ISYE 6748: Applied Analytics Practicum

Student name: Ng Tai Hiang, Jeremy
GTID: 903744572

Project Name: Vibration Signal Analysis to Predict Failure of Rotating Shaft

1. Abstract

The goal of this project is to explore Machine Learning techniques to detect unbalanced loads on rotating shafts based on the vibration data collected. In the real world, such signals could be a sign of wear and tear of the shaft and motor components which leads to an imbalance and hence causes vibration. Early or proactive detection of such signals can allow preventive maintenance to be carried out before failure of the rotation shaft, saving potential losses on unintended downtime.

The focus of this project will be to deep-dive into feature extraction from the vibration data and establish relationships with input variables.The desired outcome is to use these features to develop high accuracy model(s) with tolerable false positives that can advise on preventive maintenance before failure.

From the analysis, a convolutional neural network (CNN) was applied on the dataset and achieved an average accuracy of 75%. This implies that the model can detect unbalanced loads (vibrations) preemptively to allow maintenance/repair to be carried out before the machine breaks down.

2. Background Research

An approach to analyse the signals to allow the detection of ‘degraded’ vs ‘non-degraded’ signals is via signal decomposition. In essence, these are techniques to analyse the components of these signals which can be used as features in machine learning to train the model to identify degraded signals.

Fast Fourier Transform (FFT): Primarily a technique for analysing frequency content of a signal, to transform time-domain signal into its frequency components

Empirical Mode Decomposition (EMD): To decompose non-linear and non-stationary signals into functions known as intrinsic mode functions (IMFs).

Variational Mode Decomposition (VMD): A technique to decompose a signal into different modes, each mode represents specific oscillatory behaviour or pattern.

Principal Component Analysis (PCA): PCA could be used to identify principal components or the mode of variations in the vibration data. This can lead us to identify ‘signature patterns’ which can be suggestive of unbalanced loads

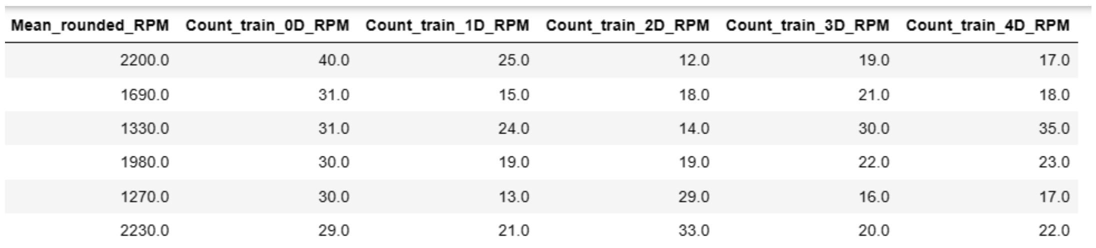
3. Exploratory Data Analysis

3.1. Brief Description of Data and Method of Collection

The data collected are vibration data from a rotating shaft driven by a motor. This serves as an analog for many real-world machines such as turbines, motors, impellers etc. The dataset is collected by varying the voltage input to the motor and measuring the vibrations picked up by 3 different vibration sensors placed at different locations. The revolutions-per-minute (RPM) of the shaft is also measured

The data are collected by varying the ‘unbalance’ factor by adding 4 different weights to the rotor.

There are 8 datasets, labelled ‘D’ and ‘E’ where D stands for development for training and E stands for evaluation. The datasets are labelled as 0D, 1D, 2D, 3D and 4D, where 0 represents data taken without unbalance and 4 represents data taken with the highest amount of unbalance load. Vibration data is collected by stepping up the voltage input (“V_in) and the RPM of the shaft and 3 sets of vibration data were measured. The dataset was collected at an interval of 4096 readings per second.



Mean_rounded_RPM	Count_train_0D_RPM	Count_train_1D_RPM	Count_train_2D_RPM	Count_train_3D_RPM	Count_train_4D_RPM
2200.0	40.0	25.0	12.0	19.0	17.0
1690.0	31.0	15.0	18.0	21.0	18.0
1330.0	31.0	24.0	14.0	30.0	35.0
1980.0	30.0	19.0	19.0	22.0	23.0
1270.0	30.0	13.0	29.0	16.0	17.0
2230.0	29.0	21.0	33.0	20.0	22.0

Figure 1: Sample of the raw data 0D (motor with no unbalance)

3.2 Cleansing/Preparation of Data

The idea for data preprocessing is that the vibration signals are related to the RPM. Hence, the preparation aims to find datasets between the training and evaluation where it corresponds to similar RPM values.
To achieve that, data exploration showed that some data have to be removed as the vibration values are much higher than the rest of the dataset.

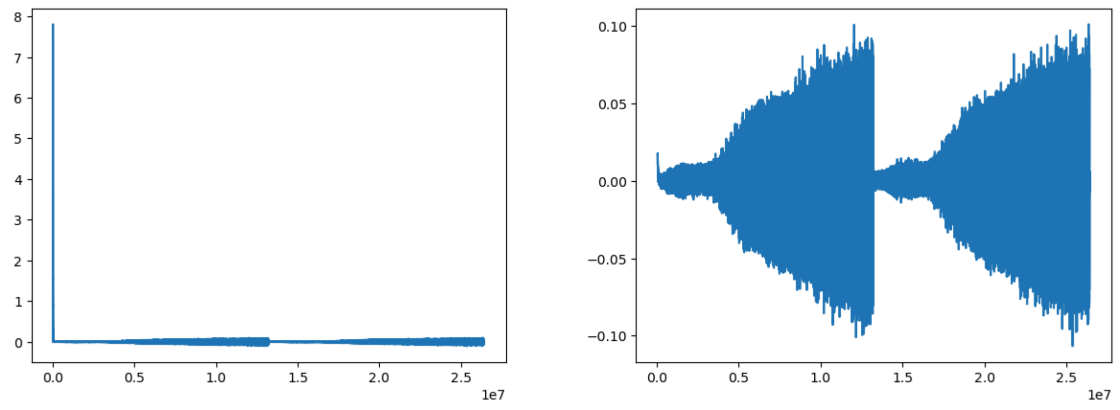


Figure 2: Plot of 'Vibration_1' data showing data irregularities in the earlier part of the data collection process on the left and Plot of 'Vibration_1' data showing a nice double pattern of ramp up and down.

Next, within each V_in, it was mentioned that each step was held for 20 seconds. By plotting that on a chart of RPM vs time, it can be observed that the RPM values are unstable near the front and end of the 20 seconds window.

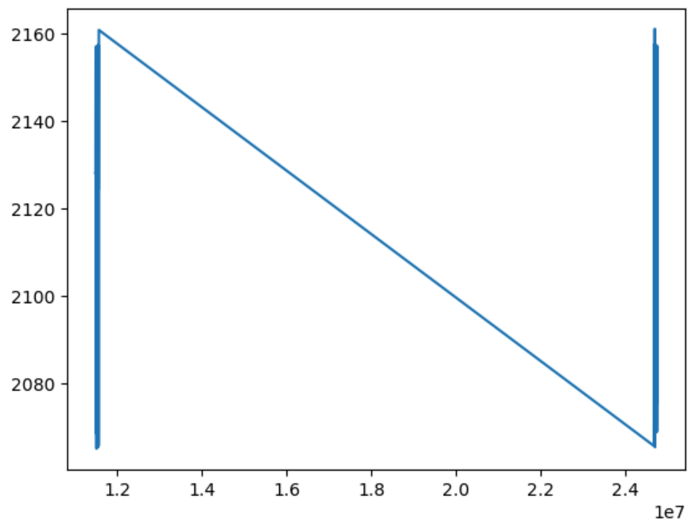
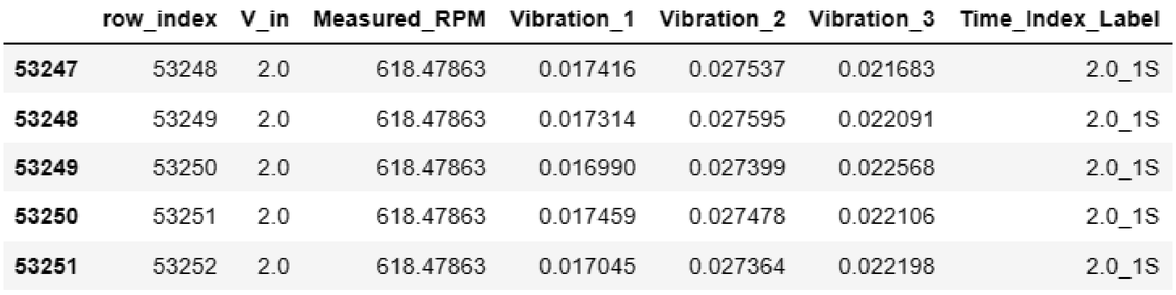


Figure 3: Plot of 20 second window for V_in of 9.0V against measured RPM. Note the irregular front and end portions of the 20 second window

Hence, a choice was made to remove the first 5 and last 5 seconds of the window. To do that, each 20 second window is labelled with an index named as

“Time_Label_Index” where it has the format of “V_in_xS”, where x represents the seconds number in each 20 second window. For example, for a V_in value of 2.0, there will be 20 labels from 2.0_1S to 2.0_20S. Each label will contain 4,096 rows of data.



	row_index	V_in	Measured_RPM	Vibration_1	Vibration_2	Vibration_3	Time_Index_Label
53247	53248	2.0	618.47863	0.017416	0.027537	0.021683	2.0_1S
53248	53249	2.0	618.47863	0.017314	0.027595	0.022091	2.0_1S
53249	53250	2.0	618.47863	0.016990	0.027399	0.022568	2.0_1S
53250	53251	2.0	618.47863	0.017459	0.027478	0.022106	2.0_1S
53251	53252	2.0	618.47863	0.017045	0.027364	0.022198	2.0_1S

Figure 4: Snap-shot of raw data with added Time_Index_Label on the rightmost column

As there were 2 run-ups which are labelled continuously for each V_in, we only kept those from 5S to 15S and 25S to 35S. First run up is labelled from 1 to 20S while 2nd run up is labelled from 20 to 40S

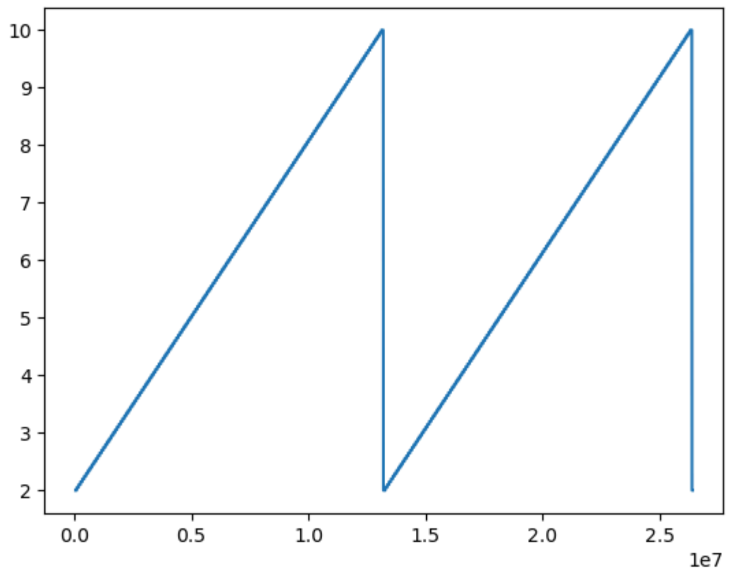
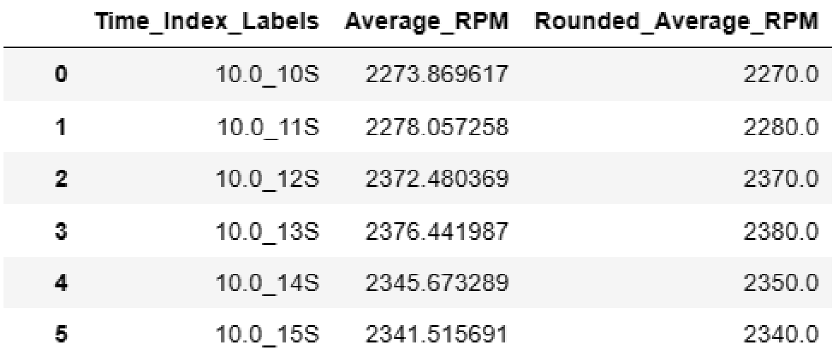


Figure 5: Plot of V_in throughout the whole dataset, showing 2 runs from 2.0V to 10V

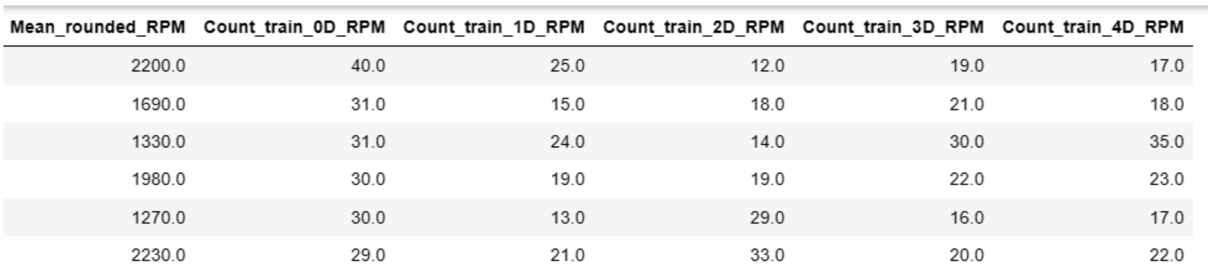
The approach is to find the average RPM for each 1 second window as there are some minor variations and round them to the nearest 10 RPM. As both the training and evaluation dataset had different RPM at different V_in, a matching was done.



	Time_Index_Labels	Average_RPM	Rounded_Average_RPM
0	10.0_10S	2273.869617	2270.0
1	10.0_11S	2278.057258	2280.0
2	10.0_12S	2372.480369	2370.0
3	10.0_13S	2376.441987	2380.0
4	10.0_14S	2345.673289	2350.0
5	10.0_15S	2341.515691	2340.0

Figure 6: Sample table showing Average_RPM calculated for each Time_Index_Labels (essentially an average of 4096 values of measured RPM) and the RPM rounded to the nearest tenth value.

Based on the matching, the average RPM values that match all the dataset were filtered out and from there, the corresponding “Time_Label_Index” for each dataset were obtained.



Mean_rounded_RPM	Count_train_0D_RPM	Count_train_1D_RPM	Count_train_2D_RPM	Count_train_3D_RPM	Count_train_4D_RPM
2200.0	40.0	25.0	12.0	19.0	17.0
1690.0	31.0	15.0	18.0	21.0	18.0
1330.0	31.0	24.0	14.0	30.0	35.0
1980.0	30.0	19.0	19.0	22.0	23.0
1270.0	30.0	13.0	29.0	16.0	17.0
2230.0	29.0	21.0	33.0	20.0	22.0

Figure 7: Sample table showing count of time_label_index for all average rounded RPM values across all the training tables. A similar table was also done for the evaluation table.

For the vibration variables to choose and use for the training, we plotted and look at the data across each vibration sensor over a single second window (4096 readings)

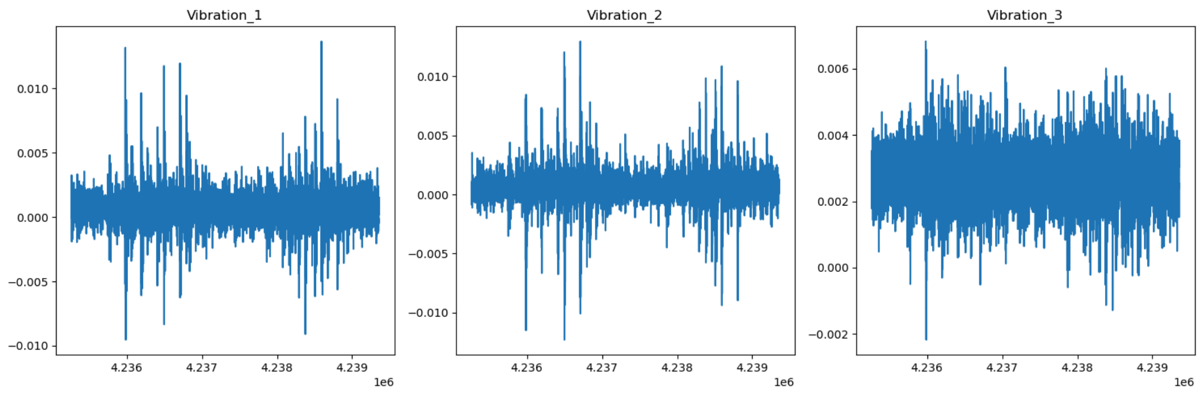


Figure 8: Based on the charts above, we can see that the vibration data for Vibration_1 and Vibration_2 have higher amplitude compared to Vibration_3.

From figure 8 above, we will use Vibration_1 as the data to train the models as it has higher amplitude than Vibration_3. However, depending on the context of the sensor placed, it might make sense to average out the values at each window so we can remove sensor to sensor variations and noise that are not due to the vibration of the shaft itself.

4. Feature Engineering

Feature Engineering involved applying Fast Fourier Transform (FFT) to break down the signal into its constituent domain frequencies.The idea is that by doing that, the machine learning model will be able to identify/differentiate between ‘balanced’ vs ‘unbalanced’ datasets.

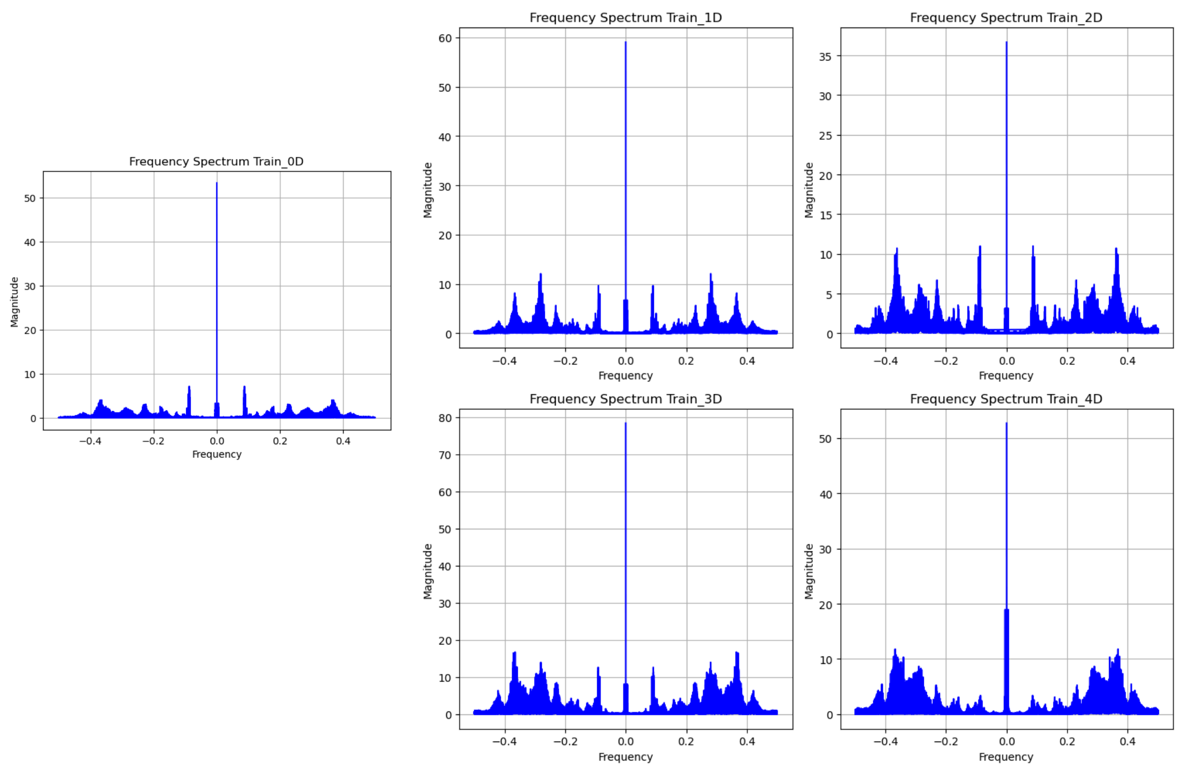


Figure 9: Plot of frequencies vs magnitude for 0D (balanced) followed by 1D, 2D, 3D, 4D (unbalanced data).

From figure 9, we can observe that there were higher magnitudes for the unbalanced (1D, 2D, 3D, 4D) set compared to the balanced set (0D) which is expected due to the vibrations. There are also some patterns observed as peak magnitudes can be observed across the whole frequency.

Visually, an interesting observation is that for higher unbalances (3D and 4D), the magnitudes are more seen at the higher frequency levels (near the tails) while the vibration for the lower unbalances (1D and 2D) are more well distributed throughout the whole frequency spectrum.

5. Machine Learning with Convolutional Neural Networks (CNN)

5.1 Initial Evaluation of Machine Learning

Based on the pre-processing above, an initial evaluation was done by choosing a specific RPM value (1200 RPM) and filtering out data which has an average measured RPM of 1200. From there, Fast-Fourier Transform (FFT) algorithm was applied to Vibration_1 in the dataset.

For 1200 RPM, the volume of data obtained from both training and evaluation set are as below, where the volume represents number of sets of 1 second windows (4096 rows per 1 second window)

Table 1: Number of datasets for each training and evaluation data where RPM measured is approximately 1200 RPM

Training 0D	Training 1D	Training 2D	Training 3D	Training 4D	Evaluation 0E	Evaluation 1E	Evaluation 2E	Evaluation 3E	Evaluation 4E
20	21	16	28	19	7	5	8	10	4

Applying a Convolution Neural Network (CNN) which is known to process signal related data well, we obtained the following summarised output

Table 2: Final validation loss and validation accuracy for CNN with 10, 20, 30 and 40 layers for dataset with a specific measured RPM of 1200 RPM

Model	**Final Validation Loss**	**Final Validation Accuracy**
10 convolution layers	0.76	18.18%
20 convolution layers	2	18.18%
30 convolution layers	0.76	18.18%
40 convolution layers	4.04	18.18%

Based on figure 2, we can see validation accuracy remains low as convolution layers increase. A potential reason could be limited data which leads to overfitting.

Another attempt was carried out using the full dataset regardless of measured RPM values. In this case, there were much more dataset, with an average of 3500 sets for training and 900 sets for evaluation.

Table 3: Final validation loss and validation accuracy for CNN with 10, 20, 30 and 40 layers running on 50 epochs for full dataset

Model	**Final Validation Loss**	**Final Validation Accuracy**
10 convolution layers	1.16	76.05%
20 convolution layers	0.59	74.69%
30 convolution layers	0.67	65.69%
40 convolution layers	4.04	78.59%

From the validation accuracy, we saw a big improvement in the accuracy. Thus, we can imply that a larger dataset is always favourable to improve training accuracy. An attempt to train on data with limited RPM resulted in low accuracy for the model. It can also be implied that the factors which influenced the vibrations due to the rotation speed of the shaft were not as significant.

5.2 Tuning the model

From the output above, the training time is relatively long as there are 50 epochs in the neural networks. An attempt was made to use lesser epochs to determine if it affects the accuracy or not while at the same time improving training and evaluation speeds.

Table 4: Final validation loss and validation accuracy for CNN with 10, 20, 30 and 40 layers running on 10 epochs for full dataset

Model	**Final Validation Loss**	**Final Validation Accuracy**
10 convolution layers	1.48	80.68%
20 convolution layers	0.47	75.59%
30 convolution layers	0.64	64.35%
40 convolution layers	0.41	78.93%

From the output above, we can observe that compared to 50 epochs, the validation loss and accuracy remained relatively similar.

6. Conclusion and Recommendations

From the above, we can imply that by applying FFT to the signal, the model performs relatively well in identifying degraded vs non-degraded signals, with accuracy of 75% on average. The key step is to extract only the useful windows of data from the whole range, as evident in the data-preprocessing step where the first 5 and last 5 seconds of data was removed due to the stepping up of the voltage input (which affects the vibration data). Applying this to a real world system/machine, a similar setup constantly monitors and logs the vibration data, and feeds it to the machine learning algorithm on a daily basis to monitor its similarity to a ‘normal’ vibration profile. Once the vibration profile goes beyond the defined threshold of similarity, it will be flagged out and the operator will determine if a check or maintenance is required to be carried out. In this way, maintenance schedules will be done on a predictive basis rather than on a scheduled/fixed basis with potential to limit downtime and reduce cost of operations and maintenance.

Further work could be to attempt other methods of signal decomposition and to deep-dive into picking out key frequencies which are of significance in the detection of unbalanced loads. The challenge would be that such an approach might remove useful data from the vibration signals and make the model too specific for a particular use case/system. For example, if a particular frequency was identified to be significant for the setup use in this analysis, it might not work for another system in other applications.

References

1. Oliver Mey, Willi Neudeck, Andre Schneider, & Olaf Enge-Rosenblatt. (2020, July 31). Machine Learning-Based Unbalance Detection of a Rotating Shaft Using Vibration Data. *ArViv*. https://arxiv.org/pdf/2005.12742.pdf
2. Eriksen, T., & Rehman, N. U. (2023, January 31). *Data-driven nonstationary signal decomposition approaches: a comparative analysis*. Scientific Reports. https://doi.org/10.1038/s41598-023-28390-w
3. Dataset downloaded from: https://www.kaggle.com/datasets/jishnukoliyadan/vibration-analysis-on-rotating-shaf