Automated System for Bearing Fault Detection in MATLAB

Hurtado V. Angel J., Student Member, *Department of Robotics and Cybernetics, Czech Technical University in Prague*.

***Abstract*—Bearings are critical components in rotating machinery, and their faults can lead to severe operational failures if undetected. This project presents an automated system for bearing fault detection in MATLAB, capable of classifying healthy bearings and those with inner or outer race faults. Vibration data underwent preprocessing with resampling, data augmentation via sliding windows, and frequency domain analysis using envelope analysis and kurtogram-based filtering. Features such as kurtosis, dominant frequency, and crest factor were extracted and used to train machine learning models. The random forest classifier achieved over 99% accuracy, demonstrating the system's effectiveness. This work highlights a robust framework for integrating signal processing and machine learning for fault detection, offering potential for predictive maintenance applications.**

***Index Terms*—bearing fault detection, envelope analysis, kurtogram, machine learning, signal processing, vibration analysis.**

I. INTRODUCTION

Bearings are fundamental components of rotating machinery, ensuring the smooth operation of mechanical systems. Despite their critical role, they are prone to faults such as inner race defects, outer race defects, and general wear and tear. These issues, if left unaddressed, can lead to unexpected equipment failures and costly downtime. For industries reliant on such machinery, early fault detection isn't just a technical necessity—it's a practical one, critical for maintaining efficiency and avoiding operational disruptions. [1]

Among the various methods available, vibration signal analysis stands out as a practical and effective tool for diagnosing bearing faults. When bearings develop faults, they produce distinct vibration patterns that correspond to specific fault frequencies, such as the Ball Pass Frequency of the Outer Race (BPFO) and the Ball Pass Frequency of the Inner Race (BPFI). However, isolating these patterns from noisy, raw data can be a challenge, especially in real-world industrial environments where data is often voluminous and cluttered. [1]

Traditional methods for vibration signal analysis, such as Empirical Mode Decomposition (EMD) and Wavelet Transforms, rely heavily on manual feature extraction and are often sensitive to noise [2], [3]. While these techniques are effective in controlled scenarios, their adaptability is limited under varying operating conditions. Machine learning (ML) has addressed some of these limitations by automating fault classification based on extracted features, offering improved accuracy and scalability. Popular ML methods such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have shown success but require large, labeled datasets to achieve optimal performance, posing challenges when data is scarce, and computational resources are limited [3], [4]. Recent methods, like Graph Neural Networks (GNN) with Horizontal Visibility Graphs (HVG), show promise in extracting complex relationships in time series data, though they also unfortunately demand high computational resources [5].

This project addresses these challenges by developing an automated system for bearing fault detection in MATLAB, utilizing random forest classifiers. By focusing on a small but meaningful set of features, such as dominant frequency, kurtosis, and crest factor, research shows that random forests and decision trees approach optimizes the classification process through efficient feature thresholding [6]. This ensures reduced computation time and memory usage while maintaining high accuracy. The system effectively classifies bearing conditions into three categories—healthy, inner race fault, and outer race fault—by integrating advanced signal processing techniques, such as kurtogram analysis and envelope analysis, with machine learning.

In this paper, I present the steps of data collection, augmentation, preprocessing, and model training involved in creating an automated system for bearing fault detection. The framework's reliability is validated experimentally through a confusion matrix, demonstrating its effectiveness in accurately classifying bearing faults. This scalable system provides a practical foundation for real-world applications in predictive maintenance, helping to mitigate the risks of machinery failures and minimize operational disruptions.

II. METHODOLOGY

This section outlines the steps involved in designing the automated system for bearing fault detection. The process begins with data collection, where vibration signals from bearings in various conditions were gathered. To address challenges such as noise and data scarcity, a sliding window approach was employed for data augmentation. Signal preprocessing, including kurtogram analysis and envelope detection, was used to extract fault-related features. Finally, these features were used to train machine learning models to classify the bearing conditions accurately.

A. Design Requirements

The automated fault detection system was designed with the following requirements in mind:

1. The system must classify vibration signals into three categories: healthy, inner race fault, and outer race fault.
2. The system should accommodate signals with varying levels of noise and detect fault-related patterns effectively.
3. The system must process and classify vibration signals with a minimum duration of 0.5 seconds, automatically assigning them to the appropriate category.
4. The classifications must then be validated through model evaluation to ensure accuracy and reliability.

B. Data Collection

The dataset for this project consisted of vibration signals collected from bearings operating under three different conditions: healthy bearings, bearings with inner race faults, and bearings with outer race faults. Each signal represents a specific operating condition and was sampled at two distinct rates: 48,828 Hz and 97,656 Hz.

The dataset includes:

- **Baseline (Healthy Bearings):** Three signals, each 6 seconds long.
- **Inner Race Faults:** Three signals, each 3 seconds long.
- **Outer Race Faults:** Six signals, each 6 seconds long.

When faults occur in bearings, they produce characteristic frequencies conditioned by the physical properties of the bearing [1], such as the number of rolling elements, the rotational speed, and the bearing dimensions. For this project, the relevant fault frequencies were:

$$F_{BPFO}=(n/2)f_r(1-(d/D)\cos\theta) \quad (1)$$
$$F_{BPFI}=(n/2)f_r(1+(d/D)\cos\theta) \quad (2)$$

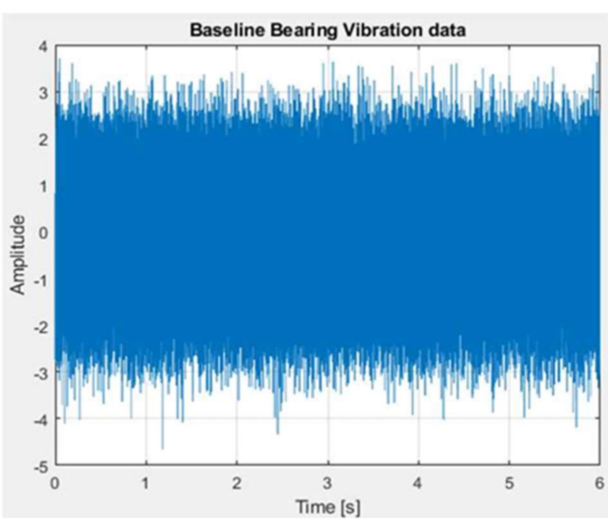

Fig. 1. Baseline signal representing healthy bearing vibrations over 6 seconds. The signal shows minimal impulsive patterns, primarily noise.

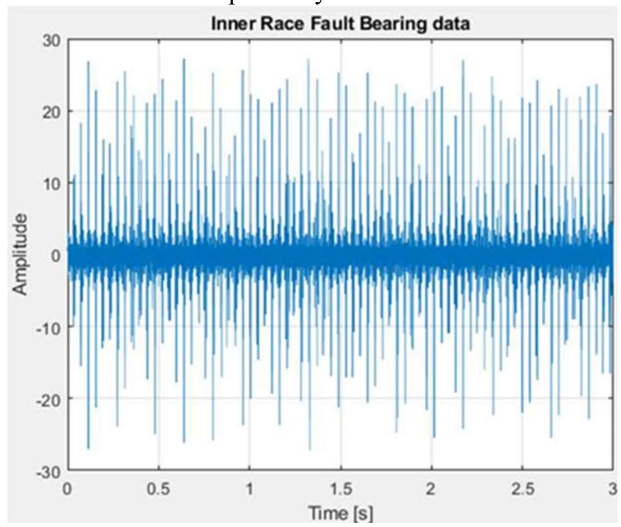

Fig. 2. Inner race fault signal captured over 3 seconds. Notice the visible peaks caused by fault-related impulsive vibrations

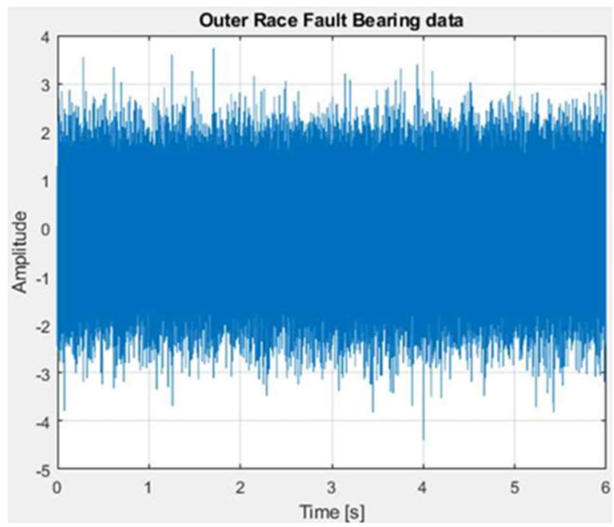

Fig. 3. Outer race fault signal recorded over 6 seconds. The signal appears noisy, masking fault-specific patterns.

Ball Pass Frequency of the outer race (1) and Ball Pass Frequency of the Inner race (2). Where n is the number of rolling elements, f_r is the shaft rotational frequency, d is the diameter of the rolling element, D the pitch diameter of the bearing and θ the contact angle. For the bearings in this study, (1) and (2) yield fault frequencies BPFO: 81.12 Hz and BPFI: 118.88 Hz.

However, as you can see from Figures 1, 2, and 3, visualizing any fault-related frequency patterns in the raw vibration data is challenging due to the inherent noise present in the signals. For instance, the baseline signal (Figure 1) primarily consists of random noise, while the inner race fault (Figure 2) and outer race fault (Figure 3) signals exhibit a mix of impulsive components and noise. These noisy signals necessitate further preprocessing steps, such as denoising and feature extraction, to uncover the fault-specific frequencies effectively.

To uncover these hidden fault-specific patterns, robust preprocessing techniques, such as filtering and envelope analysis, are essential. This ensures that the relevant fault frequencies, like BPFO and BPFI, can be effectively extracted and analyzed. To address this challenge, we applied preprocessing techniques as detailed in the next section.

C. Data Augmentation

To create a robust and diverse dataset for training the fault detection system, data augmentation techniques were employed. The raw dataset, as described in the previous section, was limited in size, with only 12 signals across three bearing conditions. This posed a challenge for training machine learning models, as a small dataset increases the risk of overfitting and reduces model generalization. To address this, two primary augmentation methods were implemented: resampling and sliding window segmentation.

The original signals were recorded at two distinct sampling rates: 48,828 Hz and 97,656 Hz. To ensure uniformity across the dataset and maintain consistency during preprocessing and feature extraction, all signals were resampled to the highest rate of 97,656 Hz. This choice was made to preserve the Nyquist frequency and ensure that high-frequency components relevant to fault detection, such as impulsive characteristics captured in kurtosis calculations, were not lost during analysis.

To further augment the dataset, a sliding window technique was applied to generate multiple overlapping samples from each signal. Each window was designed to capture 0.5 seconds of data, corresponding to 48,828 samples at the standard sampling rate. The window was then slid across the signal with a step size of 1,000 samples, resulting in significant overlap between consecutive windows. While a smaller step size could have been used to generate even more samples, it was avoided to prevent the dataset from becoming excessively large, which would increase computational demands unnecessarily.

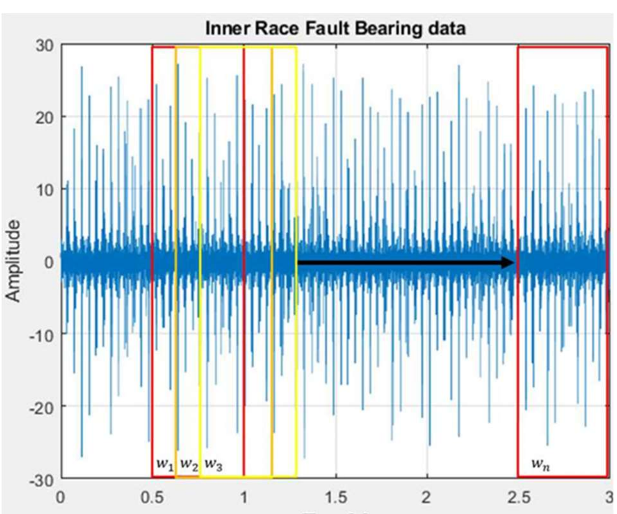

Fig. 4. Illustration of the sliding window technique applied to an inner race fault signal. Each window captures 0.5 seconds of data with a step size of 1000 samples with their overlapping segments.

Figure 4 illustrates the sliding window technique applied to divide the vibration signal into overlapping segments. This method ensured the retention of critical fault-specific patterns within each window while significantly increasing the number of samples for model training. By applying this approach, the dataset expanded from the original 12 signals to a total of 4,160 samples, offering enhanced diversity and robustness. This augmented dataset served as a solid foundation for training the machine learning model, improving its ability to generalize across different fault conditions.

D. Preprocessing

The vibration signals collected for this project, as described in the previous sections, exhibited significant levels of noise and overlapping frequency components, particularly in the outer race fault signals. Preprocessing was therefore essential to enhance the signal-to-noise ratio and isolate the fault-specific frequencies given by (1) and (2).

The preprocessing began with the application of envelope analysis, a technique well-suited for identifying fault-specific frequencies in relatively clean signals. By demodulating the signal, envelope analysis highlights amplitude modulations caused by fault vibrations. For inner race fault signals, this method worked effectively, revealing distinct peaks at harmonics of the BPFI (1). Figure 5 illustrates an envelope spectrum of an inner race fault signal, where these harmonics stand out clearly.

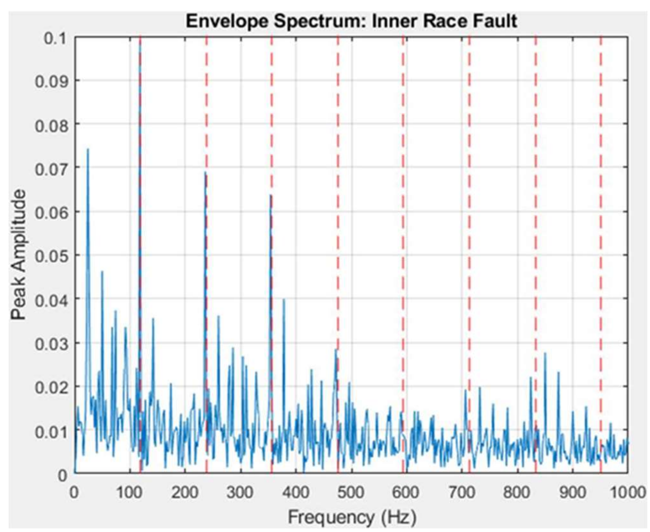

Fig. 5. Envelope spectrum of an inner race fault signal showing distinct peaks at the harmonics of BPFI (Ball Pass Frequency of the Inner Race), approximately 118 Hz, marked by red dashed lines.

In contrast to the inner race fault signals, the outer race fault signals posed a more significant challenge due to their higher noise levels, as observed in Figure 3. When envelope analysis was directly applied to these signals, it failed to reveal the clear fault frequencies characteristic of the Ball Pass Frequency of the Outer Race (BPFO). The noise within these signals masked the fault-specific frequency patterns, as shown in Figure 6. Notably, the harmonics of BPFO (approximately 81 Hz, as marked by red dashed lines) do not align with any prominent peaks in the envelope spectrum.

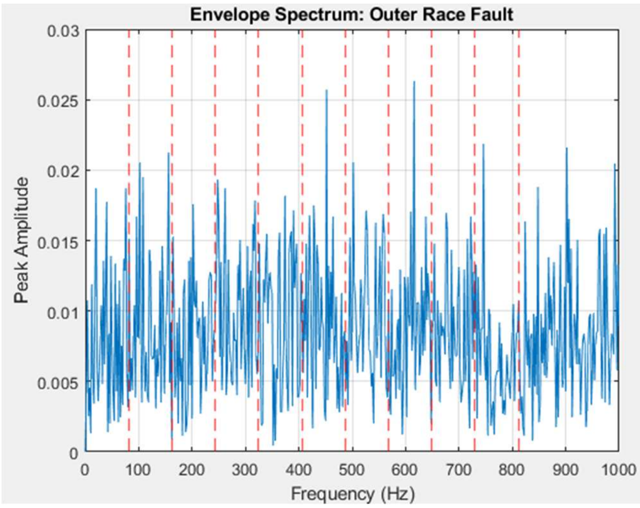

Fig. 6. Envelope spectrum of an outer race fault signal where harmonics of BPFO (Ball Pass Frequency of the Outer Race), approximately 81 Hz, marked by red dashed lines, are obscured by noise.

To address the challenge of noisy signals, kurtogram analysis was introduced. Unlike directly identifying the BPFO or BPFI frequencies, the kurtogram pinpoints the carrier frequency band with the highest kurtosis, which is indicative of impulsive energy typically associated with bearing faults [7]. This band represents the range of frequencies where fault-related features are most likely to be concentrated.

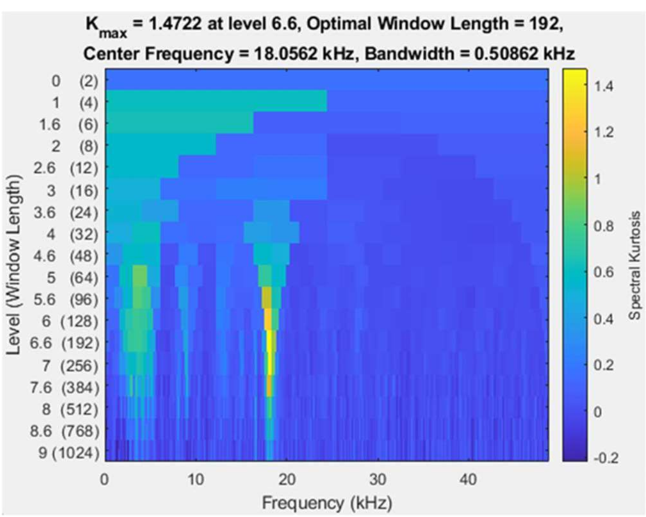

Fig. 7. Kurtogram of an outer race fault signal identifying the optimal band centered at 18.0562 kHz with a bandwidth of 0.50862 kHz.

The identified frequency band was then used to design a bandpass filter, tailored to isolate the fault-related components within the signal. During this step, care was taken to ensure that the filter design adhered to the Nyquist criterion, limiting the upper frequency cutoff to half the sampling rate to avoid aliasing. If the identified band extended beyond the Nyquist frequency, adjustments were made to confine the filter to a valid frequency range.

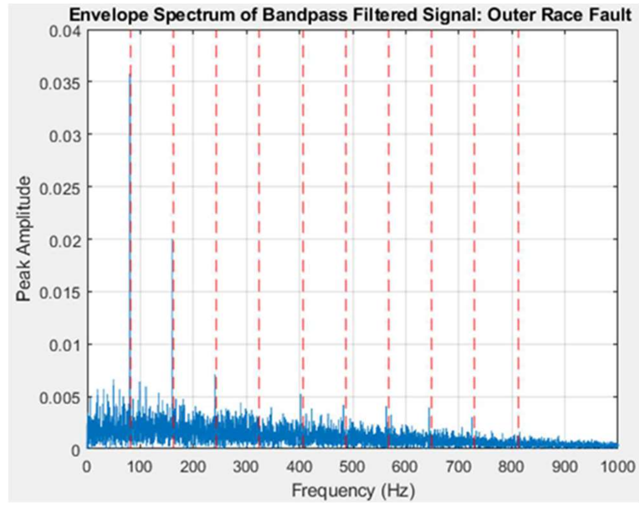

Fig. 8. Envelope spectrum of the bandpass filtered signal for an outer race fault, revealing prominent peaks at harmonics of BPFO (2), marked by red dashed lines.

Figure 7 shows a kurtogram for an outer race fault signal. The optimal band, centered at 18 kHz with a bandwidth of 0.5 kHz, was selected based on the highest kurtosis value. After filtering using this band, the fault frequencies became more prominent, as shown in Figure 8. In this filtered spectrum, peaks at harmonics of the BPFO can now be clearly observed.

Finally, the classification system must process signals without prior knowledge of their fault type, consequently, a uniform preprocessing pipeline was implemented. Kurtogram analysis and bandpass filtering were applied to all signals in the dataset, regardless of their type.

E. Feature Extraction

Once the signals were preprocessed to isolate fault-specific characteristics, the next step was to extract meaningful features that could represent the unique behavior of each signal. These features were selected based on their ability to differentiate between healthy, inner race fault, and outer race fault conditions. Three key features were chosen:

1. **Kurtosis**: Kurtosis measures the peakedness or impulsive nature of a signal.
2. **Dominant Frequency**: This feature represents the most prominent frequency in the envelope spectrum of the signal.
3. **Crest Factor**: The crest factor is the ratio of the peak amplitude to the RMS value of the signal.

The extracted features were organized into a feature matrix, where each row represented a signal, and columns corresponded to the extracted features. This matrix served as the input dataset for training the machine learning models. A sample of the feature matrix is shown in **Fig. 9**. Moreover, to understand the distribution of these features across the samples, a histogram was plotted for one of the features, normalized to show percentages (Fig. 10).the numerical representation of the extracted features.

Signal	Label	kurtosis	domfreq	crest
{48828×1 double}	1	56.9	118	15.576
{48828×1 double}	0	3.2661	50	4.1671
{48828×1 double}	2	5.7643	80	4.9468

Fig. 9. Sample feature matrix showing kurtosis, dominant frequency, and crest factor extracted from the vibration signals. Labels: 0 = Baseline, 1 = Inner Race Fault, 2 = Outer Race Fault.

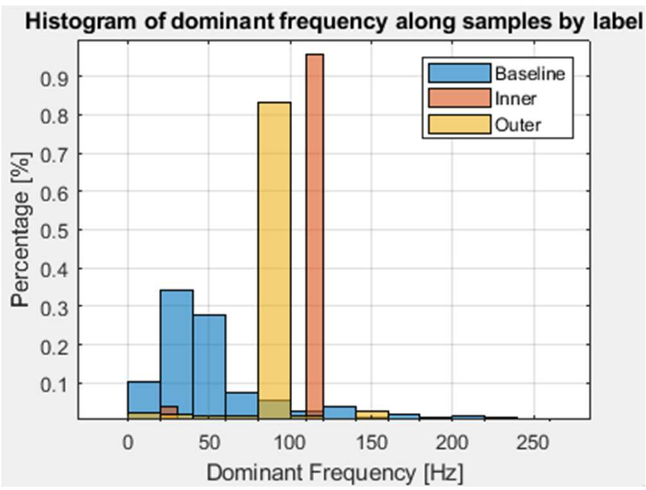

Fig. 10. Normalized histogram of dominant frequency across signal samples for each label, illustrating the separation between healthy bearings, inner race faults, and outer race faults.

F. Model Training

The final stage in developing the automated fault detection system was training machine learning models to classify the extracted features into three categories: baseline (healthy), inner race fault, and outer race fault. Two models were proposed: a decision tree and a random forests model.

The Decision Tree algorithm constructs a single tree by iteratively splitting the dataset based on the most informative feature at each step, however, its tendency to overfit, particularly with smaller datasets, limits its robustness. To overcome this, the Random Forest algorithm employs an ensemble approach, training multiple decision trees on randomly sampled subsets of the data and aggregating their predictions. [8]

The extracted feature matrix was divided into training and testing subsets using a 70:30 split ratio. Seventy percent of the data was used to train the models, while the remaining 30% was reserved for testing and validation.

III. Results

The performance of the automated fault detection system was evaluated using the confusion matrices for both the decision tree and random forest models, as shown in **Fig. 11** and **Fig. 12**. These matrices summarize the classification results by presenting the percentage of correctly and incorrectly classified samples for each fault category.

Fig. 11. Confusion matrix for the Decision Tree model, displaying the percentage of samples classified correctly and incorrectly for each class.

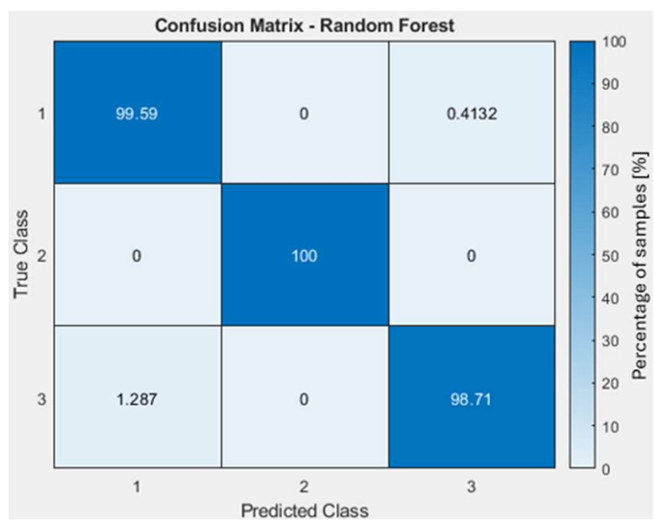

Fig. 12. Confusion matrix for the Random Forest model, showing improved classification accuracy compared to the Decision Tree, with minimal misclassifications across all classes.

The decision tree model demonstrated strong classification performance, particularly for the inner race fault class, achieving 100% accuracy. For baseline and outer race fault samples, the model achieved accuracies of 98.55% and 98.9%, respectively. However, a small percentage of baseline signals

(1.446%) were misclassified as outer race faults, and a few outer race fault signals (0.1838%) were misclassified as inner race faults.

The random forest model outperformed the decision tree by further reducing misclassifications across all classes. Baseline signals were classified with an accuracy of 99.59%, and outer race fault signals achieved 98.71% accuracy. Inner race fault signals retained perfect classification accuracy at 100%.

IV. Conclusion

This project successfully developed an automated system for bearing fault detection using MATLAB. By integrating signal processing techniques such as kurtogram analysis and envelope analysis with machine learning models, the system was able to accurately classify vibration signals into healthy, inner race fault, and outer race fault conditions. The use of resampling and sliding window techniques ensured a diverse and robust dataset, while the preprocessing steps allowed for the extraction of meaningful features from noisy signals. Key findings demonstrate that the Random Forest model outperformed the Decision Tree in terms of classification accuracy and robustness, achieving over 99% accuracy across all classes. This highlights the effectiveness of ensemble methods in handling noisy and variable data. The decision tree, while simpler, still provided valuable insights into the classification process and served as a baseline for comparison.

While the system performs well, several limitations and opportunities for improvement were identified. The dataset used in this study, although augmented, remains relatively small, which may limit the system's generalizability to unseen scenarios or other fault types. Expanding the dataset to include a broader range of operating conditions and fault scenarios would enhance the system’s robustness. Additionally, the current approach assumes that all signals are stationary, which may not hold true for real-world applications where varying loads and speeds could introduce non-stationary behavior. Incorporating time-frequency analysis techniques, such as wavelet transforms, could address this limitation.

References

[1] Kurfess, T.R., Billington, S., and Liang, S. Y., “Advanced diagnostics and prognostic techniques for rolling element bearings”. In *Condition Monitoring and control for intelligent manufacturing.* Springer, London, 2006. pp. 137-165.

[2] Kumar, R., Singh, J., Sharma, S., et al. (2022). "Identification of localized defects and fault size estimation of taper roller bearing (NBC_30205) with signal processing using the Shannon entropy method in MATLAB for automobile industries applications." *Heliyon*, 8(e12053).

[3] Attoui, I., Oudjani, B., Boutasseta, N., Bouakkaz, M.-S., Fergani, N., & Bouraiou, A. (2022). "Bearing Fault Detection and Classification Based on Vibration Signal Analysis and ANFIS Classifier." *IEEE International Multi-Conference on Systems, Signals & Devices (SSD'22)*, pp. 846-853.

[4] Yang, Y., & Tang, W. (2011). "Study of Remote Bearing Fault Diagnosis Based on BP Neural Network Combination." *2011 Seventh International Conference on Natural Computation*, 618-622.

[5] Mo, L., & Yan, R. (2020). "Rolling Bearing Fault Diagnosis Based on Horizontal Visibility Graph and Graph Neural Networks." *IEEE Transactions on Instrumentation and Measurement*, 69(7), 275–284.

[6] F. B. Abid, M. Sallem and A. Braham, "Optimized SWPT and Decision Tree for Incipient Bearing Fault Diagnosis," *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 2019, pp. 231-236*, doi: 10.1109/STA.2019.8717197.

[7] MATLAB, "Rolling Element Bearing Fault Diagnosis," MathWorks Documentation. [Online].
Available: https://www.mathworks.com/help/predmaint/ug/Rolling-Element-Bearing-Fault-Diagnosis.html. [Accessed: Jan. 12, 2025].

[8] G. Louppe, "Understanding Random Forests: From Theory to Practice," Ph.D. dissertation, Univ. of Liège, Liège, Belgium, 2014.