MARMARA UNIVERSITY
FACULTY OF ENGINEERING

BEARING FAULT DIAGNOSIS BY EXPLAINABLE DEEP LEARNING

Zeliha Hanım KARAPINAR

GRADUATION PROJECT REPORT
Department of Mechanical Engineering

Supervisor
Assoc. Prof. IBRAHIM SINA KUSEYRI

ISTANBUL, 2025

MARMARA UNIVERSITY
FACULTY OF ENGINEERING

Bearing Fault Diagnosis by Explainable Deep Learning

by

Zeliha Hanım Karapınar

June 30, 2025, Istanbul

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

BACHELOR OF SCIENCE

AT

MARMARA UNIVERSITY

Signature of Author(s) ....

Department of Mechanical Engineering

Certified By ....

Project Supervisor, Department of Mechanical Engineering

Accepted By ....

Head of the Department of Mechanical Engineering

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartfelt gratitude to my esteemed supervisor, Assoc. Prof. Ibrahim Sina Kuseyri. His unwavering support, insightful guidance, and inspiring encouragement have been instrumental throughout the entire process of this thesis.

Not only did he share his deep knowledge and experience with great generosity, but he also offered moral strength during times of uncertainty. His patience and trust motivated me to keep pushing forward and to believe in my work even when I had doubts.

It has been a privilege to learn under his mentorship, and I will always carry the lessons I learned from him throughout my academic and professional life.

Thank you, sincerely, for everything.

June,2025 Zeliha Hanım Karapınar

ABSTRACT

Bearing Diagnosis with Explainable Deep Learning

Early detection of bearing faults in industrial settings is vital to ensure reliability, cost-efficiency, and operational continuity in machine maintenance processes. Bearings are fundamental components of rotating machinery, facilitating smooth motion by minimizing frictional resistance. However, due to their continuous operation and mechanical load, they are highly susceptible to wear and unexpected failures. Each year, millions of bearings either fail or are replaced as part of preventive maintenance strategies, resulting in substantial costs and disruptions in production workflows.

Conventional diagnostic methods, which typically rely on signal processing and expert-driven analysis, often struggle to generalize effectively under varying operating conditions. Although deep learning-based approaches offer superior accuracy compared to traditional machine learning techniques, their lack of interpretability poses a significant barrier to trust and adoption in industrial environments.

In response to these constraints, the goal of this project is to create a unique system for bearing fault detection that combines high diagnostic accuracy with a transparent and interpretable decision-making framework designed specifically for engineers. The ultimate goal is to improve industrial problem diagnosis systems in terms of both technical accuracy and usability.

Özet

Açıklanabilir Derin Öğrenme ile Rulman Teşhisi

Sanayide rulman arızalarının erken teşhisi, makine bakım süreçlerinde güvenilirlik, maliyet etkinliği ve operasyonel sürekliliğin sağlanması açısından hayati bir öneme sahiptir. Rulmanlar, döner makinelerin temel bileşenleri olup, sürtünmeyi azaltarak hareketin düzgün bir şekilde gerçekleşmesini sağlar. Ancak sürekli çalışmaları ve maruz kaldıkları mekanik yükler nedeniyle, aşınmaya ve beklenmedik arızalara karşı oldukça hassastırlar. Her yıl milyonlarca rulman ya arıza nedeniyle ya da önleyici bakım kapsamında değiştirilmekte, bu da ciddi maliyetlere ve üretim süreçlerinde aksamalara yol açmaktadır.

Geleneksel teşhis yöntemleri genellikle sinyal işleme ve uzman bilgisine dayalı analizlerle sınırlı kalmakta; değişken çalışma koşulları altında genelleme yapma konusunda yetersiz kalmaktadır. Derin öğrenme tabanlı yaklaşımlar ise geleneksel makine öğrenimi tekniklerine kıyasla daha yüksek doğruluk sunsa da, karar alma mekanizmalarının yorumlanabilir olmaması, endüstriyel uygulamalarda güvenilirlik açısından önemli bir engel teşkil etmektedir.

Bu sorunlara yanıt olarak, bu projede rulman arızalarının tespiti için hem yüksek doğruluk sağlayan hem de mühendisler açısından anlaşılır ve yorumlanabilir bir karar mekanizması sunan yenilikçi bir çözüm geliştirilmesi hedeflenmektedir. Nihai amaç, sanayide kullanılan arıza teşhis sistemlerini teknik doğruluk ve pratik kullanılabilirlik açısından ileri bir noktaya taşımaktır.

LIST OF ABBREVIATIONS

2D:	Two - Dimensional
AI:	Artificial Intelligence
NN:	Neural Network
ANN:	Artificial Neural Network
AS:	Accuracy Score
BN:	Batch Normalization
CL:	Convolution Layer
CM:	Confusion Matrix
1D-CNN:	One-Dimensional Convolutional Neural Network
CNN:	Convolutional Neural Network
CV:	Cross Validation
CWRU:	Case Western Reverse University
DL:	Deep Learning
ML:	Machine Learning
FT:	Fourier Transform
FFT:	Fast Fourier Transform
DFT:	Discrete Fourier Transform
STFT:	Short-Term Fourier Transform
FN:	False Negative
FP:	False Positive
TN:	True Negative
TP:	True Positive
IFT:	Inverse Fourier Transform
FTF:	Fundamental Train Frequency
CWT:	Continuous Wavelet Transform
BA:	Base Plate Acceleration
BSF:	Ball Spain Frequency
BPFI:	Ball Pass Frequency, Inner Race
BPFO:	Ball Pass Frequency, Outer Race
D:	Bearing Pitch Diameter
d:	Rolling Element Diameter
DE:	Drive end Acceleration
f_r:	Shaft Speed
IR:	Inner Race
OR:	Outer Race
SHAP:	Shapley Additive Explanations
LIME:	Local Interpretable Model-Agnostic Explanations
XAI:	Explainable Artificial Intelligence

LIST OF SYMBOLS

L_{10}:	Basic rating life [millions of revolutions]
L_{10h}:	Basic rating life [operational hours]
C:	Kaydon dynamics load rating
P:	Equivalent dynamic bearing load
n:	Rotational speed
p:	Exponent of the life equation
N_B:	Number of ball
θ:	Bearing contact angle
d:	d is the ball diameter
D:	D is the pitch diameter
f_r:	Shaft speed
n:	Number of rolling elements
x_i:	Value of the vibration signal at time index *i*
N:	Total number of samples
$\hat{x}$:	Mean of the signal
σ:	Standard Deviation
w_i:	Weights
b:	Bias term
y_i:	True value
$\hat{y}_i$:	Predictive value
α:	Learning rate
$\theta_i(f)$:	Shapley values
S:	Subset of the features not including *i*
$f(S)$:	Model's prediction when only features in *S* are known
θ_0:	Denotes the baseline prediction
θ_i:	Represent the attribution of the i^{th} feature
$\xi(x)$:	Loss function
$\Omega(g)$:	Regularization term that penalizes the compexity of *g*
Z:	The set of perturbed instances around x
$f(z)$:	is the output of the black-box model for input z
$g(z)$:	Prediction from the interpretable model
$\pi_x(z)$:	The similarity kernel, which assigns weights to *z* based on its distance from x
$D(x, z)$:	Distance metric
σ:	Kernel width
T:	Frame size
F_s:	Sampling rate
$\delta(f)$:	Frequency resolution
$x(n)$:	Input signal at time n
$w(n)$:	Length M window function
$w_m(w)$:	DTFT of windowed data centered about time R
$\hat{x}(t)$:	Hilbert transform of $x(t)$

Contents

1 ACKNOWLEDGEMENT i

3 ABSTRACT ii

3 ÖZET iii

3 Iist of Abbreviations iv

3 List of Symbols v

LIST OF FIGURES x

LIST OF TABLES xiii

1 INTRODUCTION 1

1.1 Literature Review 1

1.2 Originality and Innovative Aspects of the Project 2

2 DATA PROCESSING METHODOLOGY 4

2.1 Literature Review 6

2.1.1 Importance of Bearing Fault Detection in Industry 8

2.2 Bearings 8

2.2.1 Structure of Bearings 9

2.2.2 Types of Bearings 10

2.2.3 Bearing Life 10

2.2.4 Bearing Characteristic Fault Frequencies 12

2.3 Bearing Fault Stages 17

2.3.1 Bearing Fault Stage I: Initiation and Early Detection 18

2.3.2 Bearing Fault Stage II: Progressive Damage and Resonant Excitation 19

2.3.3 Bearing Fault Stage III: Established Patterns in Low-Frequency Spectrum 19

2.3.4 Bearing Fault Stage IV: Terminal Degradation and Impending Failure 20

2.3.5 Feature in Time-Domain 21

2.3.5.1 Root Mean Square (RMS) 21

2.3.5.2 Standard Deviation(SD) 22

2.3.5.3 Kurtosis 22

2.3.5.4 Peak-to-Peak Value 22

2.3.5.5 Skewness 22

2.3.5.6 Crest Factor(CF) 23
2.3.5.7 Impulse Factor(IF) 23
2.3.5.8 Clearance Factor(CLF) 23
2.3.6 Sensor Technology in Bearings 24
2.4 CWRU Dataset Preprocessing 26
2.4.1 Classification by Fault Type 26
2.4.2 Segmentation of Time-Series Data 26
2.4.3 Feature and Label Extraction 27
2.4.4 Encoding of Target Labels 27
2.4.5 Training and Testing Partition 28
2.4.6 Normalization of Input Features 28
2.4.7 Reshaping for Model Compatibility 28
3 MODEL ARCHITECTURES 29
3.1 How Artificial Neural Networks (ANN) Work? 29
3.1.1 Fundamental Units and Structure 29
3.1.2 Activation Functions 29
3.1.3 Network Layers 30
3.1.4 Forward Propagation 30
3.1.5 Loss Functions 30
3.1.6 Backpropagation and Weight Update 31
3.1.7 Training Loop 32
3.2 CNN (Convolutional Neural Network) 32
3.2.1 Architecture of a Traditional CNN 32
3.2.2 Types of Layer 33
3.2.3 Filter Hyperparameters 34
3.2.4 Tuning Hyperparameters 36
3.2.5 Commonly Used Activation Functions 36
3.3 One-Dimensional Convolutional Neural Network (1D-CNN) 37
3.4 Model Architecture and Training Framework 38
3.4.1 Architectural Overview 39
3.4.2 Training Procedure 40
3.4.3 Overfitting Prevention and Generalization 40
3.4.4 Evaluation Metrics 40
4 EXPLAINABLE ARTIFICIAL INTELLIGENCE 41
4.1 Importance of Explainability in Industrial Fault Diagnosis 41
4.2 SHAP (Shapley Additive Explanations) 41
4.2.1 Underlying Assumptions of SHAP 42
4.2.2 Additive Feature Attribution Model 42
4.2.3 Axioms Guaranteed by SHAP 42

4.3 Interpretable Local Explanations with LIME 43
4.3.1 Loss Function 43
4.3.2 Step-by-Step Explanation Procedure 44
4.4 Explainability in This Project 44
4.4.1 Kernel SHAP: A Model-Agnostic Approximation 45
4.4.1.1 The Shapley Kernel Weighting Function 45
4.4.2 Differences Between LIME and Shapley Values 45
4.4.3 Explainability Analysis Using Shapley Values on Frequency-Domain Features 46
5 MECHANICAL SYSTEM and DIGITAL SIGNAL PROCESSING 47
5.1 Mechanical System and Vibration Signal 47
5.2 Time Domain Terms 48
5.3 Frequency Domain Terms 49
5.4 Aliasing 50
5.5 Fourier Series 51
5.5.1 Discrete Fourier Transform 53
5.5.2 Short-Term Fourier Transform 55
5.5.3 Nyquist Frequency 58
5.6 Envelope Analysis 58
5.6.1 Role of Modulation and Demodulation in Envelope Analysis 60
5.6.2 Is Envelope Spectrum Generation Invertible for Real Discrete-Time Signals? 62
5.6.3 Hilbert Transform in Envelope Analysis 62
5.7 The Continuous Wavelet Transform 63
5.7.1 Scaling 64
5.7.2 Shifting 64
5.8 Windows and Spectral Leakage 65
5.8.1 Leakage 65
5.8.2 Windows 67
5.8.3 Window Types 69
6 PROPOSED METHODOLOGY and EXPERIMENTAL DESIGN 70
6.1 Prediction Counts and Final Diagnosis 71
6.1.1 Evaluation of Raw-Signal CNN with Characteristic Fault Frequencies 71
6.1.2 Diagnosis Results and Interpretability Insights 72
6.2 Finding the Most Important Window for Explainability 73
6.2.1 Harmonic RMS Energy Computation 73
6.2.1.1 Methodology: 74
6.2.2 Identification of Important Signal Windows 74

6.2.2.1 Methodology: 75
6.3 Special DFT Layer Added for Explainability 75
6.3.1 DFT Layer Design and Model Integration 76
6.3.2 Spectral Preprocessing and SHAP Values 76
7 RESULTS and DISCUSSION 78
8 CONCLUSION 82

List of Figures

1 CWRU bearing test rig 5

2 Bearing components and the experimental configuration of the ball bearing system on the CWRU bearing test rig[10] 5

3 A ball bearing 9

4 Types of bearings 10

5 Bearing life 11

6 Bearing Specifications 13

7 Record 135DE (48k, 0.007 in. drive end outer race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal 13

8 Record 109DE (48k, 0.007 in. drive end inner race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal 14

9 Record 189DE (48k, 0.014 in. drive end outer race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal 14

10 Record 97DE (48k, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal 15

11 Fault frequencies 16

12 Fault types 17

13 Bearing fault - stages I 18

14 Bearing fault - stages II 19

15 Bearing fault - stages III 20

16 Bearing fault - stages IV 21

17 Each sample from 4096 consecutive time points 27

18 Activation Functions 30

19 Understanding Forward and Backward Propagation in Neural Networks 32

20 Example of a network with many convolutional layers. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer. 32

21 Convolution Layer (CONV) 33

22 Pooling 34

23 Fully Connected(FC) 34

24 Dimensions of a filter 35

25 Stride 35

26 Zero-Padding 35

27 Understanding the complexity of the model 36
28 Rectified Linear Unit Functions 37
29 Illustration of a One-Dimensional Convolutional Neural Network (1D-CNN) Architecture for Sequential Data Processing 38
30 Model Architecture 39
31 Differences Between LIME and Shapley Values 45
32 Effect of Sampling Frequency on 10Hz Sine Wave 48
33 Time domain terms used in performing a digital Fourier transform 49
34 Frequency domain terms used in performing a digital Fourier transform 50
35 The red sine wave is the original signal. The blue dots represent how often the signal is being sampled. MIDDLE: The blue line is how the signal will appear due to the low sampling rate. BOTTOM: What the user will see in the time domain. Notice the acquired frequency is much lower than the actual frequency.[11] 51
36 The top graph shows a non-periodic function $s(x)$ in blue defined only over the red interval from 0 to P. The function can be analyzed over this interval to produce the Fourier series in the bottom graph. The Fourier series is always a periodic function, even if original function $s(x)$is not. 52
37 Complex waveform vs simpler sine and cosine waves 52
38 Conversion to frequency domain 53
39 A random signal looks like in the original time-domain, after multiplying with the overlapping windows 58
40 Time waveform vs frequency spectrum 59
41 Filtered signal vs frequency of interest 59
42 Demodulated signal vs envelope spectrum 59
43 Spectral Examples from Real Measurements 60
44 Waveform recorded on a defective bearing 61
45 Waveform recorded on a defective bearing 61
46 Continuous wavelet transfors of a signal 63
47 Low and high scales in wavelets 64
48 Scaling in wavelet transform 64
49 Shifting in wavelet transform 64
50 Periodic signal – Repeating and appending captured signal recreates original signal[11] 65
51 Example of a non-periodic measurement time[11] 66
52 Sharp transient in time domain (left) has broad frequency response (right)[11] . 66
53 Sharp transients in repeated signal create leakage (green curve)[11] 67
54 A signal (top) is multiplied by a window (middle) resulting in windowed signal (bottom)[11] 67

55 After applying a window to the captured signal, the sharp transients are eliminated Because the sharp transients are reduced and smoothed, the broadband frequency of the spectral leakage is also reduced.[11] 68

56 Periodic sine wave without leakage (red), non-periodic sine wave with leakage (green), and windowed non-periodic sine wave with reduced leakage (blue) . . 68

57 Flowchart of the proposed diagnostic process. 70

58 Normalized confusion matrix of the 1DCNN model trained on frequency domain. 78

59 Normalized confusion matrix of the 1DCNN model trained on raw data. 79

60 Dft resolution and shapley values of a sample 80

61 Dft resolution and shapley values of a sample 81

List of Tables

1 48k normal baseline data f_s =48kHz 5

2 48 k drive end bearing fault data; f_s = 48 kHz; variables recorded: DE and FE. 6

3 Data sets not diagnosable - Filtered for Drive end bearing faults 48 kHz data. . 7

4 Bearing details and fault frequencies. 16

5 Bearing specifications [inches] 16

6 Accuracy performance of the 1DCNN model trained with Raw and FFT-transformed data 78

1 INTRODUCTION

This project aims to develop a high-accuracy one-dimensional Convolutional Neural Network (1D CNN) model that operates directly on raw vibration data to enable early detection of bearing faults. In addition to achieving strong classification performance, the project seeks to incorporate explainable artificial intelligence (XAI) techniques to enhance the interpretability of the model's decision-making process from an engineering perspective. During the testing phase, the model will be trained on raw time-domain signals. To improve transparency, a Fast Fourier Transform (FFT) of the input signal will be integrated into the first layer of the network, allowing engineers to interpret the model's behavior in a frequency domain context that aligns with conventional diagnostic practices.

Furthermore, an alternative model will be trained using FFT-transformed data as input, enabling a comprehensive comparison between models trained on raw and frequency-domain features. This dual-approach will allow for an in-depth analysis of performance differences and potential trade-offs between the two data representations. The scope of the project also includes evaluating the model's robustness and generalization capabilities under varying operating conditions. The ultimate goal is to design a deep learning-based diagnostic system that is not only accurate but also explainable and practically meaningful for industrial applications.

1.1 Literature Review

In recent years, artificial intelligence (AI), and particularly deep learning-based models, have found increasing application across various industrial problems. Among the most prominent of these is the early detection of bearing faults in rotating machinery. Bearings are fundamental mechanical components used in rotary systems to enable smooth motion with minimal friction. Approximately 10 billion bearings are produced each year, with nearly 1 billion being replaced due to failure or preventive maintenance. Consequently, the early diagnosis of bearing faults is of significant economic and operational importance.

Historically, signal modeling and signal processing-based methods have been developed for this purpose. However, their performance has remained limited—even under controlled experimental conditions. In contrast, data-driven approaches—especially those employing deep learning models—have demonstrated promising results in fault detection tasks.

Despite these successes, the widespread industrial adoption of such models is hindered by their lack of transparency in decision-making. Deep neural networks are often regarded as "black-box" systems, where the reasoning behind a given prediction cannot be clearly understood. This opaqueness raises concerns regarding the model's reliability and introduces risks in terms of safety, regulation, and accountability. At this point, the concept of Explainable Artificial Intelligence (XAI) comes into play. XAI encompasses a collection of methods designed to make the decision-making processes of AI systems interpretable and understandable by humans.

In this context, the initial focus of this study was to interpret the decision mechanisms of deep learning models applied to bearing fault detection. By employing envelope analysis and the Hilbert transform, the frequency-focused behavior of the model was translated into an engineering-relevant domain. Metrics such as IAS (Important Area Score) and ISNR (Improved Signal-to-Noise Ratio) were used to assess how well the model's decisions align with expert domain knowledge. Rather than working directly with raw time-series inputs, the model's decision process was evaluated in the envelope spectrum domain, which is more intuitive and interpretable for engineers. This methodology enhances model transparency and contributes to the reliability of AI-driven systems in real-world industrial applications. [1]

Literature reviews have also revealed that 1D Convolutional Neural Networks (1D CNNs) are highly effective in engineering tasks due to their low computational cost and flexible architecture. These models have shown strong performance in various domains such as automatic speech recognition, electrocardiogram (ECG) analysis, and structural health monitoring. Their efficiency and suitability for real-time applications make them an ideal solution in practice. The prevalence of 1D CNNs in the literature highlights their practicality and relatively low data requirements. [2]

Furthermore, the literature demonstrates a wide range of methods for bearing fault detection, each with its strengths. While 2D CNNs and DTCNNs (Deep Temporal Convolutional Neural Networks) offer high classification accuracy, 1D CNN-based approaches present a more practical alternative with significantly lower computational demands. Additionally, the integration of explainable AI (XAI) techniques allows for the internal logic of these models to be interpreted within a mechanical engineering context, thereby increasing trust and understanding. [3,4,5]

Together, these studies provide critical insights into enhancing both the interpretability and performance of deep learning models for industrial applications. They serve as a valuable guide for researchers and practitioners aiming to deploy AI systems that are not only accurate but also transparent and trustworthy.

1.2 Originality and Innovative Aspects of the Project

This project proposes an innovative approach aimed at improving both the explainability and performance of 1D Convolutional Neural Networks (1D CNNs) in the task of bearing fault detection and analysis. Unlike most existing studies in the literature, this project not only trains a model directly on raw time-series data but also develops and evaluates a parallel model trained on data transformed using the Fast Fourier Transform (FFT). A comparative analysis will be conducted between the two models to explore performance and interpretability differences.

From an explainability perspective, the model trained on raw data will be re-evaluated by feeding it FFT-transformed inputs into its initial layers. The key idea is that, since FFT is an invertible function, it allows for the model's decision mechanisms to be translated back into a domain that is more intuitive and interpretable for engineers. In contrast, previous works-such as those that

employed envelope analysis-have faced criticism due to the non-invertibility of the envelope spectrum, which raises questions about the reliability and traceability of their results. [1]
One of the core innovations of this project lies in its direct comparison between FFT-based and raw data-based approaches. It is hypothesized that the model trained on raw data will demonstrate superior performance, as FFT inherently loses phase information, which may negatively affect classification accuracy. However, this limitation will be carefully studied, and the trade-offs between explainability and performance will be analyzed in detail.
Ultimately, the goal is to develop a framework that delivers both higher model performance and greater interpretability, tailored to the needs of practicing engineers. This dual-focused strategy makes the proposed project highly relevant for industrial applications and distinguishes it from prior work in the field.

2 DATA PROCESSING METHODOLOGY

Rolling Element Bearings, one of the most commonly used components in rotating machinery, are critical elements whose failure can lead to machine shutdowns and costly downtime. This reality has driven the development of vibration-based diagnostic methods for bearing fault detection over the past few decades, leading to the emergence of many powerful diagnostic techniques. [6]

A widely accepted benchmark in the field of bearing fault diagnosis is the Case Western Reserve University (CWRU) Bearing Data Center dataset, which is a publicly available and extensively used dataset. [7] The CWRU dataset exhibits several unique characteristics: while some signals clearly exhibit classical bearing fault features, others are more ambiguous or present with varying fault symptoms. Therefore, a comprehensive benchmarking study is essential to accurately assess the effectiveness of new diagnostic algorithms.

The CWRU dataset is categorized into four main groups based on sampling frequency and fault location:

- 48k Baseline (normal operation),
- 12k Drive End Fault,
- 48k Drive End Fault, and
- 12k Fan End Fault

Within each category, the dataset includes various types of fault data such as ball defects, inner race faults, and outer race faults. The outer race faults are further subdivided into three subcategories based on the fault position relative to the load zone:

- Centered (6 o'clock position),
- Orthogonal (3 o'clock position), and
- Opposite direction (12 o'clock position)

Additionally, the dataset is organized according to fault sizes (ranging from 0.007 to 0.028 inches) and motor load levels (0 to 3 horsepower). These variations make the dataset rich and challenging, providing a robust ground for testing diagnostic models under diverse and realistic conditions.

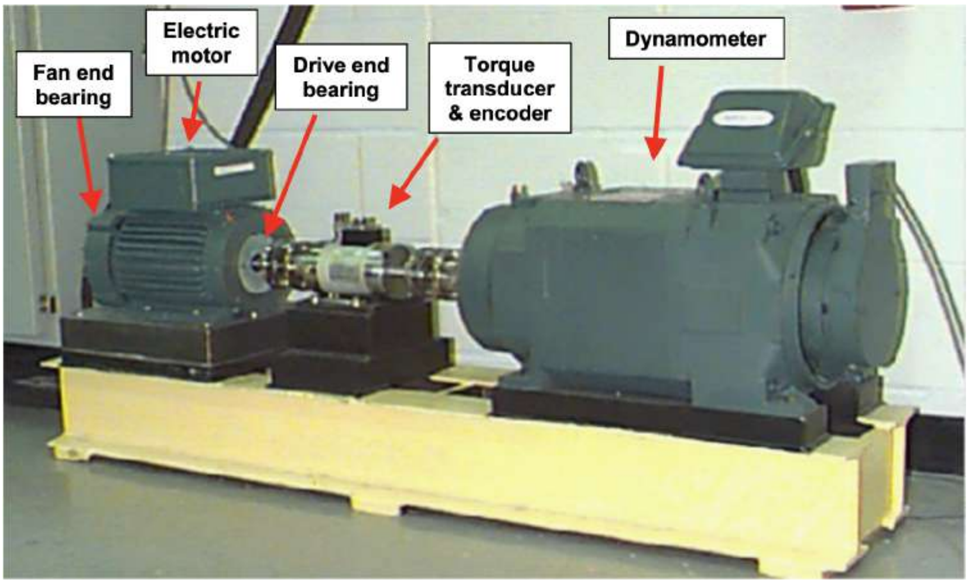


Figure 1: CWRU bearing test rig

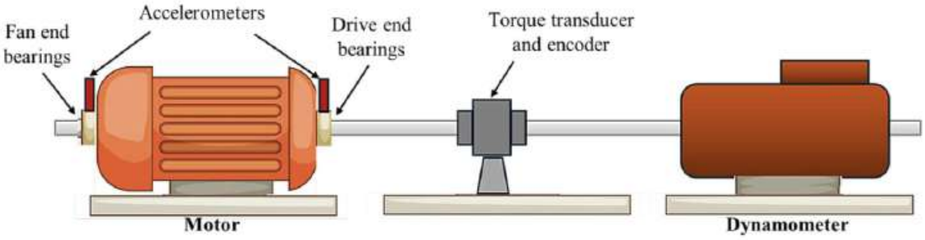


Figure 2: Bearing components and the experimental configuration of the ball bearing system on the CWRU bearing test rig[10]

In this study, the 48k Drive End Bearing Fault Data from the CWRU dataset was utilized. One of the primary reasons for this choice is the high sampling frequency of 48 kHz, which enables more detailed signal analysis. A high sampling rate is particularly beneficial in vibration-based fault diagnosis, as it preserves signal details that might otherwise be lost at lower frequencies. Moreover, drive end bearing faults are typically among the most critical components, directly affecting motor performance and overall system reliability. Therefore, models developed using this dataset are expected to have higher applicability in industrial settings.

Table 1: 48k normal baseline data f_s =48kHz

Motor Load hp	Shaft Speed rpm	Data Set Number
0	1797	97
1	1772	98
2	1750	99
3	1730	100

Table 2: 48 k drive end bearing fault data; f_s = 48 kHz; variables recorded: DE and FE.

Fault width (in (mm))	Motor load (hp)	Shaft speed (rpm)	IR	Ball	OR centred	OR orthogonal	OR opposite
0.007 (0.18)	0	1797	109	122	135	148	161
	1	1772	110	123	136	149	162
	2	1750	111	124	137	150	163
	3	1730	112	125	138	151	164
0.014 (0.36)	0	1797	174	189	201	–	–
	1	1772	175	190	202	–	–
	2	1750	176	191	203	–	–
	3	1730	177	192	204	–	–
0.021 (0.53)	0	1797	213	226	238	250	262
	1	1772	214	227	239	251	263
	2	1750	215	228	240	252	264
	3	1730	217	229	241	253	265

DE = drive end acceleration; FE = fan end acceleration.

2.1 Literature Review

Although the CWRU dataset is a widely used benchmark for bearing fault detection, it contains some critical misconceptions that are often overlooked in the literature. The most significant of these is the mistaken assumption that motor loads (torque loads) are directly correlated with the radial loads applied to the bearings. While the dataset includes recordings under different motor loads, these variations are not supported by any mechanism (such as gear systems) that would alter the radial load on the bearings. As a result, the actual influence of load variation on bearing faults remains ambiguous. The primary effect of changing the motor load is merely a slight reduction in shaft speed—approximately a 4% decrease—which is not a decisive factor in the detectability of bearing faults [8].

More importantly, the only theoretical radial load acting on the bearings in the dataset is due to gravity from the 6 o'clock position. However, some parts of the CWRU database and several studies in the literature have incorrectly stated this load to be at the 3 o'clock position. Such erroneous assumptions have led to fault classifications that are not based on solid physical principles [8]. Indeed, it has been observed that many machine learning algorithms employed in the literature do not directly capture the actual physical characteristics of bearing faults, but rather detect deviations from the reference dataset used. This raises concerns about the generalizability of the models and their applicability in real-world conditions. A common but often overlooked issue in the literature is the classification of bearing faults based solely on fault size. In the CWRU dataset, faults are typically categorized according to predefined size levels such as 0.007, 0.014, and 0.021 inches. While this approach may seem straightforward, it carries several drawbacks and can lead to misleading conclusions [9].

Firstly, the physical size of a fault alone is not a sufficient criterion for accurate diagnosis. The severity and detectability of a bearing fault are influenced not only by its size but also by its location, the operating load, system dynamics, and vibration characteristics. For instance, a fault of the same size may produce more distinct signals if it occurs on the inner race, while faults on the outer race or rolling elements may manifest with different spectral components. Relying solely on fault size disregards these dynamic and physical influences.

Secondly, fault-size-based classification methods generally have low generalization capability. Bearing faults often develop progressively over time, and forcing them into sharp size categories (e.g., 0.007 vs. 0.014 inches) imposes artificial boundaries on what is essentially a continuous phenomenon. This can make the model overly sensitive to minor variations and reduce its performance on real-world data.

Lastly, several studies have highlighted the limitations of size-based classification and have instead emphasized alternative strategies that focus on the fault location, spectral features, or general vibration patterns [9]. In line with these findings, this study adopts a classification approach that goes beyond fault size, taking into account both the fault location and its frequency characteristics to develop a more reliable and physically meaningful diagnostic model. In the literature, it has been observed that certain methods applied to some datasets failed to accurately diagnose faults [8]. Additionally, in most datasets, specific fault types were found not to exhibit classical features. A significant finding of the analysis is that the assembly of the test device influenced diagnosis results more than the faults themselves, with mechanical looseness observed in many datasets. Furthermore, many datasets showed that faults manifested only in small segments of the signals, thereby exhibiting non-stationary characteristics. Due to detected errors within the datasets, some containing faulty data were excluded from the training and testing phases.

Table 3: Data sets not diagnosable - Filtered for Drive end bearing faults 48 kHz data.

	Fault type				
	IR	**Ball**	**OR centred**	**OR orthogonal**	**OR opposite**
Drive end bearing faults 48 kHz data	174	122, 123, 124, 125, 192, 228DE, 229DE	202FE, 204FE	–	–

2.1.1 Importance of Bearing Fault Detection in Industry

Early detection of bearing failures in industrial systems is critically important for ensuring reliability, cost-effectiveness, and operational continuity in machine maintenance processes. Bearings are fundamental components of rotating machinery, enabling smooth motion by reducing frictional resistance. However, due to their intensive usage, they are highly susceptible to wear and unexpected failures.

Each year, millions of bearings either fail or are replaced as part of preventive maintenance programs, leading to high costs and disruptions in production processes.

In this context, the early detection and prevention of bearing faults hold great significance for industrial operations. Artificial intelligence (AI) and machine learning (ML) techniques offer revolutionary solutions in this area. In particular, predictive maintenance approaches allow for the optimization of maintenance schedules by forecasting equipment failures in advance. As a result, unplanned downtime is reduced, maintenance costs are minimized, and equipment lifespan is extended.

In predictive maintenance systems, the condition of rotating equipment such as bearings is continuously monitored using sensors. The collected data is analyzed through ML algorithms to detect fault patterns at an early stage. In this process, vibration, temperature, and acoustic data are especially valuable indicators. AI-based systems can process these data streams to predict maintenance needs and prevent unexpected equipment breakdowns.

Moreover, explainable artificial intelligence (XAI) techniques enhance the transparency of these systems by making the decision-making process interpretable. XAI helps maintenance teams understand the reasoning behind model outputs, allowing for more informed and trustworthy decisions.

Moreover, explainable artificial intelligence (XAI) techniques enhance the transparency of these systems by making the decision-making process interpretable. XAI helps maintenance teams understand the reasoning behind model outputs, allowing for more informed and trustworthy decisions.

Therefore, combining a bearing dataset with predictive maintenance and explainable AI techniques provides a highly valuable solution for industrial applications. This approach contributes to the development of sustainable maintenance strategies from both technical and economic perspectives.

2.2 Bearings

A bearing is a component of a machine that lowers friction between moving parts and limits relative motion to only the intended motion. For instance, the bearing's design may allow the moving part to freely rotate around a fixed axis or move linearly; it may also regulate the normal force vectors acting on the moving parts to prevent motion. By reducing friction, the majority of bearings enable the intended motion. Bearings can be widely categorized based on the direction of loads (forces) supplied to the parts, the type of operation, or the motions permitted.

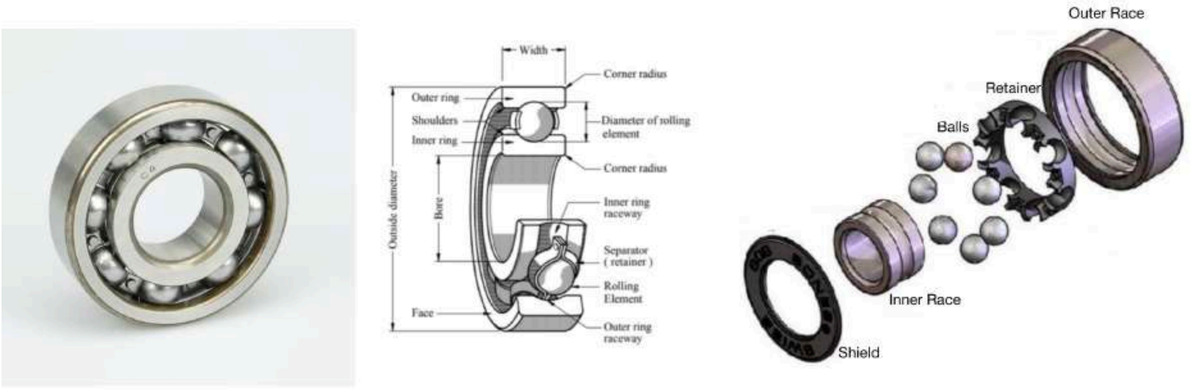


Figure 3: A ball bearing

2.2.1 Structure of Bearings

Bearings are precision-engineered machine elements designed to reduce friction between moving parts and support loads. The typical structure of an anti-friction bearing includes the following components:

- Inner Ring: Mounted on the rotating shaft, this ring provides the inner raceway for the rolling elements. Outer Ring: Fixed to the housing, it offers the outer raceway for the rolling elements.
- Rolling Elements: Balls or rollers positioned between the inner and outer rings, these elements carry the load and facilitate relative motion with minimal friction.
- Raceways: Curved grooves on both the inner and outer rings that guide the rolling elements.
- Separator (Retainer): Maintains uniform spacing between the rolling elements to avoid contact and reduce wear.
- Shoulders and Corner Radius: Structural features that support ring stability and load capacity.
- Bore and Outside Diameter: These define the internal and external dimensions for mounting the bearing.
- Face and Width: Determine axial dimensions, affecting how the bearing fits within an assembly. This architecture ensures that bearings can support radial and axial loads while allowing smooth, low-friction rotation under high-speed conditions.

2.2.2 Types of Bearings

In mechanical systems, rotary bearings support rotating parts like shafts or axles and transmit axial and radial loads from the load source to the supporting structure. A shaft revolving in a hole is the most basic type of bearing, known as a plain bearing. The purpose of lubrication is to lower friction. There are various types of lubricants, such as liquids, solids, and gases. The particular application and variables like temperature, load, and speed all influence the lubricant selection. Rolling components, such as rollers or balls with a circular cross-section, are positioned between the bearing assembly's races or journals in ball and roller bearings to lessen sliding friction. There are many different types of bearings available to ensure that the application's requirements are appropriately satisfied for optimal performance, durability, efficiency, and dependability.

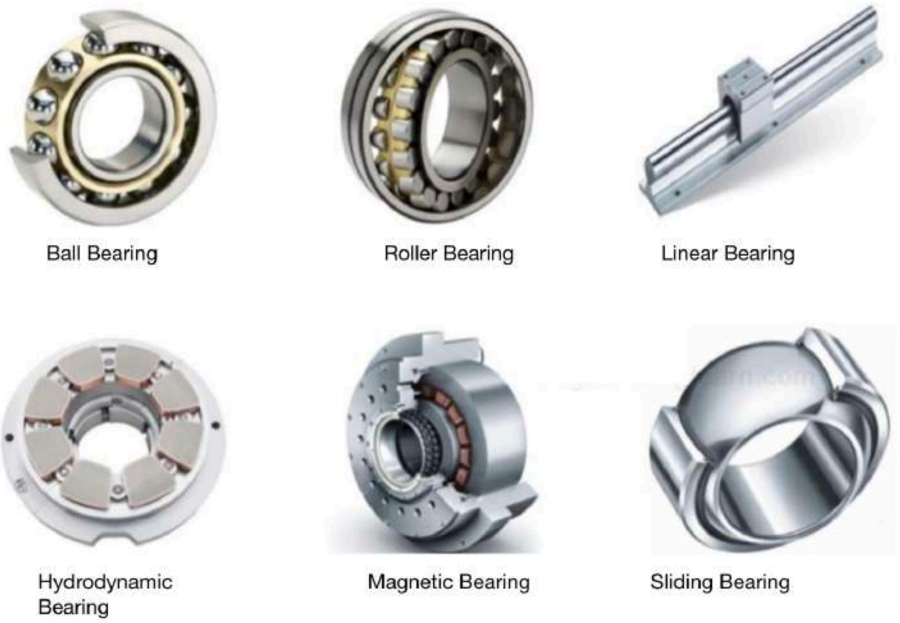


Figure 4: Types of bearings

2.2.3 Bearing Life

The service life of a bearing refers to the duration it can operate effectively under specific operating conditions before a failure occurs due to material fatigue, wear, or other degradation mechanisms. Since bearing life is statistically variable, standardized definitions are used to predict and evaluate performance:

L10 Life (Basic Rating Life):

The L10 life, also known as the basic rating life, is the number of revolutions (or operating hours) at which 90 % of a sufficiently large group of identical bearings are expected to operate without experiencing fatigue failure under a specified load and speed. It is defined by ISO 281 standard and calculated using the formula:

$$L_{10} = \left(\frac{C}{P}\right)^p \tag{1}$$

If the speed is constant, it is often preferable to calculate the life expressed in operating hours using:

$$L_{10h} = \frac{10^6}{60n} L_{10} \tag{2}$$

where:

L_{10} = basic rating life [millions of revolutions]

L_{10h} = basic rating life [operating hours]

C = Kaydon dynamic load rating (shown as C_K in the product tables)

P = equivalent dynamic bearing load

n = rotational speed [r/min]

p = exponent of the life equation; = 3 for ball bearings = 10/3 for roller bearings

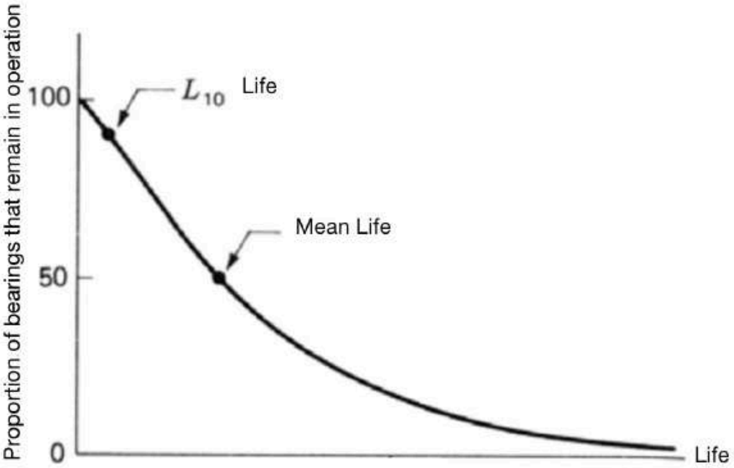


Figure 5: Bearing life

The mean life is the average operational life that bearings achieve before failure. It is typically about 5 times the L10 life, but varies depending on material, lubrication, alignment, contamination, and other operating conditions. Unlike L10, mean life does not represent a conservative estimate and is less commonly used in critical engineering applications.

2.2.4 Bearing Characteristic Fault Frequencies

Bearings is very important part in rotating machines, because they help shaft and housing to move with less friction. If bearings get damage, it can cause problems like machine stop, production delays, or even dangerous situations. So, finding the faults in bearings early is really important for maintenance planning and machine safety.

Vibration methods is often used to find these kinds of faults. When something wrong happens in the bearing — like crack on ball or damage on race surface — it makes special kind of vibration during rotation. These vibrations usually fall into two groups: forced vibrations, which comes from repeating forces during rotation, and free vibrations, which appears when system's natural frequencies gets excited after a shock.

In this study, these vibration types will be explained in detail, and how they can help to detect bearing faults early. Also, some common analysis methods will be discussed which are used for understand these vibration signals.

Force Vibrations:Each bearing has a unique set of geometric parameters (such as number of rolling elements, pitch diameter, contact angle, etc.), from which its characteristic fault frequencies can be calculated. When a bearing begins to deteriorate, these frequencies manifest as distinct peaks in the vibration spectrum. These signatures are crucial in predictive maintenance, as they allow early detection and localization of faults within the bearing.

The primary fault frequencies associated with rolling element bearings are:

- BPFO (Ball Pass Frequency Outer Race): Caused by a defect on the outer race as each rolling element passes over it.

- BPFI (Ball Pass Frequency Inner Race): Generated when a defect exists on the inner race, struck by each rolling element.

- BSF (Ball Spin Frequency): Occurs when the rolling elements themselves are damaged and spin unevenly.

- FTF (Fundamental Train Frequency): Associated with defects in the cage or separator.

These frequencies are derived using the bearing's geometric characteristics and shaft rotation speed, as shown in the standard equations. Notably, these frequencies are non-synchronous — they do not coincide with the fundamental rotating speed or its harmonics. This is due to their dependence on internal bearing geometry, which results in non-integer frequency components in the vibration spectrum.

Consequently, vibration signals resulting from bearing defects appear at distinct, non-integer frequencies that are separate from those generated by other rotating components. This makes it possible to identify bearing faults specifically and differentiate them from other types of mechanical issues.

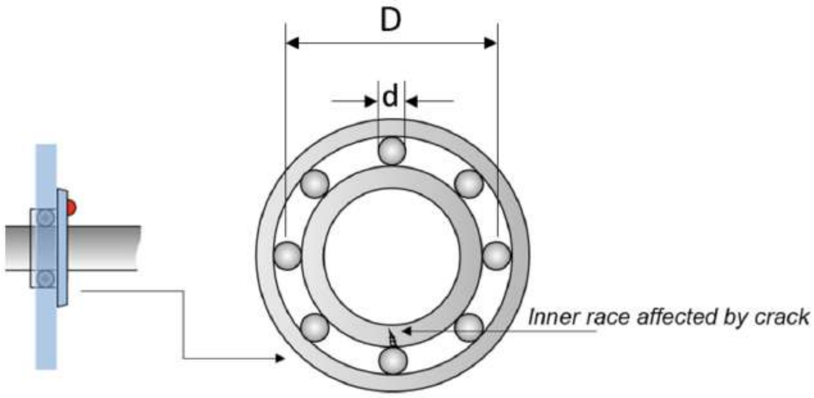


Figure 6: Bearing Specifications

Ball Pass Frequency Outer Race

This frequency corresponds to the number of times the rolling elements pass over a single point on the outer race per second. A defect on the outer race will produce vibration impulses at this frequency.

$$\text{BPFO} = n \cdot \frac{f_r}{2} \cdot \left(1 - \frac{d}{D} cos(\theta)\right) \tag{3}$$

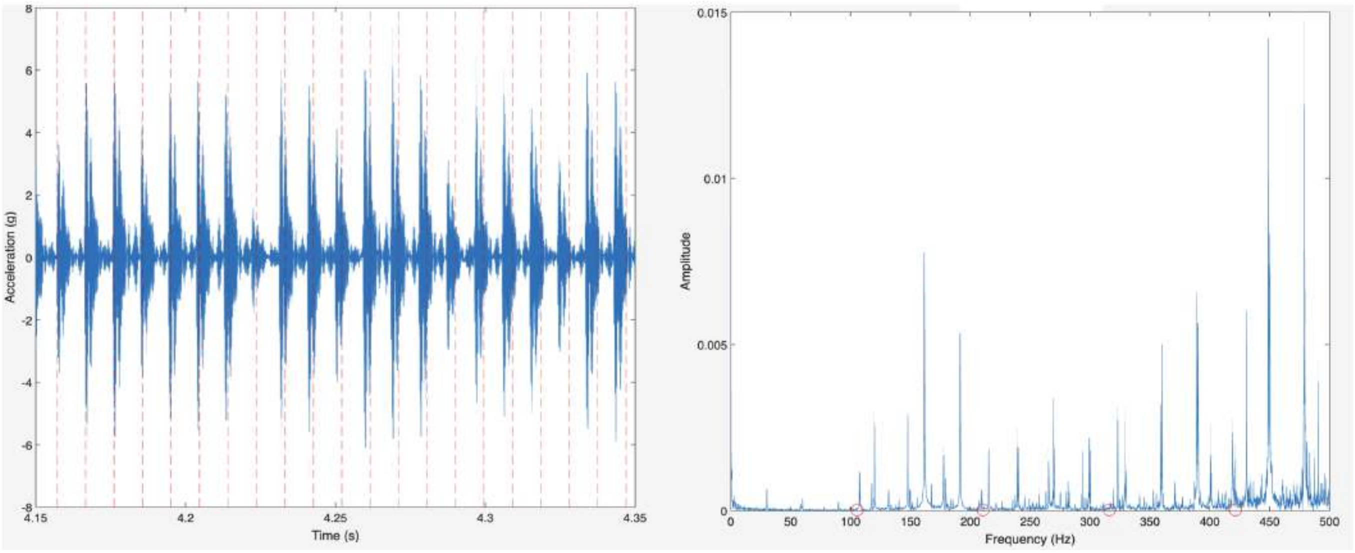


Figure 7: Record 135DE (48k, 0.007 in. drive end outer race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal

Ball Pass Frequency Inner Race

This is the frequency at which a defect on the inner race will generate vibration signals, as the rolling elements strike the damaged area.

$$\text{BPFI} = n \cdot \frac{f_r}{2} \cdot \left(1 + \frac{d}{D} cos(\theta)\right) \tag{4}$$

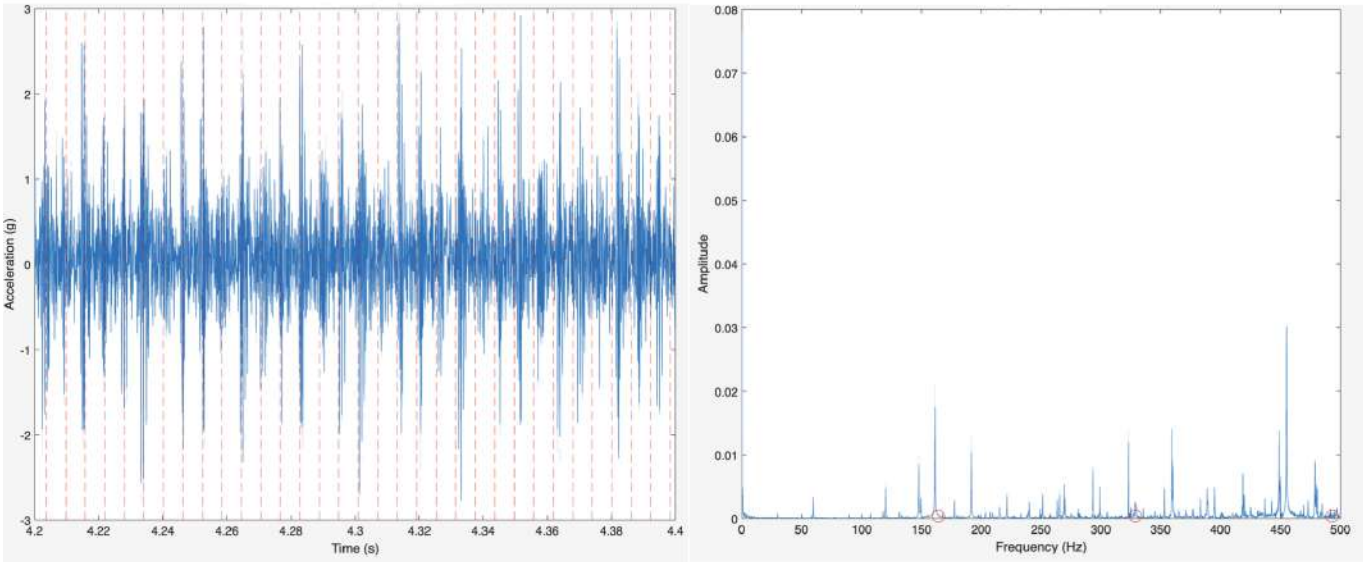


Figure 8: Record 109DE (48k, 0.007 in. drive end inner race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal

Ball Spin Frequency

If a defect is present on a rolling element, it will cause vibrations as the element itself spins around its own axis. This frequency is lower than BPFO and BPFI, and is more complex due to the spin-slip dynamics of the element.

$$\text{BSF} = \frac{D}{2d} \cdot f_r \cdot \left(1 - \left(\frac{d}{D} \cdot cos(\theta)\right)^2\right) \tag{5}$$

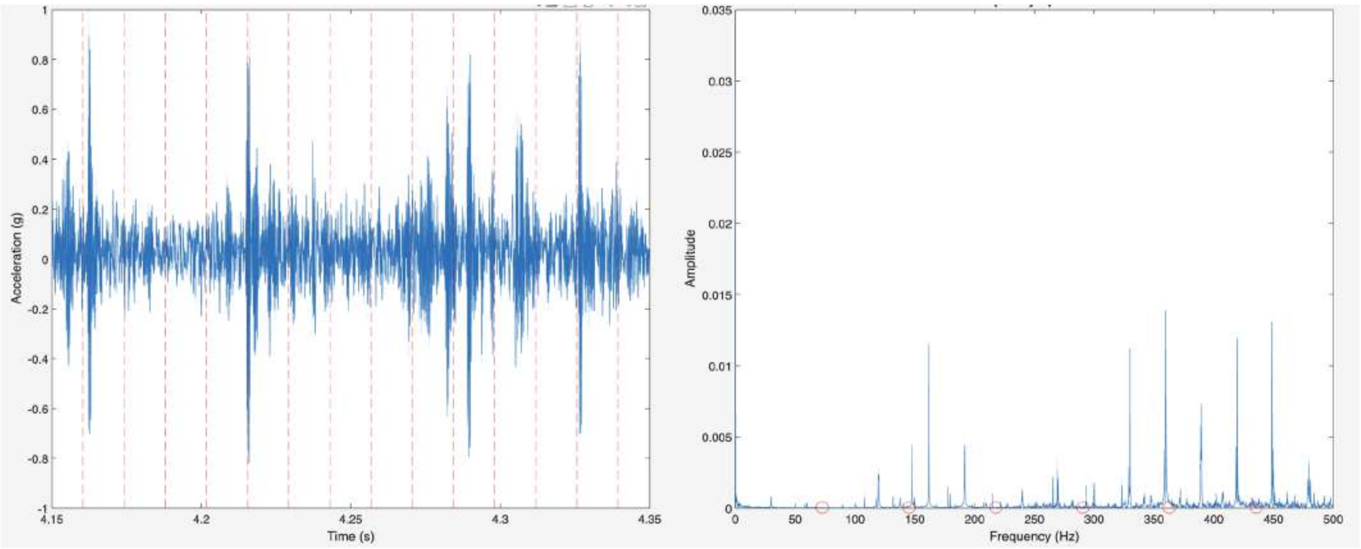


Figure 9: Record 189DE (48k, 0.014 in. drive end outer race fault, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal

Fundamental Train Frequency (Cage Frequency)

This is the frequency at which the bearing cage (retainer) rotates. Defects in the cage are less common but can be detected by monitoring this frequency.

$$\text{FTF} = \frac{f_r}{2} \cdot \left(1 - \frac{d}{D} \cdot cos(\theta)\right) \tag{6}$$

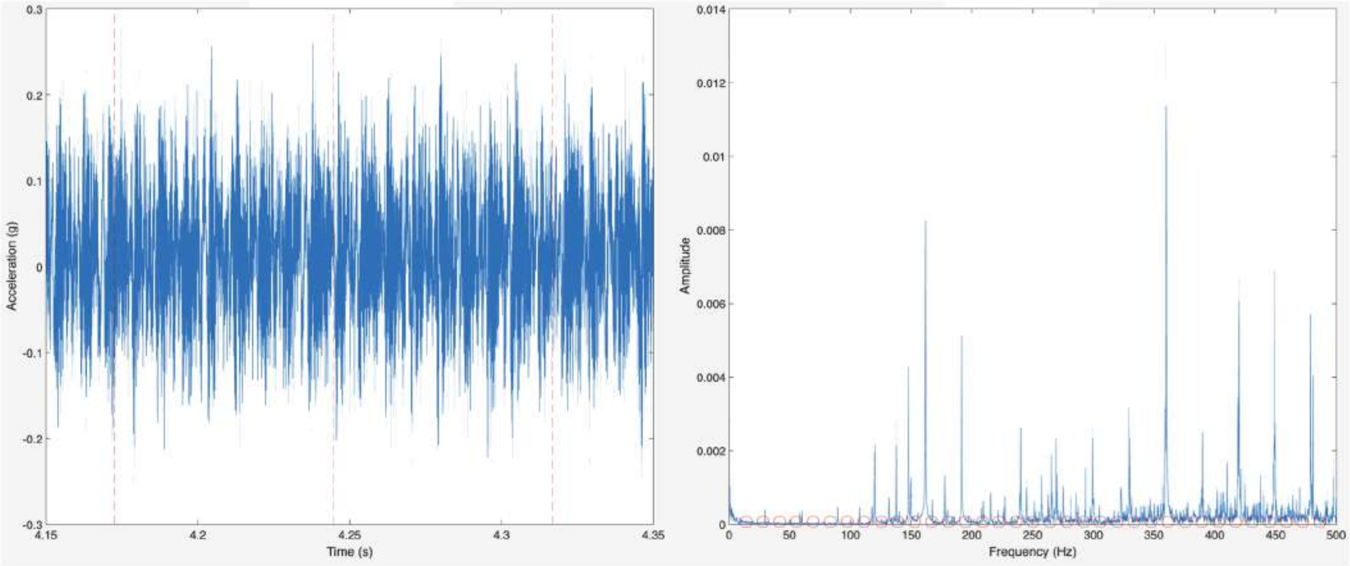


Figure 10: Record 97DE (48k, 1797 rpm). 1) Raw time signal, 2)FFT spectrum of raw signal

where:

d = Ball diameter

D = Pitch diameter

θ = Bearing contact angle

f_r = Shaft speed

Each component of a rolling element bearing exhibits characteristic fault frequencies when damaged. These frequencies are closely related to the bearing geometry and the shaft rotation speed (RPM). The approximate ranges for a bearing with 8 balls (N = 8) are as follows:

- BPFO (Ball Pass Frequency Outer Race): 2.8 × RPM
- BPFI (Ball Pass Frequency Inner Race): 5.2 × RPM
- BSF (Ball Spin Frequency): 1.45 × RPM
- FTF (Fundamental Train Frequency - Cage): 0.35 × RPM

These fault frequencies often appear in the vibration spectrum as peaks, not only at their fundamental frequency but also at their harmonics and sidebands, especially in advanced stages of damage. Because they are generally non-integer multiples of the shaft frequency, they can be clearly distinguished from synchronous machine components.
This frequency-based approach enables early fault detection and localization, which is essential for predictive maintenance applications. The failure frequencies and bearing specifications of the CWRU data set are given in Table 4 and Table 5.

Table 4: Bearing details and fault frequencies.

Position on rig	Model number	BPFI	BPFO	FTF	BSF
Drive end	SKF 6205-2RS JEM[a]	5.415	3.585	0.3983	2.357

Table 5: Bearing specifications [inches]

Inside diameter	Outside diameter	Thickness	Ball Diameter	Pitch diameter
0.9843	2.0472	5.5906	0.3126	1.537

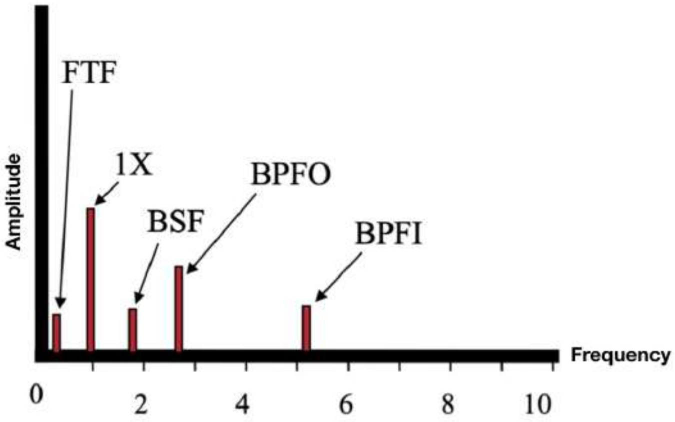


Figure 11: Fault frequencies

Note: Peaks are also seen in the spectrum at the harmonics and sidebands of the damage frequencies.

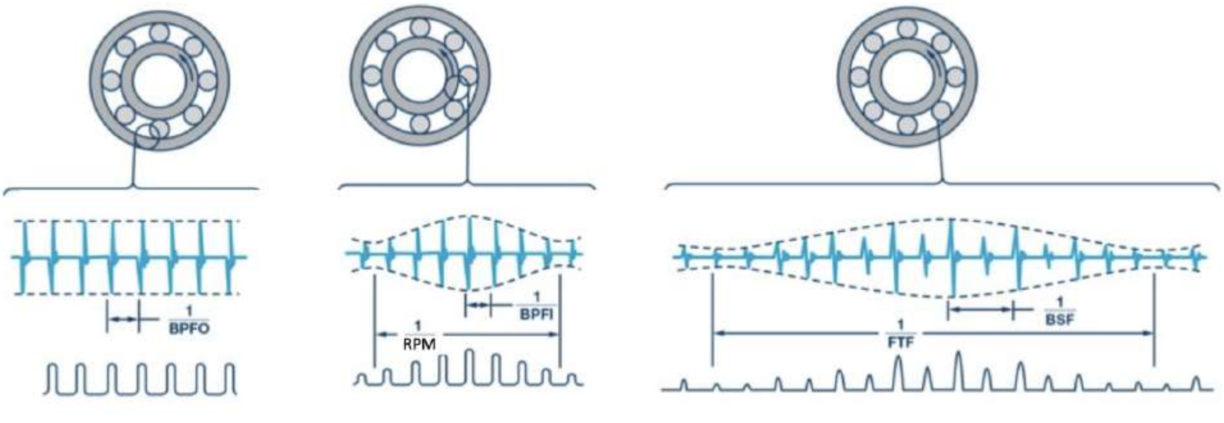


Figure 12: Fault types

Free Vibration:Free vibrations are generated when a mechanical system responds to an external impulse and begins to oscillate near its natural frequencies. In the context of bearing defects, such vibrations often appear as a result of repeated impulsive forces created during the rotation process. Damaged components such as cracked rolling elements or surface irregularities can introduce sudden impacts, which may excite the system's inherent structural modes.

When the frequency content of the impulsive excitation is close to one of the system's natural frequencies, a resonance condition can arise. In such situations, vibration amplitudes may increase significantly, especially in systems where damping is low. As a result, the machine structure may exhibit prolonged oscillations corresponding to one of its modal shapes, even if the forcing frequency is not directly matched with the natural frequency.

Unlike forced vibrations, which typically show narrowband frequency characteristics, free vibrations can present more dispersed spectral behavior. These signals are often non-stationary and contain frequency components that evolve over time. Therefore, analysis based solely on Fourier Transform might not be sufficient. Instead, time-frequency domain methods are generally more effective to understand these transient phenomena.

It is also observed that free vibrations associated with bearing faults can sometimes appear at higher frequency regions in the spectrum. This occurs due to the transmission of impact energy throughout the system, which activates global resonant responses rather than local periodic excitations. As a result, even relatively slow rotational speeds can lead to high-frequency structural vibrations, making accurate fault diagnosis more challenging.

2.3 Bearing Fault Stages

The diagnosing process is impacted since a bearing failure typically manifests in stages. Depending on their magnitude and the patterns they create in the frequency spectrum, bearing defects can be divided into four wear stages. It is crucial to identify and address the problems as soon as possible since, as one may anticipate, as the number of stages increases, the vibration levels rise and the system gets closer to critical failure.

2.3.1 Bearing Fault Stage I: Initiation and Early Detection

In the initial stage of bearing degradation, the defect remains microscopic and does not manifest through observable symptoms such as abnormal vibrations, temperature rise, or audible noise. At this phase, conventional low-frequency vibration analysis methods are typically insufficient for detection. Instead, diagnostic approaches that operate in high-frequency domains—particularly in the range of 20 to 40 kHz—are necessary to identify these subtle faults. Techniques such as envelope analysis exploit the natural resonant frequencies of the bearing components to amplify defect-related impacts, while methods like the Shock Pulse Method (SPM) rely on transducers calibrated to specific resonance peaks. These high-frequency approaches are particularly valuable for early-stage fault detection, as they offer enhanced sensitivity to minor surface imperfections or incipient cracks in rolling elements or raceways. Their effectiveness is especially pronounced in slow-speed rotating machinery, where traditional techniques may fail to capture the low-energy signals of an emerging defect.

Despite the utility of high-frequency methods, it is advisable that maintenance personnel correlate findings with lower-frequency observations before initiating any major intervention—especially in high-speed systems. Premature disassembly or overhaul might not be justified when the defect remains at a non-propagating stage. Instead, predictive maintenance strategies such as condition monitoring, routine acoustic emission tracking, and ensuring optimal lubrication practices are critical during this period. Proper lubrication not only mitigates friction and wear but also influences the signal clarity in high-frequency measurements, reinforcing the reliability of early-stage diagnostics.

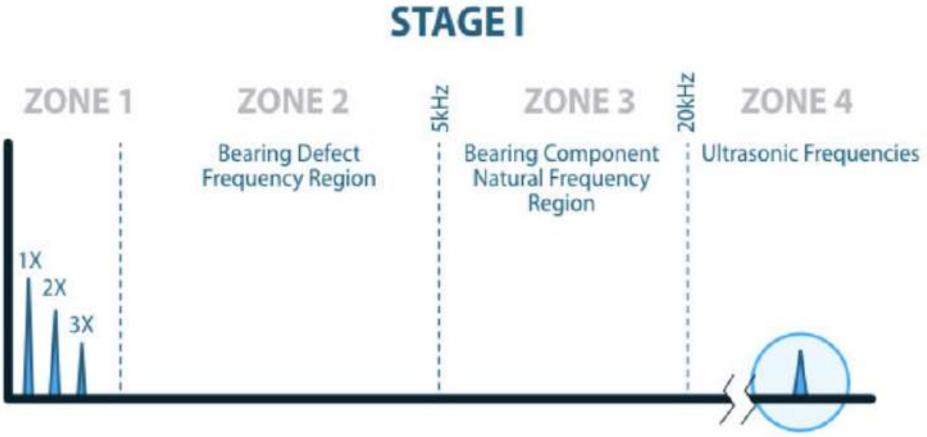

STAGE I
ZONE 1
ZONE 2
Bearing Defect Frequency Region
5kHz
ZONE 3
Bearing Component Natural Frequency Region
20kHz
ZONE 4
Ultrasonic Frequencies
1X
2X
3X

Figure 13: Bearing fault - stages I

2.3.2 Bearing Fault Stage II: Progressive Damage and Resonant Excitation

As the bearing defect evolves beyond the incipient phase, it transitions into the second stage of fault development. During this progression, the localized damage—such as pitting or spalling—generates high-energy impacts that begin to excite the bearing's natural frequencies. These resonances, which are predominantly influenced by the bearing's geometry, material properties, and mounting conditions, typically manifest in frequency ranges above 5 kHz.

At this stage, the vibrational energy within the system increases noticeably, particularly in the high-frequency spectrum. One of the most effective tools for detection at this point is envelope analysis, which enables the extraction of modulated signals caused by repetitive impacts. By demodulating the high-frequency carrier signal, this technique reveals characteristic fault frequencies and spectral peaks associated with specific bearing components (e.g., ball pass frequency of outer race - BPFO, ball spin frequency - BSF, etc.). However, due to its signal processing complexity, envelope analysis demands both advanced computational capacity and precise filtering strategies to minimize the effects of noise and unrelated vibrations.

As the mechanical degradation intensifies, spectral sidebands begin to emerge symmetrically around the defect frequency peaks. These sidebands are typically a result of amplitude modulation phenomena and are strong indicators that the system is approaching Stage III, where damage becomes more severe and machine reliability is significantly compromised.

At this point, the bearing is no longer in a stable condition, and continued operation without intervention may lead to catastrophic failure. Thus, intensified monitoring is imperative, and predictive maintenance planning should be accelerated to prevent unscheduled downtime or secondary damage to adjacent components.

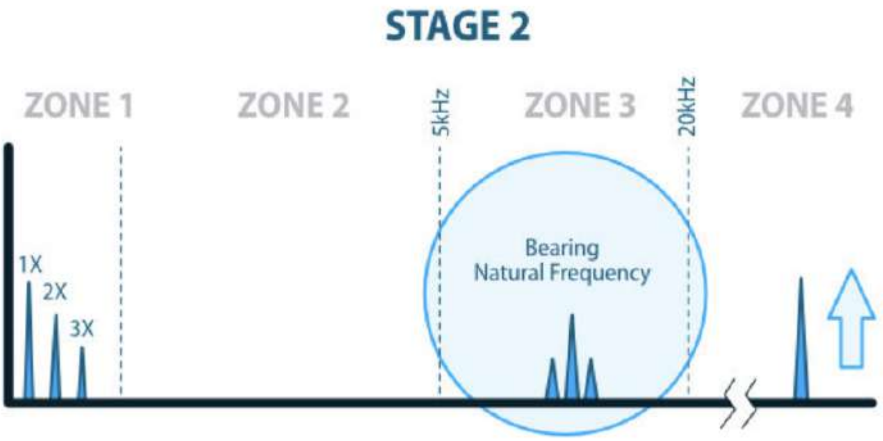


Figure 14: Bearing fault - stages II

2.3.3 Bearing Fault Stage III: Established Patterns in Low-Frequency Spectrum

In the third stage of bearing deterioration, characteristic fault patterns become distinctly observable within the low-frequency vibration spectrum. Although high-frequency energy content continues to rise—and advanced techniques such as envelope analysis remain applicable—the most definitive and classically documented signatures of bearing defects now emerge clearly in the velocity-based spectral analysis.

In the case of an outer race defect, harmonics of the Ball Pass Frequency of the Outer race (BPFO) dominate the low-frequency region. Conversely, an inner race fault is characterized by harmonics of the Ball Pass Frequency of the Inner race (BPFI) accompanied by prominent sidebands spaced at the shaft's rotational frequency. These sidebands are indicative of amplitude modulation, which occurs as the localized defect cyclically enters and exits the load zone, producing a modulated vibration response.

It is important to note that the aforementioned spectral characteristics are based on the assumption of a stationary outer race. If the outer race rotates, the spectral patterns associated with BPFO and BPFI are effectively reversed. Additionally, ball or roller element defects typically manifest through harmonics of the Ball Spin Frequency (BSF), often modulated by the Fundamental Train Frequency (FTF) due to interactions with the cage or retainer structure.

In practical diagnostics, the identification of three or more significant harmonics is commonly accepted as a threshold for confirming a critical fault condition, warranting immediate bearing replacement to avoid secondary system damage or unplanned machine shutdown.

This stage presents a strategic advantage in terms of cost-effective monitoring, as the fault signatures lie predominantly below the 5 kHz range. Standard vibration transducers equipped with integrated frequency analysis capabilities (such as built-in Fast Fourier Transform modules) are generally sufficient for capturing the required data. This facilitates timely diagnostics without the need for highly specialized high-frequency equipment, allowing for efficient integration into predictive maintenance programs.

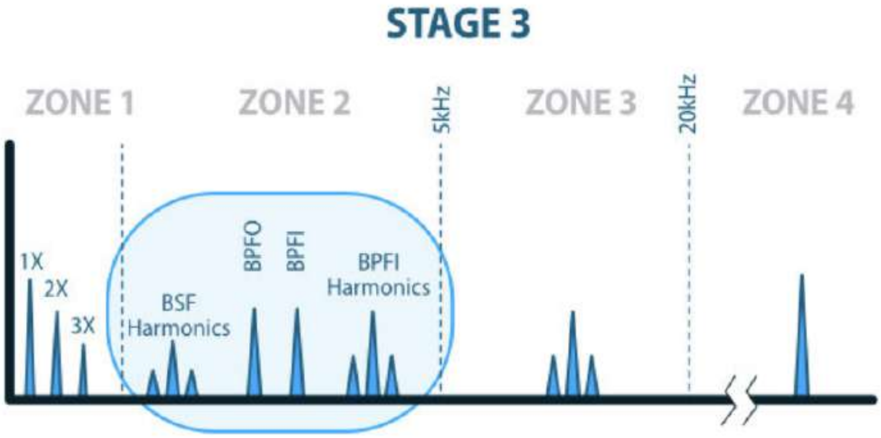


Figure 15: Bearing fault - stages III

2.3.4 Bearing Fault Stage IV: Terminal Degradation and Impending Failure

The fourth and final stage of bearing degradation signifies the onset of imminent failure and represents a critical threshold in the machine's operational health. At this point, several counterintuitive phenomena begin to emerge. Although severe internal damage continues, high-frequency diagnostic indicators—such as those used in envelope analysis or crest factor measurements—may start to decline in amplitude. This reduction is attributed to the progressive rounding of defect edges on the raceways and rolling elements, which softens the impact impulses that previously energized high-frequency responses.

Additionally, displaced metal debris generated from spalling or fragmentation within the bearing may become compacted and temporarily fill surface defects, further damping the impact intensity and diminishing the clarity of defect-related spectral features. As the system degrades, bearing internal clearances increase significantly, resulting in mechanical looseness. This looseness introduces broad-spectrum vibration characterized by integer multiples of shaft rotational speed (harmonics of 1× RPM) and contributes to a sharp rise in overall RMS vibration levels.

One of the most defining features of Stage IV is the disappearance of discrete fault frequencies such as BPFO, BPFI, BSF, and FTF. Instead, the vibration spectrum becomes dominated by randomized broadband noise, manifesting as an elevated noise floor. This loss of diagnostic clarity signals that structural integrity has been compromised beyond recoverable limits. Continued machine operation under such conditions poses a severe risk of catastrophic failure, and immediate shutdown and bearing replacement are imperative to prevent collateral damage.

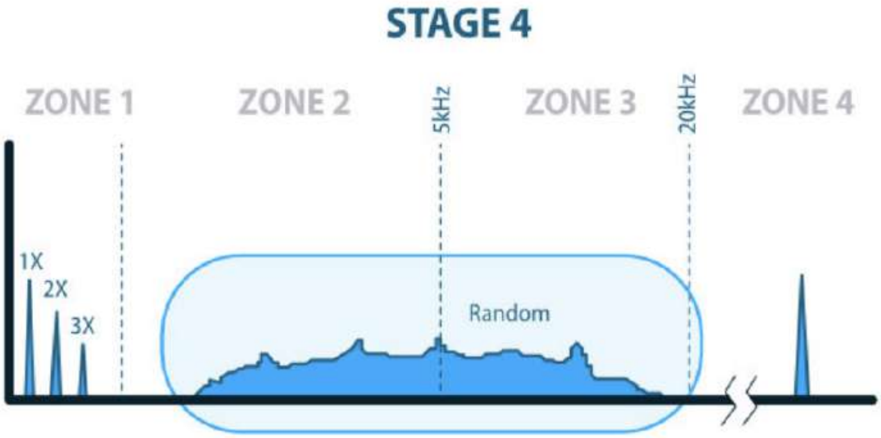


Figure 16: Bearing fault - stages IV

2.3.5 Feature in Time-Domain

In vibration-based fault diagnosis of bearings, time domain features are widely used to extract statistical information from the raw acceleration signals. These features are computed directly from the time waveform and provide insights into the amplitude, energy, and statistical structure of the signal, without requiring

2.3.5.1 Root Mean Square (RMS) : The root mean square (RMS) is one of the most commonly used time-domain features in bearing fault detection. It reflects the overall energy content of the signal and is particularly sensitive to both continuous vibrations and impulsive events. RMS increases with the severity of the fault, making it a reliable health indicator for rotating machinery.

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{7}$$

Where x_i is the value of the vibration signal at time index i, and N is the total number of samples.

2.3.5.2 Standard Deviation(SD) :Standard deviation measures the spread or dispersion of the signal values around the mean. In mechanical systems, an increase in SD may indicate rising levels of irregular vibration due to looseness, imbalance, or fault initiation. It is closely related to RMS but lacks the squaring effect, making it less sensitive to extreme values.

$$\mathrm{SD} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (x_i - \overline{x})^2} \tag{8}$$

Where $\overline{x}$ is the mean of the signal.

2.3.5.3 Kurtosis :Kurtosis describes the "peakedness" of a signal's distribution and is highly effective in detecting impulsive events such as bearing faults. A high kurtosis value suggests the presence of rare, sharp impacts — often a sign of local defects. It is a fourth-order statistical moment and is zero for a normal (Gaussian) distribution.

$$\mathrm{Kurtosis} = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{x_i - \overline{x}}{\sigma}\right)^4 \tag{9}$$

Where σ is the standard deviation. For healthy machinery, kurtosis values generally stay near 3 (Gaussian). Values significantly higher than 3 indicate impulsive behavior.

2.3.5.4 Peak-to-Peak Value :The peak-to-peak value represents the total range of vibration by calculating the difference between the maximum and minimum values of the signal. It is useful in identifying sudden bursts in vibration amplitude, which may arise from crack propagation or severe impacts inside the bearing.

$$\text{Peak-to-Peak} = \text{max(x)-min(x)} \tag{10}$$

This measure is especially helpful in capturing non-periodic spikes in vibration signals.

2.3.5.5 Skewness :Skewness is a measure of signal asymmetry. A perfectly symmetrical signal has a skewness of zero. Positive skewness indicates the presence of more extreme high values, whereas negative skewness suggests more extreme low values. In the context of bearing diagnostics, it can help reveal directional bias in impacts or uneven wear patterns.

$$\text{Skewness} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \overline{x}}{\sigma}\right) \tag{11}$$

Skewness can be useful when combined with other metrics like kurtosis to get a fuller picture of the signal behavior.

2.3.5.6 Crest Factor(CF)

: Crest factor is a widely used feature in condition monitoring to detect impulsive behaviors in vibration signals. It is defined as the ratio between the signal's maximum absolute value and its root mean square (RMS). An increase in crest factor may indicate the presence of localized defects, such as cracks or pitting in bearing elements.

$$\text{CF} = \frac{\max|x_i|}{\text{RMS}} \tag{12}$$

Where $\max|x_i|$ represents the absolute peak of the signal, and RMS is the root mean square value.

2.3.5.7 Impulse Factor(IF)

: Impulse factor quantifies the sharpness of vibration impacts by comparing the peak value to the mean value of the signal. It is useful for detecting sudden, transient shocks that are often associated with early-stage bearing faults or structural looseness.

$$\text{IF} = \frac{\max|x_i|}{\overline{x}} \tag{13}$$

2.3.5.8 Clearance Factor(CLF)

: Clearance factor reflects the relationship between the peak value and the average rectified value of the signal. It is particularly sensitive to clearance-related faults and looseness within the bearing structure, such as play between the rolling elements and races.

$$\text{CLF} = \frac{\max|x_i|}{\frac{1}{N}\sum_{i=1}^{N}\sqrt{|x_i|}} \tag{14}$$

This feature increases when high-magnitude impacts occur within a generally low-energy signal. **NOTE:**In this study, handcrafted feature extraction methods were deliberately not applied. Instead, raw vibration signals were directly utilized as input to the deep learning architecture. This approach eliminates the need for prior statistical or time-domain feature engineering, enabling the model to autonomously learn informative patterns from the raw data. No additional preprocessing, such as filtering, normalization, or transformation, was performed on the signal. The rationale behind this decision is rooted in the inherent ability of deep learning models—particularly convolutional and recurrent networks—to capture both local and temporal structures without relying on predefined features. By avoiding manual feature selection, the risk of losing potentially meaningful information was minimized, and the architecture was allowed to focus entirely on learning optimal representations directly from the signal. Such end-to-end learning strategies have shown significant promise in vibration-based fault diagnosis, particularly in scenarios where feature relevance may vary across different fault types or operating conditions.

2.3.6 Sensor Technology in Bearings

In predictive maintenance applications, especially for rotating machinery like bearings, the use of appropriate sensors is essential for reliable fault detection. The most commonly used sensors include:

1 Accelerometers (Vibration Sensors)

Accelerometers are the primary sensors used in bearing condition monitoring. They measure vibration in terms of acceleration (m/s^2 or g). Accelerometers offer high sensitivity and are especially effective for detecting early-stage faults such as minor cracks or pitting on the races or balls.

- High-frequency response makes them ideal for identifying high-frequency fault components such as ball spin frequency (BSF) or harmonics of BPFI/BPFO.
- Mounted close to the bearing housing to minimize signal loss and noise.
- Can be used in both time-domain and frequency-domain analysis (via FFT).

2 Velocity and Displacement Spectrums

Vibration signals can be analyzed in different domains depending on the fault severity and frequency content:

- Acceleration Spectrum (g or m/s^2): Best suited for detecting early-stage faults. It captures high-frequency content but can be noisy.
- Velocity Spectrum (mm/s or in/s): Often used in industry standards (e.g., ISO 10816). It is ideal for moderate faults and gives a clearer picture of the fault severity.
- Displacement Spectrum (µm): Used for identifying late-stage, low-frequency faults such as misalignment, looseness, or unbalance. However, it is less sensitive to bearing defects compared to acceleration. Conversions among these domains are possible through integration or differentiation (e.g., integrating acceleration gives velocity).

3 Temperature Sensors

Bearings tend to heat up as friction increases due to internal defects. Therefore, thermocouples or RTDs (Resistance Temperature Detectors) are used to monitor abnormal temperature rises.

- Temperature rise is a slow-developing indicator and typically used in conjunction with vibration data.

4 Sampling Frequency and Data Acquisition

- The sampling rate must be at least 2.5–3 times higher than the highest frequency of interest (according to Nyquist theorem).
- For bearing analysis, typical sampling frequencies range from 10 kHz to 50 kHz, depending on machine speed and sensor type.
- Data is usually acquired periodically (e.g., every few seconds to minutes) or continuously in critical applications (e.g., turbines).

5 Sensor Placement and Data Selection

In rotating machinery, vibration sensors are typically mounted on both sides of the motor to monitor the condition of the bearings. These two standard locations are:

- Drive End (DE): The end of the motor where the mechanical load is applied (e.g., shaft, gearbox, or pulley connection).
- Fan End (FE): The end of the motor where the cooling system (fan) is located.

In the bearing dataset used in this study, vibration signals were recorded from both the Drive End (DE) and the Fan End (FE) using accelerometers. However, in the scope of this project, only the Drive End data is utilized.

The rationale for this selection is based on the fact that the Drive End bearing is subjected to higher mechanical stress due to its direct connection with the load. Consequently, defects are more likely to originate and develop in this region. Moreover, vibration signals from the DE side typically contain more pronounced fault signatures, making them more informative for fault detection and condition monitoring tasks.

Focusing on the DE data enhances the model's ability to detect early-stage bearing faults and increases the reliability of the predictive maintenance framework.

2.4 CWRU Dataset Preprocessing

In predictive maintenance applications, the quality of the results is significantly influenced by how effectively the raw sensor data is prepared for analysis. The raw dataset employed in this study contains time-series vibration signals obtained from rotating machinery. Such data, in its original form, is unsuitable for direct use in machine learning models due to its high dimensionality, presence of noise, and lack of structured labeling. To enable meaningful learning and classification, several preprocessing steps were systematically applied, as outlined below.

2.4.1 Classification by Fault Type

In this study, the fault classification was carried out based on fault type, rather than fault size or load conditions. As shown in Table 2, load values 0, 1, and 2 were exclusively used for training purposes. However, the classification labels correspond to fault types (e.g., ball fault, inner race fault, outer race fault).

The reason behind this decision lies in the operational priority of identifying the location and nature of the fault, rather than its severity or the operating condition at the time. Fault type is a fundamental indicator of the mechanical issue at hand, directly linked to the faulty bearing component. Correctly identifying whether a fault originates from the ball, the inner race, or the outer race is crucial for effective diagnosis and maintenance planning. Moreover, by aggregating data across various fault sizes and load conditions under the same fault type label, the model is expected to generalize better, learning the invariant features of each fault type irrespective of operating conditions. This approach increases robustness and reflects real-world scenarios, where a diagnosis system is required to detect fault type even when load or severity varies.

2.4.2 Segmentation of Time-Series Data

In order to prepare the dataset for training and testing, vibration signals were segmented into fixed-length windows of 4096 samples. This segmentation was performed using non-overlapping windows to preserve the natural characteristics of each signal segment and to ensure uniform input dimensions for the machine learning models.

The MATLAB scripts were developed to automate the data loading, normalization, windowing, labeling, and saving processes. Raw time-series signals collected from the Case Western Reserve University (CWRU) bearing dataset were first merged and normalized using z-score normalization. This normalization step is essential to reduce the influence of signal amplitude variations, ensuring better generalization during model training.

Each signal was divided into segments of 4096 samples with a step size equal to the window length, resulting in non-overlapping windows. This strategy guarantees a consistent number of samples per input feature, facilitating the training of deep learning architectures such as

Convolutional Neural Networks (CNNs). In each iteration, the windowed segment was assigned a label corresponding to the bearing fault type (e.g., inner race, ball, or outer race fault).
The preprocessed segments were then stored in both .mat and .csv formats for flexible use in MATLAB-based and Python-based environments.

Why 4096 samples?

Segment lengths that are powers of two are particularly advantageous when working with frequency-domain methods or neural networks, especially convolutional architectures. A window size of 4096 provides a balance between preserving signal fidelity and maintaining computational feasibility. Moreover, it aligns well with the resolution needed to detect bearing fault frequencies and other periodic components.

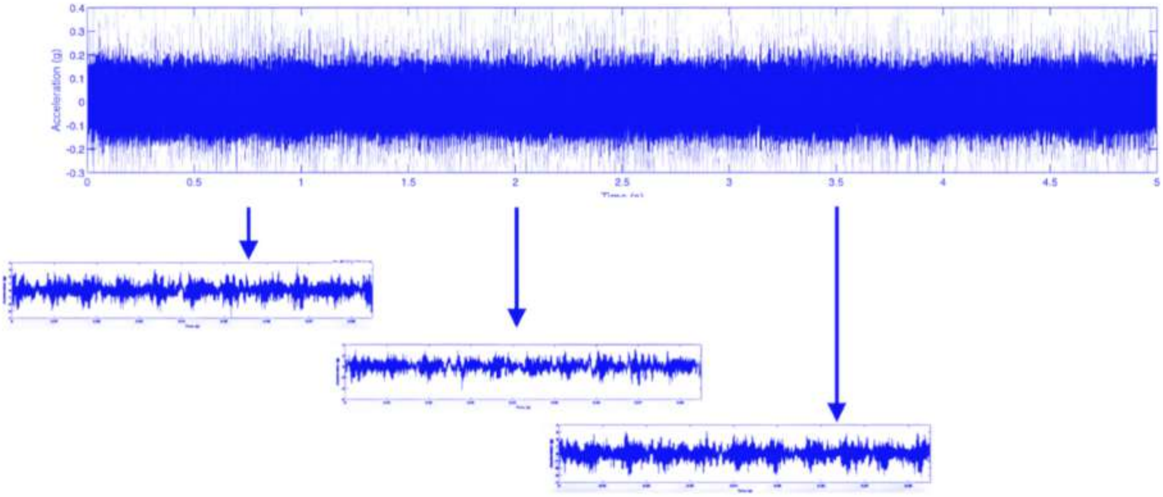


Figure 17: Each sample from 4096 consecutive time points

2.4.3 Feature and Label Extraction

From the structured dataset, the first 4096 columns correspond to vibration signal values (features), while the final column indicates the target class, such as fault severity or operational condition. The features (`X = df.iloc[:, :4096].values`) and labels (`Y = df[’Label’].values`) were extracted separately to facilitate supervised learning.

2.4.4 Encoding of Target Labels

The target labels were initially in categorical form. These were first converted to numerical indices using `LabelEncoder` to allow for compatibility with machine learning pipelines. Subsequently, `one-hot encoding` was applied to prevent the model from inferring ordinal relationships between classes. This transformation is particularly critical for multi-class classification tasks, as it ensures that each class is treated as a distinct and independent category.

2.4.5 Training and Testing Partition

To assess generalization capability, the dataset was divided into training and testing subsets in an 80:20 ratio. A stratified split was employed to preserve the proportional representation of each class in both subsets.`(stratify=Y)` This is especially important in imbalanced datasets, as it prevents the model from being biased toward majority classes and improves robustness.

2.4.6 Normalization of Input Features

Due to inherent variations in sensor readings across samples, feature scaling is essential to standardize the data. Here, a z-score normalization `(StandardScaler)` was applied such that each feature has zero mean and unit variance. Importantly, the scaler was fit only on the training set and subsequently applied to the test set, thereby preventing any form of data leakage. This normalization process enhances model convergence during training and ensures consistency across different input magnitudes.

2.4.7 Reshaping for Model Compatibility

To match the input requirements of the deep learning architecture, the feature arrays were reshaped to a standardized format `(reshape(-1, 4096).transpose(0, 1))` Specifically, each sample was reshaped as a vector of 4096 values, followed by a transposition operation when necessary. This format is particularly suited for one-dimensional convolutional networks and time-series models that rely on fixed-length sequential inputs.

3 MODEL ARCHITECTURES

3.1 How Artificial Neural Networks (ANN) Work?

Artificial Neural Networks are computational models inspired by the biological neural networks in the human brain. In predictive maintenance, they are capable of learning complex, nonlinear relationships in data, enabling critical tasks such as fault detection estimation.

3.1.1 Fundamental Units and Structure

An artificial neural network consists of multiple interconnected artificial neurons arranged in layers. Each neuron computes a weighted sum of its input signals and passes the result through an activation function. Mathematically, the operation of a neuron is defined as:

$$z = \sum_{i=1}^{n} x_i \cdot w_i + b \tag{15}$$

where x_i are the inputs, w_i the weights, and b the bias term. The neuron's output is then computed by applying an activation function $\sigma(z)$:

$$y = \sigma(z) \tag{16}$$

3.1.2 Activation Functions

In artificial neural networks, activation functions introduce non-linearity, enabling the network to learn complex patterns.

1 ReLU (Rectified Linear Unit): ReLU zeros out negative inputs and keeps positive inputs as is, increasing computational efficiency and aiding faster training.

$$\text{ReLU}(x) = max(0, x) \tag{17}$$

2 Sigmoid Function: Squashes output between 0 and 1, suitable for binary classification.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{18}$$

3 Tanh (Hyperbolic Tangent): Outputs values between -1 and 1, providing zero-centered activations.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}} \tag{19}$$

Preserves the sign of input and often performs better than sigmoid in practice.

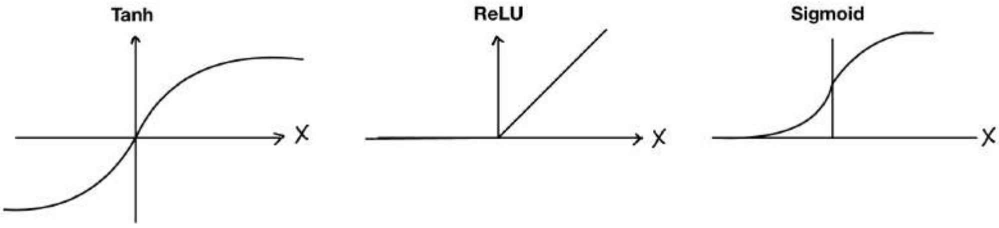


Figure 18: Activation Functions

3.1.3 Network Layers

- Input Layer: The first layer that receives raw input data such as sensor signals. No computation is done here; data is simply forwarded to the next layer.
- Hidden Layers: The core of learning occurs here. Each neuron is connected to all neurons in the previous layer (fully connected). Multiple hidden layers form what is known as deep learning architectures.
- Output Layer: Produces the final prediction or classification. For classification tasks, softmax or sigmoid activation functions are often used.

3.1.4 Forward Propagation

When input data is sent forward via a network to produce an output, this is known as forward propagation. After being approved by hidden layers and processed in accordance with the activation function, the data advances to the next layer. The purpose of the forward flow of data is to prevent circular motion, which does not provide an output.

3.1.5 Loss Functions

Quantify the difference between predicted and true values, minimized during training.

1 Mean Squared Error (MSE) Used for regression tasks; average of squared differences between true and predicted values.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{20}$$

where:

N = number of samples
y_i = true value
$\hat{y}_i$ = predictive value

2 Cross-Entropy Loss Used mainly for classification. Measures difference between true labels and predicted probabilities.

Binary classification:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)] \tag{21}$$

where:

$y_i \in 0, 1$: true label

$\hat{y}_i \in 0, 1$: predicted probability

Multi-class classification (with softmax):

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} log(\hat{y}_{i,c}) \tag{22}$$

C = number of classes

yi, c = one-hot encoded true label

$\hat{y}_{i,c}$ = predicted probability for class c

Multi-class classification (with softmax):

3.1.6 Backpropagation and Weight Update

This phase constitutes the learning mechanism. The error from the loss function propagates backward through the network via the chain rule, computing gradients of each weight's contribution to the error. Weights are then updated using gradient descent:

$$w = w - \alpha \cdot \frac{\partial L}{\partial w} \tag{23}$$

where α is the learning rate, and $\frac{\partial L}{\partial w}$ is the gradient of the loss with respect to the weight.

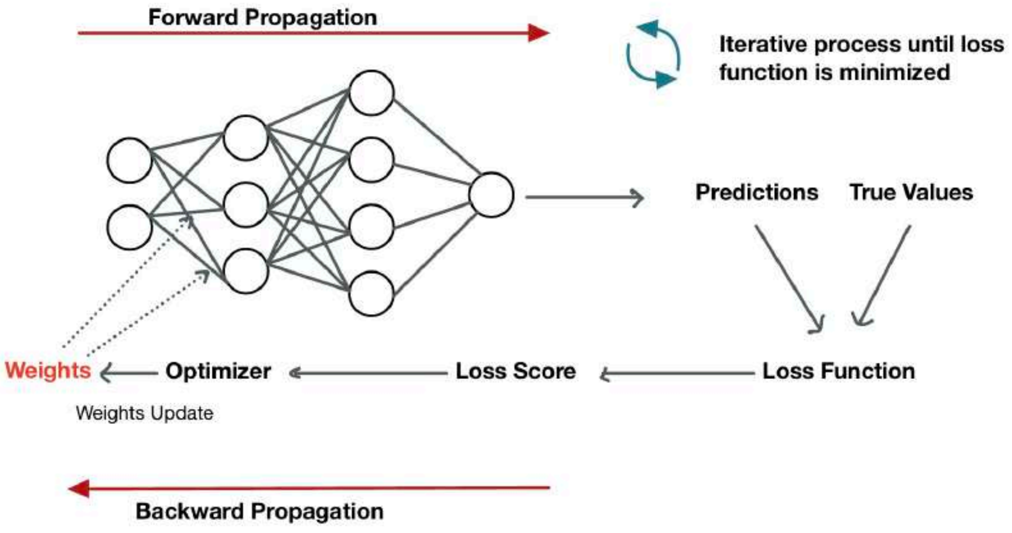


Figure 19: Understanding Forward and Backward Propagation in Neural Networks

3.1.7 Training Loop

Forward propagation, loss calculation, backpropagation, and weight updates are iterated over multiple epochs. This iterative optimization enables the model to generalize well on unseen data.

3.2 CNN (Convolutional Neural Network)

3.2.1 Architecture of a Traditional CNN

Convolutional neural networks, also known as CNNs, are a specific type of neural networks that are generally composed of the following layers:

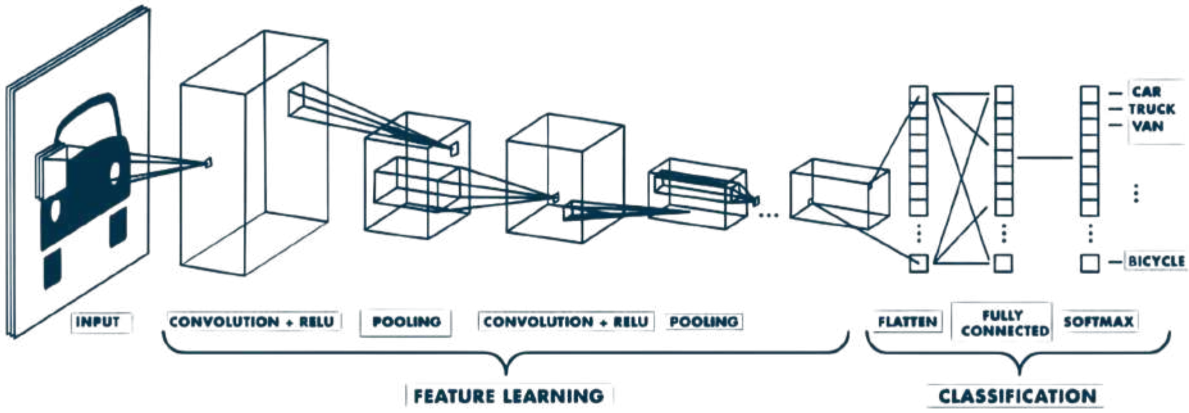


Figure 20: Example of a network with many convolutional layers. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer.

3.2.2 Types of Layer

- **Convolution Layer (CONV):** The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size F and stride S. The resulting output O is called feature map or activation map.

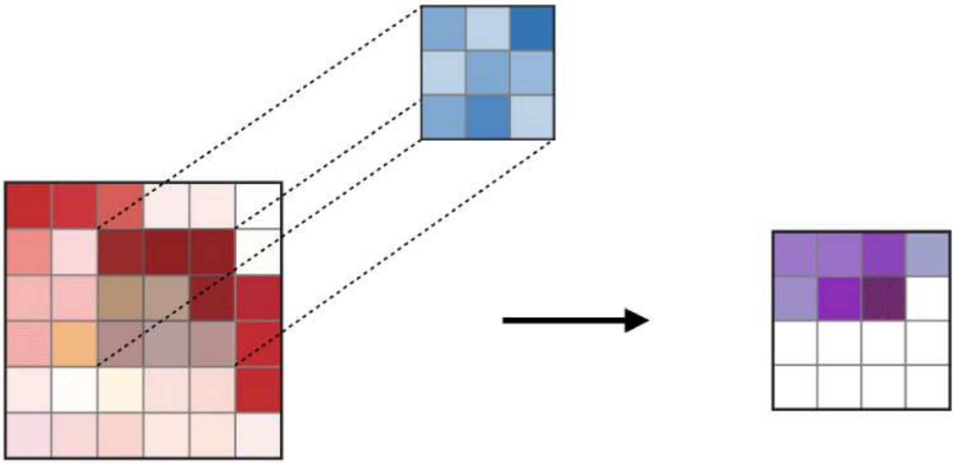
  

  Figure 21: Convolution Layer (CONV)

  The Mechanism of Convolutional Layers in Deep Learning Architectures

  A convolutional layer constitutes a fundamental component in contemporary deep learning models, particularly within the domain of image processing. The operational principle of this layer is predicated on the application of a filter, or kernel, which systematically traverses the input matrix—commonly an image—according to a predefined stride parameter. At each spatial location, the filter performs an element-wise multiplication with the corresponding segment of the input, followed by an aggregation of the resultant values, typically via summation. This computed value is then assigned to the respective position in the output feature map.

  The filter advances horizontally across the input matrix by the stride value, repeating the convolution operation at each step. Upon reaching the boundary of the input, the filter resets to the next row and continues the process until the entire input has been scanned. This systematic traversal enables the extraction of local features, such as edges or textures, which are subsequently encoded in the feature map.

  Through the hierarchical stacking of multiple convolutional layers, neural networks acquire the capacity to learn increasingly abstract and complex representations of the input data. This hierarchical feature extraction underpins the efficacy of convolutional neural networks in tasks involving spatial data, such as image classification and object detection.

- **Pooling (POOL):** The pooling layer (POOL) is a downsampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.

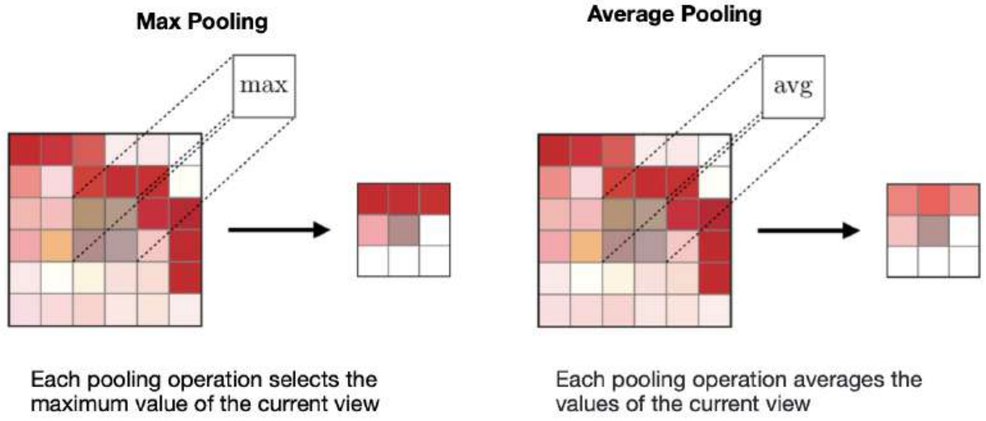

Max Pooling
max
Each pooling operation selects the maximum value of the current view
Average Pooling
avg
Each pooling operation averages the values of the current view

Figure 22: Pooling

- **Fully Connected(FC):** The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.

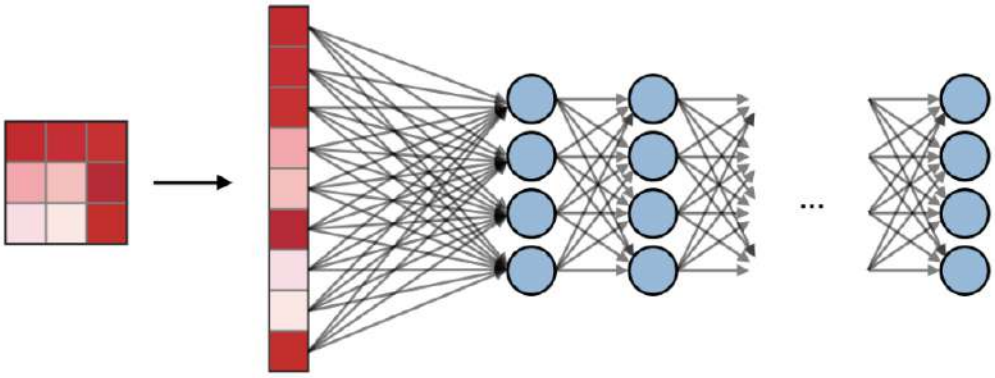


Figure 23: Fully Connected(FC)

3.2.3 Filter Hyperparameters

The convolution layer contains filters for which it is important to know the meaning behind its hyperparameters.

- Dimensions of a filter:A filter of size $F \times F$ applied to an input containing C channels is a $F \times F \times C$ volume that performs convolutions on an input of size $I \times I \times C$ and produces an output feature map (also called activation map) of size $O \times O \times 1$.

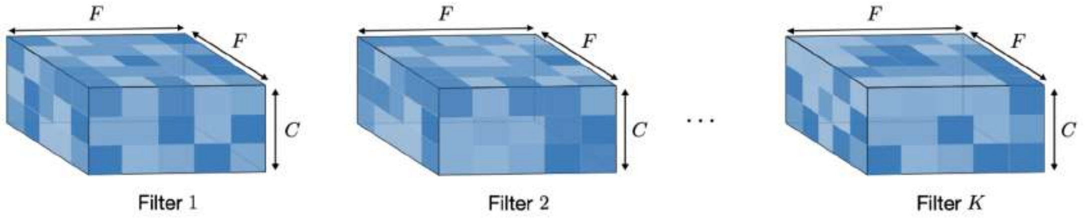


Figure 24: Dimensions of a filter

The application of K filters of size $F \times F$ results in an output feature map of size $O \times O \times K$.

- Stride: For a convolutional or a pooling operation, the stride S denotes the number of pixels by which the window moves after each operation.

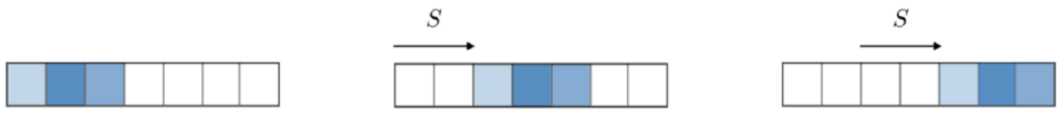
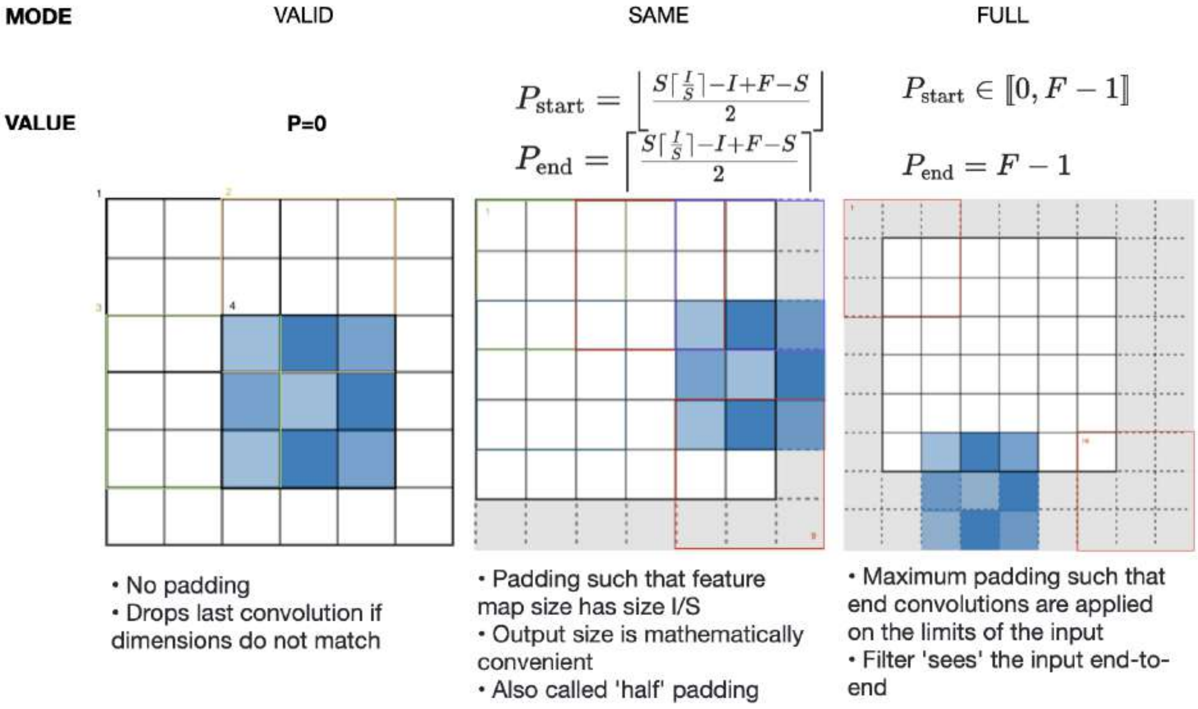


Figure 25: Stride

- Zero-Padding: Zero-padding denotes the process of adding P zeroes to each side of the boundaries of the input. This value can either be manually specified or automatically set through one of the three modes detailed below:



$$P_{\text{start}} = \left\lfloor \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rfloor$$

$$P_{\text{end}} = \left\lceil \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rceil$$

$$P_{\text{start}} \in [\![0, F-1]\!]$$

$$P_{\text{end}} = F - 1$$

- No padding
- Drops last convolution if dimensions do not match

- Padding such that feature map size has size I/S
- Output size is mathematically convenient
- Also called 'half' padding

- Maximum padding such that end convolutions are applied on the limits of the input
- Filter 'sees' the input end-to-end

Figure 26: Zero-Padding

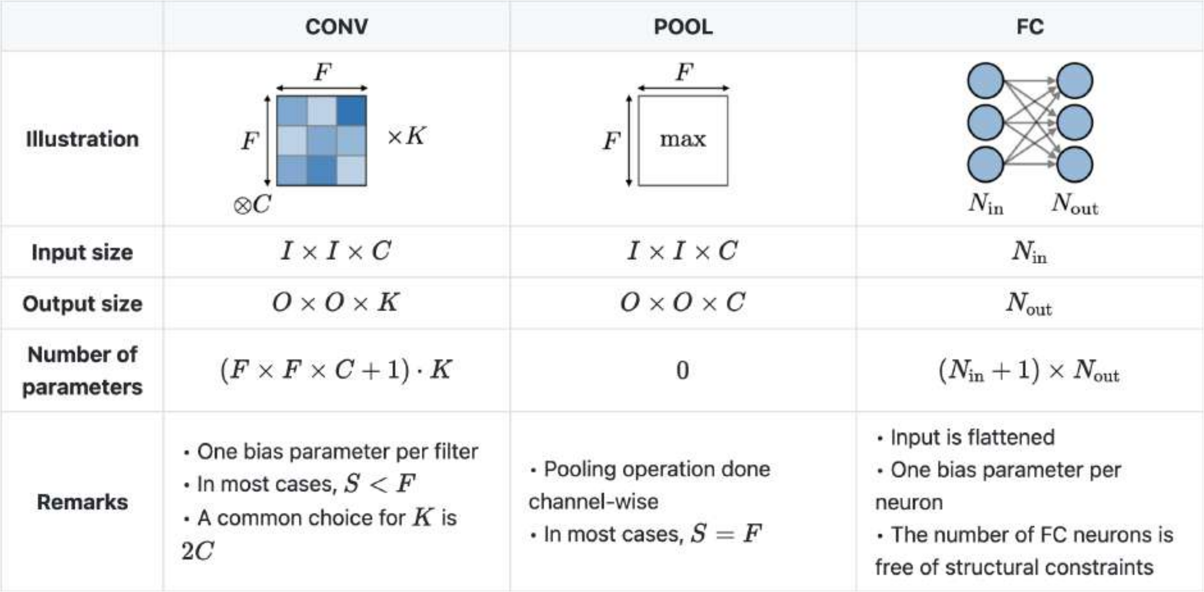
3.2.4 Tuning Hyperparameters

Parameter compatibility in convolution layer: By noting I the length of the input volume size, F the length of the filter, P the amount of zero padding, S the stride, then the output size O of the feature map along that dimension is given by:

$$O = \frac{I - F + P_{start} + P_{end}}{S} + 1 \tag{24}$$

often times, $P_{start} = P_{end} \triangleq P$ in which case we can replace $P_{start} + P_{end}$ by $2P$ in the formula above.

Understanding the complexity of the model: In order to assess the complexity of a model, it is often useful to determine the number of parameters that its architecture will have. In a given layer of a convolutional neural network, it is done as follows:

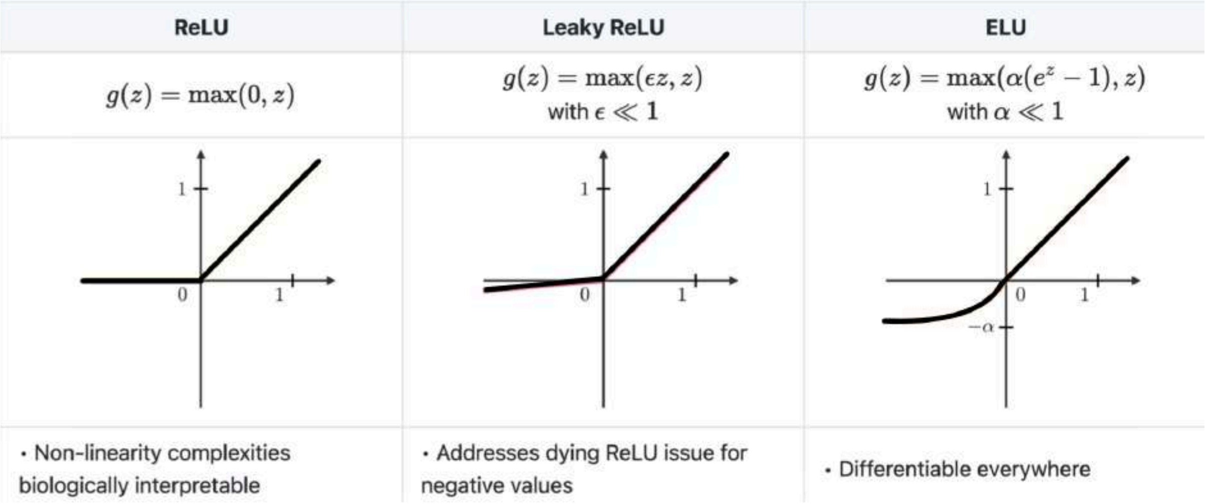


	CONV	POOL	FC
Illustration	F F $\times K$ $\otimes C$	F F max	N_{in} N_{out}
Input size	$I \times I \times C$	$I \times I \times C$	N_{in}
Output size	$O \times O \times K$	$O \times O \times C$	N_{out}
Number of parameters	$(F \times F \times C + 1) \cdot K$	0	$(N_{\text{in}} + 1) \times N_{\text{out}}$
Remarks	• One bias parameter per filter • In most cases, $S < F$ • A common choice for K is $2C$	• Pooling operation done channel-wise • In most cases, $S = F$	• Input is flattened • One bias parameter per neuron • The number of FC neurons is free of structural constraints

Figure 27: Understanding the complexity of the model

3.2.5 Commonly Used Activation Functions

- Rectified Linear Unit: The rectified linear unit layer (ReLU) is an activation function g that is used on all elements of the volume. It aims at introducing non-linearities to the network. Its variants are summarized in the Figure 24 below:



ReLU	Leaky ReLU	ELU
$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$	$g(z) = \max(\alpha(e^z - 1), z)$ with $\alpha \ll 1$
• Non-linearity complexities biologically interpretable	• Addresses dying ReLU issue for negative values	• Differentiable everywhere

Figure 28: Rectified Linear Unit Functions

- Softmax: The softmax step can be seen as a generalized logistic function that takes as input a vector of scores $x \in \mathbb{R}^n$ and outputs a vector of output probability $p \in \mathbb{R}^n$ through a softmax function at the end of the architecture. It is defined as follows:

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \tag{25}$$

$$p_i = \frac{e^{x_i}}{\sum\limits_{j=1}^{n} e^{x_j}} \tag{26}$$

3.3 One-Dimensional Convolutional Neural Network (1D-CNN)

Convolutional layers are a fundamental component of deep learning architectures, particularly in applications involving structured signal data. While two-dimensional convolutional layers have demonstrated exceptional performance in image processing tasks, one-dimensional convolutional layers (1D-CNNs) are specifically tailored for sequential data such as time series, audio waveforms, or text sequences.

A 1D convolutional layer performs convolution operations along a single spatial dimension. In this context, the input typically consists of a one-dimensional sequence—such as vibration signals, temperature readings, or current measurements acquired over time. The convolutional operation is carried out by a set of learnable kernels (filters), each designed to capture local features by sliding over the input sequence and computing the dot product at each position. The result is a feature map that encapsulates local dependencies and temporal patterns within the signal.

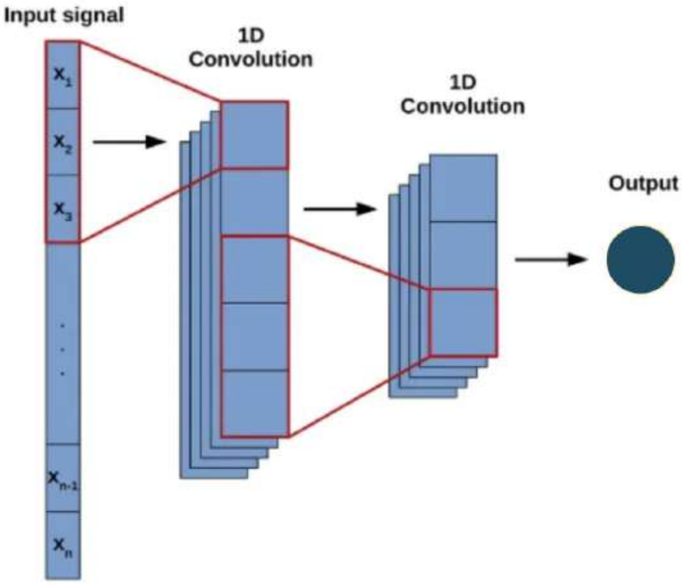


Figure 29: Illustration of a One-Dimensional Convolutional Neural Network (1D-CNN) Architecture for Sequential Data Processing

The key advantage of 1D convolutions lies in their ability to detect short-term structures, such as sudden fluctuations, peaks, or repetitive motifs, that are often indicative of system degradation or failure onset. These localized features are crucial in predictive maintenance scenarios, where early identification of anomalous behavior can prevent unexpected downtime.

In architecture, stacked 1D convolutional layers are employed, each followed by non-linear activation functions (ReLU), pooling operations, and normalization techniques. This hierarchical structure allows the network to learn increasingly abstract representations of the input signal, facilitating robust classification or regression in downstream tasks.

Typical applications of 1D-CNNs in the literature include anomaly detection in industrial sensor data, fault diagnosis in rotating machinery, and health monitoring in IoT environments. Furthermore, by maintaining a relatively low computational footprint compared to their 2D counterparts, 1D-CNNs are especially suited for real-time or embedded predictive maintenance systems.

3.4 Model Architecture and Training Framework

In this study, a deep learning-based approach was developed to solve a classification problem on one-dimensional sensor data structured in the form of time series. The proposed model architecture integrates multiple well-established components from deep learning literature to effectively capture local patterns, prevent overfitting, and enhance generalization capability.

3.4.1 Architectural Overview

The model follows a layered architecture composed of convolutional, normalization, pooling, and fully connected blocks, as detailed below:

- Initial Feature Extraction: The first layer of the model is a one-dimensional convolutional layer (Conv1D) with 64 filters and a kernel size of 3. This layer is designed to extract local temporal patterns from the raw signal. The use of `'same'` padding ensures that the output dimensions remain consistent with the input sequence length, preserving temporal resolution.

- Normalization and Regularization: To accelerate training and stabilize the learning process, batch normalization is applied immediately after the convolution. This is followed by a max pooling layer with a pool size of 2, which performs downsampling and reduces the computational load while retaining essential features. A dropout layer with a rate of 0.3 is also included to mitigate overfitting by randomly disabling a fraction of the neurons during training.

- Hierarchical Feature Learning: A second convolutional block with 128 filters is employed to enable the model to learn higher-order and more abstract representations. Similar to the previous block, batch normalization, max pooling, and dropout operations are applied in succession.

- Transition to Dense Layers: After the convolutional stages, the multi-dimensional output is flattened into a one-dimensional vector, which is then passed to a fully connected layer consisting of 128 neurons with ReLU activation. To further enhance generalization, a dropout layer with a higher rate (0.5) is included prior to the output layer.

- Output Layer: The final dense layer employs the softmax activation function to generate a probability distribution over the target classes. In this case, the output layer comprises four neurons, corresponding to the number of categories in the classification task.

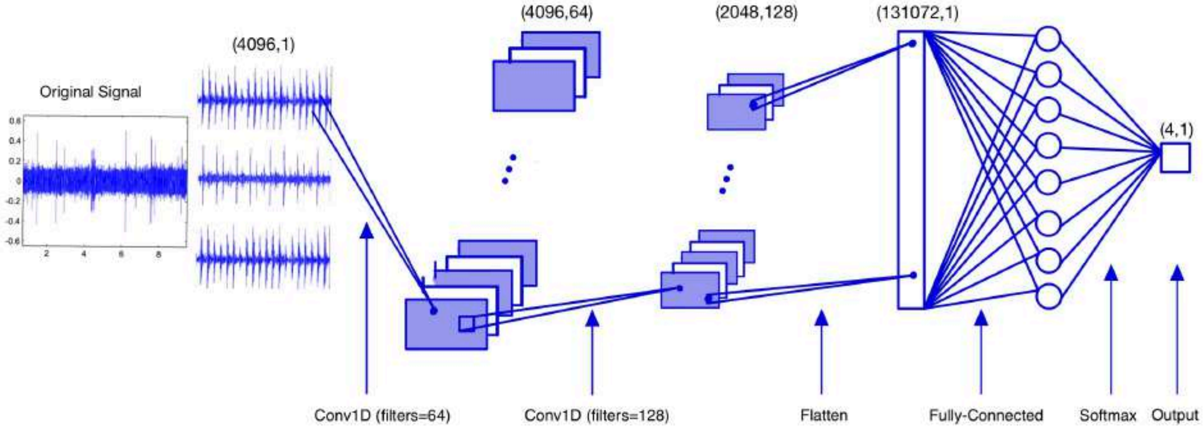


Figure 30: Model Architecture

3.4.2 Training Procedure

The model is compiled using the Adam optimizer, an adaptive gradient-based optimization algorithm known for its robustness and fast convergence in complex, high-dimensional spaces. The learning rate is set to 0.001, which was empirically determined to balance convergence speed and stability.

The categorical cross-entropy loss function is used, as it is well-suited for multi-class classification tasks with mutually exclusive class labels. The model was trained for 500 epochs with a batch size of 128, ensuring efficient gradient updates and sufficient exposure to diverse data subsets during training.

To monitor performance, the dataset is split into training and validation sets. At the end of each epoch, both training and validation metrics are recorded to evaluate convergence behavior and detect signs of overfitting.

3.4.3 Overfitting Prevention and Generalization

To enhance the model's ability to generalize to unseen data, several regularization strategies were employed:

- Dropout: Strategically placed dropout layers reduce reliance on specific neurons and promote robust feature representations.

- Batch Normalization: By normalizing activations at each batch, this technique mitigates internal covariate shift, leading to faster and more stable training.

- Validation Monitoring: The use of a validation set allows for continuous performance tracking and provides a mechanism for early stopping or model selection based on generalization performance.

3.4.4 Evaluation Metrics

Beyond standard accuracy, a comprehensive set of performance metrics was used to assess the model's classification effectiveness:

- Precision and Recall: These class-specific metrics are particularly informative in the presence of class imbalance.

- F1 Score: As a harmonic mean of precision and recall, the F1 score provides a balanced measure of classification quality.

- Confusion Matrix: Enables detailed analysis of misclassification patterns across different classes.

4 EXPLAINABLE ARTIFICIAL INTELLIGENCE

It is a set of techniques that produce more understandable models while maintaining high levels of performance, or provide external tools to better understand inherently uninterpretable models.

4.1 Importance of Explainability in Industrial Fault Diagnosis

In critical industrial applications such as predictive maintenance, interpretability is not a luxury, it is a necessity. While black-box models like deep neural networks may deliver superior predictive performance, their opaque internal mechanisms create a barrier for engineers who need to understand, verify, and act upon model outputs. Misclassifications or false alarms can result in costly downtimes, unnecessary part replacements, or, worse, unexpected failures.
Explainable Artificial Intelligence (XAI) methods offer a path forward by revealing how and why a model makes its predictions, allowing domain experts to:

- Validate model reasoning against known physics-based behaviors of machinery.
- Diagnose faults with confidence, supported by interpretable evidence.
- Integrate model outputs into existing maintenance workflows with clear thresholds or decision rules. In this study, we utilize two leading post-hoc explanation techniques—SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations)—to enhance the transparency of our deep learning-based fault diagnosis system.

4.2 SHAP (Shapley Additive Explanations)

SHAP is grounded in cooperative game theory, particularly the concept of Shapley values, which quantify the marginal contribution of each feature to the prediction by averaging over all possible feature subsets. Given a model $f(x)$ and a set of features $\{x_1, x_2, \cdots, x_n\}$ the SHAP value ϕ_i for feature x_i is defined as:

$$\phi_i(f) = \sum_{S \subseteq N/\{i\}} \frac{|S|!(n-|S|-1)!}{n!}[f(S \cup \{i\}) - f(S)] \tag{27}$$

Where:

S is a subset of the features not including i

$f(S)$ is the model's prediction when only features in S are known

n is the total number of features.

4.2.1 Underlying Assumptions of SHAP

SHAP operates under the premise that a complex model $f(x)$ can be approximated locally by a simpler, additive model $g(x')$, where x' represents a binary vector that indicates the presence or absence of each feature. The assumption is formalized as follows:

$$x \approx x' \rightarrow f(x) \approx g(x') \tag{28}$$

This implies that simplified feature inputs can sufficiently replicate the model's local behavior for explanation purposes.

4.2.2 Additive Feature Attribution Model

The SHAP explanation model is additive in nature, where the final prediction is decomposed into the sum of individual feature contributions:

$$g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i {x'}_i \tag{29}$$

Here:

ϕ_0 denotes the baseline prediction,
ϕ_i represents the attribution of the i^{th} feature,
n is the total number of features.

4.2.3 Axioms Guaranteed by SHAP

The SHAP framework is uniquely defined by three axiomatic properties, ensuring reliable and fair explanations:

- Missingness: If a feature is not present in the input, its contribution must be zero:

$$ {x'}_i = 0 \rightarrow \phi_i = 0 \tag{30}$$

- Local Accuracy: The sum of the baseline and individual feature contributions must match the original model prediction:

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i {x'}_i \tag{31}$$

- Consistency: If a model changes such that the marginal contribution of a feature increases (or stays the same), its Shapley value should not decrease.

4.3 Interpretable Local Explanations with LIME

Local Interpretable Model-agnostic Explanations (LIME) is a widely used method for interpreting the predictions of any black-box machine learning model. It addresses a core challenge in explainable AI: making complex model predictions understandable to humans. LIME achieves this by training a simple, interpretable model in the vicinity of the instance being explained, while preserving the fidelity of the black-box model's local behavior.
LIME does not depend on the internal structure of the original model. Instead, it perturbs the input space and observes the corresponding predictions, treating the model as a black box. This makes it especially valuable for cases where the underlying model is either too complex (e.g., deep neural networks) or inaccessible (e.g., proprietary systems). The objective of LIME is to approximate a complex function f, representing the original model, with a simpler function $g \in G$, chosen from a set of interpretable models (typically sparse linear models), such that g behaves similarly to f in the local neighborhood of a data point x. This is formulated as:

$$\xi(x) = \text{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{32}$$

Where:

$L(f, g, \pi_x)$ measures how well the simpler model g approximates f in the neighborhood of x,
π_x is a proximity measure that weighs instances based on their closeness to x,
$\Omega(g)$ is a regularization term that penalizes the complexity of g.

4.3.1 Loss Function

The local loss function L is defined as:

$$L(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z)(f(z) - g(z))^2 \tag{33}$$

Where:

Z is the set of perturbed instances around x,
$f(z)$ is the output of the black-box model for input z,
$g(z)$ is the prediction from the interpretable model,
$\pi_x(z)$ is the similarity kernel, which assigns weights to z based on its distance from x.

This weighted squared error encourages the surrogate model g to approximate f closely in the vicinity of x, without being concerned about the model's global behavior.

4.3.2 Step-by-Step Explanation Procedure

The process can be summarized as follows:

- Global View:
  - The entire data space is modeled by a complex non-linear function.
  - A specific instance (e.g., one with diabetes) is selected for explanation.
- Local Region Selection:
  - The instance is isolated, and its immediate neighborhood is defined.
- Perturbation and Weighting:
  - Synthetic data points are created by perturbing the original input.
  - These perturbed samples are weighted using a proximity kernel:

    $$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right) \tag{34}$$

    Where:

    $D(x,z)$ is a distance metric (e.g., cosine or Euclidean),
    σ controls the kernel width (how sharply the weights drop off with distance).
- Training a Local Model:
  - A sparse linear model is trained on the weighted dataset $\{z, f(z)\}$, with the goal of explaining the prediction $f(x)$.
  - The resulting coefficients of $g(z) = \omega_g \cdot z$ show the relative influence of each feature on the prediction.

4.4 Explainability in This Project

In this project, the Kernel SHAP method was utilized as an explainable AI technique. Kernel SHAP can be considered a hybrid approach that combines the strengths of LIME (Local Interpretable Model-Agnostic Explanations) and Shapley values derived from cooperative game theory. While maintaining model-agnostic properties similar to LIME, it leverages the theoretical foundation of Shapley values to provide fair and consistent attributions of each feature's contribution to the prediction. By evaluating all possible combinations of input features, Kernel SHAP offers a robust and mathematically grounded explanation mechanism. This makes it particularly suitable for interpreting complex black-box models without requiring access to their internal architecture, thereby enhancing model transparency and trustworthiness.

4.4.1 Kernel SHAP: A Model-Agnostic Approximation

Computing exact Shapley values becomes computationally infeasible as the number of features increases. To overcome this, Kernel SHAP provides an efficient, model-agnostic approximation. It estimates Shapley values by solving a weighted linear regression problem over sampled feature subsets.

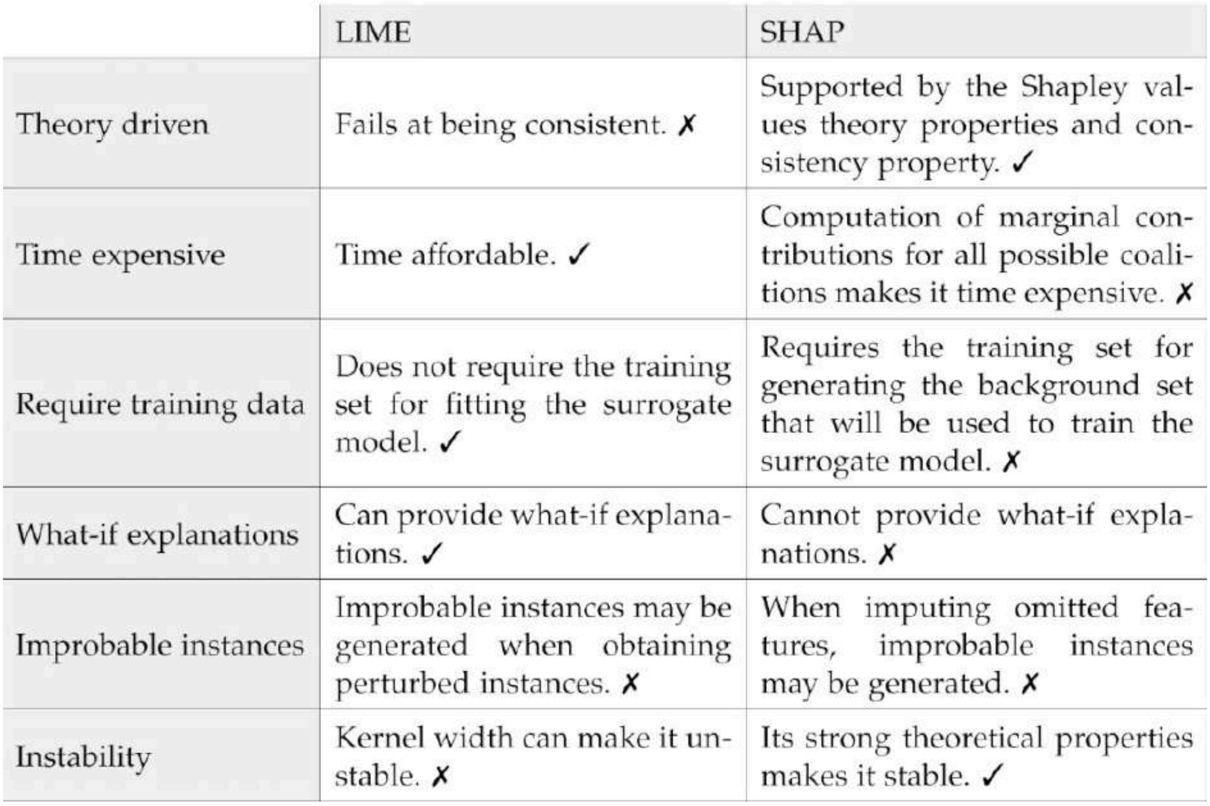
4.4.1.1 The Shapley Kernel Weighting Function : The weighting of each sampled feature subset during Kernel SHAP's regression step is determined by the following kernel function:

$$\pi_x(z') = \frac{M-1}{|z'|(M-|z'|)} \binom{M}{|z'|}^{-1} \tag{35}$$

This formulation ensures that subsets of different sizes are fairly represented, balancing the regression and improving the estimation of Shapley values across varying feature combinations.

4.4.2 Differences Between LIME and Shapley Values

To better illustrate the theoretical and practical differences between LIME and Shapley values, the following table summarizes their key characteristics. This comparison highlights how Kernel SHAP integrates the strengths of both methods to offer more reliable and interpretable explanations in model-agnostic settings.



	LIME	SHAP
Theory driven	Fails at being consistent. ✗	Supported by the Shapley values theory properties and consistency property. ✓
Time expensive	Time affordable. ✓	Computation of marginal contributions for all possible coalitions makes it time expensive. ✗
Require training data	Does not require the training set for fitting the surrogate model. ✓	Requires the training set for generating the background set that will be used to train the surrogate model. ✗
What-if explanations	Can provide what-if explanations. ✓	Cannot provide what-if explanations. ✗
Improbable instances	Improbable instances may be generated when obtaining perturbed instances. ✗	When imputing omitted features, improbable instances may be generated. ✗
Instability	Kernel width can make it unstable. ✗	Its strong theoretical properties makes it stable. ✓

Figure 31: Differences Between LIME and Shapley Values

4.4.3 Explainability Analysis Using Shapley Values on Frequency-Domain Features

In this study, SHAP (SHapley Additive exPlanations) analysis is employed to uncover the internal decision mechanisms of the one-dimensional convolutional neural network (1D-CNN) trained on frequency-domain representations of vibration signals. Specifically, the `KernelSHAP` algorithm is utilized due to its model-agnostic property and its capability to approximate Shapley values through a locally weighted linear regression approach, making it suitable for explaining black-box models such as deep neural networks.

The raw time-series signals are first transformed into the frequency domain via the Fast Fourier Transform (FFT). Only the frequency components within the 0–500 Hz range are retained, since this band typically includes the most diagnostically relevant information for bearing fault detection. The truncated frequency-domain representation serves as the *background dataset*, which is essential for SHAP value computation.

The `KernelExplainer` object is initialized with the syntax:

```
shap.KernelExplainer(lambda x: model.predict(x.reshape(-1, num_bins+1, 1)),
                     background.reshape(118, -1), link="identity")
```

Here, the prediction function is reshaped to fit the CNN's expected input dimensions. The background data, preprocessed and FFT-transformed, is also reshaped accordingly.

After initialization, SHAP values for a selected test sample are computed using:

```
explainer.shap_values(sample)
```

This function returns class-specific contribution scores for each frequency bin. These values quantify the extent to which individual frequency components contribute positively or negatively toward a particular classification outcome (e.g., 'outer race fault').

The extracted SHAP values are especially informative when visualized, as they reveal the specific spectral components that the model relies on for its decision-making. This form of interpretability not only increases confidence in the model's outputs but also provides actionable engineering insights into which frequencies are indicative of different fault types.

Furthermore, since FFT is an invertible transformation, the interpretability provided by SHAP remains traceable to the original time domain. This reversibility offers a critical advantage over non-invertible techniques such as envelope analysis, which have limited diagnostic transparency.

Overall, the combination of frequency-domain filtering and Kernel SHAP analysis offers a robust and explainable AI framework tailored for predictive maintenance applications in industrial settings.

5 MECHANICAL SYSTEM and DIGITAL SIGNAL PROCESSING

5.1 Mechanical System and Vibration Signal

Rotating machinery systems—such as electric motors, gearboxes, turbines, and especially rolling element bearings—are fundamental components of industrial environments. Among them, rolling bearings are used to reduce friction between rotating parts and to support radial and axial loads. However, due to continuous operation under high loads, misalignment, lubrication failure, or material fatigue, bearings are prone to developing defects over time. These defects can occur on the inner race, outer race, rolling elements, or cage, and they manifest physically as localized impacts or repetitive disturbances during rotation.

When a bearing defect forms, even at a microscopic level, it induces a cyclic impulse each time the damaged part comes into contact during rotation. These impulses excite the natural frequencies of the system, generating vibration signals that carry rich information about the mechanical health of the equipment.

To capture these signals, accelerometers (vibration sensors) are typically mounted on the bearing housing. These sensors produce analog signals that reflect the time-domain behavior of the mechanical vibrations. However, these raw signals are often contaminated with noise, and the fault-related information is usually hidden within modulated high-frequency components.

In mathematical terms, the measured signal $x(t)$ can be modeled as a combination of several physical phenomena:

$$x(t) = s(t) + n(t) \tag{36}$$

- $s(t)$: the deterministic part related to fault-induced vibrations,
- $n(t)$: random noise and system disturbances.

Since fault signals are often non-stationary and may exhibit time-varying frequency content, time-domain analysis alone is insufficient. Therefore, digital signal processing (DSP) techniques are employed to extract meaningful information from these signals.

Moreover, impulsive components in the signal, caused by bearing faults, usually manifest as high-frequency bursts superimposed on a low-frequency background. These can be better characterized in the frequency domain through spectral analysis, which requires transformation techniques such as FFT.

Additionally, resonance frequencies excited by faults depend on the mechanical properties of the system, including:

- Mass and stiffness of machine elements,
- Mounting conditions of the bearings,

- Structural damping characteristics.

Hence, a deep understanding of the mechanical resonance modes and dynamic behavior of the machine is essential to accurately interpret the vibration signals and isolate fault signatures.

5.2 Time Domain Terms

- Sampling Rate (F_s):Number of data samples acquired per second
- Frame Size (T):Amount of time data collected to perform a Fourier transform
- Block Size (N):Total number of data samples acquired during one frame

Frequency: Sampling rate (sometimes called sampling frequency or F_s) is the number of data points acquired per second. A sampling rate of 2000 samples/second means that 2000 discrete data points are acquired every second. This can be referred to as 2000 Hertz sample frequency. The sampling rate is important for determining the maximum amplitude and correct waveform of the signal as shown in Figure 28.

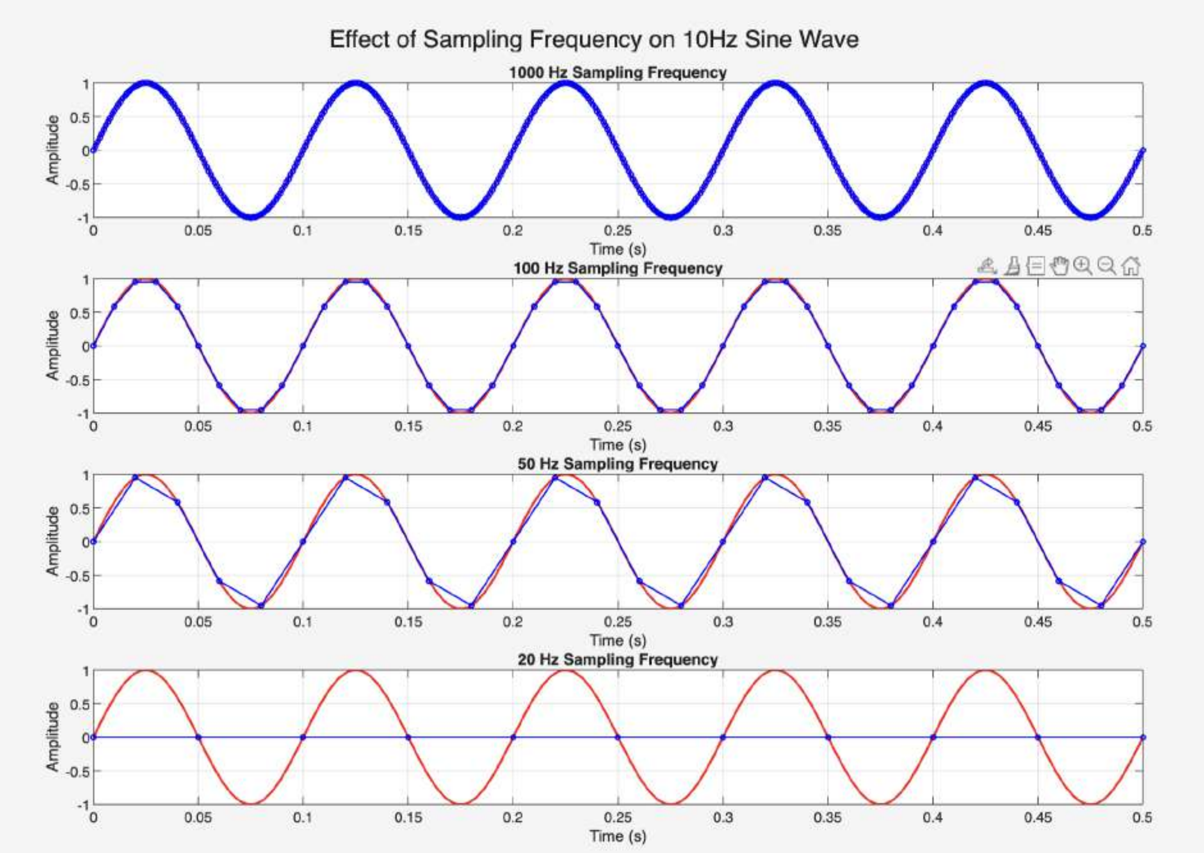


Figure 32: Effect of Sampling Frequency on 10Hz Sine Wave

To get close to the correct peak amplitude in the time domain, it is important to sample at least 10 times faster than the highest frequency of interest. For a 100 Hertz sine wave, the minimum sampling rate would be 1000 samples per second. In practice, sampling even higher than 10x helps measure the amplitude correctly in the time domain.

The inverse of sampling frequency (F_s) is the sampling interval or Δt. It is the amount of time between data samples collected in the time domain.

$$\frac{1}{\Delta t} = F_s \tag{37}$$

Block Size (N): The block size (N) is the total number of time data points that are captured to perform a Fourier transform. A block size of 2000 means that two thousand data points are acquired, then a Fourier transform is performed.

Frame Size (T): The frame size is the total time (T) to acquire one block of data. The frame size is the block size divided by sample frequency.

$$T = \frac{N}{F_s} = N \cdot \Delta t \tag{38}$$

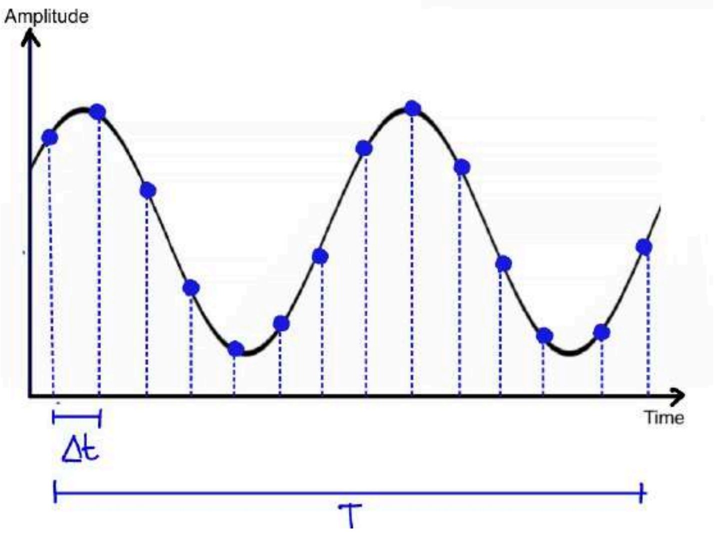


Figure 33: Time domain terms used in performing a digital Fourier transform

5.3 Frequency Domain Terms

- Bandwidth F_{max}: Highest frequency that is captured in the Fourier transform, equal to half the sampling rate
- Spectral Lines(SL): After Fourier transform, total number of frequency domain samples
- Frequency Resolution(Δf): Spacing between samples in the frequency domain

Bandwidth F_{max}:

$$F_{max} = \frac{1}{2} \cdot F_s \tag{39}$$

Spectral Lines (SL): After performing a Fourier transform, the spectral lines (SL) are the total number of frequency domain data points.

$$SpectiralLines = SL = \frac{1}{2} \cdot N \tag{40}$$

Frequency Resolution (Δ_f): The frequency resolution (Δ_f) is the spacing between data points in frequency. The frequency resolution equals the bandwidth divided by the spectral lines.

$$\Delta_f = \frac{Bandwidth}{SpectralLine} \tag{41}$$

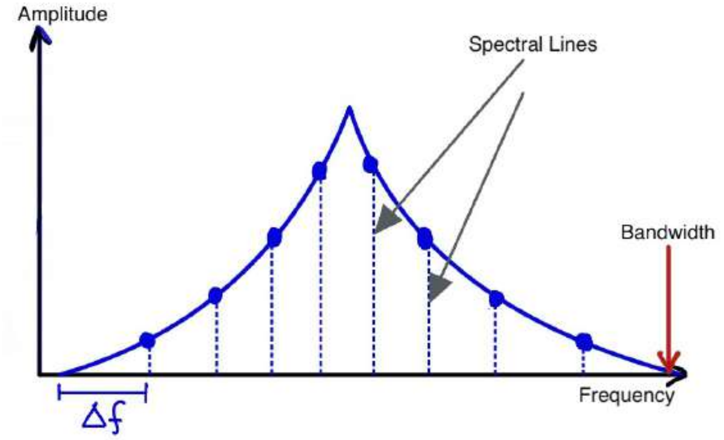


Figure 34: Frequency domain terms used in performing a digital Fourier transform

Digital Signal Processing Relationships Putting the above relationships together, the different digital signal processing parameters can be related to each other (Equation 7)

$$\frac{1}{T} = \frac{Bandwidth}{SpectralLine} = \frac{SamplingFrequency}{BlockSize} = \Delta_f \tag{42}$$

5.4 Aliasing

Aliasing is a phenomenon that occurs when converting an analog signal into a digital one, especially during the sampling process. If the sampling rate (the rate at which the analog signal is measured) is too low, it cannot accurately capture the higher frequencies in the signal.

As a result, high-frequency components of the signal appear as lower frequencies in the digitized version. This misleading representation is called an alias, which is why the effect is named aliasing. It creates distortion in the frequency spectrum and can lead to inaccurate interpretations of the signal.

To avoid aliasing, we must follow the Nyquist theorem, which states that the sampling rate should be at least twice the highest frequency component in the analog signal.

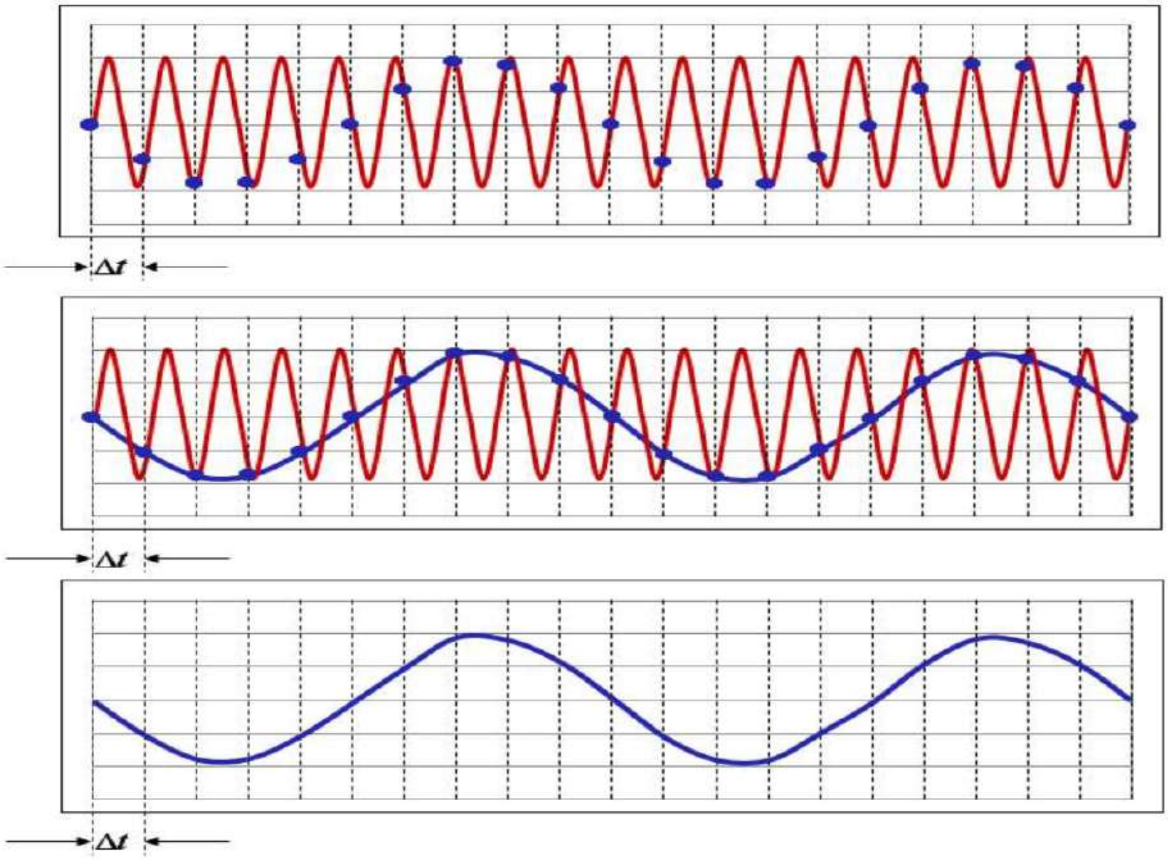


Figure 35: The red sine wave is the original signal. The blue dots represent how often the signal is being sampled. MIDDLE: The blue line is how the signal will appear due to the low sampling rate. BOTTOM: What the user will see in the time domain. Notice the acquired frequency is much lower than the actual frequency.[11]

5.5 Fourier Series

A Fourier series is an expansion of a periodic function into a sum of trigonometric functions. The Fourier series is an example of a trigonometric series. By expressing a function as a sum of sines and cosines, many problems involving the function become easier to analyze because trigonometric functions are well understood.

A Fourier series can be written in several equivalent forms, shown here as the N^{th} partial sums $S_N(x)$ of the Fourier series of $S(x)$:

Sine-Cosine Form:

$$S_N(x) = a_0 + \sum_{n=1}^{N} \left(a_n \cos\left(2\pi \frac{n}{P} x\right) + b_n \sin\left(2\pi \frac{n}{P} x\right)\right) \tag{43}$$

Exponential Form:

$$S_N(x) = \sum_{n=-N}^{N} c_n e^{i2\pi \frac{n}{P} x} \tag{44}$$

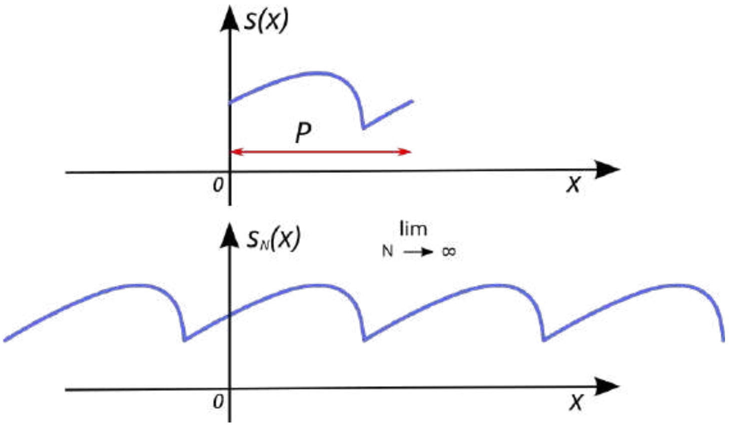


Figure 36: The top graph shows a non-periodic function $s(x)$ in blue defined only over the red interval from 0 to P. The function can be analyzed over this interval to produce the Fourier series in the bottom graph. The Fourier series is always a periodic function, even if original function $s(x)$is not.

Any periodic signal can be represented as a sum of sinusoids with different frequencies and amplitudes.

- The left side (Figure 33) shows a complex periodic signal.
- The right side (Figure 33) shows the individual frequency components (sine waves) that add up to form the original signal.

This principle is key in signal processing, enabling us to analyze and reconstruct signals based on their frequency content.

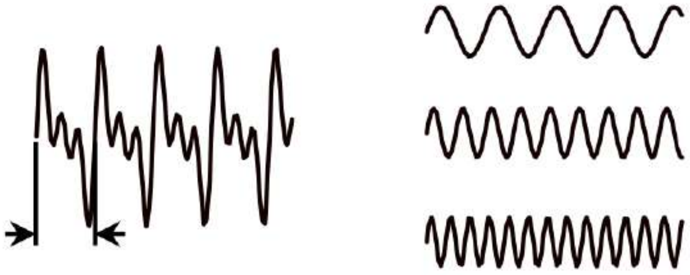


Figure 37: Complex waveform vs simpler sine and cosine waves

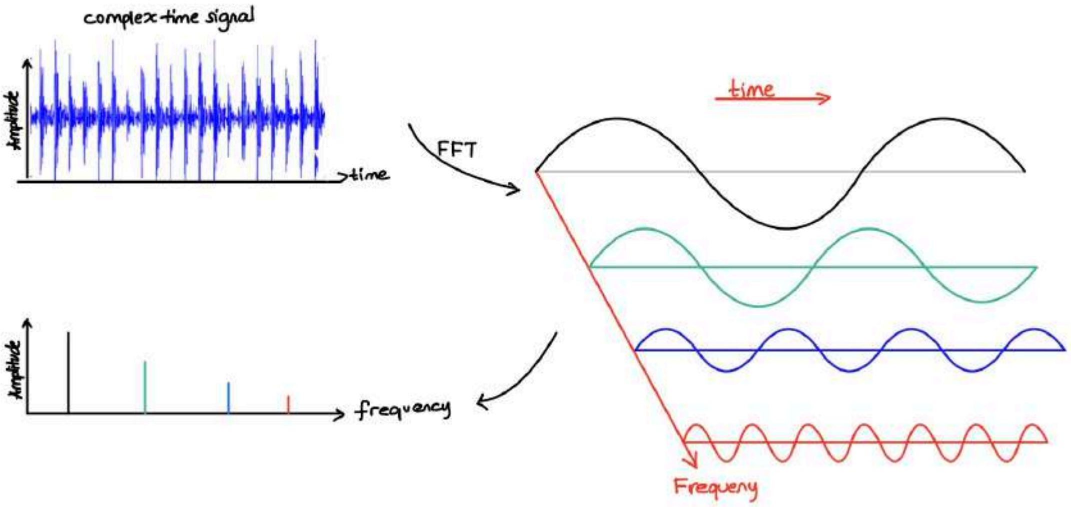


Figure 38: Conversion to frequency domain

5.5.1 Discrete Fourier Transform

In mathematics, the discrete Fourier transform (DFT) converts a finite sequence of equally-spaced samples of a function into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency.
The discrete Fourier transform transforms a sequence of N complex numbers $\{x_n\} := x_0, x_1, \cdots, x_{N-1}$ into another sequence of complex numbers $\{X_k\} := X_0, X_1, \cdots, X_{N-1}$, which is defined by:
For a real-valued signal $x \in \mathrm{R}^N$, the DFT coefficients are:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{k}{N} n} \tag{45}$$

Frequency Spectrum Symmetry in Real-Valued Signals and Single Sideband Representation

For real-valued signals, the frequency spectrum exhibits conjugate symmetry, which is mathematically expressed as:

$$X(f) = X(-f)^* \tag{46}$$

This property results in an even magnitude spectrum and an odd phase spectrum, such that:

$$|X(f)| = |X(-f)|, arg(X(f)) = -arg(X(-f)) \tag{47}$$

Due to this symmetry, the spectrum is commonly referred to as a Double Sideband (DSB) spectrum. However, since all information about the signal is inherently present in only one side of the spectrum (either positive or negative frequencies), it is possible to represent a real-valued signal using only a Single Sideband (SSB).

Various methods exist to extract either the upper or lower sideband. One direct approach is half-band filtering, which allows for the isolation of one half of the spectrum, enabling more efficient signal representation and transmission. Such techniques are especially valuable in communication systems where bandwidth efficiency is critical.

One-Sided Magnitude Spectrum

The one-sided Discrete Fourier Transform (DFT) is a technique used to analyze the frequency components of a real-valued signal. Unlike the full DFT, which provides both positive and negative frequency components, the one-sided DFT focuses only on the positive frequencies. This is particularly useful for real signals where the negative frequencies are redundant due to symmetry.

- **Positive Frequencies :**The one-sided DFT retains only the positive frequency components, which are sufficient to represent the signal's frequency content for real-valued inputs.
- **Magnitude Calculation:** The magnitudes of the DFT coefficients are calculated using the formula:

$$|X[k]| = \sqrt{Re(X[k]^2 + Im(X[k]^2))} \tag{48}$$

By focusing on the positive frequencies, the one-sided DFT simplifies the analysis and interpretation of real signals, making it a valuable tool in various engineering and scientific applications.

Extract One-Sided Magnitudes

Retain the first $[N/2] + 1$ magnitudes:

$$\text{One-Sided Magnitudes} = \begin{cases} \{|X[0]|, |X[1]|, \ldots, |X[N/2]|\}, & \text{even } N, \\ \{|X[0]|, |X[1]|, \ldots, |X[\lfloor N/2 \rfloor]|\}, & \text{odd } N. \end{cases} \tag{49}$$

Scale Magnitudes (Except DC/Nyquist)

The one-sided DFT magnitude spectrum is computed as:

$$\text{Scaled Magnitudes [k]} = \begin{cases} |X[0]|, & k = 0, \\ 2|X[k]|, & k = 1, 2, \ldots, N/2, \\ |X[N/2]|, & k = N/2\text{(even N)}. \end{cases} \tag{50}$$

Why Scaling is Necessary?

1 **Energy Conservation**

The two-sided DFT includes energy from both k. For real signals:

$$|X[k]| = |X[N-k]|(\text{due to conjugate symmetry}). \tag{51}$$

Scaling by 2 accounts for the energy in the discarded symmetric component.

2 **Physical Interpretation**
The scaled one-sided magnitude spectrum reflects the total energy at each frequency:

$$\text{Amplitude of sinusoid at} k \text{Hz} = \frac{ScaledMagnitude[k]}{N} \tag{52}$$

Summary:

- **Input:**Real-valued time-domain signal $x \in \mathrm{R}^N$
- **Output:**Scaled one-sided magnitude spectrum $\in \mathrm{R}^{N/2+1}$
- **Applications:** Power spectral density (PSD), audio analysis, vibration monitoring.

Inverse Discrete Fourier Transform
DFT converts a signal from the time domain to the frequency domain. However, this transformation is not one-way. The Inverse DFT (IDFT) allows us to reconstruct the original time-domain signal from its frequency components. This reversibility is crucial for several reasons:

- It ensures lossless signal analysis and synthesis.
- It allows modifications made in the frequency domain (e.g., filtering or compression) to be applied back to the time-domain signal.
- It forms the foundation of many digital signal processing (DSP) applications.

In short, while the DFT is used to analyze the signal, the IDFT is used to reconstruct it—enabling complete two-way transformation between domains.
Inverse transform:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{j2\pi \frac{k}{N} n} \tag{53}$$

5.5.2 Short-Term Fourier Transform

The Short-Time Fourier Transform (STFT) (or short-term Fourier transform) is a powerful general-purpose tool for audio signal processing. It defines a particularly useful class of time-frequency distributions which specify complex amplitude versus time and frequency for any signal. We are primarily concerned here with tuning the STFT parameters for the following applications:

- Approximating the time-frequency analysis performed by the ear for purposes of spectral display.
- Measuring model parameters in a short-time spectrum.

Mathematical definiton of the STFT:

$$X_m(w) = \sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-jwn} \tag{54}$$

$$= DTFT_w(x \cdot SHIFT_{mR}(w)) \tag{55}$$

where:

$$\begin{aligned} x(n) &= \text{input signal at time } n \\ w(n) &= \text{length } M \text{ window function (e.g., Hamming)} \\ X_m(w) &= \text{DTFT of windowed data centered about time } mR \\ R &= \text{hop size, in samples, between successive DTFT's} \end{aligned}$$

STFT is invertible for real discrete-time signals under specific conditions.

Conditions for Invertibility

- Proper Windowing and Overlap
  - **Constant Overlap-Add (COLA)** Condition : The analysis window ($w[n]$) and overlap must satisfy the COLA condition, which ensures that the sum of overlapping windows equals a constant value. This is crucial for perfect reconstruction of the signal.

$$\sum_m w[n-mR] = \text{constant}, \forall n \tag{56}$$

  - Common COLA-Compliant Windows : Examples include the Hann window with 50% overlap and the Hamming window. These windows are designed to minimize spectral leakage and ensure smooth transitions between frames.
- No Lossy Downsampling :
  - Retention of Time-Frequency Bins : The STFT must retain all time-frequency bins without decimation.
  - Downsampling, such as reducing time frames or frequency resolution, can introduce aliasing, which disrupts the invertibility of the STFT.

- Real-Signal Symmetry :
  - Conjugate-Symmetric STFT Spectra : Real signals have conjugate-symmetric spectra in the STFT domain. During inversion, the redundant negative-frequency components are discarded by taking the real part of the reconstructed signal.

Inversion Methods

- Overlap-Add (OLA) :
  - Inverse Transform and Overlap-Add : Each STFT frame is inverse-transformed, and the results are overlap-added. This method requires COLA-compliant windows to ensure that the original signal is perfectly reconstructed.
- Weighted Overlap-Add (WOLA) :
  - Synthesis Window Application : A synthesis window ($v[n]$) is applied before overlap-add to minimize spectral leakage. The product ($w[n] \cdot v[n]$) must satisfy the COLA condition for effective reconstruction.

Why Invertibility Holds

- Preservation of Phase and Magnitude Information : The STFT preserves full phase and magnitude information in each time-frequency bin, which is essential for accurate signal reconstruction.
- Perfect Reconstruction with COLA-Compliant Windows : When using COLA-compliant windows, the original signal can be perfectly reconstructed, as the overlapping windows sum to a constant value, ensuring no loss of information.

$$x[n] = \frac{\text{Real}(\sum_m \text{ISTFT}\{X[m,k]\})}{\text{constant}} \tag{57}$$

By adhering to these conditions and methods, the STFT can be effectively inverted, allowing for accurate reconstruction of real discrete-time signals. This process is fundamental in applications such as audio processing, speech analysis, and other signal processing tasks where time-frequency representation is crucial.

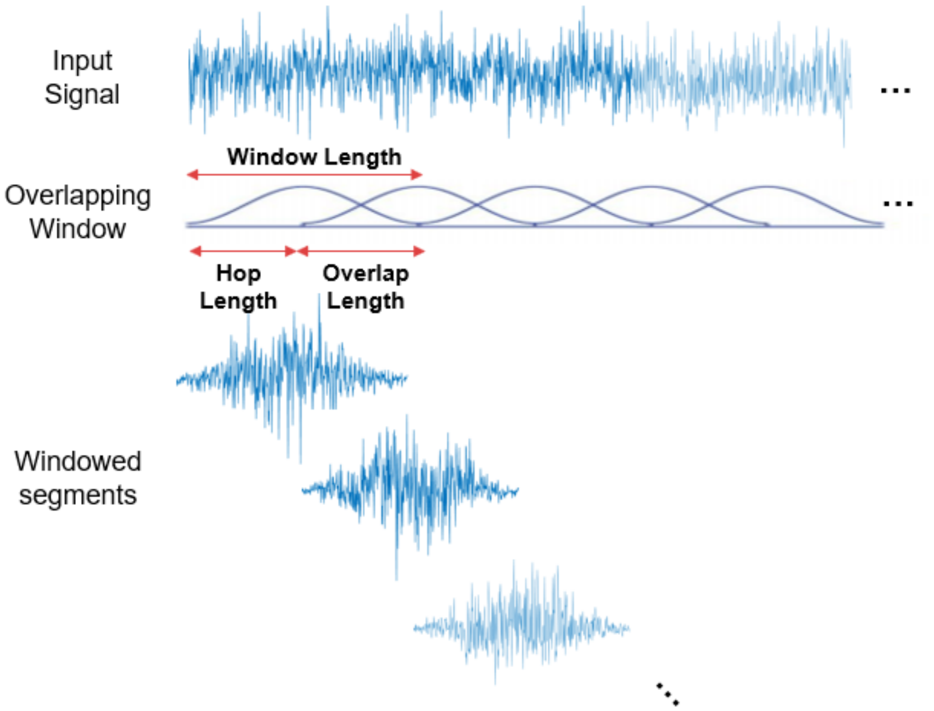


Figure 39: A random signal looks like in the original time-domain, after multiplying with the overlapping windows

5.5.3 Nyquist Frequency

In the context of Fast Fourier Transform (FFT), the Nyquist frequency plays a critical role in defining the upper frequency limit that can be reliably analyzed. It is equal to half of the sampling rate and serves as a boundary: frequency components above this limit cannot be distinguished correctly and will appear distorted due to aliasing. Therefore, to capture the true spectral content of a signal during FFT analysis, the sampling rate must be chosen to be at least twice the highest frequency of interest. This ensures that fault-related components—such as bearing defect frequencies—are preserved without misrepresentation.

5.6 Envelope Analysis

Envelope analysis is a signal processing technique widely employed in vibration-based condition monitoring, particularly for detecting localized defects in rolling element bearings. These defects typically generate impulsive excitations when rolling elements interact with the damaged surfaces. However, due to the system's resonant properties, these impulses are often buried within high-frequency signals, making them difficult to detect directly in the time or frequency domain.

Envelope analysis addresses this challenge by demodulating the vibration signal to extract the amplitude modulation pattern caused by repetitive mechanical impacts. This enables the identification of characteristic defect frequencies associated with the bearing components, such as the outer race, inner race, rolling elements, or cage.

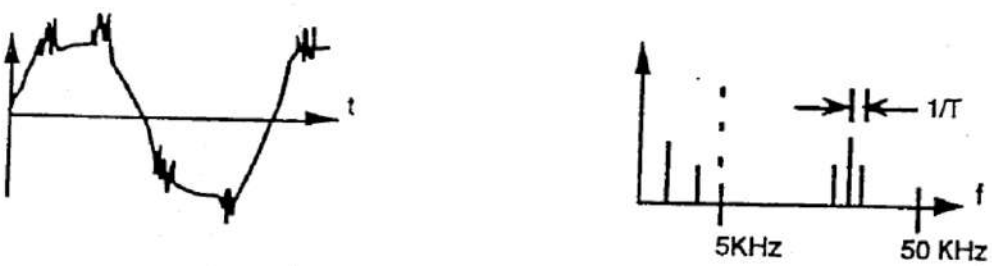


Figure 40: Time waveform vs frequency spectrum

This waveform in Figure 36 represents the raw time-domain vibration signal acquired from a rotating machine. It contains a mixture of periodic and random components, including structural resonances, mechanical noise, and potential fault-induced impacts. However, defect-related modulations are not clearly discernible in this raw signal due to the complex overlay of frequencies. This spectrum shows the frequency content of the original raw signal. While high-frequency resonances (e.g., around 50 kHz) are prominent, the actual fault frequencies—being much lower in amplitude and masked by other components—are typically not observable here.

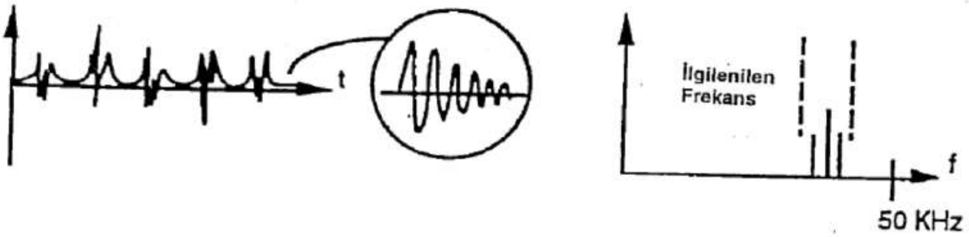


Figure 41: Filtered signal vs frequency of interest

The raw signal is band-pass filtered around a resonant frequency where the energy from bearing defects is most prominent. The filtering process isolates the high-frequency resonances that are excited by repeated impacts, enhancing the signal-to-noise ratio and making the modulated components more visible.

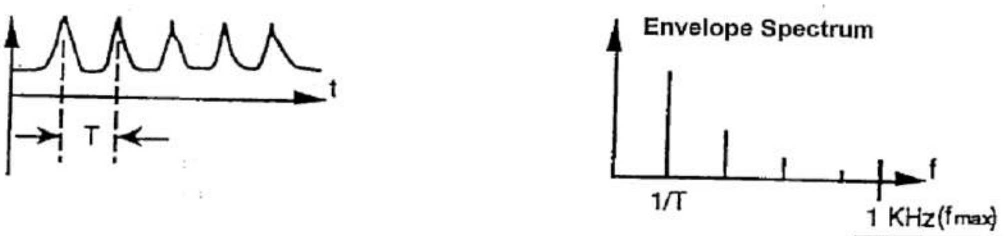


Figure 42: Demodulated signal vs envelope spectrum

The filtered signal is then subjected to an envelope extraction (typically using Hilbert transform or rectification followed by low-pass filtering). The resulting envelope signal reveals periodic amplitude variations corresponding to repeated mechanical events. The time interval T between successive peaks can be used to estimate the fault frequency $f = 1/T$

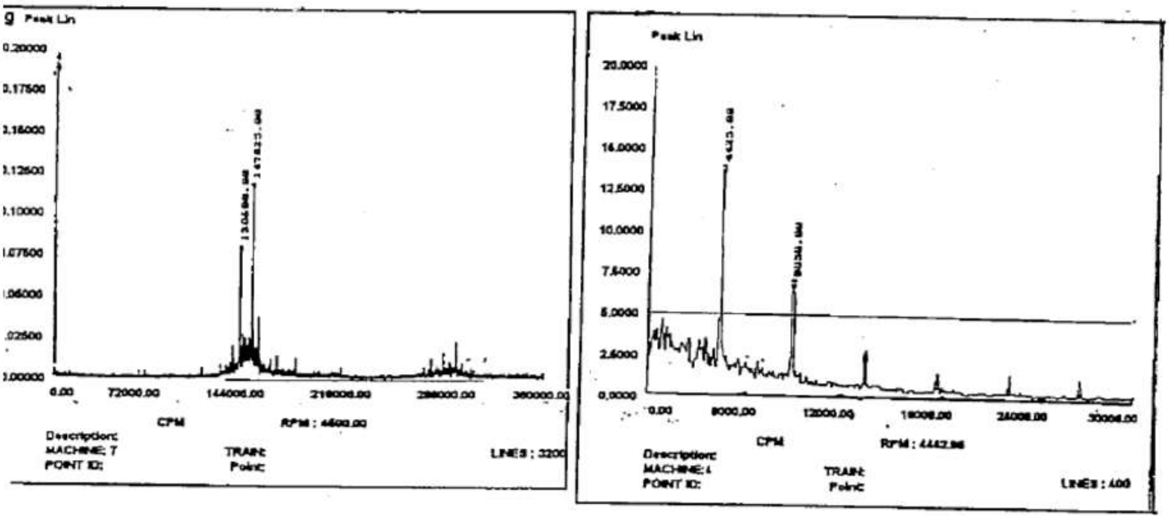


Figure 43: Spectral Examples from Real Measurements

- Left Plot: Displays the envelope spectrum in terms of CPM (Cycles Per Minute). A prominent peak around a characteristic fault frequency confirms the presence of a bearing defect. This spectrum corresponds to a machine running at 4600 RPM, using high-resolution analysis (3200 lines).
- Right Plot: Another envelope spectrum example showing peaks aligned with specific fault frequencies. The dominant frequency component likely corresponds to a bearing defect, providing critical insights for predictive maintenance.

Each plot includes measurement metadata such as machine ID, measurement point, resolution, and rotational speed, all of which are essential for correlating observed frequencies with theoretical defect frequencies.

5.6.1 Role of Modulation and Demodulation in Envelope Analysis

Envelope analysis is a widely used technique in condition monitoring for detecting faults in rolling element bearings. Localized defects on bearing components (e.g., inner race, outer race, ball, or cage) generate periodic impact forces as the elements rotate. These impacts excite structural resonances at high frequencies, resulting in a modulated vibration signal.

This vibration can be characterized as amplitude modulation (AM), where a high-frequency carrier signal (resonance) is modulated by a lower-frequency fault signature. The fault frequencies themselves are often masked in raw time or frequency data, making them difficult to identify directly.

To extract these hidden fault components, the signal undergoes demodulation—a process that removes the carrier and reveals the underlying modulation pattern. This is typically achieved using:

- Hilbert transform, to compute the analytic signal and obtain the envelope,
- or rectification followed by low-pass filtering.

The result is a demodulated signal containing the repetitive features of bearing faults. When a Fourier Transform is applied to this signal, the envelope spectrum is obtained, clearly showing characteristic frequencies such as BPFO, BPFI, BSF, and FTF. These correspond to specific fault locations on the bearing.

Envelope analysis, supported by proper modulation and demodulation methods, enables early fault detection and is essential for predictive maintenance in rotating machinery.

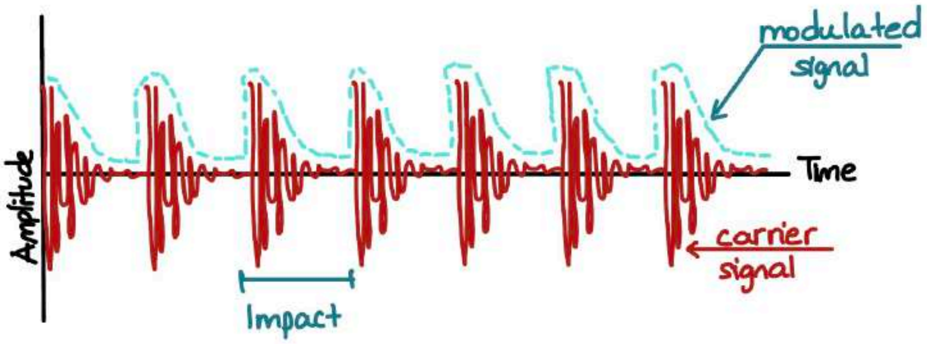


Figure 44: Waveform recorded on a defective bearing

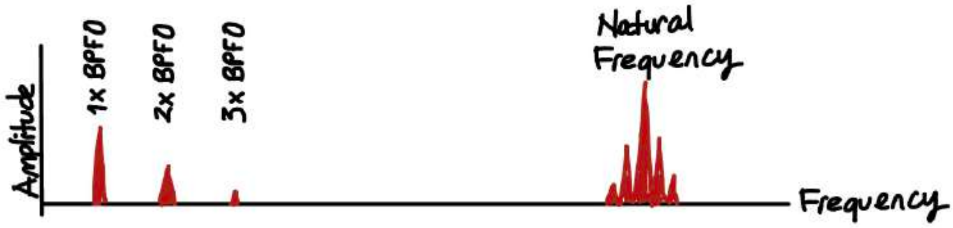


Figure 45: Waveform recorded on a defective bearing

After demodulation, the characteristic fault frequencies—such as BPFO, BPFI, and their harmonics—become clearly visible in the envelope spectrum. These low-frequency components are revealed after the removal of the high-frequency carrier, which typically corresponds to the system’s natural resonance. This process allows early detection of bearing defects that would otherwise be masked in raw vibration data. (Figure 41)

5.6.2 Is Envelope Spectrum Generation Invertible for Real Discrete-Time Signals?

Envelope spectrum generation is not an invertible transformation for real discrete-time signals. The process involves a series of nonlinear and lossy operations that prevent the original signal from being accurately reconstructed. **The key reasons for its non-invertibility are outlined below:**

- Loss of Phase Information:
  The envelope is obtained by calculating the magnitude of the analytic signal, typically using the Hilbert transform. This step discards the phase component of the original signal. Subsequent Fourier transformation of the envelope (i.e., producing the envelope spectrum) further removes any residual time-domain phase information, rendering reconstruction infeasible.

- Nonlinear Nature of Envelope Detection:
  Envelope extraction involves nonlinear operations specifically, magnitude computation. These operations eliminate key structural attributes of the original signal. Notably, different amplitude-modulated (AM) signals with varying carrier phases can yield the same envelope, introducing ambiguity.

- Irreversible Analytic Signal Compression:
  The analytic signal $x_a[n] = x[n] + jH\{x[n]\}$ becomes non-invertible when reduced to its magnitude $|x_a[n]|$. The Hilbert-transformed quadrature component $H\{x[n\}]$ is permanently lost in this process.

- Destruction of Spectral Symmetry:
  Real-valued signals possess conjugate symmetry in their Fourier spectra. However, the envelope spectrum emphasizes only the positive frequency components, effectively discarding the negative-frequency content that is essential for full reconstruction.

Practical Implications:
Envelope spectra are intended for diagnostic and analytical purposes—such as identifying fault-related frequency components in mechanical systems—not for signal reconstruction. While partial signal recovery might be possible under strict and uncommon assumptions (e.g., known carrier frequency and phase), such conditions are rarely applicable in practical scenarios.Envelope spectrum generation is fundamentally $non-invertible$, primarily due to the loss of phase information, spectral asymmetry, and the nonlinear nature of envelope extraction. As a result, the original real discrete-time signal cannot be recovered from its envelope spectrum.

5.6.3 Hilbert Transform in Envelope Analysis

The Hilbert Transform is a mathematical operation used to construct the analytic signal from a real-valued time-domain signal. It introduces a 90-degree phase shift to all frequency components, creating a complex-valued signal composed of the original signal (real part) and its Hilbert transform (imaginary part). This analytic signal enables the computation of the instantaneous amplitude, also known as the envelope, by taking its magnitude:

$$x_a(t) = x(t) + j \cdot \hat{x}(t) \Rightarrow \text{Envelope} = |x_a(t)| \tag{58}$$

Here $\hat{x}(t)$is the Hilbert Transform of $x(t)$. The envelope reflects amplitude modulation patterns in the signal and is widely used in fault detection applications, particularly for identifying bearing defects.

5.7 The Continuous Wavelet Transform

The Wavelet Transform is a widely used method for analyzing signals whose frequency content varies over time. Unlike classical Fourier-based techniques, which provide global frequency information, wavelet analysis captures both time and frequency characteristics simultaneously. This is achieved by decomposing the signal using a family of scaled and shifted versions of a chosen mother wavelet. As a result, the transform can zoom in on short-duration, high-frequency events as well as longer, low-frequency trends, making it highly suitable for non-stationary signal analysis.

In practical applications such as bearing fault detection, the Discrete Wavelet Transform (DWT) enables the identification of sudden impacts or irregularities by isolating signal components at various frequency bands. This makes WT a reliable tool for extracting diagnostic features from vibration data in rotating machinery. the continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} f(t)\psi(\text{scale}, \text{position}, t)\, dt \tag{59}$$

The results of the CWT are many wavelet coefficients C, which are a function of scale and position.

Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal:

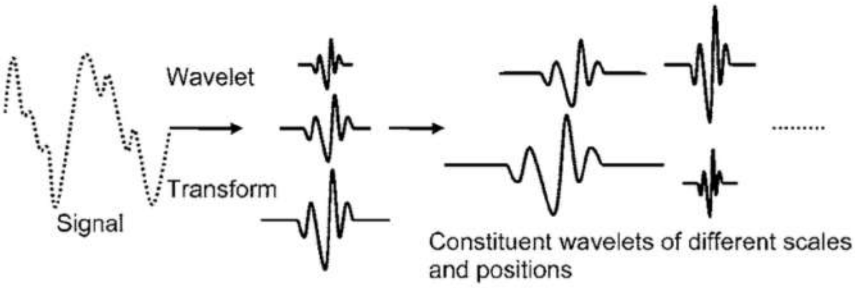


Figure 46: Continuous wavelet transfors of a signal

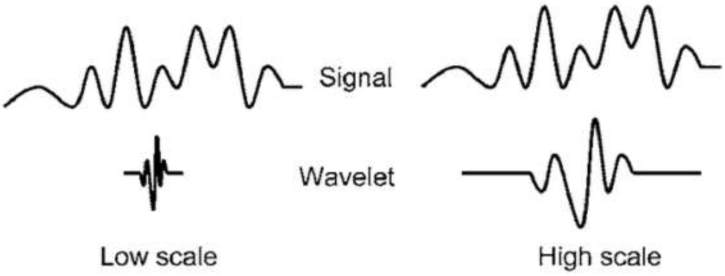


Figure 47: Low and high scales in wavelets

5.7.1 Scaling

Scaling changes the width of the wavelet: Stretching the wavelet (i.e., increasing the scale) allows it to capture low-frequency, long-duration components, while compressing it (i.e., decreasing the scale) targets high-frequency, short-duration features. This multiscale capability enables detection of both slow and rapid signal variations.

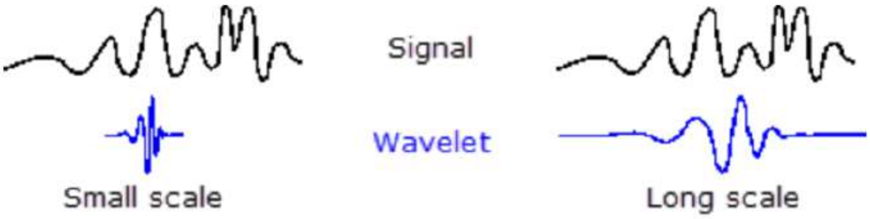


Figure 48: Scaling in wavelet transform

5.7.2 Shifting

Shifting refers to translating the wavelet along the time axis to inspect different portions of the signal. By moving the wavelet across the time domain, local features such as transients or sudden changes can be detected with high temporal precision.

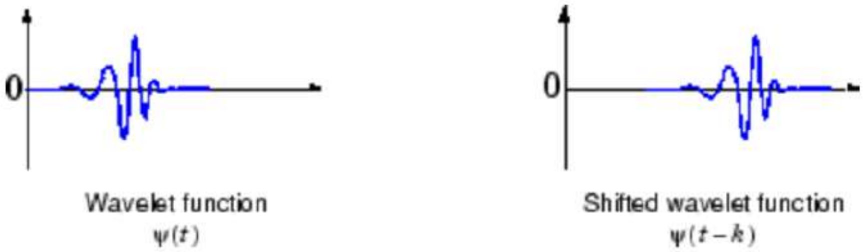


Figure 49: Shifting in wavelet transform

Scaling and shifting provide a time-frequency map of the signal, making the wavelet transform especially effective for analyzing non-stationary and transient phenomena.

5.8 Windows and Spectral Leakage

Spectral leakage is a problem that arises in the digital processing of signals. Leakage causes the signal levels to be reduced and redistributed over a broad frequency range, which must be addressed in order to analyze digital signals properly.

5.8.1 Leakage

Why does leakage effect the entire frequency range? It has to do with whether or not the signal is periodic relative to the measurement time frame.

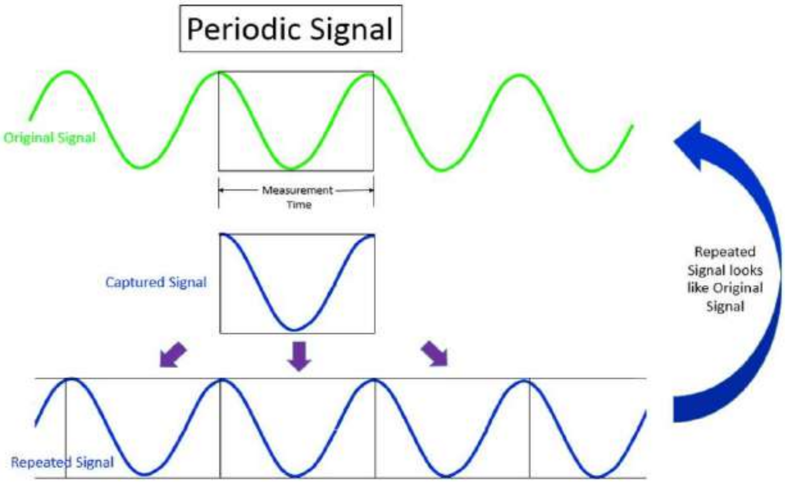


Figure 50: Periodic signal – Repeating and appending captured signal recreates original signal[11]

When measuring a sine wave, only a limited portion of the original waveform is captured, depending on the specified frame size used during acquisition. This portion is referred to as the captured signal, as illustrated in Figure 46.

Before performing frequency analysis, the captured signal is duplicated and concatenated multiple times. This process results in what is known as the repeated signal. If the repeated signal resembles the original sine wave, it indicates that the captured signal is periodic within the sampling window.

The rationale for repeating the captured signal lies in the mathematical foundation of the Fourier Transform (see Equation 60), upon which the Fast Fourier Transform (FFT) is based. The Fourier Transform assumes integration over an infinite time interval. However, in digital systems, signals are inherently finite in duration. To perform the FFT, it is assumed that the captured signal repeats periodically.

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi f t} dt \tag{60}$$

If the captured signal does not fulfill this periodicity assumption—meaning it does not start and end at the same point in its cycle—discontinuities arise at the boundaries when the signal is repeated. These discontinuities lead to a phenomenon known as spectral leakage, which distorts the frequency analysis. Therefore, ensuring periodicity of the signal within the acquisition window or applying appropriate windowing techniques is essential for accurate frequency-domain representation.

The Fourier Transform integrates over an infinite time interval, but digital systems can only capture finite-length signals. To address this, the FFT assumes that the captured signal repeats indefinitely. If the signal is periodic within the sampling window, this repetition introduces no discontinuities, and the resulting frequency spectrum is free from spectral leakage.

Just by changing the measurement time a small amount, the captured signal is not periodic, as shown in Figure 47.

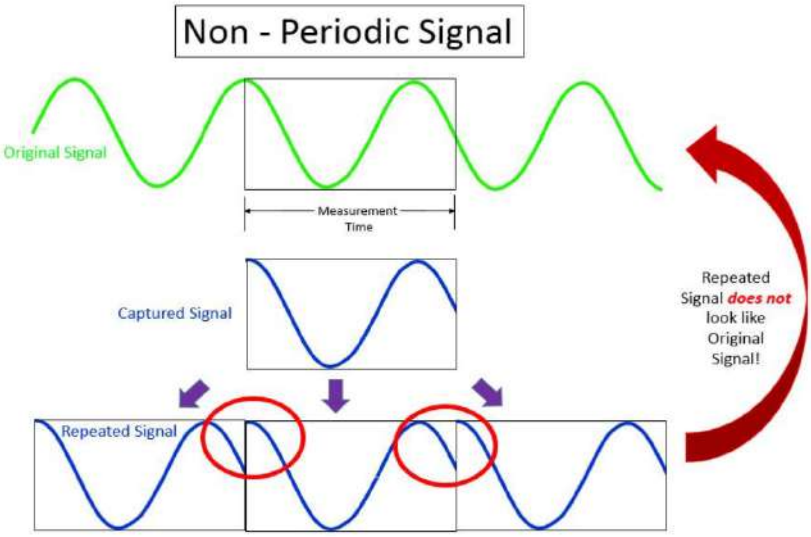


Figure 51: Example of a non-periodic measurement time[11]

There are sudden transitions at the end of each captured signal. These sharp transients, circled in red in Figure 47, have a broad frequency response. have a broad frequency response. Short transient signals in the timedomain produce high, broadband frequency content as shown in Figure 48.

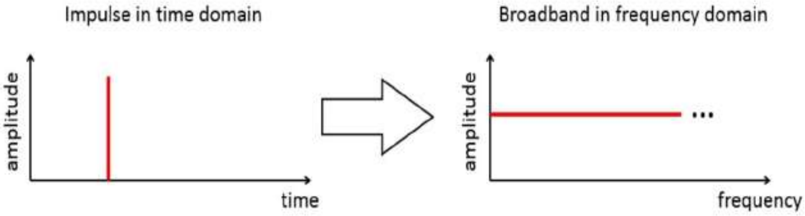


Figure 52: Sharp transient in time domain (left) has broad frequency response (right)[11]

So the resulting spectrum will have broadband response as well as a sinusoidal response as shown in Figure 49 in green.

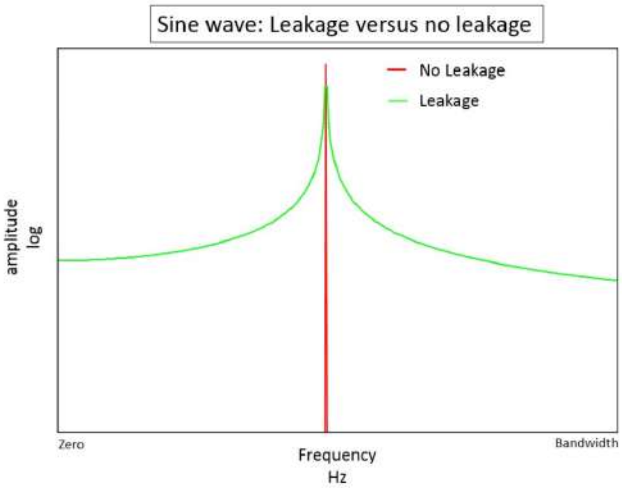


Figure 53: Sharp transients in repeated signal create leakage (green curve)[11]

5.8.2 Windows

To minimize spectral leakage, a mathematical function known as a window is applied to the captured signal. Window functions are specifically designed to reduce the abrupt transitions that occur at the boundaries of the repeated signal.

Typically, these functions begin at zero, gradually increase to one, and then taper back to zero within the duration of a single frame. The window is applied by multiplying it with the captured signal, as illustrated in Figure 50. This smooth tapering helps reduce discontinuities and improves the accuracy of the resulting frequency spectrum.

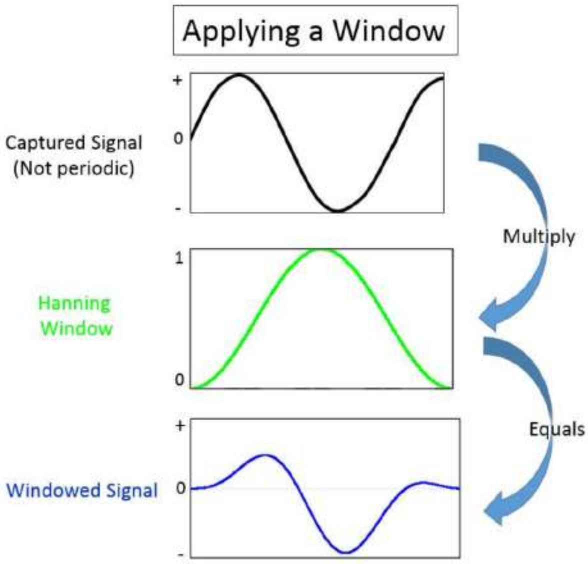


Figure 54: A signal (top) is multiplied by a window (middle) resulting in windowed signal (bottom)[11]

After windowing, the modified signal is repeated and concatenated, as illustrated in Figure 51. The window effectively eliminates the sharp transients at the boundaries—highlighted in gold—by smoothing the transitions. Although the repeated signal may no longer exactly match the original waveform, the reduction in discontinuities significantly minimizes spectral leakage.

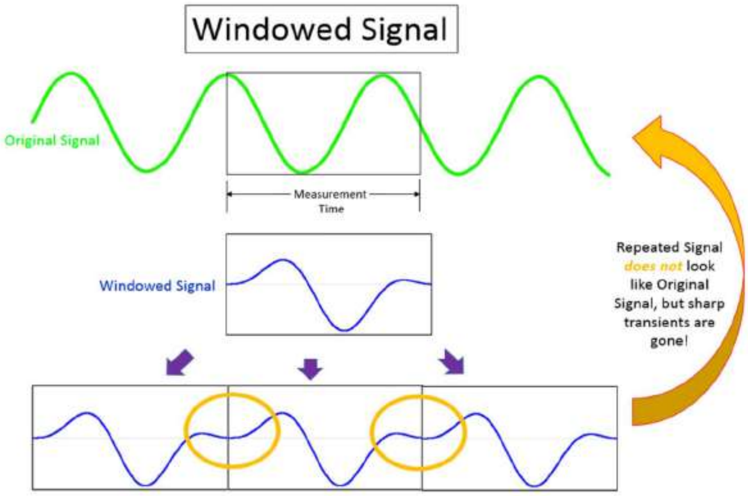

Windowed Signal
Original Signal
Measurement Time
Windowed Signal
Repeated Signal does not look like Original Signal, but sharp transients are gone!

Figure 55: After applying a window to the captured signal, the sharp transients are eliminated Because the sharp transients are reduced and smoothed, the broadband frequency of the spectral leakage is also reduced.[11]

Although windowing alters the shape of the original signal and the repeated version may no longer perfectly replicate it, the key advantage lies elsewhere. The primary benefit is the reduction of spectral leakage across the frequency spectrum. Windowing confines the leakage to a narrower frequency range, rather than allowing it to spread across the entire bandwidth. This improvement in spectral resolution is illustrated in Figure 52.

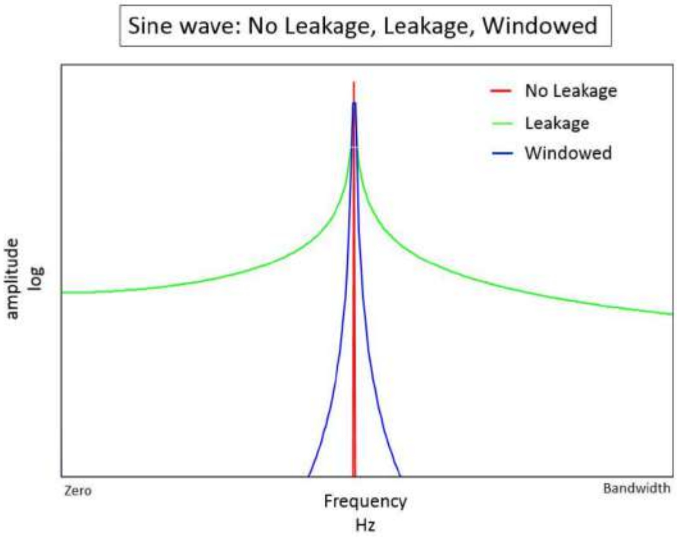


Figure 56: Periodic sine wave without leakage (red), non-periodic sine wave with leakage (green), and windowed non-periodic sine wave with reduced leakage (blue)

5.8.3 Window Types

There are many different windows, each optimized for a particular situation. Some windows:

- Hanning – Used for general data analysis, good tradeoff between frequency and amplitude accuracy
- Flattop – Excellent accuracy for amplitude, often used in calibration
- Tukey – Used for transient events
- Exponential – Used in impact hammer modal testing, be careful of adding artificial damping to measurement
- Uniform – Another way of saying “no window”
- Depending on the measurement situation, the appropriate window can be applied.

Spectral leakage is a common issue in digital signal processing, particularly when analyzing non-periodic signals using the Fourier Transform. It can be characterized by the following:

- Origin: Spectral leakage arises when the Fourier Transform is applied to signals that are not periodic within the sampled window.
- Effect: Leakage causes the energy of a single frequency component to spread across a wide range of frequencies, distorting the frequency spectrum.
- Mitigation: Applying a window function smooths the signal in the time domain, which reduces discontinuities at the window boundaries and helps confine leakage to a narrower frequency band. However, windowing cannot completely eliminate leakage.

6 PROPOSED METHODOLOGY and EXPERIMENTAL DESIGN

This study presents an innovative approach to improve the performance and explainability of 1D Convolutional Neural Networks (1D CNN) on bearing failure detection. The main objective of the project is to train the model directly with raw time series data and then analyze it with the SHAP method by applying Fourier Transform (FFT) to this data to improve explainability. Since the FFT is a lossless transform, it allows model decisions to be moved to a more intuitive domain (frequency space) for engineers.

In addition, this study proposes to add an additional layer directly to the trained model instead of the separate interpretive models usually used for the calculation of Shapley values. These two steps - training with raw data and directly annotating the model - are unique contributions of the project.

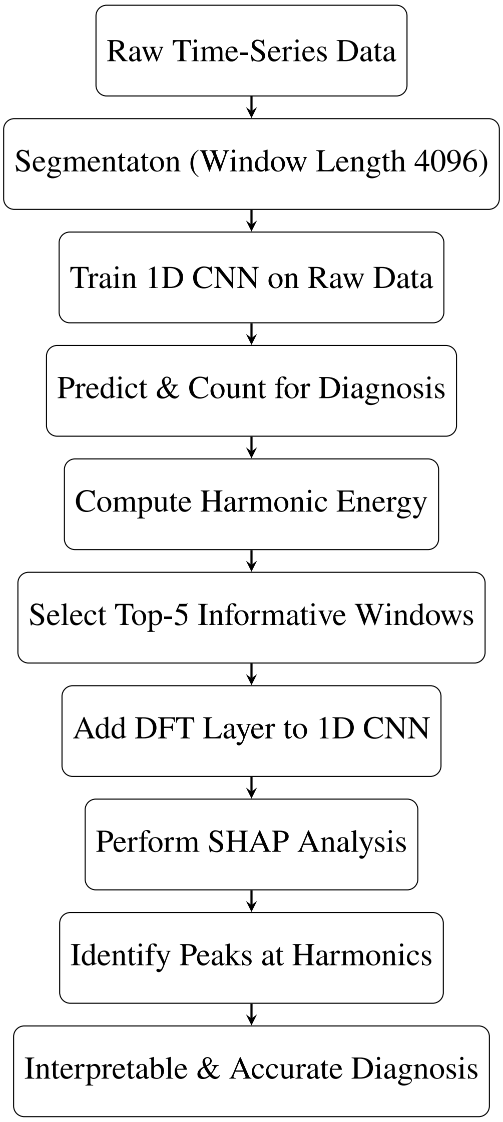

Raw Time-Series Data
Segmentaton (Window Length 4096)
Train 1D CNN on Raw Data
Predict & Count for Diagnosis
Compute Harmonic Energy
Select Top-5 Informative Windows
Add DFT Layer to 1D CNN
Perform SHAP Analysis
Identify Peaks at Harmonics
Interpretable & Accurate Diagnosis

Figure 57: Flowchart of the proposed diagnostic process.

6.1 Prediction Counts and Final Diagnosis

The use of a predict-and-count technique for reliable defect diagnosis is one of the study's main accomplishments. The model generates distinct predictions for each test set by processing it in fixed-length windows, or batches, rather than depending on a single forecast. The approach guarantees a more dependable and comprehensible classification decision by combining these predictions and determining the fault class that occurs most frequently. By enabling a more thorough examination of the most relevant signal segments that correspond to the dominant class, this method improves explainability and greatly boosts robustness against transient noise or outlier predictions. A vital connection between data-driven categorization and domain-specific interpretability is made possible by the predict-and-count methodology, which also makes it easier to choose diagnostically significant windows that are subsequently utilized in later harmonic energy analysis.

6.1.1 Evaluation of Raw-Signal CNN with Characteristic Fault Frequencies

To validate the classification capability of the raw-data-trained model, fault-specific samples containing known fault frequencies are fed into the network. For example, the test dataset `outer265racefault_testrawdata.csv` contains 118 samples, each consisting of 4096 time-domain measurements.
The diagnostic pipeline operates as follows:

- The test dataset is loaded using `pandas`.
- A previously saved `scaler` is loaded with `joblib` to normalize the raw input.
- The scaled data is reshaped to match the input format expected by the trained CNN model.
- The model produces a probability distribution across the classes for each sample.
- The `LabelEncoder` decodes predictions back into human-readable fault categories.
- Finally, a class-wise frequency count is computed, and a majority vote determines the overall diagnosis.

The following code was used to implement the test-time inference and voting procedure:

```
def predict_and_count(test_file_path, model, encoder, scaler_path):
    df = pd.read_csv(test_file_path)
    X = df.iloc[:, :4096].values
    scaler = joblib.load(scaler_path)
    X_scaled = scaler.transform(X)
    X_reshaped = X_scaled.reshape(-1, 4096)

    predictions = model.predict(X_reshaped)
    predicted_indices = np.argmax(predictions, axis=1)
```

```python
    predicted_labels = encoder.inverse_transform(predicted_indices)
    confidence = np.max(predictions, axis=1)

    class_counts = {cls: np.sum(predicted_labels == cls) for cls in encoder.classes_}
    majority_class = max(class_counts, key=class_counts.get)
    diagnosis = f"{majority_class} fault" if majority_class != 'healthy' else 'healthy'

    print("\nPrediction Counts:")
    for cls, count in class_counts.items():
        print(f"{cls}: {count}")

    print(f"\nFinal Diagnosis: {diagnosis}")

    return {
        "diagnosis": diagnosis,
        "majority_class": majority_class,
        "predicted_labels": predicted_labels,
        "confidence": confidence,
        "X_raw": X,
        "encoder": encoder,
    }
```

Listing 1: Predict and Count

6.1.2 Diagnosis Results and Interpretability Insights

When evaluated on the `outer265racefault_testrawdata.csv` file that model correctly identified the fault pattern as predominantly associated with the "Outer" class. The following output was recorded:

```
Prediction Counts:
Ball: 0
Healthy: 0
Inner: 0
Outer: 118

Final Diagnosis: OUTER fault
```

This result demonstrates the model's high sensitivity to characteristic fault features in raw time-domain data, without the need for manual frequency engineering.

6.2 Finding the Most Important Window for Explainability

The detection and diagnosis of faults in rotating machinery rely heavily on the identification of characteristic frequency components within vibration or velocity signals. One effective approach is to analyze the signal energy concentrated around harmonics of known fault frequencies. This document outlines a methodology for computing the Root Mean Square (RMS) energy of velocity signals centered on the harmonics of a fault frequency and subsequently identifying the most significant time windows for interpretability in fault diagnosis.

6.2.1 Harmonic RMS Energy Computation

The key objective is to quantify the signal energy around multiple harmonics of a fault characteristic frequency extracted from vibration data. The velocity signal batch is first transformed into the frequency domain using the Real Fast Fourier Transform (RFFT), which efficiently handles real-valued inputs and outputs the positive frequency spectrum.

Given the fault fundamental frequency f_f, the harmonics are defined as integer multiples nf_f for $n = 1, 2, \ldots, N$, where N is the number of harmonics analyzed. For each harmonic frequency, a narrow frequency band around its corresponding FFT bin is selected, and the energy is accumulated by summing the squared magnitude of FFT coefficients within this band. The final RMS value is computed by normalizing the total energy considering the FFT scaling.

```python
def compute_harmonic_rms(velocity_batch, fault_freq, sampling_freq=48000,
                         num_harmonics=6, sideband_bins=2):
    num_samples = len(velocity_batch)
    fft_result = np.fft.rfft(velocity_batch)
    freq_res = sampling_freq / num_samples
    total_energy = 0.0

    for harmonic in range(1, num_harmonics + 1):
        center_freq = harmonic * fault_freq
        if center_freq > sampling_freq / 2:
            continue
        bin_center = int(round(center_freq / freq_res))
        lower = max(bin_center - sideband_bins, 0)
        upper = min(bin_center + sideband_bins, len(fft_result)-1)

        for bin_idx in range(lower, upper + 1):
            magnitude = np.abs(fft_result[bin_idx])
            scale_factor = 1 if bin_idx == 0 else 2
            total_energy += (magnitude ** 2) * scale_factor

    return np.sqrt(total_energy) / num_samples
```

Listing 2: Function to compute RMS energy around fault frequency harmonics

6.2.1.1 Methodology:

- The scaling factor accounts for the symmetry in the FFT of real signals, doubling the energy contribution except at DC (0 Hz).
- The sideband parameter defines the number of frequency bins around each harmonic to include in the energy sum, providing robustness to frequency spread.
- Harmonics above the Nyquist frequency are omitted to avoid aliasing effects.

6.2.2 Identification of Important Signal Windows

After computing the harmonic RMS energy for multiple signal batches, the windows (or batches) exhibiting the highest energy corresponding to the fault frequency are selected for further analysis. This step enables focusing on the most diagnostically relevant segments of the data, thereby facilitating interpretability and explainability in fault diagnosis.

```python
def analyze_important_windows(results, X, shaft_speed_rpm=1761, top_n=5):
    majority_class = results["majority_class"]
    shaft_speed_hz = shaft_speed_rpm / 60
    fault_frequencies = {
        'Outer': 3.585 * shaft_speed_hz,
        'BPFI': 5.415 * shaft_speed_hz,
        'FTF': 0.3983 * shaft_speed_hz,
        'BSF': 2.357 * shaft_speed_hz
    }
    fault_freq = fault_frequencies.get(majority_class, 0)

    fault_mask = results["predicted_labels"] == majority_class
    batch_indices = np.where(fault_mask)[0]

    rms_values = []
    for idx in batch_indices:
        batch_data = X.iloc[idx].values
        rms = compute_harmonic_rms(batch_data, fault_freq)
        rms_values.append(rms)

    top_indices = np.argsort(rms_values)[-top_n:][::-1]
    top_batches = batch_indices[top_indices]

    return {
        "diagnosis": f"Detected␣{majority_class}␣fault",
        "fault_frequency": f"{fault_freq:.2f}␣Hz",
        "harmonics_analyzed": 5,
        "top_batches": pd.DataFrame({
            'Batch␣Index': top_batches,
            'RMS␣(mm/s)': np.array(rms_values)[top_indices],
            'Confidence': results["confidence"][top_batches]
        }),
        "average_rms": np.mean(rms_values)
    }
```

Listing 3: Function to analyze and select top diagnostic windows

The following output was recorded:

```
=== FAULT DIAGNOSIS ===
Detected Healty fault | Fundamental Frequency: 103.38 Hz

Top 5 Diagnostic Batches (3 harmonics):
   Batch Index  RMS (mm/s)  Confidence
0           32    0.218137         1.0
1           31    0.177803         1.0
2          103    0.175910         1.0
3           34    0.173253         1.0
4          104    0.170083         1.0
```

6.2.2.1 Methodology:

- The function first calculates the fault characteristic frequency based on the shaft speed and fault type.
- It filters the signal batches predicted as containing the fault and computes the harmonic RMS for each.
- The top N batches with the highest harmonic RMS values are identified, as these represent the most significant windows related to the fault.
- This subset facilitates explainability by isolating key instances from the large dataset.

The described method allows for the quantification of fault-related frequency components through harmonic RMS energy metrics and provides a systematic way to identify the most relevant data windows for interpretability. Such targeted analysis is critical in condition monitoring and predictive maintenance frameworks, enabling focused investigation and effective decision-making.

6.3 Special DFT Layer Added for Explainability

In this study, a dedicated *Discrete Fourier Transform (DFT)* layer is incorporated into the deep learning architecture to enable direct spectral processing and enhance the explainability of the model's decision-making process.

6.3.1 DFT Layer Design and Model Integration

The proposed approach performs the transformation from the time domain to the frequency domain within the model during training. The DFT layer implemented in TensorFlow Keras is as follows:

```
class DFTLayer(layers.Layer):
    def __init__(self, **kwargs):
        super(DFTLayer, self).__init__(**kwargs)

    def call(self, inputs):
        inputs_complex = tf.cast(tf.squeeze(inputs, axis=-1), tf.complex64)
        dft_output = tf.signal.fft(inputs_complex)
        return tf.stack([tf.math.real(dft_output), tf.math.imag(dft_output)], axis=-
```

This layer converts the input signal into complex format, applies Fourier transform, and returns the real and imaginary parts as separate channels. Thus, spectral features can be learned directly by subsequent deep learning layers.

The model architecture processes spectral data hierarchically through convolutional, normalization, and pooling layers following the DFT layer.

6.3.2 Spectral Preprocessing and SHAP Values

Prior to training and interpretability analysis, input signals are transformed from time domain to frequency domain using Fourier Transform. To reduce the influence of high-frequency components and improve analysis efficiency, the spectrum is truncated to a specific frequency cutoff (e.g., 500 Hz).

SHAP (SHapley Additive exPlanations) is utilized to quantify the contribution of each frequency component to the model's predictions. The inclusion of the DFT layer enables SHAP values to be calculated specifically for spectral components.

The KernelExplainer is defined as follows to explain model predictions based on frequency components:

```
explainer = shap.KernelExplainer(
    lambda x: model_DFT.predict(x.reshape(-1, num_bins+1,1)),
    background.reshape(num_background_samples, -1),
    link="identity"
)
```

Note:

In summary, integrating a dedicated DFT layer that transforms time domain signals to frequency domain within the model enables direct spectral feature learning. This facilitates SHAP-based interpretability that reveals which frequency bands the model relies upon for its decisions, significantly improving explainability. This approach contributes notably to increasing the trustworthiness and interpretability of deep learning models in fault diagnosis applications.

7 RESULTS and DISCUSSION

Table 6: Accuracy performance of the 1DCNN model trained with Raw and FFT-transformed data

Model	**Accuracy Metric**	**Raw Data** (%)	**FFT-Transformed Data** (%)
1DCNN	Training Accuracy	99.59	99.36
	Validation Accuracy	99.69	99.63
	Testing Accuracy	96.14	82.22

Table 6 summarizes the classification performance of a one-dimensional convolutional neural network (1DCNN) model trained on two different data formats: raw time-domain signals and signals processed via the Fast Fourier Transform (FFT).

The results demonstrate that the model trained with raw data achieves high accuracy across all stages—training, validation, and testing.

In contrast, while the FFT-transformed dataset maintains high training and validation accuracy, the testing accuracy drops significantly to 82.22%. This decline is expected and can be attributed to the loss of phase information during the FFT process. Phase plays a critical role in time-series signal interpretation, especially in transient and non-stationary behaviors, which are often crucial for accurate classification.

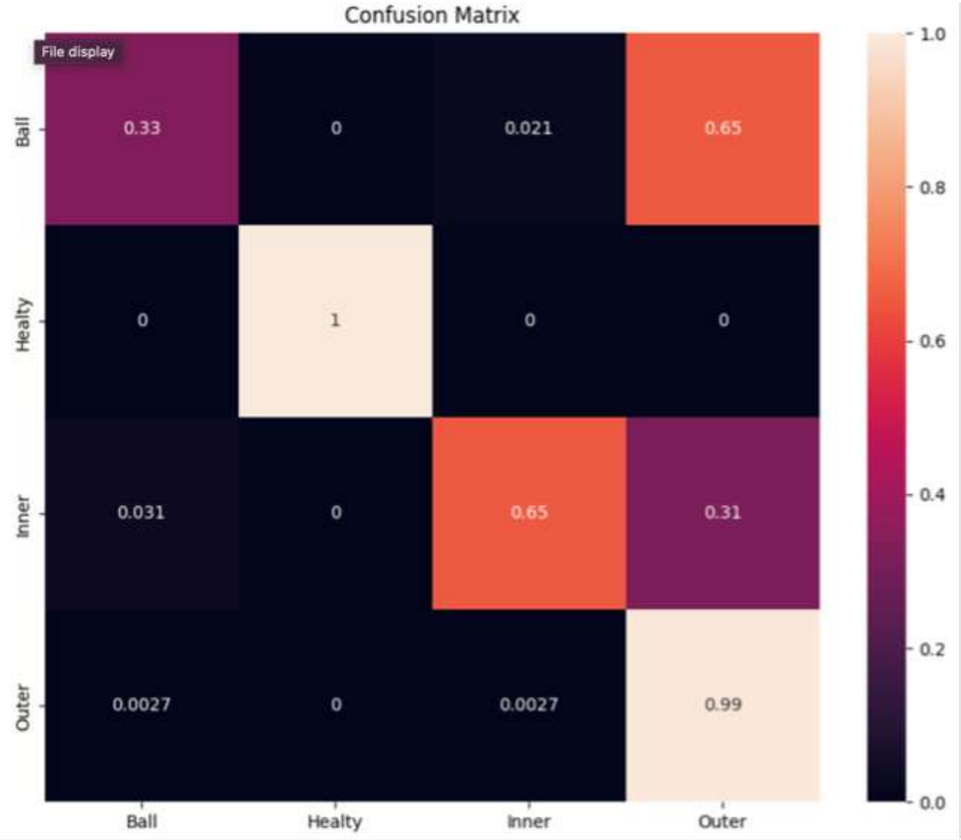


Figure 58: Normalized confusion matrix of the 1DCNN model trained on frequency domain.

As shown in Figure 58, the 1DCNN model trained on the frequency domain does not perform well for all fault types. This is expected.

As such, although frequency-domain features can help emphasize certain periodic characteristics, their use alone—particularly without phase context—may hinder the model's ability to generalize effectively. This comparison highlights the importance of selecting appropriate data representations based on the nature of the learning task.

As shown in Figure 59, the 1DCNN model trained on raw data performs well across all fault types. The model perfectly classifies the `Healthy` condition, and achieves high accuracy for the `Outer` (99%) and `Inner` (91%) fault classes. Misclassifications mainly occur between `Inner` and `Outer` faults, likely due to similar vibration patterns.

The lowest performance is observed in the `Ball` fault class, with an accuracy of 89%. Mislabeling occurs mostly as `Inner` (4.2%) and `Outer` (6.3%) faults. Despite this, the model maintains strong overall performance.

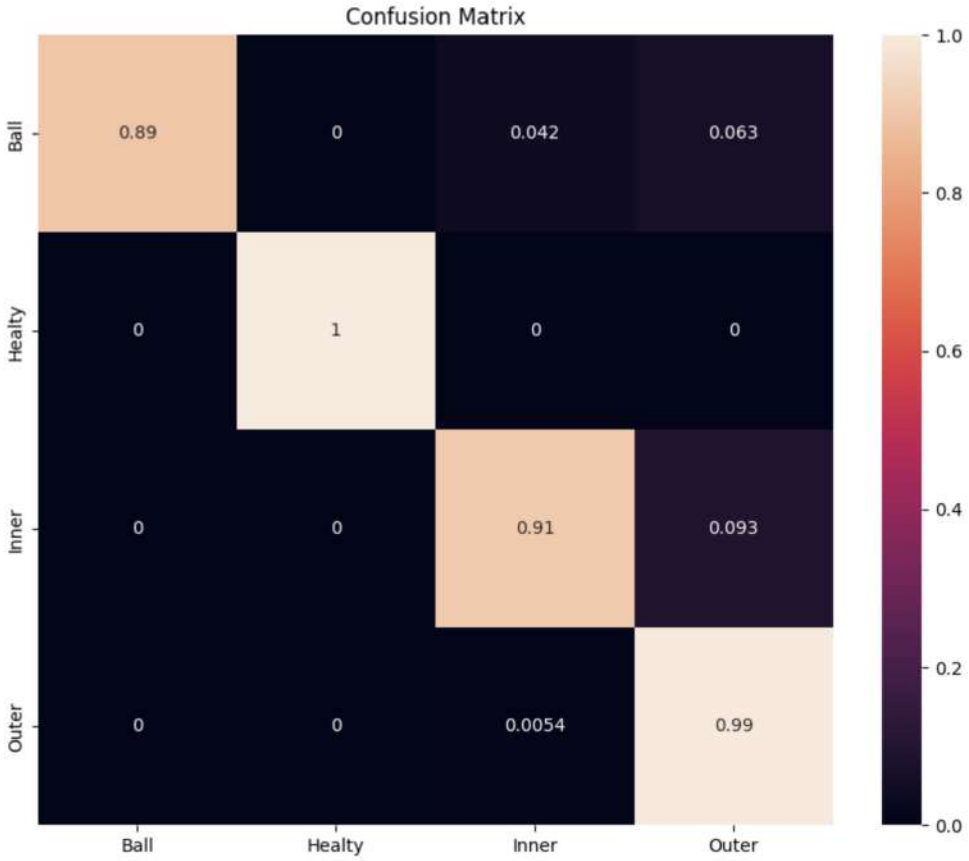


Figure 59: Normalized confusion matrix of the 1DCNN model trained on raw data.

These results confirm that using raw data preserves critical temporal features, leading to more accurate fault classification compared to FFT-transformed data.

In this study, the proposed model was validated using the test dataset `outer265racefault_testrawdata.csv` which contains time-series vibration signals corresponding to outer race defects in bearings. The model was evaluated on this dataset, and it successfully identified all 118 instances related to outer race faults. This result indicates that the model achieves a high level of accuracy in the fault diagnosis task.

To enhance the explainability of the classification results, the most influential temporal segments (windows) contributing to the model's predictions were analyzed. Based on this evaluation, the five most informative windows were identified, and two representative examples are illustrated in Figure 60 and 61.

These windows not only highlight which portions of the time-series signal were most influential for classification, but also offer a crucial advantage in terms of interpretability from an engineering perspective. Unlike conventional methods that often rely on the entire signal, this approach facilitates the localization of decision-critical components. Furthermore, due to the invertible nature of the FFT, the frequency-domain representations of these windows were also examined, thereby improving the traceability and interpretability of the model's decision-making process.

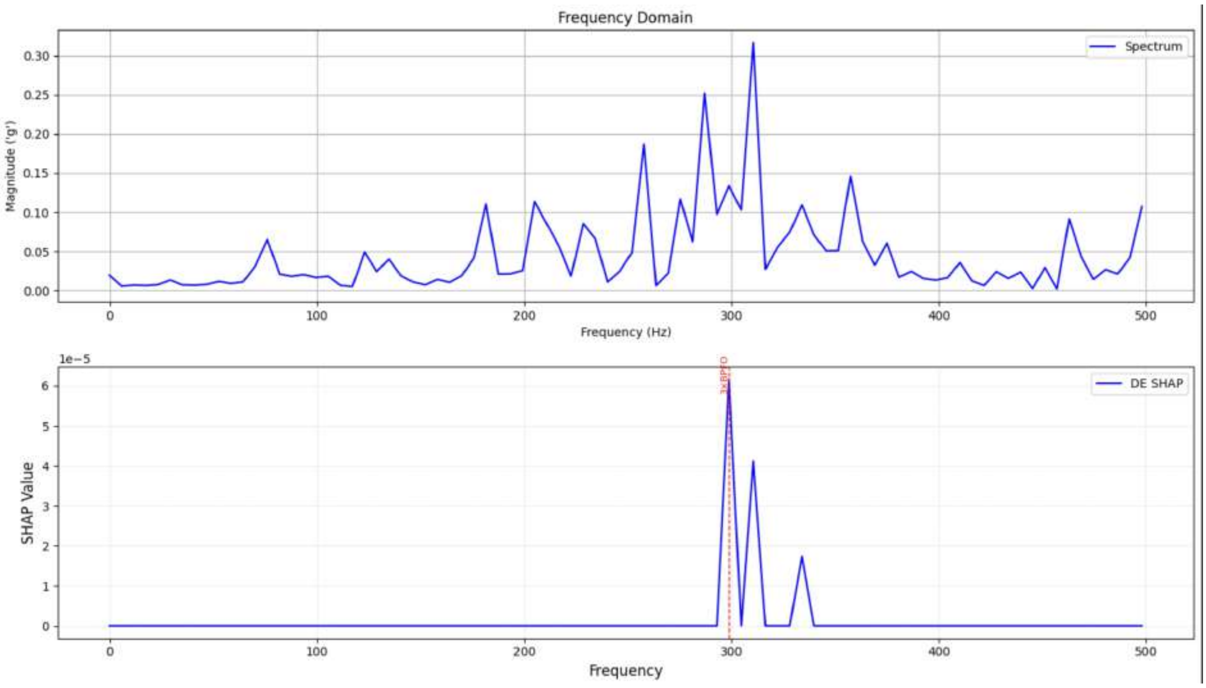


Figure 60: Dft resolution and shapley values of a sample

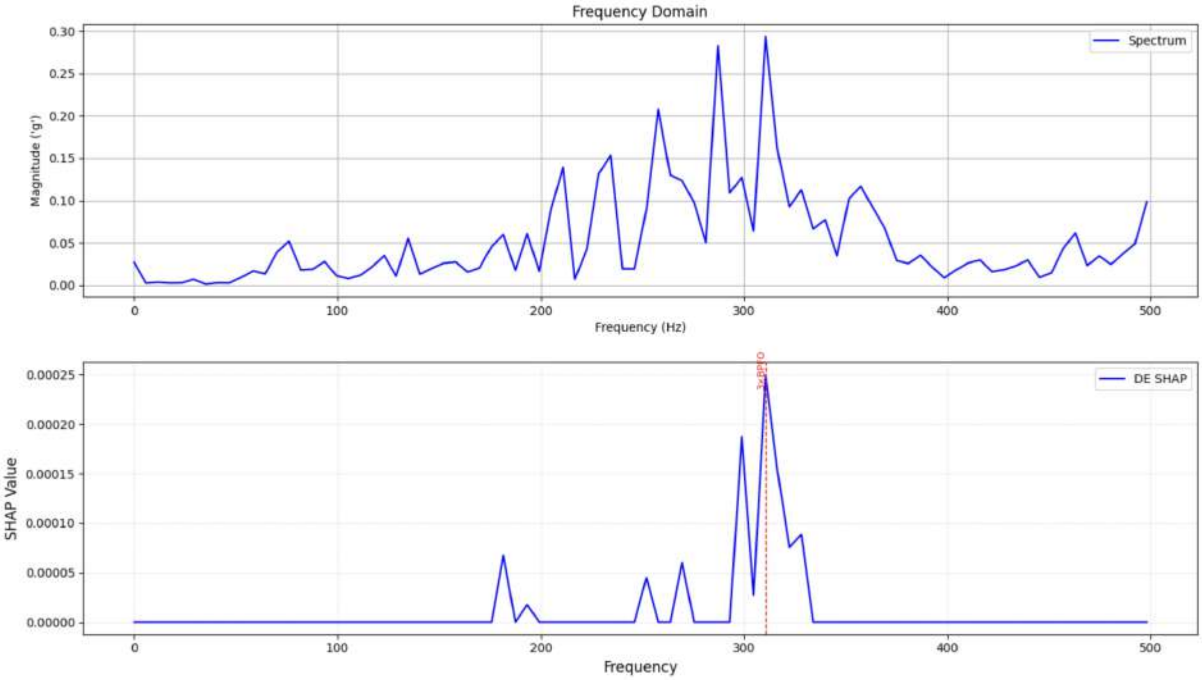


Figure 61: Dft resolution and shapley values of a sample

Frequency-domain signal and corresponding SHAP values. A peak is clearly observed at the third harmonic, as expected. The associated high SHAP value confirms the model's sensitivity to this defect-related frequency component. In contrast, nearby low SHAP values may be attributed to spectral leakage effects, which do not contribute significantly to the model's prediction.

In conclusion, the proposed model not only demonstrates strong classification performance but also provides meaningful insights into its internal reasoning by bridging time and frequency domain representations. This dual perspective distinguishes the present work from prior studies and underscores the significance of FFT-based invertible transformations for interpretable and engineer-oriented fault diagnosis applications.

8 CONCLUSION

This study suggested a comprehensive methodology to tackle the bearing fault diagnosis issue by integrating high-performance classification with interpretability. The primary contributions of the work can be summarized as follows:

- A one-dimensional convolutional neural network (1DCNN) was built on raw time-domain vibration data, achieving good accuracy in training, validation, and testing phases.
- A sliding-window method was employed for inference on the test signal. Each segment (window) was independently forecasted, and a majority-vote procedure was employed to reach the final diagnostic outcome.
- A spectral energy analysis was conducted on each window to determine which segments made the biggest contributions to the model's predictions. In particular, the top five windows with the highest total energy consumption were chosen as the most informative after the energy distribution of the first five harmonics was calculated.
- The trained 1DCNN model was supplemented with a customized Discrete Fourier Transform (DFT) layer to improve explainability. This layer permits end-to-end differentiability while maintaining the signal's frequency characteristics.
- The spectral components primarily responsible for the model's judgments were then revealed by using the updated model for Shapley value-based interpretation.
- Finally, SHAP analysis on the selected window segments revealed distinct peaks at the first three harmonic frequencies—consistent with known fault characteristics—demonstrating that the model's internal representations align with domain knowledge.

Overall, the findings emphasize that, mainly because temporal and phase information is retained, raw time-domain signals perform better in classification than FFT-transformed data. Additionally, by bridging the gap between time and frequency domain representations, the suggested approach enables transparent, frequency-informed reasoning in addition to precise fault identification. For deployment in industrial settings, where engineers need diagnostic outputs that are traceable and justified, this interpretability is crucial.

Bibliography

[1] Decker, T., Lebacher, M., & Tresp, V. (n.d.). *Does your model think like an engineer? Explainable AI for bearing fault detection with deep learning*. Ludwig Maximilians Universität & Siemens AG.

[2] Hasan, M. J. (2021). *Artificial intelligence techniques for bearing fault diagnosis*.

[3] Eren, L., Ince, T., & Kiranyaz, S. (2018). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Mechanical Systems and Signal Processing*.

[4] Herwig, N., & Borghesani, P. (2023). Explaining deep neural networks processing raw diagnostic signals. *Mechanical Systems and Signal Processing*.

[5] Chen, H.-Y., & Lee, C.-H. (2023). Vibration signals analysis by explainable artificial intelligence (XAI) approach: Application on bearing faults diagnosis. *Mechanical Systems and Signal Processing*.

[6] Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics–A tutorial. *Mechanical Systems and Signal Processing, 25*(2), 485–520.

[7] Case Western Reserve University Bearing Data Center. (n.d.). *Bearing data center website*.

[8] Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mechanical Systems and Signal Processing*.

[9] Hendriks, J., Dumond, P., & Knox, D. A. (2022). Towards better benchmarking using the CWRU bearing fault dataset. *Mechanical Systems and Signal Processing*.

[10] Ghorbel, A., Eddai, S., Limam, B., Feki, N., & Haddar, M. (2025). Bearing fault diagnosis based on artificial intelligence methods: Machine learning and deep learning. *Mechanical Systems and Signal Processing*.

[11] Siemens. (n.d.). Digital Signal Processing: Knowledge booklet .