Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/SA6.20250047
Published: May 12, 2025

Fault Analysis in Self-Aligning Roller Bearings Using Machine Learning Techniques with kNN and Decision Tree Algorithms

Rogerio Daniel Dantas
Professor of Industrial Automation
Federal Institute of Sao Paulo (IFSP)
Guarulhos, Sao Paulo, Brazil
rogerio.dantas@ifsp.edu.br

Marcelo George Griese
São Paulo State Faculty of Technology (FATEC)
São Bernardo do Campo, Sao Paulo, Brazil
marcelo.griese@fatec.sp.gov.br

João Victor Pelegrino
Master's student in mechanical engineering
Federal Institute of Sao Paulo (IFSP)
São Paulo, Sao Paulo, Brazil
pelegrino.jv@gmail.com

Caique Movio Pereira de Souza
Professor of mechanical, electrical and mechatronic engineering
Santo Amaro University (UNISA)
São Paulo, Sao Paulo, Brazil
caiquemovio@gmail.com

Vanessa Seriacopi
Professor of mechanical engineering
Mauá Institute of Technology (IMT)
São Caetano do Sul, Sao Paulo, Brazil
vanessa.seriacopi@gmail.com

Wilson Carlos da Silva Junior
Professor of Industrial Automation
Federal Institute of Sao Paulo (IFSP)
Guarulhos Sao Paulo, Brazil
wilsoncarlos@ifsp.edu.br

Abstract

This paper presents a study on fault identification in self-aligning roller bearings. For this purpose, a test bench was used for experiments and data acquisition, utilizing a system composed of a microcontroller board and a MEMS-type accelerometer sensor. The main objective is to determine which Machine Learning techniques are most efficient in fault identification. The kNN (K-Nearest Neighbors) and Decision Tree algorithms were tested, with statistical descriptors applied to the acquired data. Additionally, a feature reduction study was conducted using the PCA (Principal Component Analysis) technique and the ANOVA (Analysis of Variance) statistical method for time-domain data analysis.The data were also transformed into the frequency domain before being applied to the algorithms. The kNN algorithm with frequency-domain data achieved the best result, with an accuracy of 98%.

Keywords
Fault Analysis, Bearings, Machine Learning, Accelerometer, Sensor.

1. Introduction
In modern industries, rotating machinery accounts for more than 90% of the equipment in use (Han et al. 2021). Rotating machines encompass a wide variety of equipment such as motors, compressors, spindles, and mechanical gearboxes (Jiangquan et al. 2020).

As one of the main machine elements that compose a rotating machine, the rolling bearing is designed to support axial and radial loads, as well as to reduce rotational friction with the shaft (Wang et al. 2019). In general, rotating machines are used for extended periods under complex and variable workload conditions. As a result, bearings are frequently subjected to premature mechanical wear (Deveci et al. 2021). This characteristic makes bearings a critical component, accounting for approximately 55% of failures in rotating machinery (Duan et al. 2019).

When bearings fail unexpectedly, they can cause significant damage to equipment, leading to high repair and replacement costs, as well as unplanned production downtime. This directly affects the operational efficiency of the equipment and causes substantial financial losses for companies (Deveci et al. 2021).

Vibration signal analysis is the most widely used technique for assessing the mechanical condition of machinery, as vibration is one of the first detectable symptoms of degradation. It is capable of identifying around 70% of common mechanical failures associated with rotating machines (Alhams et al. 2024).

Machine Learning algorithms have opened new possibilities for analyzing large volumes of data in real time, without the need for subject matter experts. Moreover, low-cost sensors such as MEMS are revolutionizing the way predictive maintenance is carried out. Combined with Artificial Intelligence (AI) techniques and the Internet of Things (IoT), prescriptive maintenance is emerging as a new paradigm.

1.1 Objectives
The objective of this study is to investigate the application of vibration signal analysis combined with machine learning techniques for fault detection in rolling bearings used in rotating machinery. The work aims to contribute to the development of predictive and prescriptive maintenance strategies by leveraging low-cost sensors, such as MEMS accelerometers, and advanced data analysis methods. This approach seeks to enhance the operational efficiency and reliability of industrial equipment by enabling early detection of mechanical degradation and reducing unexpected machine downtime.

1. Methods
In this study, data acquisition was carried out using a bearing failure test bench located at the Instituto Federal de São Paulo – Campus Guarulhos. This test bench comprises a geared motor connected to two bearing housings via an elastic coupling and a split shaft. Additionally, the base of the setup is equipped with a Vibra-Stop vibration isolator, designed to isolate the experimental bench from external noise and vibrations. The speed control can be adjusted via a frequency inverter, allowing for experiments to be conducted at different rotational speeds.
For this study, tests were performed using two types of manually introduced defects in the bearings: roller defect and outer race defect. Figure 1 shows the test bench along with the defective bearings.

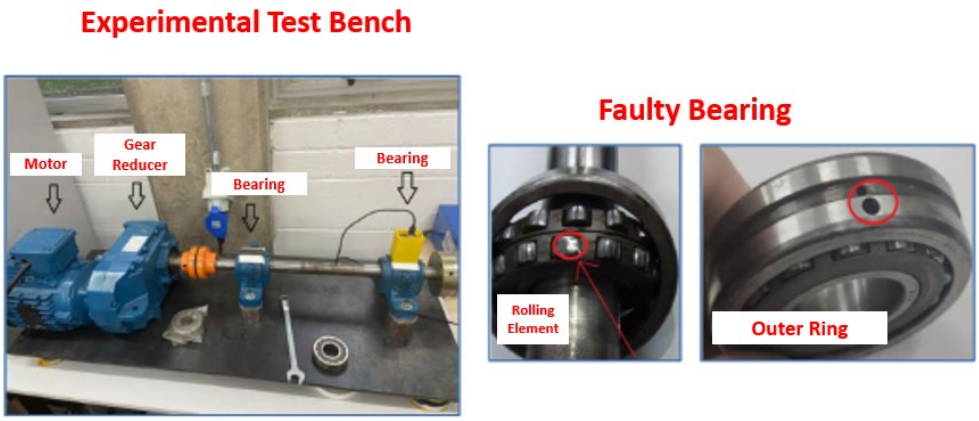

Figure 1. Experimental Test Bench and Bearings with Defects

Data acquisition was conducted using a circuit composed of an ESP32 microcontroller connected to a digital MEMS (Micro-Electro-Mechanical Systems) accelerometer sensor, model ADXL345. This sensor is a small, thin, ultra-low-power triaxial device that provides high-resolution (13-bit) measurements up to ±16g (m/s²), communicating via a digital SPI (3- or 4-wire) or I2C interface. The data are formatted as 16-bit two's complement.

In this study, only the Z-axis data were considered. A program was developed in C/C++ using the Arduino IDE so the ESP32 could read the sensor data and transmit it via serial communication to a computer. The data sent by the ESP32 were stored in .CSV format using a plugin called ArduSpreadsheet, available in the Arduino IDE. Figure 2 shows the vibration data acquisition circuit and the interface displaying the received data in .CSV format.

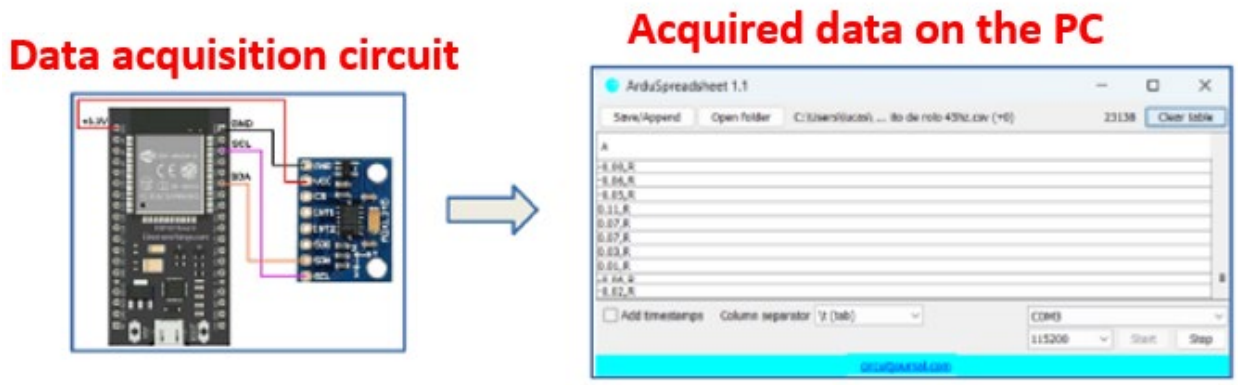

Figure 2. Data acquisition circuit.

At the end of the experiments, three data classes were acquired, focusing exclusively on the bearing in the output shaft housing: Normal (N), Roller Defect (R), and Outer Race Defect (OR). The acquisition settings used in this study were: output shaft speed of 54.8 rpm, sampling rate of 530 Hz, and 20 minutes of testing duration. Each test yielded 600,000 data samples per class. Working with raw data is not always the most effective approach. Therefore, it was necessary to preprocess the data for optimal treatment and feature extraction for classification algorithms.

1. Statistical Methods:

For feature extraction from the vibration signals acquired from the bearings, the following descriptors were used:

$$Xm = \text{mean}(x) \quad (1)$$

$$Xpeak = \max(|x|) \quad (2)$$

$$\text{Xroot} = \sqrt{mean\ (\sqrt{|x|})^2},\ \text{Xclearance} = \frac{Xpeak}{Xroot} \quad (3)$$

$$\text{Xroot} = \sqrt{mean\ (\sqrt{|x|})^2} \quad (4)$$

$$\text{Xkurtosis} = \text{kurtosis(x,fisher=False)} \quad (5)$$

$$\text{Ximpulse} = \frac{Xpeak}{\frac{1}{N}\sum_I^N |x1|} \quad (6)$$

$$\text{Xstd=std(x,ddof=1)} \quad (7)$$

$$\text{Xskewness} = \text{skew(x)} \quad (8)$$

$$\text{Xrms} = \sqrt{mean\ (\sqrt{(x)^2}},\ \text{Xshape} = \frac{Xpeak}{\frac{1}{N}\sum_I^N |x1|} \quad (9)$$

$$\text{Xrms} = \sqrt{mean\ (\sqrt{(x)^2}} \quad (10)$$

$$\text{Xcrest} = \frac{Xpeak}{Xrms} \quad (11)$$

$$\text{Xpeak2peak} = 2.\ \text{Xpeak} \quad (12)$$

$$\text{Xrssq} = \sqrt{\sum_{i=1}^N x_i^2} \quad (13)$$

2. PCA - Principal Component Analysis:

PCA is a powerful technique used to extract structure from potentially high-dimensional datasets. It involves extracting the q eigenvectors associated with the largest q eigenvalues of the input data distribution (Jacomini et al. 2012). Given an original dataset of l samples (features), forming elements of a vector x ∈ R1, the objective is to apply a linear transformation to obtain a new set of samples:

$$y = A^T X \quad (14)$$

so that the components of y are uncorrelated. In a second stage, the most significant components are selected. This process is summarized. Estimate the covariance matrix C. Normally, the mean value is assumed to be zero, i.e., E[x]=0. In this case, the covariance and autocorrelation matrices coincide, R=E[x(xT)]=C. Otherwise, the mean is subtracted. Given N feature vectors xi ∈ Rl, I = 1,2,….., N, the autocorrelation matrix estimation is given by:

$$R = \frac{1}{N}\sum_{i=1}^N xi\ x_i^T \quad (15)$$

3. ANOVA - Analysis of Variance:

ANOVA is a statistical method used to compare the means of two or more experiments. This evaluation checks whether the differences between means are statistically significant. It tests whether factors produce systematic changes in a variable of interest (Vieira 2006). The proposed factors can be either quantitative or qualitative, while the dependent variable must be quantitative (interval scale) and observed within the factor classes (treatments). Model hypotheses:

Three basic assumptions must be satisfied for ANOVA to be applicable:

1. Samples must be random and independent.
2. Samples must be drawn from normally distributed populations.
3. Populations must have equal variances.

The hypotheses are H0 and H1. If H0 is true, there is no significant difference between the groups. However, if H1 is accepted, a post-hoc test is required to compare the means and determine which are statistically significant. The Tukey test is used for this purpose, as shown in Equation (16):

$$H0 : \mu dist1 = \mu dist2 = \mu dist4 = \mu dist8 \quad (16)$$

C. Classification Algorithms

1. KNN – K-Nearest Neighbors:

KNN is a supervised learning algorithm that groups data into coherent subsets and classifies newly introduced data based on its similarity to previously trained data. Figure 3 illustrates a two-class example, in which a sample (blue dot)

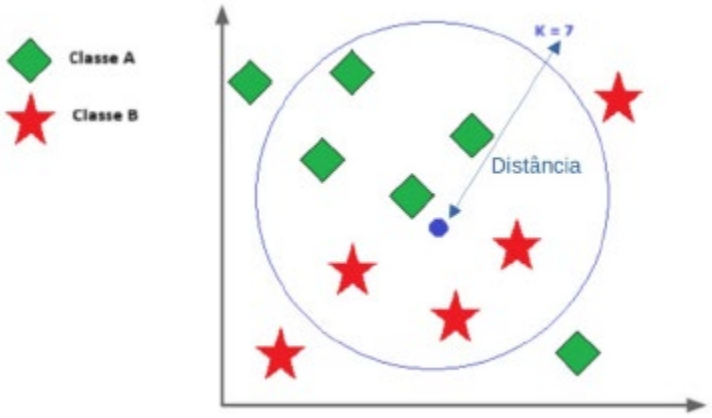

Figure 3. Example of classification with kNN with two classes and K=7

is being classified. As shown, since there are more elements in the circle belonging to class A, the blue dot will be classified as a member of class A.

$$\text{DE (p,q)} = \sqrt{(p_1 - q_1)^2} + \cdots + (p_n - q_n)^2 \tag{17}$$

Or more compactly:

$$\text{DE (p,q)} = = \sqrt{\sum_{i=1}^{n}(p_1 - q_1)^2} \tag{18}$$

Where p = (p1,…,pn) and q = (q1,…,qn) are two n-dimensional points. The constant r must be chosen based on the problem. In Figure 3, the distance from the unknown element (blue dot) to all points within the blue radius must be calculated. The value of K depends on the dataset and should preferably be an odd number. In multi-class scenarios, the graphical representation is in n-dimensional space [4]. Optimization algorithms can also be used to determine the optimal K value.

2. Decision Trees:

Decision Trees are non-parametric supervised learning algorithms used for both classification and regression. The structure resembles a tree: "nodes" represent decision points and "branches" are the paths connecting them. Each node poses a question about the data, and based on the answer (typically “yes” or “no”), the path proceeds along one of the branches. This continues until a terminal node, or “leaf node,” is reached, where a final classification is made.

The splitting process at each node is crucial to the tree’s learning. These splits aim to maximize the purity of the resulting branches, meaning each branch should ideally contain instances from the same class. Common measures include entropy and the Gini index.

While simple problems may involve only a few nodes and branches, real-world problems often lead to more complex trees with deeper hierarchies.

3. Results and Discussion

To perform all the tests, the Google Colab platform was used to run the Python codes, employing the scikit-learn machine learning library as the main tool, as presented in Figure 4.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

import time
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix,
↪  accuracy_score, classification_report
from sklearn.feature_selection import f_classif
from scipy.stats import kurtosis, skew
```

Figure 4. Programming performed in Google Colab

To enable the application of statistical descriptor calculations and the Fourier Transform, the data were segmented into windows of 100 samples, as presented in Figure 5.

```
def dividir_em_amostras(array, tamanho_amostra=100):
    if len(array) % tamanho_amostra != 0:
        raise ValueError("O tamanho do array deve ser
        ↪  múltiplo do tamanho da amostra.")
    amostras = np.array_split(array, len(array) //
    ↪  tamanho_amostra)
    return amostras
amostras_n = dividir_em_amostras(data_n)
amostras_pe = dividir_em_amostras(data_pe)
amostras_r = dividir_em_amostras(data_r)
```

Figure 5. Data segmented into windows of 100 samples in Google Colab.

The algorithms used in the experiments were the kNN and decision tree classifiers. The dataset was split into two subsets: 70% for training and 30% for testing. In Figure 6 are the programming lines used for data splitting, creation and training of each algorithm, and execution of predictions using the test data.

```
# Separar os dados em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X_pca,
↪  y, test_size=0.3, random_state=42)

# Treinar o classificador KNN
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Fazer predições no conjunto de teste
predictions_knn = knn.predict(X_test)

# Treinar o modelo de árvore de decisão
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
predictions_AD = model.predict(X_test)
```

Figure 6. Training and testing of kNN and Decision Tree classifiers using 70% of the dataset for training and 30% for testing.

A total of five tests were conducted, as follows:
• Test 1: The raw vibration data was used directly as the sole feature in the kNN classifier;
• Test 2: The 13 statistical descriptors described in the previous section were used as features for both the kNN and Decision Tree classifiers;
• Test 3: Principal Component Analysis (PCA) was applied to reduce the 13 statistical descriptors from Test 2 to just two principal components; the classification was then repeated using the kNN and Decision Tree algorithms;
• Test 4: The ANOVA technique was applied to determine which of the 13 statistical descriptors from Test 2 were the most relevant; the classification was then repeated using the kNN and Decision Tree algorithms;
• Test 5: The Discrete Fourier Transform (DFT) was computed to extract frequency-domain features from the segmented data windows; classification was then repeated using the kNN and Decision Tree algorithms.
The following sections present the confusion matrices and evaluation metrics for each test.

A. Test 1: Use of Raw Data
This test consisted essentially of using the raw vibration values as the sole feature, applying the kNN classifier. After several tests varying the value of *k*, the best result was obtained with *k = 50*, as shown in Table 1.

Table 1. Classification Report: Test 1 - kNN

k = 50	Precision	Recall	F1 - score	Support
Normal	0.52	0.60	0.56	179862
Outer Ring Fault	0.93	0.95	0.94	180094
Roller Fault	0.56	0.47	0.51	180044
Accuracy			0.67	540000
Macro Average	0.67	0.67	0.67	540000
Weighted Average	0.67	0.67	0.67	540000

With an achieved accuracy of 67%, Figure 7 presents the confusion matrix for this test.

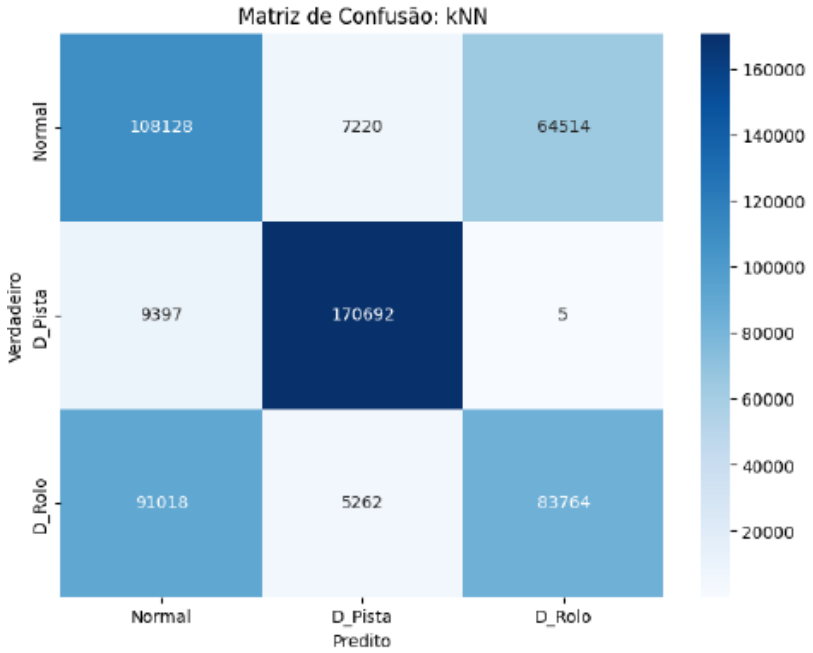

Figure 7. Confusion Matrix Test 1 - kNN Algorithm

The conclusion from the results is that the approach adopted in this test is not suitable for the analysis objectives, and better attributes must be sought to feed the algorithms.

B. Test 2: Using Statistical Descriptors

In order to achieve better accuracy compared to Test 1, 13 statistical descriptors were used as attributes for the kNN and Decision Tree algorithms. The performance of the classifiers, as shown in Tables 2 and 3, shows significant improvement, reaching 92% accuracy with kNN and 88% with the Decision Tree. For the kNN classifier, different values of k were tested, with k=30 yielding the best performance. Upon analyzing the confusion matrices in Figures 5 and 6, it is clear that both classifiers had difficulty classifying the Normal and Defect on the roller classes.

Table 2. Classification Report: Test 2 - kNN

k = 30	Precision	Recall	F1 - score	Support
Normal	0.86	0.91	0.88	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.90	0.85	0.88	1798
Accuracy			0.92	5400
Macro Average	0.92	0.92	0.92	5400
Weighted Average	0.92	0.92	0.92	5400

Table 3. Classification Report: Test 2 (Decision Tree)

	Precision	Recall	F1 - score	Support
Normal	0.82	0.82	0.82	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.82	0.82	0.82	1798
Accuracy			0.88	5400
Macro Average	0.88	0.88	0.88	5400
Weighted Average	0.88	0.88	0.88	5400

C. Test 3: Reducing Features through PCA

The goal of this test is to address the difficulty encountered in Test 2, where both algorithms struggled to classify the Normal and Roller Defect classes. To achieve this, a PCA analysis was performed to reduce the 13 statistical features to just 2 of the most relevant ones. The result of this analysis can be observed in Figure 7, where it is evident that the data from the Normal and Roller Defect classes do not present a clear and distinct boundary.

However, the tests were repeated using the kNN and Decision Tree classifiers, with the performance results presented in Tables 4 and 5. It is possible to observe a reduction in the accuracy values in both cases.

Table 4. Classification Report: Test 3 (kNN)

k = 3	Precision	Recall	F1 - score	Support
Normal	0.83	0.88	0.85	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.87	0.82	0.85	1798
Accuracy			0.90	5400
Macro Average	0.90	0.90	0.90	5400
Weighted Average	0.90	0.90	0.90	5400

D. Test 4: Reducing Attributes through ANOVA

In order to improve the performance presented in test 2, the ANOVA method was applied to determine the most relevant attributes, thus optimizing the attributes used in the classifiers. Figure 8 shows the confusion matrix for test 2, and Figure 9 shows the confusion matrix for the decision tree. Table 5 and 6 presents the results of the ANOVA method.

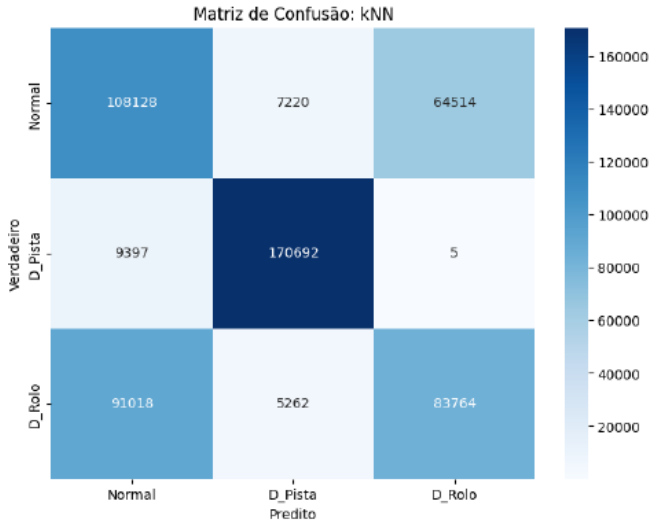

Figure 8. Confusion Matrix Test 4 - kNN Algorithm

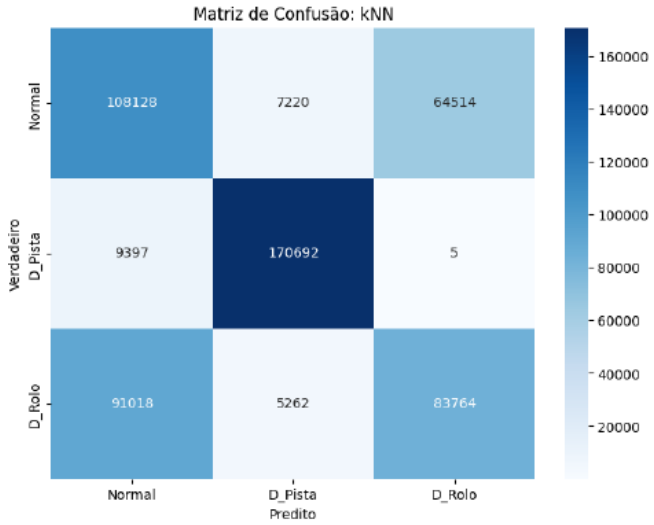

Figure 9. Confusion Matrix Test 4 – Decision Tree

Table 5. Classification Report: Test 4 (Decision Tree)

	Precision	Recall	F1 - score	Support
Normal	0.79	0.79	0.79	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.79	0.79	0.79	1798
Accuracy			0.86	5400
Macro Average	0.86	0.86	0.86	5400
Weighted Average	0.86	0.86	0.86	5400

Table 6. ANOVA Results

Attribute	F-Value	P-Value
Root Amplitude (Xroot)	379700.596690	0.000000e+00
Root Mean Square (Xrms)	178336.891917	0.000000e+00
Root of Sum of Squares (Xrssq)	178336.891917	0.000000e+00
Standard Deviation (Xstd)	175393.212028	0.000000e+00
Peak Value (Xpeak)	81234.116103	0.000000e+00
Peak-to-Peak Value (Xpeak2peak)	81234.116103	0.000000e+00
Clearance Factor (Xclearance)	73456.273223	0.000000e+00
Mean Value (Xm)	6038.797510	0.000000e+00
Shape Factor (Xshape)	104.653642	6.482861e-46
Kurtosis (Xkurtosis)	102.949665	3.494260e-45
Impulse Factor (Ximpulse)	92.042312	1.696623e-40
Crest Factor (Xcrest)	73.263528	2.045890e-32
Skewness (Xskewness)	23.717060	5.168469e-11

The tests were redone using the 4 most relevant attributes, and the results are presented in Tables 7 and 8. Where we observe that there was no change in the performance of the classifiers.

Table 7. Classification Report: Test 4 (kNN)

k = 30	Precision	Recall	F1 - score	Support
Normal	0.85	0.90	0.88	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.90	0.84	0.87	1798
Accuracy			0.92	5400
Macro Average	0.92	0.92	0.92	5400
Weighted Average	0.92	0.92	0.92	5400

Table 8. Classification Report: Test 4 (Decision Tree)

	Precision	Recall	F1 - score	Support
Normal	0.82	0.81	0.81	1782
Outer Ring Fault	1.00	1.00	1.00	1820
Roller Fault	0.81	0.82	0.82	1798
Accuracy			0.88	5400
Macro Average	0.88	0.88	0.88	5400
Weighted Average	0.88	0.88	0.88	5400

Although there was no improvement in classification accuracy, the reduction from 13 to 4 attributes brings significant computational cost savings for both classifiers.

E. Test 5: Using DFT (Discrete Fourier Transform)

In the fifth test, a new approach was adopted for attribute extraction. At this stage, the frequency signature of the sampled signals was used by applying the Discrete Fourier Transform (DFT) technique. The results obtained are summarized in Tables 9 and 10, where a significant improvement in the accuracy of both classifiers can be observed. Compared to Test 2, the kNN classifier improved its accuracy from 92% to 98%, while the Decision Tree classifier improved from 88% to 95%.

Table 9. Classification Report: Test 5 (kNN)

k = 30	Precision	Recall	F1 - score	Support
Normal	0.97	0.97	0.97	1204
Outer Ring Fault	1.00	1.00	1.00	1220
Roller Fault	0.97	0.97	0.97	1176
Accuracy			0.98	3600
Macro Average	0.98	0.98	0.98	3600
Weighted Average	0.98	0.98	0.98	3600

Table10. Classification Report: Test 5 (Decision Tree)

	Precision	Recall	F1 - score	Support
Normal	0.94	0.92	0.93	1204
Outer Ring Fault	1.00	1.00	1.00	1220
Roller Fault	0.92	0.94	0.93	1176
Accuracy			0.95	3600
Macro Average	0.95	0.95	0.95	3600
Weighted Average	0.95	0.95	0.95	3600

By analyzing the confusion matrices of both classifiers in Figures 10 and 11, it is clear that the new frequency-based attributes improved the separation between the Normal and Roller Defect classes. As this test yielded the best results, a comparison of the execution time for each classifier was carried out, since execution time can be related to computational cost. The results of this analysis are shown in Table 10, where it can be seen that the kNN classifier has a shorter training time. However, the Decision Tree classifier proves to be faster in the prediction phase.

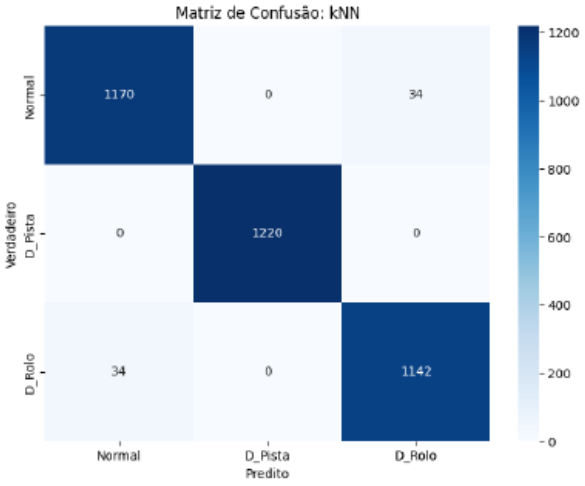

Figure 10. Confusion Matrix Test 5 – Knn

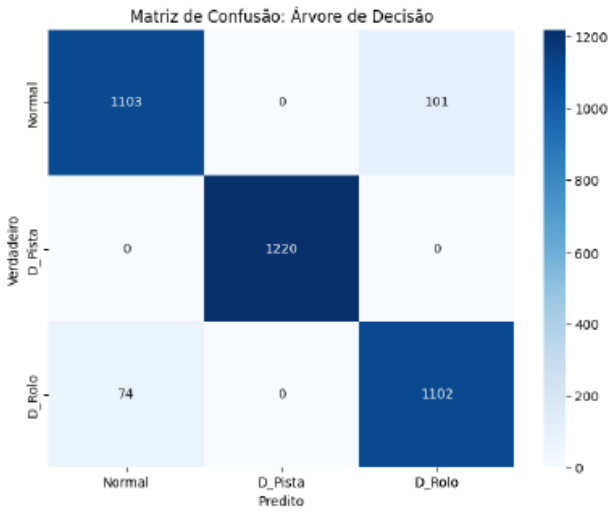

Figure 11. Confusion Matrix Test 5 – Decision Tree.

4. Conclusion

It can be concluded that both the k-Nearest Neighbors (kNN) and Decision Tree classifiers can be effectively applied in real-world bearing fault analysis, with kNN demonstrating higher accuracy for the dataset used in this study. The analysis of the frequency-domain signature of the sampled signal proved highly effective in identifying subtle characteristics that are not easily perceived in time-domain data analysis, presenting itself as an excellent alternative in situations where statistical descriptors alone are insufficient to distinguish between classes. In addition to the objective conclusions drawn from this fault analysis study, the execution of the work revealed several relevant insights. The use of Google Colab and the scikit-learn library greatly facilitated the implementation and testing of the classification algorithms. Furthermore, data preprocessing and feature selection were shown to be critical in achieving the best results among the techniques evaluated. This study presented a clear evolution in performance, progressing from an initial test with an accuracy of 67% to a final result of 98%. It is also important to note that the data used in this work were derived from a study with similar objectives [8], which utilized a Convolutional Neural Network (CNN). That study reported an accuracy above 99%. Although this result is slightly better, the CNN algorithm is considerably more complex compared to kNN and Decision Tree classifiers, indicating the need for further investigation into computational cost to determine whether the improvement in performance justifies the added complexity. Finally, vibration-based bearing fault analysis is a widely used approach, with numerous related studies in the literature. This method enables early fault detection, increasing the reliability and service life of equipment, while also reducing maintenance costs.

References

Alhams, A., Enhanced Bearing Fault Diagnosis Through Trees Ensemble Method and Feature Importance Analysis, *Journal of Vibration Engineering & Technologies*, 2024, https://doi.org/10.1007/s42417-024-01405-0.

Deveci, B. U., A Comparison of Deep Transfer Learning Methods on Bearing Fault Detection, *Proceedings of the 2021 International Conference on Future Internet of Things and Cloud (FiCloud 2021)*, IEEE, pp. 285–292, 2021, https://doi.org/10.1109/FiCloud49777.2021.00048.

Duan, J., A Novel Bearing Health Prognostic Method Based on Time-Frequency Analysis and LSTM, *Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Qingdao 2019)*, IEEE, pp. xx-xx, 2019, https://doi.org/10.1109/PHM-Qingdao46334.2019.8942821.

Han, T., Rolling Bearing Fault Diagnosis with Combined Convolutional Neural Networks and Support Vector Machine, *Measurement: Journal of the International Measurement Confederation*, vol. 177, p. 109022, 2021, https://doi.org/10.1016/j.measurement.2021.109022.

Jacomini, R. S., Yin, H., Costa, J. A. F. and Barreto, G., Comparison of PCA and ANOVA for Information Selection of CC and MLO Views in Classification of Mammograms, *Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2012)*, pp. 117–126, Berlin, Germany, 2012.

Vieira, S., *Análise de Variância: ANOVA*, Edição em Português, Capa comum, Jan. 2006.

Wang, J., A Deep Learning Method for Bearing Fault Diagnosis Based on Time-Frequency Image, *IEEE Access*, vol. 7, pp. 42373–42383, 2019.
Willenshofer, L. A., Fault Detection in Bearings Based on Convolutional Neural Networks Using Low-Cost MEMS Accelerometers, *Proceedings of the COB-2023*, Federal Institute of São Paulo, São Paulo, Brazil, 2023.
Zhang, J., A New Bearing Fault Diagnosis Method Based on Modified Convolutional Neural Networks, *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 439–447, Feb. 2020, https://doi.org/10.1016/j.cja.2019.05.007.

Biographies

Rogerio Daniel Dantas is a professor at the Federal Institute of São Paulo (IFSP), specializing in industrial automation. He holds a Bachelor's degree in Industrial Mechatronics Technology from Faculdade de Tecnologia Termomecanica (2005) and a Master's degree in Information Engineering from the Federal University of ABC (2010). Currently, he teaches at the basic, technical, and technological levels at the Federal Institute of São Paulo. He has experience in the fields of Robotics, Mechatronics, and Automation, with emphasis on topics such as Principal Component Analysis (PCA), Analysis of Variance (ANOVA), information selection, mammogram analysis, data acquisition, diagnostics, mammography, embedded machine learning, and Industry 4.0.

Marcelo George Griese as a university professor at prestigious institutions in the ABC Paulista region, teaching courses in automation and cloud computing. He also provides consultancy and services in PLC programming and embedded systems and owns a company that sells products and projects in the field of Radio Frequency Identification (RFID). An enthusiast of the Maker movement and entrepreneurship, he is involved in various projects in this area both as a hobby and professionally. Additionally, he actively participates in projects that promote and mentor entrepreneurial initiatives within the schools where he works.

João Victor Pelegrino is a Master's student in Mechanical Engineering at the Instituto Federal de São Paulo. He completed his High School (2016) at Colégio São João Ilhabela, graduated with a Bachelor's degree in Control and Automation Engineering from the Instituto Federal de Educação, Ciência e Tecnologia de São Paulo-Guarulhos (2022), and has proficiency in Python, Javascript, Node-RED, Ladder Logic for PLC, C++, Machine Learning, SCADA systems, and Excel. He has completed an English course with WiseUP Online, an Artificial Intelligence course with Huawei, and a basic Power BI course with ENG DTP Multimedia. Caíque was awarded a Teaching Scholarship (2022) for the development of robotics applications using RobotStudio and CoppeliaSim. He was a speaker on Node-RED and Industry 4.0 at Campus Party (2022, 2023).

Caique Movio Pereira de Souza is a university professor in Mechanical Engineering at Universidade de Santo Amaro. He holds a PhD in Materials Engineering and Nanotechnology from Universidade Presbiteriana Mackenzie and a Master's in Mechanical Engineering from Instituto Federal de São Paulo. He graduated in Industrial Mechatronics from Faculdade de Tecnologia Termomecanica. Caique specializes in systems automation, industrial process automation, and materials engineering, with a focus on graphene-based composites and prosthesis failure analysis.

Vanessa Seriacopi is an Associate Professor and Researcher at the Instituto Mauá de Tecnologia, teaching courses related to Manufacturing Processes, Quality Control, and Materials. She holds a Bachelor's degree in Mechatronic Engineering from Universidade Bandeirante de São Paulo (2010), a Master's in Mechanical Engineering from Universidade de São Paulo (2013), and a PhD in Mechanical Engineering from Universidade de São Paulo (2017). In 2018, she completed her Postdoctoral studies at the Polytechnic School of USP, in the Department of Mechatronic and Mechanical Systems Engineering, focusing on computational modeling of micro-milling. She has expertise in Mechanical Engineering, with an emphasis on Manufacturing Processes. She also taught Automation at the Instituto Federal de São Paulo - Guarulhos Campus (2018-2019). In 2024, she completed a Postdoctoral research with an emphasis on Multiphysics Modeling applied to Tribology at the Department of Mechanical Engineering at USP's Polytechnic School.

Wilson Carlos da Silva Junior holds a Bachelor's degree in Mechanical Engineering (1983) from the University of Mogi das Cruzes, a Master's degree in Engineering from USP (2000), and a Ph.D. in Biomedical Engineering from the same university (2010). He completed postdoctoral research at IPEN (2017). He has worked as an engineer and consultant in failure analysis, with expertise in both destructive and non-destructive testing, and has experience in material strength, mechanical vibrations, and biomaterials. He has been a faculty member at UMC, UNIBAN, and

held various academic leadership positions. Currently, he is a professor at IFSP, a coordinator in the Master's program, and an advisor for scientific initiation projects, focusing on prosthetic failures, additive manufacturing, and material simulations. Additionally, he is an associate researcher in projects with institutions such as POLI-USP and Instituto Mauá de Tecnologia, funded by FAPESP.