

**Homework 2 - ENME 691**

Report - Group 3

February 26th, 2024

Brian O’Malley

Ashwin Sudarshan

Sai Gelam

# 

**Table of Contents**

1. Introduction
2. Solution Methodology
3. Results
4. Discussion
5. Conclusions
6. Appendix
7. **Introduction**
8. **Background information and description about the problem**

**Rotor-Bearing System:** This system typically consists of a rotating shaft (the rotor) supported by bearings. Bearings are critical components that enable smooth rotation while minimizing friction and wear. Monitoring the health of these bearings is essential for ensuring the reliability and efficiency of rotating machinery.

**Unbalance Defects:** Unbalance occurs when the center of mass of a rotating component (such as the rotor in this case) is not aligned with the axis of rotation. This can lead to vibration and potential damage to the bearings and other components of the system. In the testbed, unbalanced defects were induced by adding screws to the disk of the shaft.

**Vibration Analysis:** Vibration signals provide valuable insights into the condition of rotating machinery. Changes in vibration patterns can indicate various faults or anomalies, including unbalance, misalignment, bearing defects, and more. Accelerometers are commonly used to measure vibration levels in rotating machinery.

**Data Collection:** Vibration data were collected using an accelerometer mounted on the bearing block. The data were sampled at a high rate of 2560 Hz to capture detailed vibration characteristics. This high sampling rate is necessary to capture the fast-changing vibration signals accurately

1. **Raw data description**

A rotor-bearing testbed (Figure 1) was built to analyze the health condition of the shaft with unbalance defects.The rotating speed of the test-bed is 20 Hz, and two screws were added on the disk of the shaft to induce unbalance defects as shown on right hand side of Figure 1. An accelerometer was mounted on the bearing block and to measure the bearing's vibrations. The vibration data were collected under balanced and unbalanced conditions at a rate of 2560 Hz

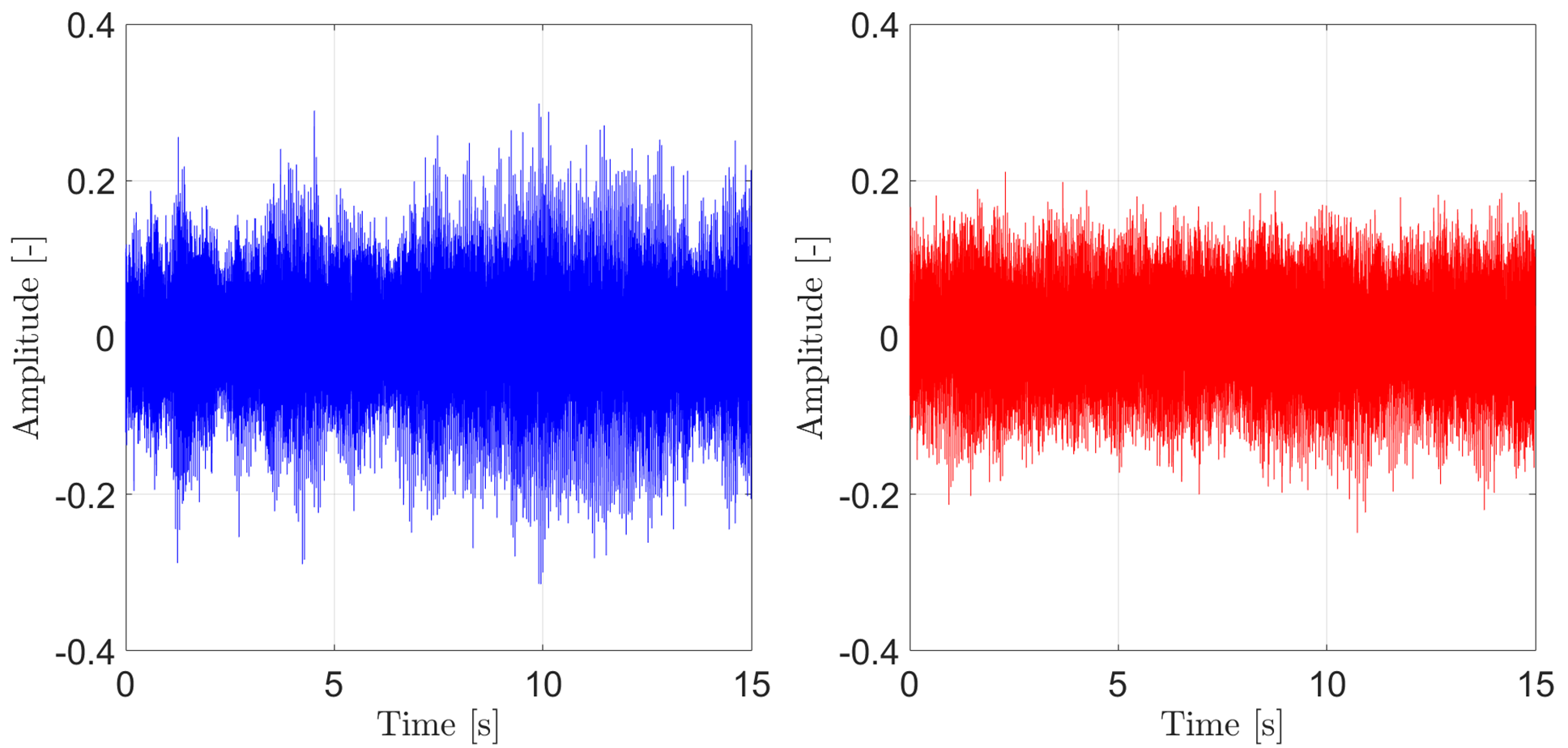


**Figure 1:** Balanced (left) and unbalanced (right) rotor-bearing test beds.

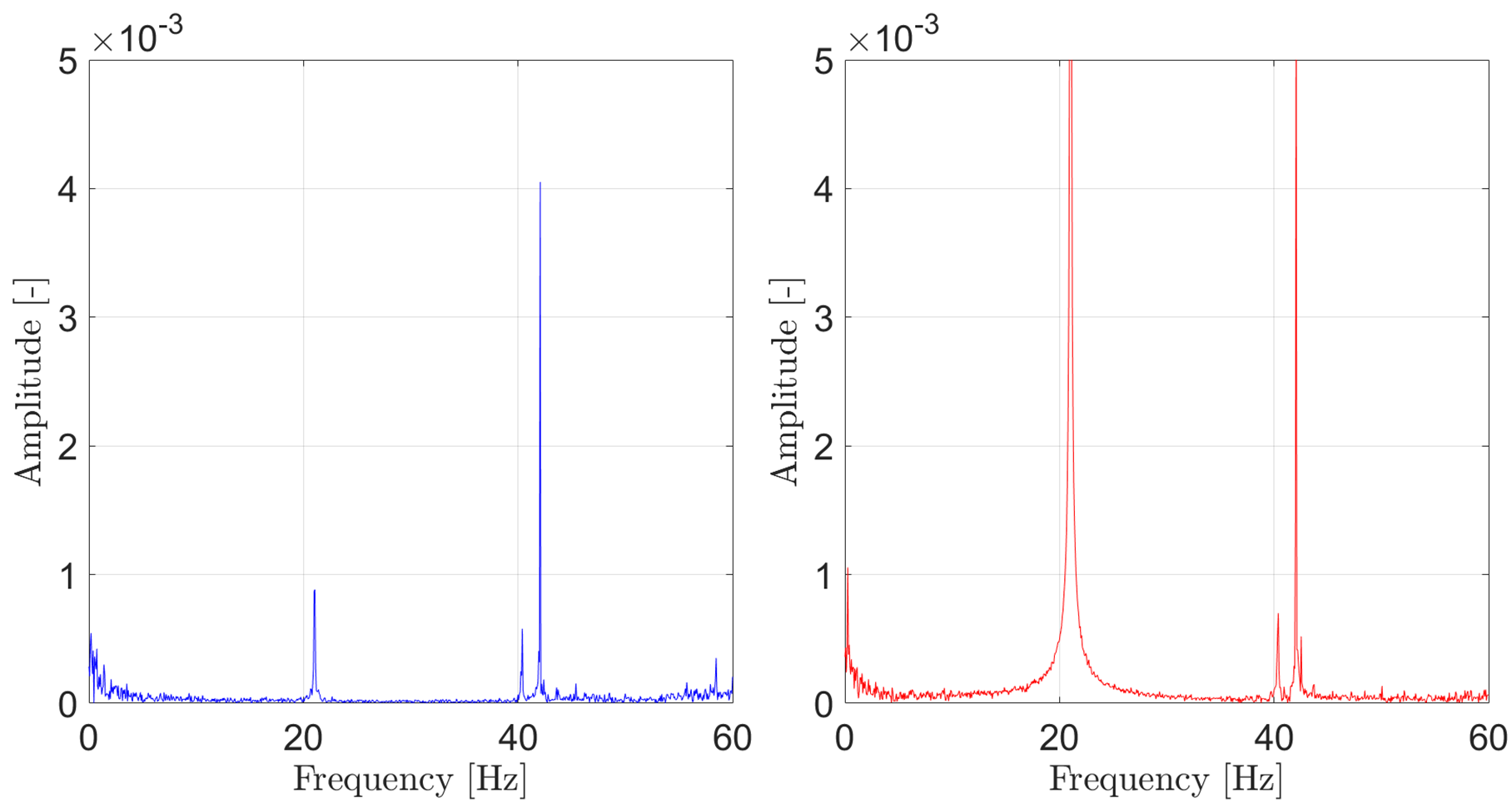
The resultant data was split into training and testing sets for the machine learning model being developed as a part of this work. Forty tests were allocated for the training set and thirty for the testing set. The training set was composed of 50% balanced and faulty bearing tests while the proportion of balanced to unbalanced tests in the testing set was unknown.

1. **Solution Methodology**
   1. **Data Analysis**

The solution process begins with loading the vibration data from the healthy, faulty, and testing datasets. Time domain analysis is conducted by plotting both healthy and faulty data sets across the data collection period, but the difference in magnitude is hard to discern and actually looks worse for the healthy data set (Figure 2). The data is transformed using the fast fourier transform as shown in Figure 3, where the peak amplitudes are visible. The amplitude of the signal is scaled down by ½ the number of points in each signal and results are only shown up to 60 Hz. The difference in amplitude between the healthy and faulty samples is now clear. The same data analysis process is applied to the test data.



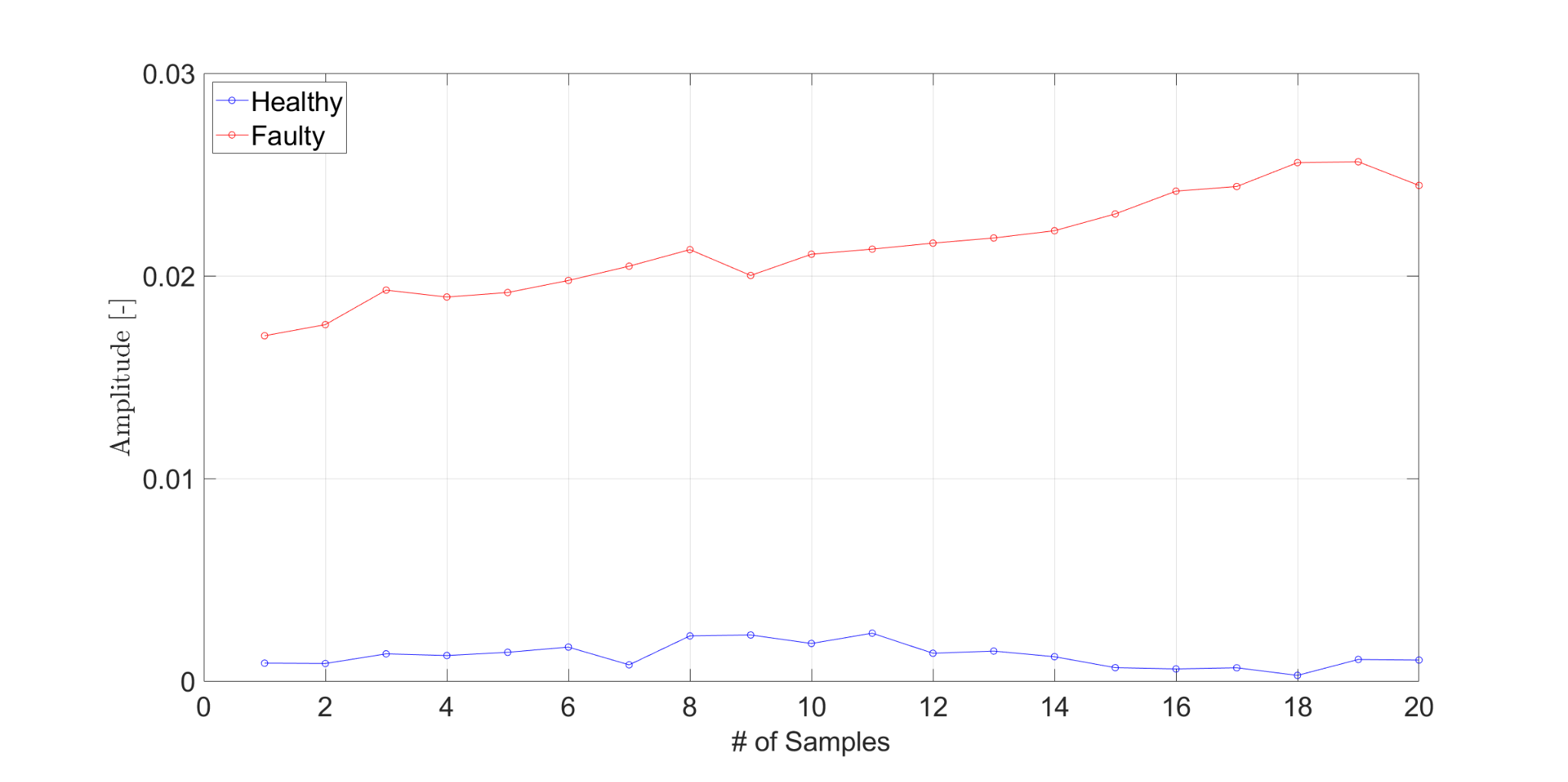
**Figure 2:** Time domain analysis of the balanced (left) and faulty (right) samples.



**Figure 3:** Frequency domain analysis of the balanced (left) and unbalanced (right) samples.

* 1. **Feature Extraction**

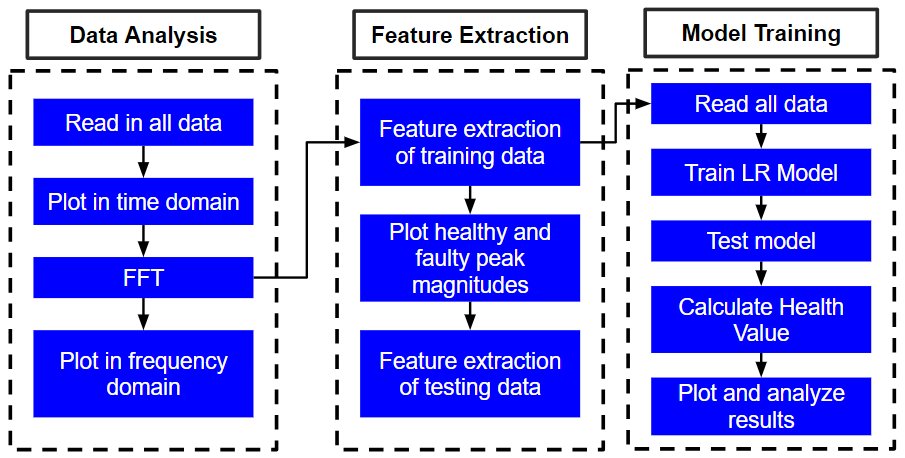
The amplitude of the signal at the 1X harmonic (20 Hz) is selected by looping through the data between 15 and 25 Hz and returning the maximum value. All of the peaks (like those shown in Figure 3) can be grouped together and trends can be established. As shown in Figure 4, Faulty data displays an amplitude between 0.017 and 0.026 while healthy data has magnitudes in the range of 2.8e-4 to 2.4e-3. The same process is applied to the test data.



**Figure 4** : Feature of amplitude of 1X harmonic in frequency domain

* 1. **Flow Chart**

The workflow of the entire process including the data analysis, feature extraction, and model development can be seen in Figure 5.

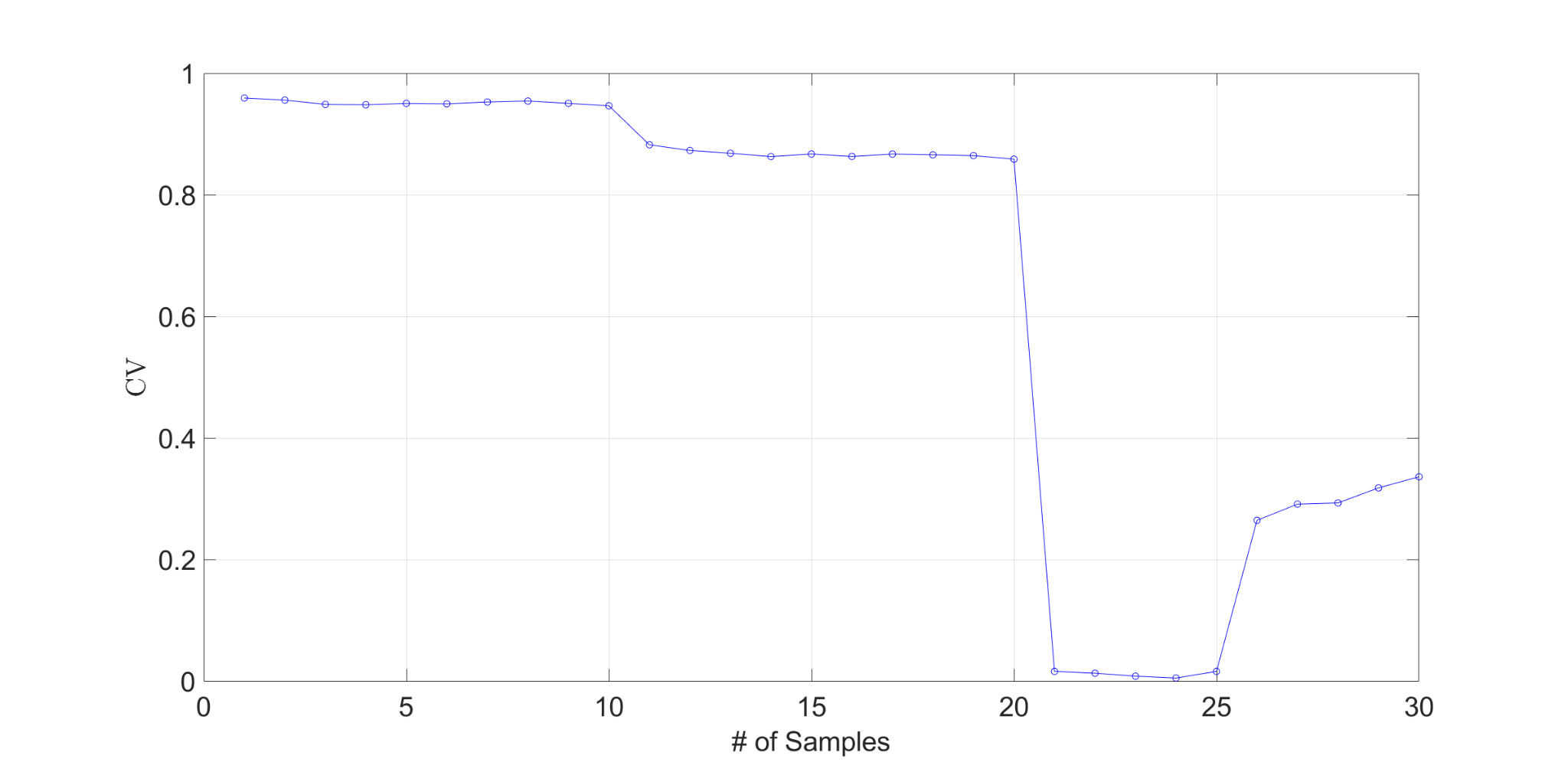


**Figure 5:** Workflow of the entire process

1. **Results and Discussion**

Logistic Regression - Logistic regression is employed as the classification algorithm in this project. It is chosen for its simplicity and effectiveness in binary classification problems. The model is trained on the extracted features, learning to predict the probability of a signal being from a healthy or faulty machine. Through training and validation, the model's parameters are optimized to maximize classification accuracy.

When training the model using both healthy and faulty data and subsequently testing it on a set of 30 samples, we observed that the model detected faulty bearing vibrations in the test dataset after the 20th sample. This detection was accompanied by a substantial drop in confidence values, indicating the presence of faulty behavior in the bearings. This can be shown in the below figure



**Figure 6:** Confidence Value of testing data.

1. **Conclusions**

In conclusion, the results of the study demonstrate the effectiveness of the proposed methodology in detecting faulty bearing vibrations in rotor-bearing systems. The utilization of vibration analysis coupled with machine learning, specifically logistic regression, yielded promising outcomes in distinguishing between healthy and faulty conditions. The model's ability to accurately identify faulty behavior in the test dataset, accompanied by a discernible drop in confidence values, underscores its potential for practical application in real-world scenarios. By leveraging features extracted from vibration data, particularly focusing on the amplitude of the 1X harmonic in the frequency domain, the model achieved reliable classification performance. These findings have significant implications for predictive maintenance strategies, offering a non-invasive and proactive approach to monitoring the health of rotating machinery. Early detection of faults, such as unbalance defects, can facilitate timely maintenance interventions, thereby minimizing downtime, reducing maintenance costs, and enhancing operational efficiency

1. **Appendix**
   1. **Data and Feature Extraction**

**%Brian O'Malley**

**%ENME 691 - Industrial AI**

**% Spring 2024**

**clc;clear;close;**

**%% Top Matter**

**format long; format compact;**

**set(0,'defaultTextInterpreter','latex'); %trying to set the default**

**sz = 60; %Marker Size**

**szz = sz/35;**

**lw = 1;**

**ms=8;**

**fs=25;**

**txtsz = 30;**

**txtFactor = 0.8;**

**ax = [0.9,1.4,0.0,2.0];**

**loc = 'southwest';**

**pos = [218,114,1478,796];**

**% txtsz = 24;**

**%%**

**%load in the data**

**dirTrainH = "F:\NOTES\Classes\Industrial AI\HW2\HW2\Homework 2\Training\Training\Healthy";**

**dirTrainF = "F:\NOTES\Classes\Industrial AI\HW2\HW2\Homework 2\Training\Training\Faulty";**

**dirTest = "F:\NOTES\Classes\Industrial AI\HW2\HW2\Homework 2\Testing\Testing";**

**filesH = dir(dirTrainH +'\\*.txt');**

**filesF = dir(dirTrainF +'\\*.txt');**

**filesT = dir(dirTest +'\\*.txt');**

**shaftspeed = 20;% Hz**

**Fs = 2560; % Sampling frequency**

**T = 1/Fs; % Sampling period**

**L = 38400; % Length of signal**

**time = (0:L-1)\*T; % Time vector**

**FsRange = Fs/L\*(0:L-1);**

**%set the range of interest (shaft speed is ~20 Hz)**

**lowerBound = find(FsRange==15);**

**upperBound = find(FsRange==25);**

**%%**

**%transform healthy data**

**for i=1:length(filesH)**

**tempStr = dirTrainH +"\"+filesH(i).name**

**trainingDataH{i} =readtable(tempStr);**

**t=abs(fft(trainingDataH{i}.Date3\_26\_2014));**

**transFFT\_H{i} = t; clear t;**

**end**

**%%**

**%transform faulty data**

**for i=1:length(filesF)**

**tempStr = dirTrainF +"\"+filesF(i).name**

**trainingDataF{i} =readtable(tempStr);**

**t=abs(fft(trainingDataF{i}.Date3\_26\_2014));**

**transFFT\_F{i} = t; clear t;**

**end**

**%%**

**%transform testing data**

**for i=1:length(filesT)**

**tempStr = dirTest +"\"+filesT(i).name**

**testData{i} =readtable(tempStr);**

**t=abs(fft(testData{i}.Date3\_26\_2014));**

**transFFT\_T{i} = t; clear t;**

**end**

**%%**

**%pick out the peak for the healthy data**

**for i=1:20**

**temp = transFFT\_H{i}/L\*2;**

**healthyMag(i) = max(temp(lowerBound:upperBound));**

**clear temp;**

**end**

**%%**

**%pick out the peak for the fault data**

**for i=1:20**

**temp = transFFT\_F{i}/L\*2;**

**faultyMag(i) = max(temp(lowerBound:upperBound));**

**clear temp;**

**end**

**%%**

**%pick out the peak for the Testing data**

**for i=1:30**

**temp = transFFT\_T{i}/L\*2;**

**testingMag(i) = max(temp(lowerBound:upperBound));**

**clear temp;**

**end**

**%%**

**%plots of interest follow, 1 and 2 are the time domain amplitudes of**

**%the healthy and faulty data, 3 and 4 are the frequency domain plots**

**%plot 5 demonstrates the feature extraction**

**%%**

**figure(1)**

**grid on; hold on; box on; axis square;**

**ax=gca;**

**ax.FontSize = fs;**

**xlim([0 15]);**

**ylim([-0.4 0.4]);**

**yticks([-0.4 -0.2 0 0.2 0.4]);**

**temp = table2array(trainingDataH{1});**

**plot(time,temp,'-b');**

**xlabel('Time [s]','FontSize',fs);**

**ylabel('Amplitude [-]','FontSize',fs);**

**clear temp;**

**% legend('Healthy','Faulty','Location','Northwest','FontSize',fs);**

**%%**

**figure(2)**

**grid on; hold on; box on; axis square;**

**ax=gca;**

**ax.FontSize = fs;**

**xlim([0 15]);**

**ylim([-0.4 0.4]);**

**yticks([-0.4 -0.2 0 0.2 0.4]);**

**temp = table2array(trainingDataF{1});**

**plot(time,temp,'-r');**

**xlabel('Time [s]','FontSize',fs);**

**ylabel('Amplitude [-]','FontSize',fs);**

**clear temp;**

**% legend('Healthy','Faulty','Location','Northwest','FontSize',fs);**

**%%**

**figure(3)**

**grid on; hold on; box on; axis square;**

**ax=gca;**

**ax.FontSize = fs;**

**xlim([0 60]);**

**ylim([0 0.005]);**

**yticks([ 0 0.001 0.002 0.003 0.004 0.005]);**

**temp = transFFT\_H{1}/L\*2;**

**plot(FsRange, temp,'-b');**

**xlabel('Frequency [Hz]','FontSize',fs);**

**ylabel('Amplitude [-]','FontSize',fs);**

**clear temp;**

**%%**

**figure(4)**

**grid on; hold on; box on; axis square;**

**ax=gca;**

**ax.FontSize = fs;**

**xlim([0 60]);**

**ylim([0 0.005]);**

**yticks([ 0 0.001 0.002 0.003 0.004 0.005]);**

**temp = transFFT\_F{1}/L\*2;**

**plot(FsRange, temp,'-r');**

**xlabel('Frequency [Hz]','FontSize',fs);**

**ylabel('Amplitude [-]','FontSize',fs);**

**clear temp;**

**% legend('Healthy','Faulty','Location','Northwest','FontSize',fs);**

**% end**

**%%**

**figure(5)**

**grid on; hold on; box on; axis square;**

**ax=gca;**

**ax.FontSize = fs;**

**yticks([ 0 0.01 0.02 0.03]);**

**xlim([0 20]);**

**ylim([0 0.03]);**

**plot(healthyMag,'-ob');**

**plot(faultyMag,'-or');**

**legend('Healthy','Faulty','Location','Northwest','FontSize',fs);**

**xlabel('# of Samples','FontSize',fs);**

**ylabel('Amplitude [-]','FontSize',fs);**

* 1. **Model Training**

%E-Manufacturing 2013

% modified by Brian O'Malley (2024)

%Logistic Regression

%How to Use This Method

clc;clear;close;

%% Top Matter

format long; format compact;

set(0,'defaultTextInterpreter','latex'); %trying to set the default

sz = 60; %Marker Size

szz = sz/35;

lw = 1;

ms=8;

fs=25;

txtsz = 30;

txtFactor = 0.8;

ax = [0.9,1.4,0.0,2.0];

loc = 'southwest';

pos = [218,114,1478,796];

% txtsz = 24;

healthyMag = table2array(readtable("healthyMag.txt"));

faultyMag = table2array(readtable("faultyMag.txt"));

testingMag = table2array(readtable("testingMag.txt"));

FeatureMatrix(1:20,1) = healthyMag;

FeatureMatrix(21:40,1) = faultyMag;

FeatureMatrix(41:70,1) = testingMag;

%% Select Training Portion and PCA (front/back)

%samples

GoodSampleIndex=1:20;

DegradedSampleIndex=21:40;

%Baseline Data

BaselineData=FeatureMatrix(GoodSampleIndex,:);

DegradedData=FeatureMatrix(DegradedSampleIndex,:);

%% Train LR Model

%Label Vector (0.95 for good samples, 0.05 for bad samples

Label=[ones(size(BaselineData,1),1)\*0.95; ones(size(DegradedData,1),1)\*0.05];

%fit LR Model (glm-fit)

beta = glmfit([BaselineData; DegradedData],Label,'binomial');

%% Calculating Health Value (using LR Model)

TestFeatureMatrix=FeatureMatrix(41:70,:);

%calculate CV (Health Value)

CV\_Test = glmval(beta,TestFeatureMatrix,'logit') ; %Use LR Model

%%

figure(1)

grid on; hold on; box on;

axis square;

ax=gca;

ax.FontSize = fs;

plot(CV\_Test,'-ob');

% legend(,'Location','Northwest');

xlabel('# of Samples','FontSize',fs);

ylabel('CV','FontSize',fs);