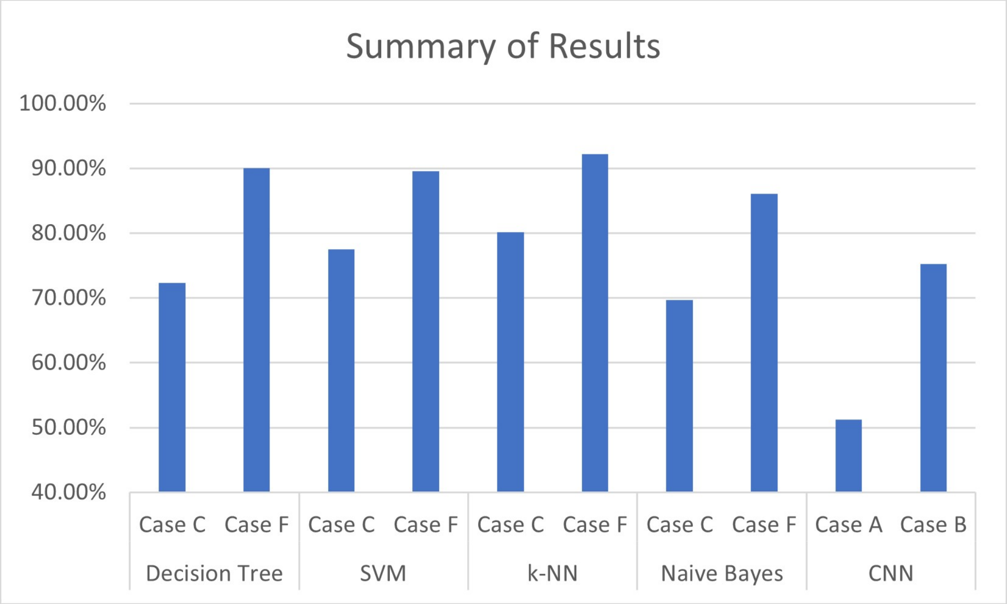


Figure 20: *Summary of results for the ML and DL algorithms.*

Appendices

A MATLAB® Functions

MATLAB® Script 6: Feature extraction function.

```matlab
function [all_12k_sets_feature_extraction_table_2] = FeatureExtraction(Signal, ...
    Sample_rate, RPM, BPFO_coeff, BPFI_coeff, BSF_coeff, Level, Filter_Order, Band, ...
    Fault_code)

%'FeatureExtraction' produces a table containing the following features/signal: ...
    [BPFO_Amplitude, BPFI_Amplitude, BSF_Amplitude, LOG_BPFI_Amplitude_BPFO_Amplitude, ...
    LOG_BSF_Amplitude_BPFO_Amplitude, LOG_BPFI_Amplitude_BSF_Amplitude].

%This function is a modified version of the 'bearingFeatureExtraction' function shown ...
    in [MathWorks(R), 2013].

%Inputs' description

    %'Signal' is a cell array containing time-domain vibration signals of healthy and ...
        faulty Rolling Element Bearings (REBs). Faults are located on the inner race, ...
        outer race and balls.
    %'Sample_rate' is a cell array of the same dimensionality as the latter, containing ...
        the sampling frequency in Hz of each signal.
    %'RPM' is a cell array of the same dimensionality as the latter, containing the ...
        shaft rotating speed in rpm of each signal.
    %'BPFO_coeff' is a cell array of the same dimensionality as the latter, containing ...
        a coefficient for each signal, such that the shaft rotating speed times that ...
        coefficient gives the outer race fault's characteristic frequency in Hz.
    %'BPFI_coeff' is a cell array of the same dimensionality as the latter, containing ...
        a coefficient for each signal, such that the shaft rotating speed times that ...
        coefficient gives the inner race fault's characteristic frequency in Hz.
    %'BPFO_coeff' is a cell array of the same dimensionality as the latter, containing ...
        a coefficient for each signal, such that the shaft rotating speed times that ...
        coefficient gives the ball fault's characteristic frequency in Hz.
    %'Level' is a cell array of the same dimensionality as the latter, containing the ...
        Kurtogram's level for each signal.
    %'Filter_Order' is a cell array of the same dimensionality as the latter, ...
        containing the characters 'FilterOrder' for each signal, required for the ...
        bandpass filtered Envelope Spectrum.
    %'Band' is a cell array of the same dimensionality as the latter, containing the ...
        characters 'Band' for each signal, required for the bandpass filtered Envelope ...
        Spectrum.
    %'Fault_code' is a cell array of the same dimensionality as the latter, containing ...
        labels for each type of signal (healthy REB, inner race fault, outer race ...
        fault, ball fault), for later classification.

%Bibliography

    %[MathWorks(R), 2013]  MathWorks(R). (2013). Rolling Element Bearing Fault Diagnosis.
    %Retrieved from: https://es.mathworks.com/help/predmaint/ug/Rolling-Element-Bearing-
    %Fault-Diagnosis.html?s_tid=srchtitle

%------------------------------
%Author: David Cascales Fulgencio
%Last revision: 28/03/2022
%------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Transform cell arrays into numeric arrays

sample_rate = cell2mat(Sample_rate);
rpm = cell2mat(RPM);
bpfo_coeff = cell2mat(BPFO_coeff);
bpfi_coeff = cell2mat(BPFI_coeff);
bsf_coeff = cell2mat(BSF_coeff);
```

```
%Extract condition indicators from bearing data

fshaft = rpm./60;

%Critical Frequencies

BPFO = bpfo_coeff.*fshaft;
BPFI = bpfi_coeff.*fshaft;
BSF = bsf_coeff.*fshaft;

%Kurtogram

[¬, ¬, ¬, fc, ¬, BW] = cellfun(@kurtogram, Signal, Sample_rate, Level, 'UniformOutput', ...
    false);
f_c = cell2mat(fc);
B_W = cell2mat(BW);

%Bandpass filtered Envelope Spectrum

a = (f_c)-(B_W)./2;
b = zeros(307,1);
c = arrayfun(@max, a, b);
d = (f_c)+(B_W)./2;
e = 0.999.*(sample_rate)./2;
f = arrayfun(@min, d, e);

band =[c, f];
BAND = cell(size(band,1),1);
for ii = 1:size(band,1)
    BAND{ii,1} = band(ii,:);
end

filterorder = zeros(307,1) + 200;
FilterOrder = num2cell(filterorder);

[pEnvpBpf, fEnvBpf] = cellfun(@envspectrum, Signal, Sample_rate, Filter_Order, ...
    FilterOrder, Band, BAND, 'UniformOutput', false);

G = cellfun(@(x) x(2), fEnvBpf, 'UniformOutput', false);
H = cellfun(@(x) x(1), fEnvBpf, 'UniformOutput', false);
Δf = cellfun(@minus, G, H);

%Features

i = BPFO-(5.*Δf);
I = num2cell(i);
j = BPFO+(5.*Δf);
J = num2cell(j);

O = cellfun(@gt, fEnvBpf, I, 'UniformOutput', false);
P = cellfun(@lt, fEnvBpf, J, 'UniformOutput', false);

U = cellfun(@and, O, P, 'UniformOutput', false);

X = cellfun(@(x,y) x(y), pEnvpBpf, U, 'UniformOutput', false);

BPFOAmplitude = cellfun(@max, X, 'UniformOutput', false);

k = BPFI-(5.*Δf);
K = num2cell(k);
l = BPFI+(5.*Δf);
L = num2cell(l);

Q = cellfun(@gt, fEnvBpf, K, 'UniformOutput',false);
R = cellfun(@lt, fEnvBpf, L, 'UniformOutput',false);

V = cellfun(@and, Q, R, 'UniformOutput', false);

Y = cellfun(@(x,y) x(y), pEnvpBpf, V, 'UniformOutput', false);
```

```
BPFIAmplitude = cellfun(@max, Y, 'UniformOutput', false);

m = BSF-(5.*Δf);
M = num2cell(m);
n = BSF+(5.*Δf);
N = num2cell(n);

S = cellfun(@gt, fEnvBpf, M, 'UniformOutput', false);
T = cellfun(@lt, fEnvBpf, N, 'UniformOutput', false);

W = cellfun(@and, S, T, 'UniformOutput', false);

Z = cellfun(@(x,y) x(y), pEnvpBpf, W, 'UniformOutput', false);

BSFAmplitude = cellfun(@max, Z, 'UniformOutput', false);

A = cellfun(@(x,y) x/y, BPFIAmplitude, BPFOAmplitude, 'UniformOutput', false);
B = cellfun(@(x,y) x/y, BSFAmplitude, BPFOAmplitude, 'UniformOutput', false);
C = cellfun(@(x,y) x/y, BPFIAmplitude, BSFAmplitude, 'UniformOutput', false);

LOG_BPFIAmplitude_BPFOAmplitude = cellfun(@(x) log(x), A, 'UniformOutput', false);
LOG_BSFAmplitude_BPFOAmplitude = cellfun(@(x) log(x), B, 'UniformOutput', false);
LOG_BPFIAmplitude_BSFAmplitude = cellfun(@(x) log(x), C, 'UniformOutput', false);

%Transform cell arrays into numeric arrays

Fault_Code = char(Fault_code);
BPFO_Amplitude = cell2mat(BPFOAmplitude);
BPFI_Amplitude = cell2mat(BPFIAmplitude);
BSF_Amplitude = cell2mat(BSFAmplitude);
LOG_BPFI_Amplitude_BPFO_Amplitude = cell2mat(LOG_BPFIAmplitude_BPFOAmplitude);
LOG_BSF_Amplitude_BPFO_Amplitude = cell2mat(LOG_BSFAmplitude_BPFOAmplitude);
LOG_BPFI_Amplitude_BSF_Amplitude = cell2mat(LOG_BPFIAmplitude_BSFAmplitude);

%Table

all_12k_sets_feature_extraction_table_2 = table(Fault_Code, BPFO_Amplitude, ...
    BPFI_Amplitude, BSF_Amplitude, LOG_BPFI_Amplitude_BPFO_Amplitude, ...
    LOG_BSF_Amplitude_BPFO_Amplitude, LOG_BPFI_Amplitude_BSF_Amplitude);
end
```

B Tables

B.1 Vibration Analysis

Drive end bearing: 6205-2RS JEM SKF, deep groove ball bearing.

Size:

Inside Diameter	Outside Diameter	Thickness	Ball Diameter	Pitch Diameter
25,0000 mm (0,9843”)	52,0000 mm (2,0472”)	15,0000 mm (0,5906”)	7,9400 mm (0,3126”)	39,0398 mm (1,5370”)

Table 9: Case Western Reserve University (CWRU). (nd). *Drive end bearing geometry.* [Table]. Retrieved from [CWRU, nd].

Defect frequencies:

Inner Ring	Outer Ring	Cage Train	Rolling Element
$5,4152 \cdot f_{shaft}$ Hz	3,5848 $\cdot f_{shaft}$ Hz	$0,39828 \cdot f_{shaft}$ Hz	$4,7135 \cdot f_{shaft}$ Hz

Table 10: Case Western Reserve University (CWRU). (nd). *Drive end bearing defect frequencies.* [Table]. Retrieved from [CWRU, nd].

Fan end bearing: 6203-2RS JEM SKF, deep groove ball bearing.

Size:

Inside Diameter	Outside Diameter	Thickness	Ball Diameter	Pitch Diameter
17,0000 mm (0,6693”)	40,0000 mm (1,5748”)	12,0000 mm (0,4724”)	6,7462 mm (0,2656”)	28,4988 mm (1,1220”)

Table 11: Case Western Reserve University (CWRU). (nd). *Fan end bearing geometry.* [Table]. Retrieved from [CWRU, nd].

Defect frequencies:

Inner Ring	Outer Ring	Cage Train	Rolling Element
$4,9469 \cdot f_{shaft}$ Hz	3,0530 $\cdot f_{shaft}$ Hz	$0,3817 \cdot f_{shaft}$ Hz	$3,9874 \cdot f_{shaft}$ Hz

Table 12: Case Western Reserve University (CWRU). (nd). *Fan end bearing defect frequencies.* [Table]. Retrieved from [CWRU, nd].

Analysed Signals’ Breakdown:

Method	Type of REB	Signal’s Name	Fault Location	Fault Diameter (inches)	Motor Load (HP)	Approx. Motor Speed (rpm)	Sampling Frequency (kHz)
Method 1	Drive end REB	IR021_0_DE	Inner Race	0.021	0	1797	12
		OR007@3_0_BA	Outer Race (Orthogonal)	0.007	0	1797	12
		OR007@6_0_FE	Outer Race (Centered)	0.007	0	1797	12
	Fan end REB	IR014_0_FE	Inner Race	0.014	0	1797	12
		IR021_3_FE	Inner Race	0.021	3	1730	12
Method 2	Drive end REB	IR021_0_FE	Inner Race	0.021	0	1797	12
	Fan end REB	IR007_1_BA	Inner Race	0.007	1	1772	12
		IR014_3_BA	Inner Race	0.014	3	1730	12
		OR014@6_0_DE	Outer Race (Centered)	0.014	0	1797	12
Method 3	Fan end REB	IR007_0_FE	Inner Race	0.007	0	1797	12
		IR014_3_FE	Inner Race	0.014	3	1730	12
		OR014@3_1_DE	Outer Race (Orthogonal)	0.014	1	1772	12
		OR021@3_2_BA	Outer Race (Orthogonal)	0.021	2	1750	12

Table 13: *Analysed signals’ breakdown.*

B.2 Feature extraction

Name	Fault type	Fault code	Mean	Peak value	RMS	Standard deviation	Clearance factor	Kurtosis	Skewness	Impulse factor	Crest Factor	Shape factor
’Normal_0_DE’	’Normal’	’0’	0.012558211	0.311254154	0.073763968	0.072687248	6.144245883	2.76424279	-0.035410743	5.231687913	4.219596126	1.239855132
’Normal_0_FE’	’Normal’	’0’	0.031375814	0.357490909	0.084673148	0.07864557	6.217136994	2.760103557	0.137998772	5.276134036	4.222010365	1.249673397
’Normal_1_DE’	’Normal’	’0’	0.012564102	0.345884308	0.066352043	0.065151713	7.594014064	2.930617176	-0.173048079	6.468640034	5.212866012	1.240898964
’Normal_1_FE’	’Normal’	’0’	0.032375256	0.321947273	0.074723645	0.067345937	6.397556507	2.868279093	0.127822192	5.411384121	4.308506023	1.255976919
’Normal_2_DE’	’Normal’	’0’	0.012260601	0.359235692	0.064325396	0.063146199	8.125497238	2.925145779	-0.167060051	6.925225864	5.584663511	1.240043532
’Normal_2_FE’	’Normal’	’0’	0.031962495	0.311674545	0.072503102	0.065077704	6.382181144	2.865914578	0.128008273	5.398101703	4.298775302	1.255730138
’Normal_3_DE’	’Normal’	’0’	0.012458514	0.306456	0.065883867	0.06469527	6.806829521	2.95716868	-0.127529579	5.789082383	4.651457364	1.244573889
’Normal_3_FE’	’Normal’	’0’	0.031896471	0.384816364	0.081749458	0.075270184	6.96269956	2.991924123	0.014001315	5.89702624	4.707265031	1.252749994
’IR007_0_DE’	’Inner race fault’	’1’	0.013443559	1.739030499	0.291526045	0.29121711	10.425707	5.395863773	0.164029512	8.325715255	5.965266317	1.395698836
’IR007_0_FE’	’Inner race fault’	’1’	0.03301915	1.139656364	0.24566908	0.243441001	6.921160135	3.326424163	-0.20857414	5.849979643	4.638989835	1.261046015
’IR007_0_BA’	’Inner race fault’	’1’	0.006097541	0.372034896	0.090583518	0.090378432	6.129652536	3.082952914	0.0940088	5.17715094	4.107092604	1.260539131
’IR007_1_DE’	’Inner race fault’	’1’	0.005800988	1.580818683	0.292892281	0.292836029	9.590162763	5.542254575	0.130428682	7.613106677	5.397269859	1.410547717
’IR007_1_FE’	’Inner race fault’	’1’	0.033327004	1.1526	0.240030467	0.23770654	7.186145281	3.356215993	-0.147943712	6.06975443	4.801890423	1.264034348
’IR007_1_BA’	’Inner race fault’	’1’	0.005677852	0.370264451	0.095992311	0.095824637	5.797751702	3.107709133	0.016741837	4.88182671	3.857230298	1.265630085
’IR007_2_DE’	’Inner race fault’	’1’	0.004551921	1.6396202	0.299510119	0.299476753	9.588249971	5.563783473	0.090417422	7.673510928	5.474339915	1.401723504
’IR007_2_FE’	’Inner race fault’	’1’	0.033986933	0.957212727	0.2351426	0.232674394	6.091110012	3.124921644	0.028711358	5.140924784	4.070775471	1.262885861
’IR007_2_BA’	’Inner race fault’	’1’	0.005823562	0.372356795	0.096403237	0.096227574	5.904691665	3.314347739	-0.01277147	4.940213526	3.862492679	1.279022107
’IR007_3_DE’	’Inner race fault’	’1’	0.004718397	1.671457485	0.313606732	0.31357251	9.346994889	5.291053175	-0.013220457	7.457475282	5.329788289	1.399206662
’IR007_3_FE’	’Inner race fault’	’1’	0.033394321	1.082129091	0.227231006	0.224764678	7.038775364	3.221696022	-0.209276017	5.970042451	4.762242216	1.253620076
’IR007_3_BA’	’Inner race fault’	’1’	0.00587786	0.381048071	0.096958798	0.096780863	5.884847781	3.268169187	-0.135394918	4.968926833	3.929999957	1.264357987
’B007_0_DE’	’Ball fault’	’2’	0.012607049	0.60702008	0.139233558	0.138662189	6.446439588	2.984716023	-0.008853864	5.461056675	4.359725417	1.252614821
’B007_0_FE’	’Ball fault’	’2’	0.033162302	0.465354545	0.110642917	0.105556656	6.10312779	2.765721465	0.005560342	5.209266684	4.205913567	1.238557712
’B007_0_BA’	’Ball fault’	’2’	0.005553873	0.161231217	0.035810484	0.03537733	6.711063569	3.125094326	0.028167755	5.672135927	4.502346719	1.259817664
’B007_1_DE’	’Ball fault’	’2’	0.003891835	0.659649062	0.139068323	0.139014428	6.990525202	2.96375413	0.007453333	5.931159451	4.743345206	1.250416993
’B007_1_FE’	’Ball fault’	’2’	0.032981525	0.384610909	0.088101565	0.081695531	6.410408929	2.901767765	-0.008633202	5.443664634	4.365540024	1.246962484
’B007_1_BA’	’Ball fault’	’2’	0.005192882	0.145981246	0.035228164	0.034843471	6.135424653	3.002619045	0.024981278	5.197107529	4.143878971	1.254164894
’B007_2_DE’	’Ball fault’	’2’	0.004564115	0.604583553	0.147251841	0.147181696	6.009732252	2.831413349	0.027123577	5.108463782	4.105779256	1.244212965
’B007_2_FE’	’Ball fault’	’2’	0.032758774	0.386254545	0.091599866	0.085542144	6.198360612	2.923313992	-0.000667637	5.262657952	4.216758842	1.248033893
’B007_2_BA’	’Ball fault’	’2’	0.005721228	0.162196914	0.037373021	0.036932662	6.498666355	3.137106772	0.047190945	5.483272721	4.339946521	1.263442463
’B007_3_DE’	’Ball fault’	’2’	0.004199705	0.720562236	0.153634849	0.153578069	6.886993068	2.889729715	0.020412484	5.847831457	4.690096286	1.246846781
’B007_3_FE’	’Ball fault’	’2’	0.032817589	0.476449091	0.125630108	0.121268507	5.415281867	2.718490315	-0.110236749	4.650687966	3.792475387	1.226293513
’B007_3_BA’	’Ball fault’	’2’	0.00574752	0.208590623	0.045539118	0.045175149	6.801667793	2.981991972	0.105800715	5.754798397	4.580471329	1.256376906
’OR007@6_0_DE’	’Outer race fault’	’3’	0.023171476	3.63042515	0.669505588	0.66910723	12.6360571	7.649434671	0.056946021	8.947741688	5.422546452	1.650099592
’OR007@6_0_FE’	’Outer race fault’	’3’	0.032555324	1.105345455	0.256690124	0.254618345	6.806802712	3.572616451	0.065038767	5.622884013	4.306147191	1.305780728
’OR007@6_0_BA’	’Outer race fault’	’3’	0.005277252	0.395050682	0.076915843	0.076734906	8.274768168	4.068723666	0.152719103	6.799416549	5.136141906	1.323837361
’OR007@6_1_DE’	’Outer race fault’	’3’	0.004061054	3.112257086	0.591947423	0.59193591	11.6886291	7.594989679	0.033401318	8.492661763	5.25765797	1.615293694
’OR007@6_1_FE’	’Outer race fault’	’3’	0.032773633	1.06672	0.258424317	0.256338747	6.498355753	3.446786449	0.067550342	5.37067612	4.127784928	1.30110367
’OR007@6_1_BA’	’Outer race fault’	’3’	0.005350632	0.370143739	0.068753086	0.068544846	8.669325096	4.01324994	0.160209876	7.123244507	5.383667286	1.323121235
’OR007@6_2_DE’	’Outer race fault’	’3’	0.003912254	3.101292715	0.570170372	0.570159298	12.31580075	7.852217866	0.019542968	8.862415057	5.43923863	1.629348455
’OR007@6_2_FE’	’Outer race fault’	’3’	0.032340456	1.049872727	0.227458839	0.225148919	7.347377726	3.69271994	0.038556418	6.052987617	4.61566028	1.311402324
’OR007@6_2_BA’	’Outer race fault’	’3’	8.715129429	5.34379194	7.127661246	4.16450096	0.00500342	0.3753746	0.07024499	1.33382088	0.17046261	0.070066857
’OR007@6_3_DE’	’Outer race fault’	’3’	12.68455931	5.57570902	9.109222904	7.96374551	0.00452908	3.23611387	0.58039504	1.63373355	-0.0021069	0.580379738
’OR007@6_3_FE’	’Outer race fault’	’3’	7.618342053	4.76438348	6.264380141	3.67184704	0.03170506	1.22615273	0.25735811	1.31483542	0.08233996	0.255398744
’OR007@6_3_BA’	’Outer race fault’	’3’	8.267890123	5.14051315	6.798508667	4.05811853	0.00510191	0.37774861	0.07348461	1.32253502	0.12384245	0.073307587
’OR007@3_0_DE’	’Outer race fault’	’3’	7.149061224	4.35266224	5.84010337	4.23927749	0.00719164	3.34900629	0.76941561	1.34173135	0.065802	0.769385149
’OR007@3_0_FE’	’Outer race fault’	’3’	6.763693162	4.54400832	5.720430489	3.12525884	0.03245138	0.77744	0.17109124	1.25889525	0.01631517	0.16798616
’OR007@3_0_BA’	’Outer race fault’	’3’	6.856743959	4.60281822	5.798045845	3.16534534	0.00539224	0.39734421	0.08632629	1.259673	-0.1136211	0.086158067
’OR007@3_1_DE’	’Outer race fault’	’3’	6.605045982	4.19203716	5.495326159	4.05110931	0.00593853	3.55489281	0.84801081	1.31089634	0.07342687	0.847993492
’OR007@3_1_FE’	’Outer race fault’	’3’	7.583153492	5.00192251	6.370140716	3.33806042	0.03304522	0.96645818	0.19321734	1.27353847	0.09684045	0.190371356
’OR007@3_1_BA’	’Outer race fault’	’3’	6.380420888	4.36438046	5.432117354	2.94103	0.00516101	0.3945276	0.09039716	1.24464799	-0.097702	0.090250081
’OR007@3_2_DE’	’Outer race fault’	’3’	6.594652958	4.25677154	5.534979026	3.94136669	0.00570897	3.63692255	0.85438519	1.30027627	0.07317015	0.854369629
’OR007@3_2_FE’	’Outer race fault’	’3’	7.089571312	4.67622017	5.954748605	3.283313	0.03277788	0.87585273	0.18729929	1.27341066	0.07657605	0.184409639
’OR007@3_2_BA’	’Outer race fault’	’3’	5.85236919	4.02839391	4.992558061	2.86084258	0.00589702	0.36499335	0.09060518	1.23934207	-0.1908313	0.090413445
’OR007@3_3_DE’	’Outer race fault’	’3’	7.148976591	4.53744024	5.952484756	4.09025895	0.00667384	3.61905469	0.79759832	1.31185965	0.0833652	0.797573655
’OR007@3_3_FE’	’Outer race fault’	’3’	7.962227904	5.3269683	6.727055172	3.28526457	0.03279445	0.97426545	0.18289305	1.26282996	-0.075746	0.179929582

Table 14: *A 55 member portion of the data set along with their time-domain features.*

Name	Fault type	Fault code	BPFO_Amplitude	BPFI_Amplitude	BSF_Amplitude	LOG_BPFI_Amplitude_BPFO_Amplitude	LOG_BSF_Amplitude_BPFO_Amplitude	LOG_BPFI_Amplitude_BSF_Amplitude
’Normal_0_DE’	’Normal’	’0’	7.23E-06	2.60E-06	1.99E-06	-1.024185003	-1.287481413	0.263296411
’Normal_0_FE’	’Normal’	’0’	6.99E-06	2.73E-06	3.58E-06	-0.942123317	-0.669056533	-0.273066784
’Normal_1_DE’	’Normal’	’0’	2.50E-06	1.74E-06	2.00E-06	-0.362183297	-0.223362668	-0.13882063
’Normal_1_FE’	’Normal’	’0’	5.17E-05	7.12E-05	3.33E-05	0.320062819	-0.439132521	0.759195341
’Normal_2_DE’	’Normal’	’0’	6.74E-05	1.69E-05	2.24E-05	-1.381524646	-1.0998303	-0.281694346
’Normal_2_FE’	’Normal’	’0’	2.36E-06	1.27E-06	1.30E-06	-0.619577217	-0.596986428	-0.022590788
’Normal_3_DE’	’Normal’	’0’	0.000275254	6.79E-05	7.52E-05	-1.400237176	-1.296940599	-0.103296577
’Normal_3_FE’	’Normal’	’0’	0.000168897	0.000134303	4.67E-05	-0.229190509	-1.284970252	1.055779743
’IR007_0_DE’	’Inner race fault’	’1’	0.002304509	0.190874853	0.001737689	4.416750195	-0.28231177	4.699061965
’IR007_0_FE’	’Inner race fault’	’1’	0.001187405	0.022611231	0.000379137	2.946676783	-1.141626514	4.088303297
’IR007_0_BA’	’Inner race fault’	’1’	0.000218642	0.000268448	0.00030065	0.205221226	0.318509171	-0.113287945
’IR007_1_DE’	’Inner race fault’	’1’	0.001305567	0.18999801	0.00142687	4.980376275	0.088846112	4.891530162
’IR007_1_FE’	’Inner race fault’	’1’	0.000629303	0.010762674	0.000251248	2.839226734	-0.91817064	3.757397374
’IR007_1_BA’	’Inner race fault’	’1’	1.60E-05	8.50E-06	2.07E-06	-0.631447352	-2.042598766	1.411151415
’IR007_2_DE’	’Inner race fault’	’1’	0.001334589	0.18643248	0.001131044	4.939445723	-0.165482717	5.10492844
’IR007_2_FE’	’Inner race fault’	’1’	0.000254553	0.011525931	0.000227588	3.81284711	-0.111970506	3.924817616
’IR007_2_BA’	’Inner race fault’	’1’	0.000702897	0.030908537	0.000379176	3.783577258	-0.617209445	4.400786703
’IR007_3_DE’	’Inner race fault’	’1’	0.003352367	0.154999628	0.001457261	3.833756182	-0.833107961	4.666864143
’IR007_3_FE’	’Inner race fault’	’1’	0.000740042	0.009668884	0.000345336	2.569961146	-0.762190464	3.33215161
’IR007_3_BA’	’Inner race fault’	’1’	0.001003705	0.020305217	0.000190798	3.007179669	-1.660239364	4.667419033
’B007_0_DE’	’Ball fault’	’2’	0.001140458	0.000156683	5.43E-05	-1.984961024	-3.04395667	1.058995647
’B007_0_FE’	’Ball fault’	’2’	1.13E-05	4.77E-06	3.38E-06	-0.859642396	-1.202275253	0.342632857
’B007_0_BA’	’Ball fault’	’2’	5.06E-05	1.48E-05	1.89E-05	-1.232795028	-0.98549857	-0.247296458
’B007_1_DE’	’Ball fault’	’2’	9.81E-06	8.48E-06	1.93E-06	-0.146360879	-1.627590341	1.481229461
’B007_1_FE’	’Ball fault’	’2’	1.18E-05	1.20E-05	7.32E-06	0.020024928	-0.477725091	0.497750019
’B007_1_BA’	’Ball fault’	’2’	1.66E-05	4.45E-06	2.23E-06	-1.315846076	-2.007085843	0.691239767
’B007_2_DE’	’Ball fault’	’2’	6.36E-05	2.18E-05	2.86E-05	-1.07287038	-0.799176879	-0.273693501
’B007_2_FE’	’Ball fault’	’2’	9.32E-06	5.19E-06	4.50E-06	-0.58603319	-0.727826265	0.141793075
’B007_2_BA’	’Ball fault’	’2’	0.002371022	0.000280192	0.000649028	-2.13560262	-1.295600273	-0.840002346
’B007_3_DE’	’Ball fault’	’2’	2.08E-05	1.43E-05	2.64E-06	-0.374929853	-2.060562315	1.685632462
’B007_3_FE’	’Ball fault’	’2’	8.45E-05	3.73E-05	5.02E-05	-0.818617153	-0.521371448	-0.297245705
’B007_3_BA’	’Ball fault’	’2’	3.11E-05	2.20E-05	3.64E-06	-0.346684947	-2.143227714	1.796542766
’OR007@6_0_DE’	’Outer race fault’	’3’	0.561750858	0.004520649	0.001949096	-4.822402894	-5.663692949	0.841290055
’OR007@6_0_FE’	’Outer race fault’	’3’	0.008378089	7.00E-05	9.66E-05	-4.78468131	-4.462298653	-0.322382657
’OR007@6_0_BA’	’Outer race fault’	’3’	0.001220071	3.28E-05	3.37E-05	-3.615770559	-3.589609092	-0.026161466
’OR007@6_1_DE’	’Outer race fault’	’3’	0.033378625	0.001073749	0.000717899	-3.436759649	-3.839342397	0.402582748
’OR007@6_1_FE’	’Outer race fault’	’3’	0.039793147	0.001532174	0.000660134	-3.257006744	-4.099006639	0.841999895
’OR007@6_1_BA’	’Outer race fault’	’3’	0.000443401	4.72E-05	9.50E-05	-2.239523719	-1.541024622	-0.698499096
’OR007@6_2_DE’	’Outer race fault’	’3’	0.358692206	0.010600451	0.005514648	-3.52156811	-4.175056904	0.653488794
’OR007@6_2_FE’	’Outer race fault’	’3’	0.035334073	0.003795903	0.00089562	-2.230925507	-3.675087209	1.444161702
’OR007@6_2_BA’	’Outer race fault’	’3’	1.37E-05	4.50E-06	2.46E-06	-1.111274394	-1.712698251	0.601423857
’OR007@6_3_DE’	’Outer race fault’	’3’	0.473759216	0.007647316	0.002402265	-4.126344457	-5.284287043	1.157942586
’OR007@6_3_FE’	’Outer race fault’	’3’	0.043450083	0.00359482	0.00099871	-2.492118942	-3.77290382	1.280784878
’OR007@6_3_BA’	’Outer race fault’	’3’	0.035861126	0.000840531	0.0004404	-3.753374787	-4.399725724	0.646350937
’OR007@3_0_DE’	’Outer race fault’	’3’	0.007567642	6.24E-05	9.22E-05	-4.798503316	-4.407637151	-0.390866165
’OR007@3_0_FE’	’Outer race fault’	’3’	1.97E-05	9.96E-06	6.83E-06	-0.684094374	-1.061499364	0.37740499
’OR007@3_0_BA’	’Outer race fault’	’3’	0.007914829	0.000178256	7.36E-05	-3.793273149	-4.67733781	0.884064662
’OR007@3_1_DE’	’Outer race fault’	’3’	0.009356707	0.000257396	7.93E-05	-3.593233901	-4.77011291	1.176879009
’OR007@3_1_FE’	’Outer race fault’	’3’	0.033013606	0.001412787	0.000180356	-3.151355636	-5.209739796	2.05838416
’OR007@3_1_BA’	’Outer race fault’	’3’	0.020021367	0.000413744	0.000181577	-3.879308536	-4.702873979	0.823565442
’OR007@3_2_DE’	’Outer race fault’	’3’	0.009096176	0.000442214	0.000212292	-3.023815366	-3.757647451	0.733832085
’OR007@3_2_FE’	’Outer race fault’	’3’	9.75E-06	3.52E-06	3.53E-06	-1.017444916	-1.014891198	-0.002553718
’OR007@3_2_BA’	’Outer race fault’	’3’	0.018507373	0.000196625	0.000180045	-4.54462676	-4.632718563	0.088091803
’OR007@3_3_DE’	’Outer race fault’	’3’	0.008415081	0.00040808	8.43E-05	-3.026317489	-4.603974046	1.577656556
’OR007@3_3_FE’	’Outer race fault’	’3’	0.017571104	0.004100193	0.00110482	-1.455221552	-2.766573101	1.311351548

Table 15: *A 55 member portion of the data set along with their frequency-domain features.*

B.3 Hypothesis Tests

Features	Normality condition	Homoscedasticity condition		Equality of continuous distributions condition *
		W	p-value	
Clearance_Factor_0	0	10.7727	0	1
Clearance_Factor_1	1			
Clearance_Factor_2	1			
Clearance_Factor_3	1			
Crest_Factor_0	0	27.6535	0	1
Crest_Factor_1	1			
Crest_Factor_2	1			
Crest_Factor_3	1			
Impulse_Factor_0	0	15.1834	0	1
Impulse_Factor_1	1			
Impulse_Factor_2	1			
Impulse_Factor_3	1			
Kurtosis_0	0	5.4175	0.0012	1
Kurtosis_1	1			
Kurtosis_2	1			
Kurtosis_3	1			
Mean_0	1	1.3019	0.2739	1
Mean_1	1			
Mean_2	1			
Mean_3	1			
Peak_Value_0	0	4.555	0.0039	1
Peak_Value_1	1			
Peak_Value_2	1			
Peak_Value_3	1			
RMS_0	0	4.3323	0.0052	1
RMS_1	1			
RMS_2	1			
RMS_3	1			
Shape_Factor_0	0	10.6547	0	1
Shape_Factor_1	1			
Shape_Factor_2	1			
Shape_Factor_3	1			
Skewness_0	0	4.3422	0.0051	1
Skewness_1	1			
Skewness_2	1			
Skewness_3	1			
Std_0	0	4.3611	0.005	1
Std_1	1			
Std_2	1			
Std_3	1			
BPFO_Amplitude_0	1	24.707	0	1
BPFO_Amplitude_1	1			
BPFO_Amplitude_2	1			
BPFO_Amplitude_3	1			
BPFI_Amplitude_0	1	42.0366	0	1
BPFI_Amplitude_1	1			
BPFI_Amplitude_2	1			
BPFI_Amplitude_3	1			
BSF_Amplitude_0	0	3.255	0.022	1
BSF_Amplitude_1	1			
BSF_Amplitude_2	1			
BSF_Amplitude_3	1			
LOG_BPFI_Amplitude_BPFO_Amplitude_0	0	27.8576	0	1
LOG_BPFI_Amplitude_BPFO_Amplitude_1	0			
LOG_BPFI_Amplitude_BPFO_Amplitude_2	0			
LOG_BPFI_Amplitude_BPFO_Amplitude_3	1			
LOG_BSF_Amplitude_BPFO_Amplitude_0	0	10.3122	0	1
LOG_BSF_Amplitude_BPFO_Amplitude_1	1			
LOG_BSF_Amplitude_BPFO_Amplitude_2	1			
LOG_BSF_Amplitude_BPFO_Amplitude_3	1			
LOG_BPFI_Amplitude_BSF_Amplitude_0	0	10.7565	0	1
LOG_BPFI_Amplitude_BSF_Amplitude_1	1			
LOG_BPFI_Amplitude_BSF_Amplitude_2	1			
LOG_BPFI_Amplitude_BSF_Amplitude_3	1			

*The Kolmogorov-Smirnov test has been applied to every pair of groups within each feature. A negative result is provided if at least one pair of groups do not fulfil the null hypothesis.

Table 16: *Hypothesis tests applied to every group within each feature.*

B.4 Features' Ranking

One-Way ANOVA					
Case A		**Case B**		**Case C**	
Features	**F**	**Features**	**F**	**Features**	**F**
Crest factor	6.0064	LOG_BPFI_Amplitude_BPFO_Amplitude	111.9799	LOG_BPFI_Amplitude_BPFO_Amplitude	111.9799
Shape factor	5.8229	LOG_BSF_Amplitude_BPFO_Amplitude	15.9996	LOG_BSF_Amplitude_BPFO_Amplitude	15.9996
Impulse factor	3.4917	LOG_BPFI_Amplitude_BSF_Amplitude	12.2117	LOG_BPFI_Amplitude_BSF_Amplitude	12.2117
Clearance factor	2.7894	BPFI_Amplitude	11.8174	BPFI_Amplitude	11.8174
Kurtosis	1.9256	BPFO_Amplitude	7.3982	BPFO_Amplitude	7.3982
Mean	1.4055	BSF_Amplitude	1.0520	Crest factor	6.0064
Peak value	1.3931			Shape factor	5.8229
Standard deviation	0.9941			Impulse factor	3.4917
RMS	0.9637			Clearance factor	2.7894
Skewness	0.6254			Kurtosis	1.9256
				Mean	1.4055
				Peak value	1.3931
				BSF_Amplitude	1.0520
				Standard deviation	0.9941
				RMS	0.9637
				Skewness	0.6254

Table 17: *Features' ranking for cases A, B & C using one-way ANOVA.*

One-Way ANOVA					
Case D		**Case E**		**Case F**	
Features	**F**	**Features**	**F**	**Features**	**F**
Shape factor	4.0389	LOG_BPFI_Amplitude_BPFO_Amplitude	135.9887	LOG_BPFI_Amplitude_BPFO_Amplitude	135.9887
Clearance factor	2.9775	LOG_BSF_Amplitude_BPFO_Amplitude	15.4378	LOG_BSF_Amplitude_BPFO_Amplitude	15.4378
Kurtosis	2.8668	LOG_BPFI_Amplitude_BSF_Amplitude	13.3609	LOG_BPFI_Amplitude_BSF_Amplitude	13.3609
Standard deviation	2.7570	BPFI_Amplitude	11.5246	BPFI_Amplitude	11.5246
Impulse factor	2.6555	BPFO_Amplitude	5.7609	BPFO_Amplitude	5.7609
RMS	2.6403	BSF_Amplitude	1.0619	Shape factor	4.0389
Peak value	2.5531			Clearance factor	2.9775
Crest factor	1.8718			Kurtosis	2.8668
Mean	1.2457			Standard deviation	2.7570
Skewness	0.7408			Impulse factor	2.6555
				RMS	2.6403
				Peak value	2.5531
				Crest factor	1.8718
				Mean	1.2457
				BSF_Amplitude	1.0619
				Skewness	0.7408

Table 18: *Features' ranking for cases D, E & F using one-way ANOVA.*

Kruskal-Wallis Test					
Case A		**Case B**		**Case C**	
Features	**H**	**Features**	**H**	**Features**	**H**
Standard deviation	26.0577	LOG_BPFI_Amplitude_BPFO_Amplitude	154.9906	LOG_BPFI_Amplitude_BPFO_Amplitude	154.9906
RMS	25.2882	LOG_BSF_Amplitude_BPFO_Amplitude	61.7938	LOG_BSF_Amplitude_BPFO_Amplitude	61.7938
Shape factor	17.9515	BPFI_Amplitude	52.6833	BPFI_Amplitude	52.6833
Peak value	15.7401	BPFO_Amplitude	49.5724	BPFO_Amplitude	49.5724
Mean	12.3161	LOG_BPFI_Amplitude_BSF_Amplitude	38.7083	LOG_BPFI_Amplitude_BSF_Amplitude	38.7083
Kurtosis	11.9904	BSF_Amplitude	27.6716	BSF_Amplitude	27.6716
Clearance factor	5.3638			Standard deviation	26.0577
Impulse factor	5.1157			RMS	25.2882
Crest factor	4.8157			Shape factor	17.9515
Skewness	3.0968			Peak value	15.7401
				Mean	12.3161
				Kurtosis	11.9904
				Clearance factor	5.3638
				Impulse factor	5.1157
				Crest factor	4.8157
				Skewness	3.0968

Table 19: *Features' ranking for cases A, B & C using the Kruskal-Wallis test.*

Kruskal-Wallis Test					
Case D		**Case E**		**Case F**	
Features	**H**	**Features**	**H**	**Features**	**H**
Standard deviation	18.0231	LOG_BPFI_Amplitude_BPFO_Amplitude	118.1788	LOG_BPFI_Amplitude_BPFO_Amplitude	118.1788
RMS	17.5023	LOG_BSF_Amplitude_BPFO_Amplitude	43.0433	LOG_BSF_Amplitude_BPFO_Amplitude	43.0433
Peak value	15.3398	BPFI_Amplitude	31.6875	BPFI_Amplitude	31.6875
Mean	11.6661	LOG_BPFI_Amplitude_BSF_Amplitude	29.0640	LOG_BPFI_Amplitude_BSF_Amplitude	29.0640
Kurtosis	10.9086	BPFO_Amplitude	24.4599	BPFO_Amplitude	24.4599
Shape factor	10.1311	BSF_Amplitude	17.7375	Standard deviation	18.0231
Clearance factor	5.9409			BSF_Amplitude	17.7375
Impulse factor	5.4390			RMS	17.5023
Crest factor	3.8549			Peak value	15.3398
Skewness	2.7573			Mean	11.6661
				Kurtosis	10.9086
				Shape factor	10.1311
				Clearance factor	5.9409
				Impulse factor	5.4390
				Crest factor	3.8549
				Skewness	2.7573

Table 20: *Features' ranking for cases D, E & F using the Kruskal-Wallis test.*

B.5 Classification Results

One-Way ANOVA					
Features \ **Algorithms**		**Decision Tree**	**SVM**	**k-NN**	**Naive Bayes**
Case A					
1	a	50.50%	50.80%	51.50%	50.50%
1-2	b	53.70%	56.00%	59.00%	47.90%
1-2-3	c	57.70%	58.00%	60.90%	45.60%
1-2-3-4	d	57.70%	57.70%	60.60%	46.60%
1-2-3-4-5	e	54.40%	58.30%	63.20%	43.60%
1-2-3-4-5-6	f	58.00%	60.60%	61.20%	47.20%
1-2-3-4-5-6-7	g	58.30%	65.10%	70.70%	47.20%
1-2-3-4-5-6-7-8	h	61.60%	65.80%	68.40%	49.50%
1-2-3-4-5-6-7-8-9	i	60.30%	65.50%	70.70%	47.90%
1-2-3-4-5-6-7-8-9-10	j	62.90%	70.00%	71.70%	48.90%
Case B					
1	a	65.80%	66.10%	63.80%	68.10%
1-2	b	68.10%	71.00%	67.40%	67.80%
1-2-3	c	65.50%	71.30%	67.10%	69.40%
1-2-3-4	d	64.80%	70.70%	66.80%	56.70%
1-2-3-4-5	e	63.50%	69.70%	67.40%	61.60%
1-2-3-4-5-6	f	65.50%	70.40%	70.40%	59.60%
Case C					
1	a	65.80%	66.10%	63.80%	68.10%
1-2	b	68.10%	71.00%	67.40%	67.80%
1-2-3	c	65.50%	71.30%	67.10%	69.40%
1-2-3-4	d	64.80%	70.70%	66.80%	56.70%
1-2-3-4-5	e	63.50%	69.70%	67.40%	61.60%
1-2-3-4-5-6	f	71.30%	70.40%	73.60%	62.20%
1-2-3-4-5-6-7	g	67.10%	69.10%	74.30%	59.30%
1-2-3-4-5-6-7-8	h	69.10%	69.40%	72.30%	59.60%
1-2-3-4-5-6-7-8-9	i	69.10%	71.00%	73.60%	60.90%
1-2-3-4-5-6-7-8-9-10	j	69.70%	71.00%	75.60%	60.60%
1-2-3-4-5-6-7-8-9-10-11	k	66.80%	70.70%	73.30%	64.20%
1-2-3-4-5-6-7-8-9-10-11-12	l	72.60%	72.00%	76.20%	65.50%
1-2-3-4-5-6-7-8-9-10-11-12-13	m	70.70%	73.30%	77.90%	64.80%
1-2-3-4-5-6-7-8-9-10-11-12-13-14	n	68.40%	74.30%	77.90%	66.10%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	o	70.00%	75.20%	80.10%	64.80%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16	p	68.70%	76.20%	78.80%	65.10%

Table 21: *Classification results for cases A, B & C following the one-way ANOVA ranking order.*

One-Way ANOVA					
Features \ **Algorithms**		**Decision Tree**	**SVM**	**k-NN**	**Naive Bayes**
Case D					
1	a	66.70%	63.60%	67.50%	61.90%
1-2	b	69.70%	67.50%	71.90%	55.40%
1-2-3	c	71.00%	68.80%	76.20%	55.40%
1-2-3-4	d	74.00%	73.20%	78.40%	64.10%
1-2-3-4-5	e	70.60%	74.90%	80.10%	63.20%
1-2-3-4-5-6	f	68.80%	70.10%	79.20%	64.50%
1-2-3-4-5-6-7	g	74.90%	72.70%	79.70%	65.80%
1-2-3-4-5-6-7-8	h	71.00%	73.60%	78.40%	69.70%
1-2-3-4-5-6-7-8-9	i	68.40%	77.50%	84.00%	67.50%
1-2-3-4-5-6-7-8-9-10	j	72.30%	80.10%	82.30%	65.40%
Case E					
1	a	82.70%	84.00%	86.10%	84.00%
1-2	b	80.10%	83.10%	82.70%	82.70%
1-2-3	c	81.00%	86.10%	84.80%	80.50%
1-2-3-4	d	83.50%	86.10%	82.70%	73.20%
1-2-3-4-5	e	84.80%	83.50%	82.70%	74.50%
1-2-3-4-5-6	f	84.00%	84.00%	84.80%	75.30%
Case F					
1	a	82.70%	84.00%	86.10%	84.00%
1-2	b	80.10%	83.10%	82.70%	82.70%
1-2-3	c	81.00%	86.10%	84.80%	80.50%
1-2-3-4	d	83.50%	86.10%	82.70%	73.20%
1-2-3-4-5	e	84.80%	83.50%	82.70%	74.50%
1-2-3-4-5-6	f	83.50%	84.40%	85.30%	74.90%
1-2-3-4-5-6-7	g	83.50%	84.80%	84.80%	76.60%
1-2-3-4-5-6-7-8	h	86.10%	86.10%	85.70%	78.40%
1-2-3-4-5-6-7-8-9	i	87.00%	87.40%	88.30%	79.70%
1-2-3-4-5-6-7-8-9-10	j	84.40%	87.00%	87.90%	80.50%
1-2-3-4-5-6-7-8-9-10-11	k	87.00%	87.00%	89.60%	81.80%
1-2-3-4-5-6-7-8-9-10-11-12	l	87.00%	88.70%	89.20%	84.00%
1-2-3-4-5-6-7-8-9-10-11-12-13	m	82.30%	87.00%	85.30%	81.00%
1-2-3-4-5-6-7-8-9-10-11-12-13-14	n	87.90%	87.00%	90.90%	82.70%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	o	86.10%	87.90%	90.00%	83.10%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16	p	85.70%	87.40%	90.50%	83.50%

Table 22: *Classification results for cases D, E & F following the one-way ANOVA ranking order.*

Kruskal-Wallis Test					
Algorithms Features		Decision Tree	SVM	k-NN	Naive Bayes
Case A					
1	a	46.90%	49.20%	47.90%	45.90%
1-2	b	47.90%	49.20%	47.90%	46.30%
1-2-3	c	55.00%	49.20%	59.00%	49.20%
1-2-3-4	d	59.90%	61.20%	69.70%	45.00%
1-2-3-4-5	e	61.60%	65.50%	78.80%	43.00%
1-2-3-4-5-6	f	62.90%	64.50%	76.20%	45.00%
1-2-3-4-5-6-7	g	60.60%	65.80%	74.60%	45.30%
1-2-3-4-5-6-7-8	h	62.50%	65.80%	69.10%	48.50%
1-2-3-4-5-6-7-8-9	i	60.30%	65.50%	70.70%	47.90%
1-2-3-4-5-6-7-8-9-10	j	62.90%	70.00%	71.70%	48.90%
Case B					
1	a	65.80%	66.10%	63.80%	68.10%
1-2	b	68.10%	71.00%	67.40%	67.80%
1-2-3	c	65.50%	70.00%	67.80%	59.30%
1-2-3-4	d	63.80%	70.40%	68.70%	61.20%
1-2-3-4-5	e	63.50%	69.70%	67.40%	61.60%
1-2-3-4-5-6	f	65.50%	70.40%	70.40%	59.60%
Case C					
1	a	65.80%	66.10%	63.80%	68.10%
1-2	b	68.10%	71.00%	67.40%	67.80%
1-2-3	c	65.50%	70.00%	67.80%	59.30%
1-2-3-4	d	63.80%	70.40%	68.70%	61.20%
1-2-3-4-5	e	63.50%	69.70%	67.40%	61.60%
1-2-3-4-5-6	f	65.50%	70.40%	70.40%	59.60%
1-2-3-4-5-6-7	g	67.10%	71.30%	71.30%	63.50%
1-2-3-4-5-6-7-8	h	66.40%	71.70%	72.00%	62.90%
1-2-3-4-5-6-7-8-9	i	65.80%	73.60%	74.30%	62.90%
1-2-3-4-5-6-7-8-9-10	j	67.10%	72.30%	73.90%	63.80%
1-2-3-4-5-6-7-8-9-10-11	k	67.80%	72.30%	76.20%	64.50%
1-2-3-4-5-6-7-8-9-10-11-12	l	69.40%	72.30%	77.50%	65.50%
1-2-3-4-5-6-7-8-9-10-11-12-13	m	71.70%	74.30%	78.50%	65.50%
1-2-3-4-5-6-7-8-9-10-11-12-13-14	n	70.40%	74.30%	78.20%	64.20%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	o	70.00%	75.20%	80.10%	64.80%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16	p	68.70%	76.20%	78.80%	65.10%

Table 23: *Classification results for cases A, B & C following the Kruskal-Wallis test ranking order.*

Kruskal-Wallis Test					
Algorithms Features		Decision Tree	SVM	k-NN	Naive Bayes
Case D					
1	a	63.20%	63.60%	63.60%	64.10%
1-2	b	61.90%	63.60%	63.60%	60.60%
1-2-3	c	64.90%	65.40%	71.90%	61.90%
1-2-3-4	d	68.00%	74.90%	80.50%	63.20%
1-2-3-4-5	e	69.30%	73.60%	84.80%	65.80%
1-2-3-4-5-6	f	64.90%	74.00%	86.60%	67.10%
1-2-3-4-5-6-7	g	65.80%	74.00%	84.80%	66.70%
1-2-3-4-5-6-7-8	h	74.90%	75.80%	87.00%	67.10%
1-2-3-4-5-6-7-8-9	i	68.40%	77.50%	84.00%	67.50%
1-2-3-4-5-6-7-8-9-10	j	72.30%	80.10%	82.30%	65.40%
Case E					
1	a	82.70%	84.00%	86.10%	84.00%
1-2	b	80.10%	83.10%	82.70%	82.70%
1-2-3	c	80.10%	85.30%	83.50%	71.40%
1-2-3-4	d	83.50%	86.10%	82.70%	73.20%
1-2-3-4-5	e	84.80%	83.50%	82.70%	74.50%
1-2-3-4-5-6	f	84.00%	84.00%	84.80%	75.30%
Case F					
1	a	82.70%	84.00%	86.10%	84.00%
1-2	b	80.10%	83.10%	82.70%	82.70%
1-2-3	c	80.10%	85.30%	83.50%	71.40%
1-2-3-4	d	83.50%	86.10%	82.70%	73.20%
1-2-3-4-5	e	84.80%	83.50%	82.70%	74.50%
1-2-3-4-5-6	f	84.00%	87.00%	86.60%	84.80%
1-2-3-4-5-6-7	g	87.00%	87.00%	88.70%	84.40%
1-2-3-4-5-6-7-8	h	84.00%	87.00%	86.60%	78.80%
1-2-3-4-5-6-7-8-9	i	84.40%	87.00%	88.70%	82.70%
1-2-3-4-5-6-7-8-9-10	j	82.70%	86.60%	90.90%	82.70%
1-2-3-4-5-6-7-8-9-10-11	k	83.50%	87.40%	90.90%	83.10%
1-2-3-4-5-6-7-8-9-10-11-12	l	86.10%	87.90%	90.50%	84.40%
1-2-3-4-5-6-7-8-9-10-11-12-13	m	85.70%	87.40%	90.90%	82.30%
1-2-3-4-5-6-7-8-9-10-11-12-13-14	n	86.60%	86.60%	90.50%	79.20%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15	o	86.10%	87.90%	90.00%	83.10%
1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16	p	85.70%	87.40%	90.50%	83.50%

Table 24: *Classification results for cases D, E & F following the Kruskal-Wallis test ranking order.*

List of Figures

List of Tables

List of MATLAB® Scripts

References

[Antoni, 2006] Antoni, J. (2006, January 30). *Fast computation of the kurtogram for the detection of transient faults.* Mechanical Systems and Signal Processing, 21(1), pp. 108–124. Retrieved from: https://doi.org/10.1016/j.ymssp.2005.12.002

[Antoni & Randall, 2004] Antoni, J., & Randall, R. B. (2004, November 6). *The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines.* Mechanical Systems and Signal Processing, 20(2), pp. 308–331. Retrieved from: https://doi.org/10.1016/j.ymssp.2004.09.002

[Barszcz, 2019] Barszcz, T. (2019). *Vibration-based condition monitoring of wind turbines.* Applied Condition Monitoring: Springer International Publishing. ISBN 978-3-030-05969-9. Retrieved from: https://doi.org/10.1007/978-3-030-05971-2

[Borghesani et al., 2012] Borghesani, P., Pennacchi, P., Randall, R. B., Sawalhi, N., & Ricci, R. (2012, December 4). *Application of cepstrum pre-whitening for the diagnosis of bearing faults under variable speed conditions.* Mechanical Systems and Signal Processing, 36(2), pp. 370–384. Retrieved from: https://doi.org/10.1016/j.ymssp.2012.11.001

[Broecker, 1975] Broecker, W. S. (1975, August 8). *Climatic Change: Are We on the Brink of a Pronounced Global Warming?* Science, 189(4201), pp. 460–463. Retrieved from: https://doi.org/10.1126/science.189.4201.460

[Burkov, 2019] Burkov, A. (2019, January 11). *The hundred-page machine learning book.* ISBN 978-1-9995795-0-0. Retrieved from: https://themlbook.com/

[CWRU, nd] Case Western Reserve University (CWRU) . (nd). Case Western Reserve University (CWRU) bearing data center. Retrieved from: https://engineering.case.edu/bearingdatacenter/welcome

[Goralski & Tan, 2019] Goralski, M. A., & Tan, T. K., (2019, October 31). *Artificial intelligence and sustainable development.* The International Journal of Management Education, 18(1). Retrieved from: https://doi.org/10.1016/j.ijme.2019.100330

[GWEC, 2021] Global Wind Energy Council (GWEC). (2021, March 25). *Global Wind Report 2021.* Retrieved from: https://gwec.net/global-wind-report-2021/#:~:text=Today%2C%20there%20is%20now%20743,carbon%20emissions%20of%20South%20America.

[IBM, 2020] IBM Cloud Education. (2020, July 15). *What is machine learning?*. Retrieved from: https://www.ibm.com/in-en/cloud/learn/machine-learning

[Magar et al., 2021] Magar, R., Ghule, L., Li, J., Zhao, Y., & Farimani, A. B. (2021, February 3). *FaultNet: a deep convolutional neural network for bearing fault classification.* IEEE Access, 9, pp. 25189–25199. Retrieved from: https://doi.org/10.1109/access.2021.3056944

[MathWorks®, 2013] MathWorks®. (2013). *Rolling Element Bearing (REB) fault diagnosis.* Retrieved from: https://es.mathworks.com/help/predmaint/ug/Rolling-Element-Bearing-Fault-Diagnosis.html?s_tid=srchtitle

[MathWorks®, 2021] MathWorks®. (2021). *Statistics and Machine Learning Toolbox™. Hypothesis tests.* Retrieved from: https://es.mathworks.com/help/stats/hypothesis-tests-1.html

[Mitchell, 1997] Mitchell, T. M. (1997, March 1). *Machine learning.* McGraw-Hill Series in Computer Science. ISBN 978-0-0704280-7-2. Retrieved from: https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf

[NASA, 2022] National Aeronautics and Space Administration (NASA). (2022, March 15). *Carbon Dioxide Concentration | NASA Global Climate Change.* Climate Change: Vital Signs of the Planet. Retrieved from: https://climate.nasa.gov/vital-signs/carbon-dioxide/

[Neupane & Seok, 2020] Neupane, D., & Seok, J. (2020, April 27). *Bearing fault detection and diagnosis using Case Western Reserve University dataset with deep learning approaches: A review.* IEEE Access, 8: pp. 93155–93178. Retrieved from: https://doi.org/10.1109/access.2020.2990528

[NIST/SEMATECH, 2012] *NIST/SEMATECH e-Handbook of Statistical Methods.* (2012, April). Retrieved from: https://doi.org/10.18434/M32189

[Randall, 2010] Randall, R. B. (2010, December 19). *Vibration-based condition monitoring: industrial, automotive and aerospace applications.* John Wiley & Sons. ISBN 978-0-470-74785-8. Retrieved from: https://doi.org/10.1002/9780470977668

[Randall et al., 2011] Randall, R. B., Sawalhi, N., & Coats, M. (2011, June 1). *A comparison of methods for separation of deterministic and random signals.* International Journal of Condition Monitoring, 1(1), pp. 11–19. Retrieved from: https://doi.org/10.1784/204764211798089048

[Reece & Danforth, 2017] Reece, A. G., & Danforth, C. M. (2017, August 8). *Instagram photos reveal predictive markers of depression.* EPJ Data Science, 6(1). Retrieved from: https://doi.org/10.1140/epjds/s13688-017-0110-z

[Rich, 1983] Rich, E. (1983, February). *Artificial intelligence.* New York: McGraw-Hill. Artificial Intelligence, 28(1), pp. 119–121. Retrieved from: https://doi.org/10.1016/0004-3702(86)90034-2

[Rodríguez Pozueta, 2016] Rodríguez Pozueta, M. A. (2016). *Máquina asíncrona doblemente alimentada.* Universidad de Cantabria. Retrieved from: https://personales.unican.es/rodrigma/PDFs/As%C3%ADncrona%20doblemente%20alimentada.pdf

[Rudge et al., 2010] Rudge, J. F., Kleine, T., & Bourdon, B. (2010, May 10). *Broad bounds on Earth's accretion and core formation constrained by geochemical models.* Nature Geoscience, 3(6), pp. 439–443. Retrieved from: https://doi.org/10.1038/ngeo872

[Samanta & Al-Balushi, 2003] Samanta, B., & Al-Balushi, K. R. (2003, March 19). *Artificial neural network based fault diagnostics of rolling element bearings using time-domain features.* Mechanical Systems and Signal Processing, 17(2), pp. 317–328. Retrieved from: https://doi.org/10.1006/mssp.2001.1462

[Sathya & Abraham, 2013] Sathya, R., & Abraham, A. (2013, February). *Comparison of supervised and unsupervised learning algorithms for pattern classification.* International Journal of Advanced Research in Artificial Intelligence, 2(2), pp: 34-38. Retrieved from: https://doi.org/10.14569/ijarai.2013.020206

[Sawalhi & Randall, 2011] Sawalhi, N., & Randall, R. B. (2011, May 30). *Signal pre-whitening using cepstrum editing (liftering) to enhance fault detection in rolling element bearings.* In Proceedings of the 24 international congress on condition monitoring and diagnostic engineering management (comadem2011), pp. 330-336.

[Simmons & Chapell, 1988] Simmons, A. B., & Chappell, S. G. (1988, April). *Artificial intelligence-definition and practice.* IEEE Journal of Oceanic Engineering, 13(2), pp 14–42. Retrieved from: https://doi.org/10.1109/48.551

[Smith & Randall, 2015] Smith, W. A., & Randall, R. B. (2015, May 11). *Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study.* Mechanical Systems and Signal Processing, vol. 64-65, pp. 100–131. Retrieved from: https://doi.org/10.1016/j.ymssp.2015.04.021

[SOCR (a), 2002] Statistics Online Computational Resource (SOCR). (2002, January 1). *Chi-Square distribution tables.* Retrieved from: http://www.socr.ucla.edu/Applets.dir/ChiSquareTable.html

[SOCR (b), 2002] Statistics Online Computational Resource (SOCR). (2002, January 1). *F-distribution tables.* Retrieved from: http://www.socr.ucla.edu/Applets.dir/F_Table.html

[UN, 2015] United Nations (UN) General Assembly. (2015, October 21). *Transforming our world: The 2030 agenda for sustainable development.* United Nations: New York, NY, USA. Report No. A/RES/70/1. Retrieved from: https://stg-wedocs.unep.org/bitstream/handle/20.500.11822/11125/unepswiosm1inf7sdg.pdf?sequence=1

[Zhang et al., 2017] Zhang, W., Peng, G., & Li, C. (2017, February 9). *Bearings fault diagnosis based on convolutional neural networks with 2-D representation of vibration signals as input.* MATEC Web of Conferences, 95, 13001. Retrieved from: https://doi.org/10.1051/matecconf/20179513001

Universitat Politècnica de València

Escola Tècnica Superior D'Enginyeria Industrial

Vibration analysis-based diagnosis of faulty rolling element bearings with applications in condition monitoring of wind turbines. A further machine learning and deep learning-based classification

Budget

Author:

David Cascales Fulgencio

Supervisor:

Eduardo Quiles Cucarella

July 6, 2022

Document II

BUDGET

Table of Contents

1 Budget Summary

Fees	
Project personnel	10.800,00 €
Expenses	
Data processing	500,00 €
TOTAL BUDGET	
Tax base	11.300,00 €
VAT (21%)	2.373,00 €
TOTAL BUDGET	13.673,00 €

2 Budget Breakdown

2.1 Fees for Project Personnel

Role	**Number of hours**	**Hourly rate**	**Total costs**	**Justification**
Project Supervisor	300	20,00 €	6.000,00 €	Establish guidelines and supervise the correct development of the project.
Project Developer	300	16,00 €	4.800,00 €	Perform data analysis and produce a report.

2.2 Data Processing Costs

Description of item	**Quantity**	**Cost per unit**	**Total costs**	**Justification**
MathWorks® annual license (academic use) Nº 40525643 & 40525644	2	250,00 €	500,00 €	Software licence for data processing.

*These licences cannot be included in future project budgets during their validity as they are fully amortised in this project.