**Industrial Big Data Analytics and Applications**

**Capstone Project Report**

**Team 1**

**Bearing Failure Diagnosis**

**Submitted by**

Cyrus Azamfar

Krishna Kumar Chandran

Ravi Teja Telkapally

Siddhartha Patlori

**Table of Contents**

[List of Figures 3](#_Toc512468250)

[List of Tables 4](#_Toc512468251)

[Abstract 5](#_Toc512468252)

[Background 6](#_Toc512468253)

[**Data Collection** 8](#_Toc512468254)

[**Typically used techniques for bearing health analysis** 8](#_Toc512468255)

[Process Flow 10](#_Toc512468256)

[Data Pre-processing 11](#_Toc512468257)

[1. **Moving window average algorithm** 13](#_Toc512468258)

[2. **Kalman filter** 14](#_Toc512468259)

[3. **Grubbs’ Test** 15](#_Toc512468260)

[Feature Extraction 21](#_Toc512468261)

[**Time domain features:** 21](#_Toc512468262)

[**Frequency domain features** 22](#_Toc512468263)

[**Envelope Analysis:** 23](#_Toc512468264)

[**Using FFT analysis** 25](#_Toc512468265)

[**Spectral Kurtosis** 27](#_Toc512468266)

[Feature selection 28](#_Toc512468267)

[**Multi Class Fisher Criterion:** 29](#_Toc512468268)

[**Principle Component Analysis** 29](#_Toc512468269)

[Learning Models (Classification Techniques) 30](#_Toc512468270)

[**K-Fold Cross Validation** 31](#_Toc512468271)

[**KNN Method** 36](#_Toc512468272)

[**Decision Tree Method** 39](#_Toc512468273)

[**SOM Map tool** 43](#_Toc512468274)

[**SOM-MQE Study** 45](#_Toc512468275)

[Conclusions 48](#_Toc512468276)

[Future Work 49](#_Toc512468277)

[References 50](#_Toc512468278)

# List of Figures

[Figure 1: Bearing Failures Percentage 6](#_Toc512464291)

[Figure 2: Typical Bearing Failures 7](#_Toc512464292)

[Figure 3 : Process Flow Chart 10](#_Toc512464293)

[Figure 4: Bearing Frequency Equations 11](#_Toc512464294)

[Figure 5: Data after removing outliers using moving window 14](#_Toc512464295)

[Figure 6: Data after removing outliers using Kalman Filter 15](#_Toc512464296)

[Figure 7: Probability plot for Normal Distribution - Healthy Data 16](#_Toc512464297)

[Figure 8: Histogram plot for Healthy Data 16](#_Toc512464298)

[Figure 9: Probability Plot for Normal Distribution – Faulty Roller Data 17](#_Toc512464299)

[Figure 10: Histogram plot for Faulty Data 17](#_Toc512464300)

[Figure 11: Data after removing outliers using Grubbs’ test. 18](#_Toc512464301)

[Figure 12: Probability Plot for Normal Distribution – Faulty Roller Data 18](#_Toc512464302)

[Figure 13: Histogram plot for Healthy Data -After removing Outliers 19](#_Toc512464303)

[Figure 14: Probability Plot for Faulty Data after removing outliers. 19](#_Toc512464304)

[Figure 15: Histogram plot for Faulty data – After removing outliers 20](#_Toc512464305)

[Figure 16: Envelope analysis process on a vibration signal 24](#_Toc512464306)

[Figure 17: FFT analysis for finding bandpass frequency 25](#_Toc512464307)

[Figure 18: Envelope analysis on faulty data 26](#_Toc512464308)

[Figure 19: Envelope analysis on faulty data – using FFT for finding bandpass frequency 26](#_Toc512464309)

[Figure 20: Spectral Kurtosis analysis for finding bandpass frequency 27](#_Toc512464310)

[Figure 21: Envelope analysis on faulty data - using Spectral Kurtosis for finding bandpass frequency 28](#_Toc512464311)

[Figure 22: Multi Class Fisher Criterion 29](#_Toc512464312)

[Figure 23: Principal Component Analysis 30](#_Toc512464313)

[Figure 24: K-fold Cross Validation 31](#_Toc512464314)

[Figure 25: SVM results on training dataset – Using all features 33](#_Toc512464315)

[Figure 26: SVM results on test dataset– Using all features 33](#_Toc512464316)

[Figure 27: SVM results on training dataset with selected features 34](#_Toc512464317)

[Figure 28: SVM results on the test dataset with selected features 35](#_Toc512464318)

[Figure 29: KNN results on training dataset – Using all features 37](#_Toc512464319)

[Figure 30: KNN results on test dataset- Using all features 38](#_Toc512464320)

[Figure 31: KNN results on training dataset – Using selected features 38](#_Toc512464321)

[Figure 32: KNN results on test dataset- Using selected features 39](#_Toc512464322)

[Figure 33: Decision Tree results on training dataset – Using all features 41](#_Toc512464323)

[Figure 34: Decision Tree results on test dataset- Using all features 41](#_Toc512464324)

[Figure 35: Decision Tree results on training dataset – Using selected features 42](#_Toc512464325)

[Figure 36: Decision Tree results on test dataset- Using selected features 42](#_Toc512464326)

[Figure 37: SOM Map results on train dataset- Using selected features 44](#_Toc512464327)

[Figure 38: SOM Map results on test dataset- Using selected features 44](#_Toc512464328)

[Figure 39: The proposed classification method based on SOM-MQE 45](#_Toc512464329)

[Figure 40: SOM MQE results on training dataset – Healthy data as baseline - Using selected features 46](#_Toc512464330)

[Figure 41: SOM MQE results on training dataset – Class 6 as baseline - Using selected features 47](#_Toc512464331)

[Figure 42: SOM-MQE results on test dataset- Class 6 as baseline- Using selected features 47](#_Toc512464332)

# List of Tables

[Table 1: Input Data (Given Data) 10](#_Toc512464642)

[Table 2: SVM results 36](#_Toc512464643)

[Table 3: Results of all methods (moving window was used for removing outliers**)** 48](#_Toc512464644)

# Abstract

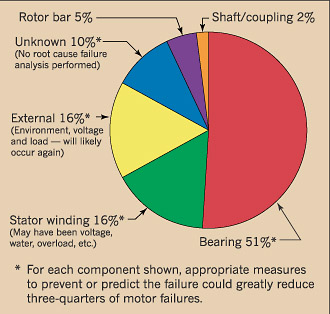
Rolling element bearing is an indispensable component in rotary machinery systems, and quite usually a critical one that costs significant expense and time when it fails. The research interest in developing improvement methods for bearing health monitoring and prognostics is warranted by the prevalent use of bearings and the impact bearing failure has on logistics, maintenance costs and downtime for various applications. This project is to assess and diagnosis bearing health using vibration data.

The first step of the process is data visualization where we plot complete data to have detailed look. The next step is data pre-processing which includes removing the outliers from the data and normalizing the data. For removing the outlier’s various methods like moving window, Kalman filter and combination of both are used. After pre-processing the data, feature extraction is performed to extract time domain features and frequency domain features. Then using Fisher criterion, best features are selected and model is trained. After training the model, cross-validation is done to validate the training model. For classification of data, support vector machine (SVM) and various other methods are used. Finally, the test data is classified supported by training model. Future works look to investigate various other methods that can be used to diagnosis bearing failures accurately.

# Background

Roller element bearings are one of the key components for rotating machinery such as machine tools, motors, gearboxes, turbomachinery etc. Bearings fail due to many reasons like inadequate lubrication, overloading, and installation errors. The research interest in developing improved methods for bearing health monitoring and prognostics is warranted by the prevalent use of bearings and the impact bearing failure has on logistics, maintenance costs and downtime for various applications.

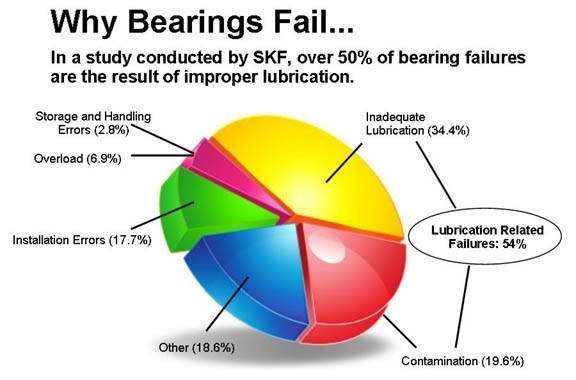
Figure 1: Bearing Failures Percentage



The above pie chart shows different failure modes of an AC servo motor. As we can see, the major share of motor failures is caused because of the bearings failure.

The below pie chart shows various causes for the bearing failures. We can see that major bearing failures are caused due to inadequate lubrication and installation errors.

Figure 2: Typical Bearing Failures



To monitor bearing health condition, various techniques have been applied including vibration analysis, oil debris and acoustic emission. Among these techniques, vibration analysis is currently most established with various signal processing methods employed to analyze the fault signatures in high-resolution vibration waveform. An inevitable challenge for bearing monitoring is the multiple failure modes and their combinations which requires advanced analytical methods to extract most relevant information and identify failure modes with high accuracy.

## **Data Collection**

To evaluate analytical methods for bearing fault diagnosis, a test-bed is set up to collect vibration data under different bearing fault conditions. An SKF 32208 tapered roller bearing (TRB) is used and an accelerometer is installed in the orthogonal direction of its housing to measure the axial vibration. A PCM3178-HG-B DAQ card, embedded in Advantech UNO-2160 box, is used to convert analog waveform to digital data.

During the tests, seven bearings with induced defects and a new bearing with no defect are used in turn to generate data under eight different conditions. The bearing models are the same, thus same specification and geometry parameters. The fault conditions are roller defect, inner-race defect, outer-race defect, the three combinations of any two independent faults and the combination of all three.

## **Typically used techniques for bearing health analysis**

* **Time Waveform Analysis:** By mounting accelerometers at strategic points on bearings, we can measure acceleration and derive velocity. These velocity and acceleration measurements are recorded, analyzed and displayed as tables and plots by condition monitoring equipment. A plot of amplitude versus time is called a time waveform. Time waveforms display a short time sample of the raw vibration. Though typically not as useful as other analysis formats, time waveform analysis can provide clues to machine condition that is not always evident in the frequency spectrum and, when available, should be used as part of your analysis program. (SKF, 2018)
* **Frequency Spectral Analysis:** Analyzing the amplitude, frequency, and phase of the sinusoidal signals is referred to as the frequency spectrum analysis of the signal. To extract these parameters, the signal is filtered, digitized, and Fourier transformed to the frequency domain. The amplitude and the phase spectrum of the signal can then be plotted based on these parameters.  (electronicsforu, 2018)
* **Stress Wave Analysis:** Stress wave analysis (SWAN) provides real-time measurement of friction and mechanical shock in operating machinery. This high-frequency acoustic sensing technology filters out background levels of vibration and audible noise and provides a graphic representation of machine health. By measuring shock and friction events, the stress wave analysis technique can detect wear and damage at the earliest stages, prior to increased vibration, and can track the progression of a defect throughout the failure process. (machinerylubrication, 2018)
* **Enveloping:** Envelope-analysis of a bandpass-filtered signal is an established technique for identifying faults in rolling-element bearings. The traditional method uses an analog bandpass-filter plus a rectifier and smoothing circuit. The filter extracts the resonance excited by the hearing fault from the frequency spectrum; the detector detects the envelope of the corresponding time-signal.

# Process Flow

Figure 3: Process Flow Chart

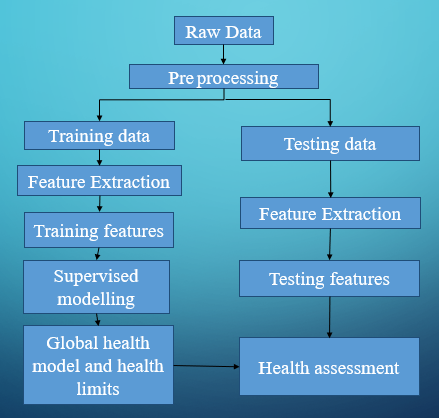
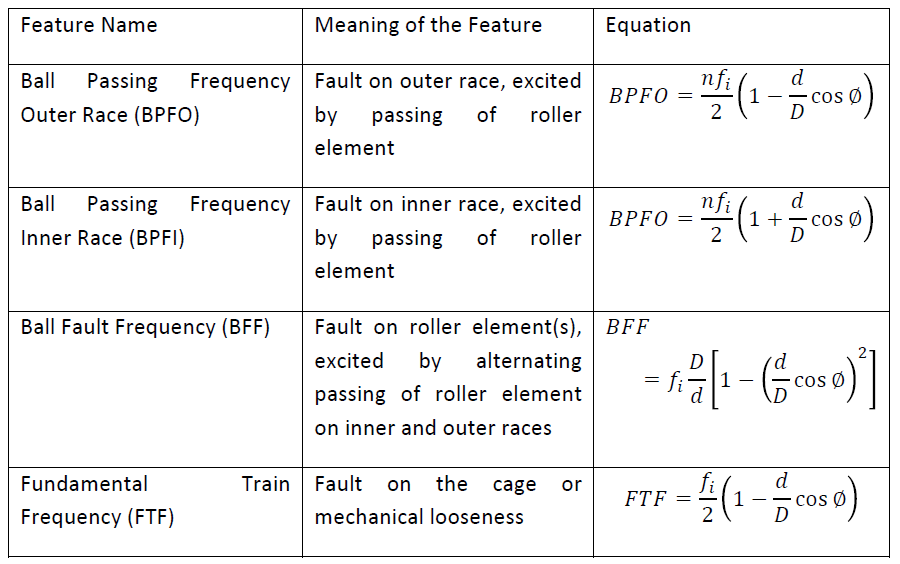


Table 1: Input Data (Given Data)

|  |  |
| --- | --- |
| **Training Data** | **Testing Data** |
| 2048 Sets | 512 Sets |
| 8-Labels | No Labels |
| Fs = 50000; -Sampling rate |  |
| fr = 800/60; -Spindle's rotating frequency (in Hz) |  |
| BPFO = 7.14 \* fr; -outer defect freq. (in Hz) |  |
| BPFI = 9.88 \* fr; - inner  BSF = 5.824 \* fr; - roller |  |

Frequencies related to bearing faults can be calculated as:

**Figure 4: Bearing Frequency Equations**



# Data Pre-processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Data preprocessing plays a major role in the data analysis, often it is observed that the data given is incomplete, inconsistent, or lacking certain behaviors or trends apart from that there are many errors associated with the data collection. Data preprocessing is used to prepare the data for further processing. Data gathering methods are often loosely controlled, resulting in values that are out of range, and there might some missing values in the data. It is very important that we deal with such kind of data otherwise there might be some misleading results. Data preprocessing improves the quality of the data by taking care of all the errors mentioned above.

In the present analysis data preprocessing is focused on the outlier removal. In statistics, outlier is an observation point that is distant from other observations. Generally, outliers in the data occur due to the variability in the measurement or due to some experimental error. An outlier can cause a serious threat to the data analysis.

**Outlier Removal Techniques:**

Some of the techniques used to remove the outliers from the data are as follows:

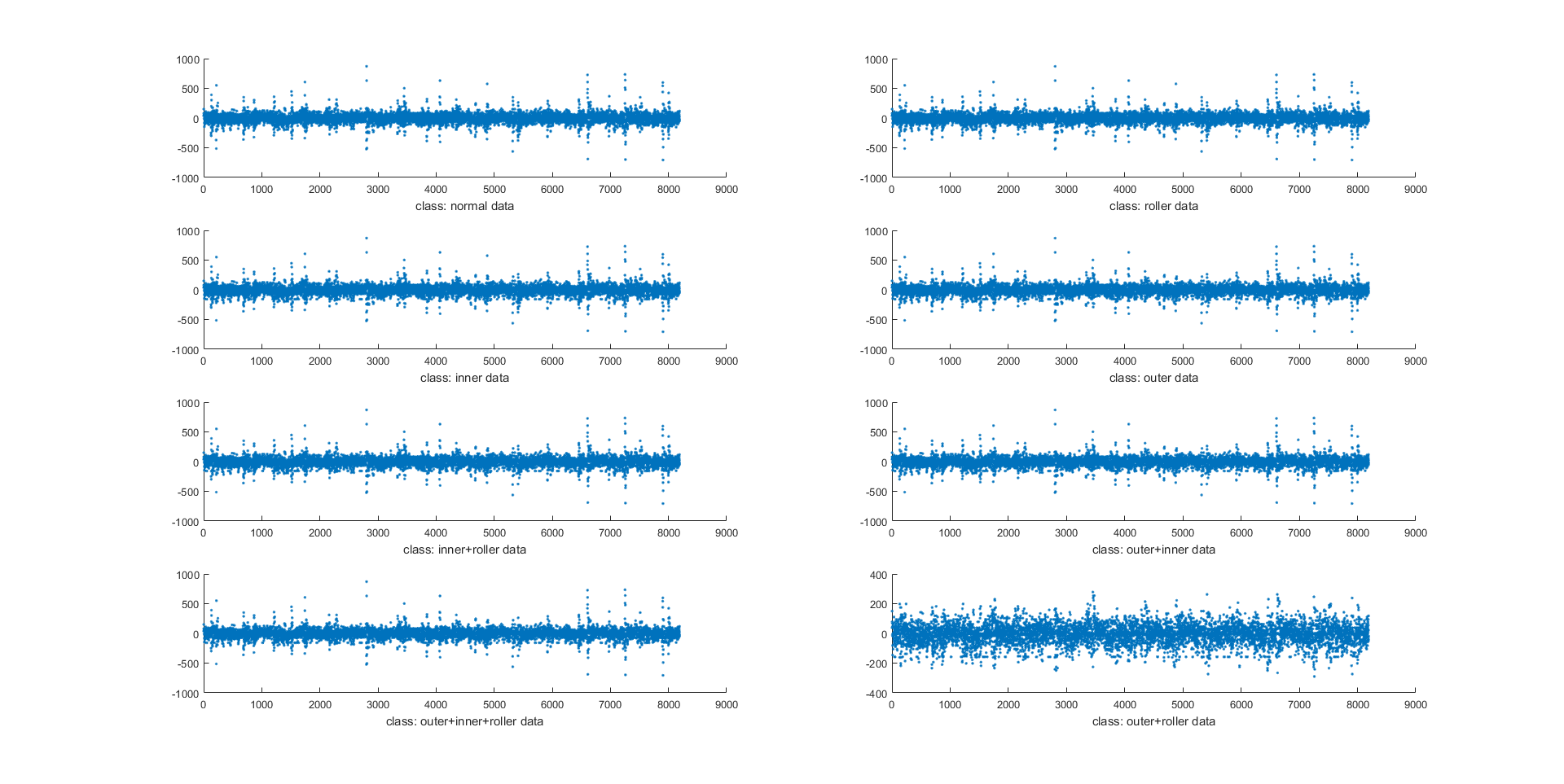
1. Extreme value analysis**:** This is the simplest method in removing the outlier from the given data, this can be done without using the statistical methods to analyze and remove the outlier. Steps used in this method are:
2. We need to focus mainly on the univariate methods.
3. Next step is to visualize the data using the scatterplots, histograms and box and whisker plots and from that, we can analyze the extreme values.
4. Next step is to make an assumption that the data distribution is Gaussian and look for the values more than 2 or 3 standard deviations from the mean or 1.5 times form the first or third quartile.
5. From there we can filter out the outliers from the given training data set and assess the model performance using the filtered data set.
6. Proximity methods:This method uses the clustering techniques to identify the outliers. Steps involved in this method are:
7. Clustering methods such as k-means algorithm can be used to identify the natural clusters.
8. Next step is to identify the cluster centroids.
9. From the cluster centroids, the data instances that are at a fixed distance or percentage distance from the cluster centroids are identified.
10. We can filter out the data outliers from the analysis of the centroids.
11. Projection methods:This method summarizes the given data into two-dimensional data and from there the outliers are removed. Steps involved in this process are:
12. The data is summarized to a two-dimension data using the algorithms such as the PCA, SOM or Sammons mapping.
13. Next step is to visualize the mapping and identify the outliers.
14. Proximity measures form projected values can be used to identify the outliers.

**Tools used for outlier removal**

1. **Moving window average algorithm:** In statistics, a moving average rolling average is a calculation used to analyze data points by creating series of averages of different subsets of the full data sets. (Wikipidea, 2018) It is called as moving mean or rolling mean and is a type of finite impulse filter. Steps involved in moving window average are:
2. From the given data set the first element of the moving average is obtained by taking the average of the initially fixed subset.
3. Then the subset is modified by shifting to the next subset, excluding the previous subset. From the average, the data points that are at a distance from the average are considered as outliers and are removed from the dataset.

A moving average is mostly used in the time series data analysis in smoothening the short-term fluctuations and highlight the long-term fluctuations.

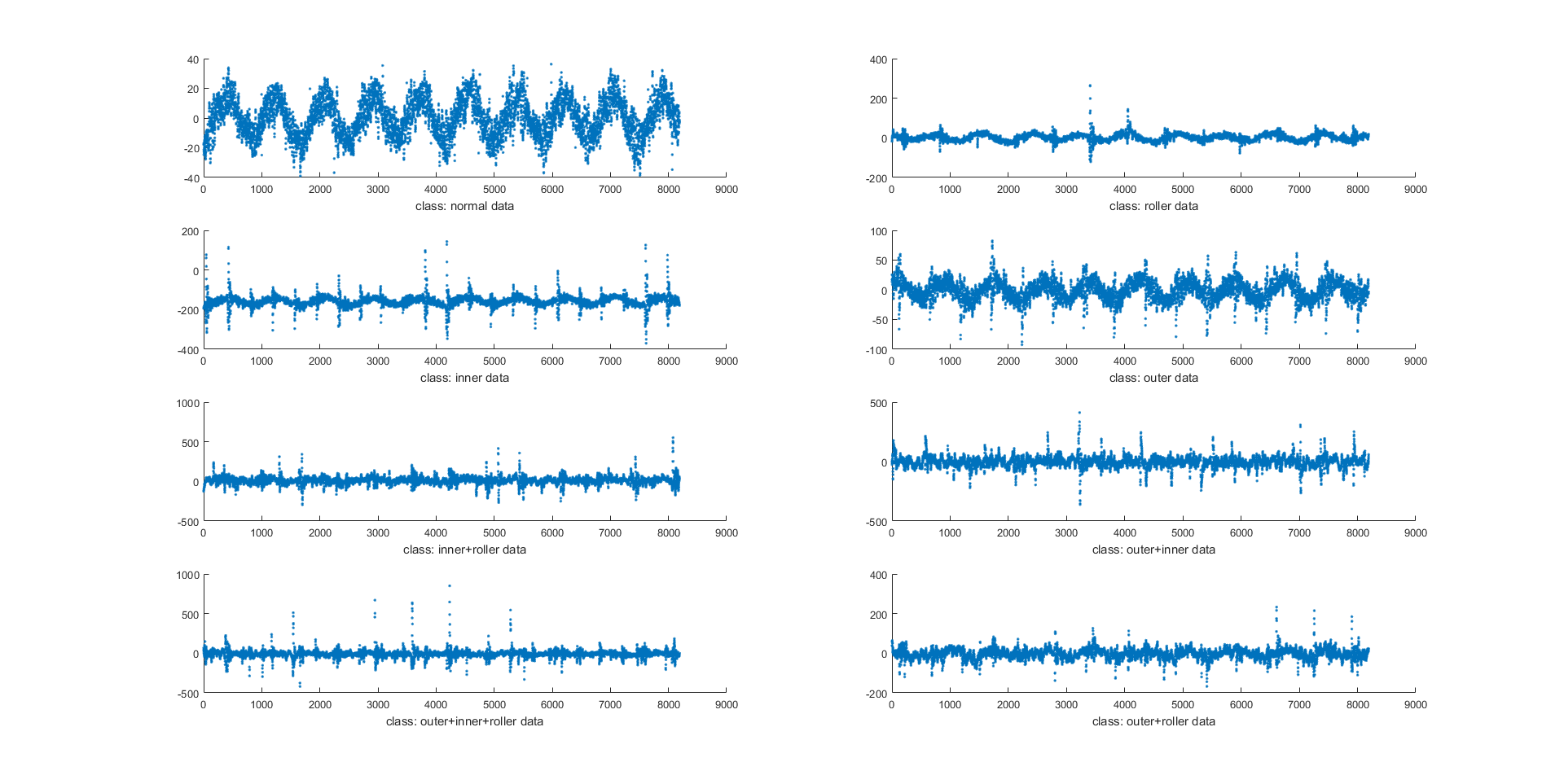
**Figure 5: Data after removing outliers using a moving window**



1. **Kalman filter:** Kalman filter is an optimal estimator that infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive. If all noise is Gaussian, the Kalman filter minuses the mean square error of the estimated parameters. Given only the mean and standard deviation of the noise, the Kalman filter is the best linear estimator. Non-linear estimators may be better. Kalman filter is so popular because it gives good results in practice due to optimality and structure, a convenient form for online real-time processing, easy to formulate and implement given a basic understanding and measurement equations need not be inverted.

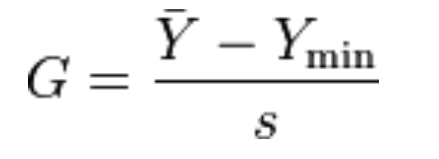
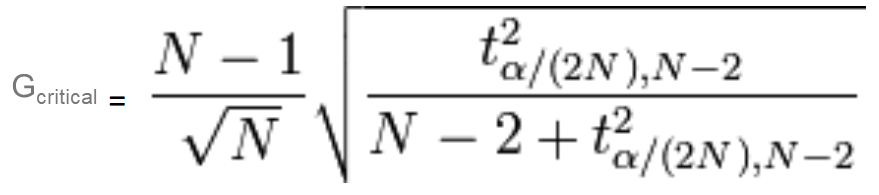
The data given for the bearing is the accelerometer sensor, which is prone to Gaussian noise. Here the acceleration can be estimated from the noisy data. This is done by computing gain. Kalman filter algorithm has two parts, prediction, and update. Prediction is where the Kalman filter predicts the next value using state space equations and later make a correction to the predictions using measured value. (Lacey, 2018)

**Figure 6: Data after removing outliers using Kalman Filter**



From the above-filtered data, we can see that for normal data, the predictions of the Kalman filter looks well. But when the Kalman filter is not tuned properly, there are chances that it removes crucial information from the given data.

1. **Grubbs’ Test:** Grubbs' test is used to detect a single outlier in a univariate data set that follows an approximately normal distribution. This outlier is expunged from the dataset and the test is iterated until no outliers are detected. However, multiple iterations change the probabilities of detection, and the test should not be used for sample sizes of six or fewer since it frequently tags most of the points as outliers. (nist, 2018)

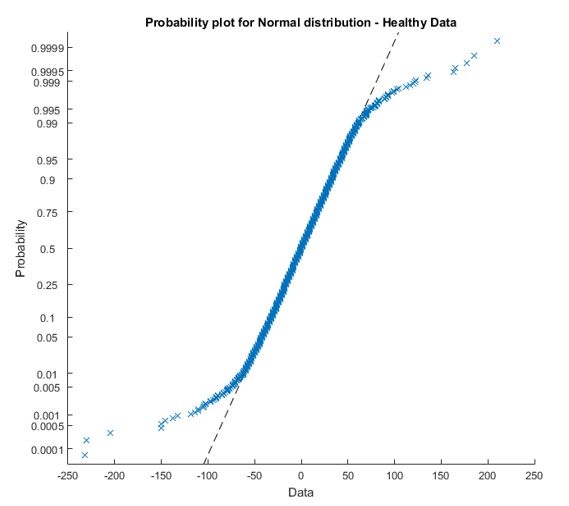
 

tα/(2N), N−2 is the upper critical value of a t-distribution.

Guest < Gcritical: keep the point in the data set; it is not an outlier.

Gtest > Gcritical: reject the point as an outlier.

**Figure 7: Probability plot for Normal Distribution - Healthy Data**



**Figure 8: Histogram plot for Healthy Data**

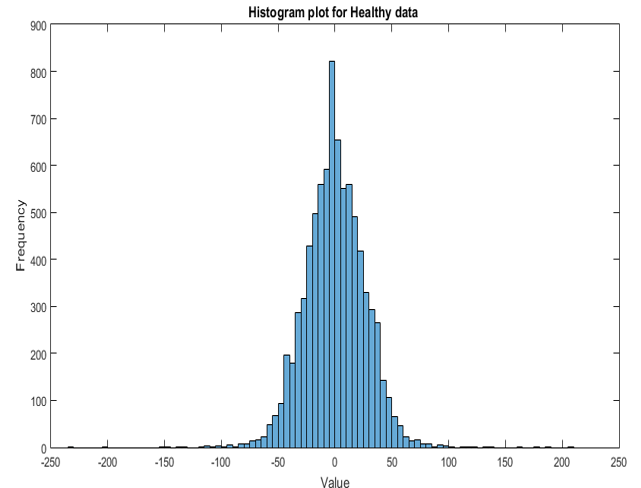
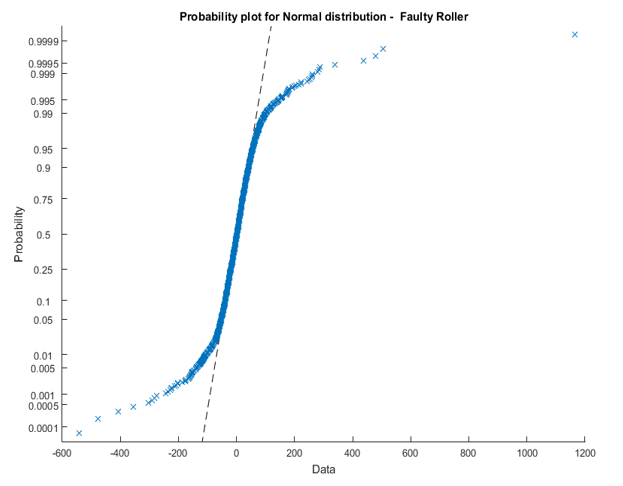
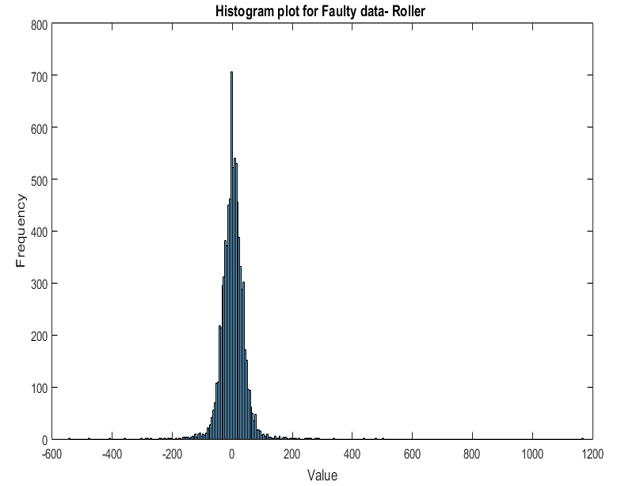


Figure shows healthy data probability plot for normal distribution. Figure shows histogram plot for healthy data.

**Figure 9: Probability Plot for Normal Distribution – Faulty Roller Data**

**Figure 10: Histogram plot for Faulty Data**

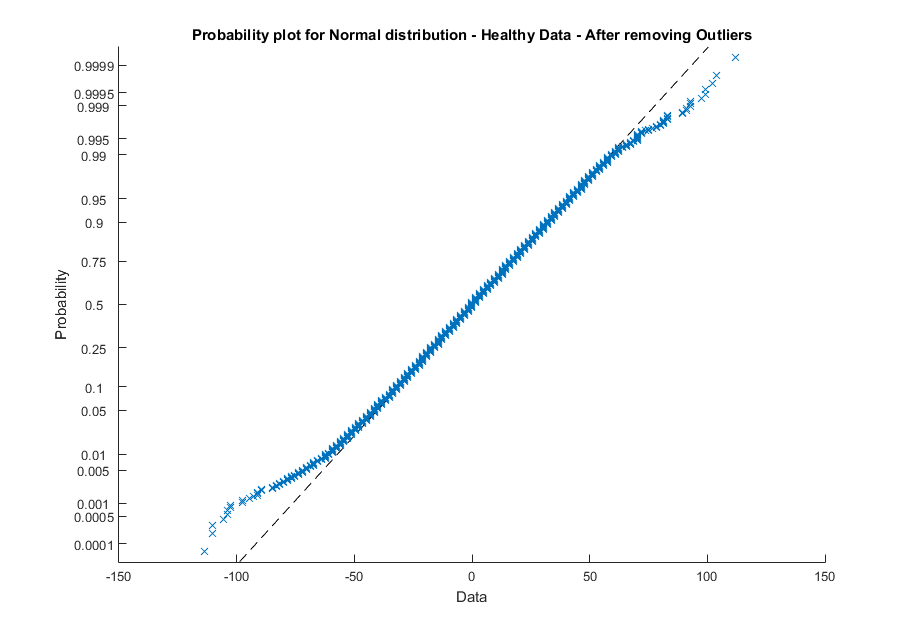


The outliers are filtered by Grubbs’ test which we can see in figure 11.

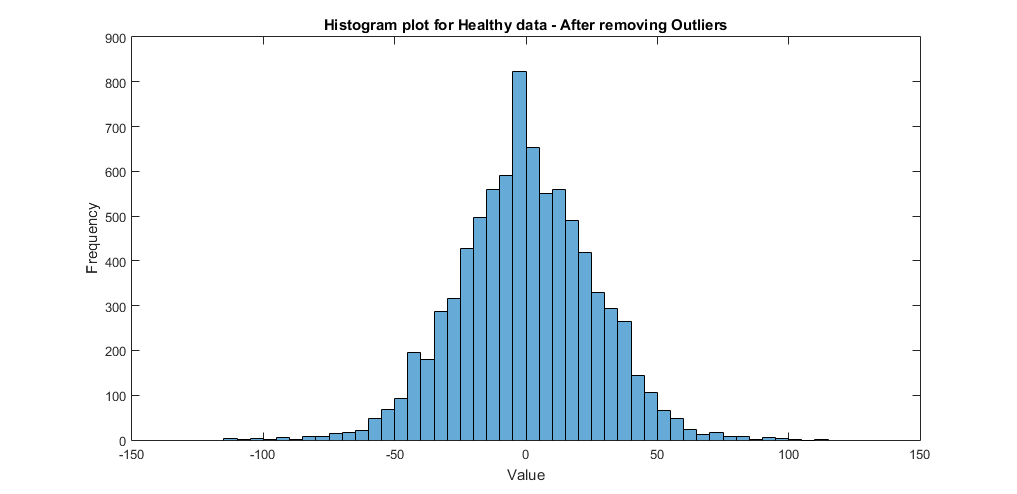
**Figure 11: Data after removing outliers using Grubbs’ test.**



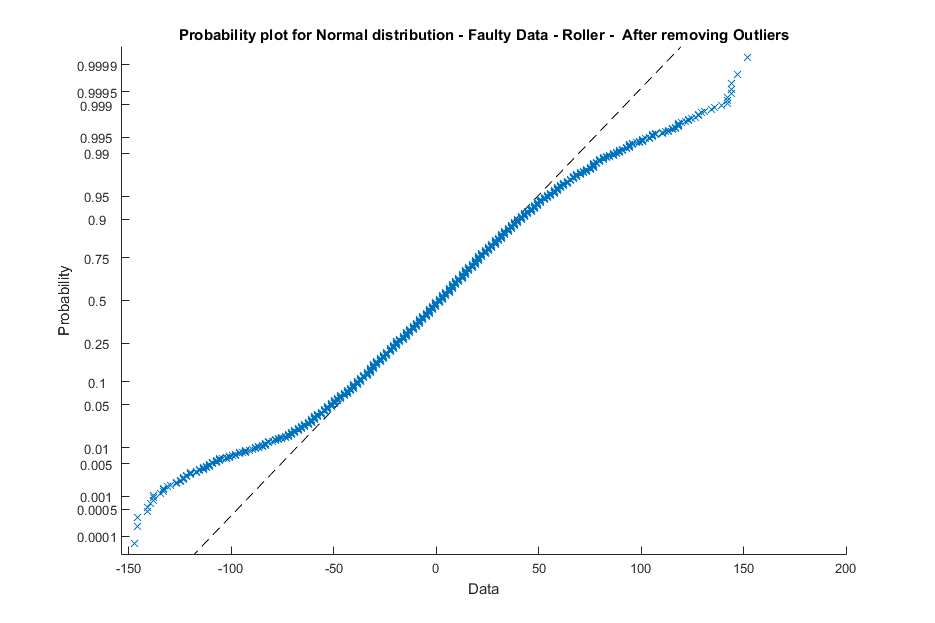
**Figure 12: Probability Plot for Normal Distribution – Faulty Roller Data**



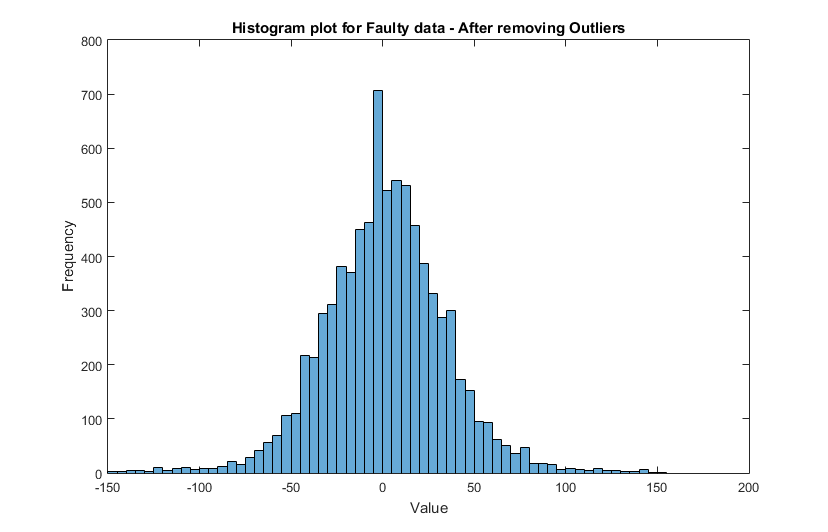
**Figure 13: Histogram plot for Healthy Data -After removing Outliers**



**Figure 14: Probability Plot for Faulty Data after removing outliers.**



**Figure 15: Histogram plot for Faulty data – After removing outliers**



# Feature Extraction

The next step of the analysis is the feature extraction. For the given classification, it is important to extract the useful information from the given data. As the data is labeled, we can extract the features which appreciate the differences in the classes well. As the given signal is time domain data, the following features can be extracted for Time domain. We can also convert the signal to the frequency domain and extract frequency features.

* Time domain features
  + Mean, Variance, Skewness, Kurtosis, RMS, Peak to Peak
* Frequency domain features
  + Amplitudes of Envelope spectrum at BSF, BPFI, and BPFO and their first 3 harmonics
  + 1- 3 X Harmonics shaft

## **Time domain features:**

1. Mean: Average or mean value of the given signal
2. Variance: is the expectation of the squared deviation of a random variable from its mean. Informally, it measures how far a set of (random) numbers are spread out from their average value. A variance is an important tool in the sciences, where statistical analysis of data is common. The variance is the square of the standard deviation
3. Skewness is a measure of the asymmetry of the [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) of a [real](https://en.wikipedia.org/wiki/Real_number)-valued [random variable](https://en.wikipedia.org/wiki/Random_variable) about its mean. The skewness value can be positive or negative, or undefined. The qualitative interpretation of the skew is complicated and unintuitive. Skew does not refer to the direction the curve appears to be leaning; in fact, the opposite is true. For a [unimodal](https://en.wikipedia.org/wiki/Unimodal) distribution, negative skew indicates that the tail on the left side of the probability density function is [longer](https://en.wikipedia.org/wiki/Long_tail) or [fatter](https://en.wikipedia.org/wiki/Fat-tailed_distribution) than the right side – it does not distinguish these two kinds of shape. Conversely, positive skew indicates that the tail on the right side is longer or fatter than the left side. In cases where one tail is long but the other tail is fat, skewness does not obey a simple rule. For example, a zero value means that the tails on both sides of the mean balance out overall; this is the case for asymmetric distribution but is also true for an asymmetric distribution where the asymmetries even out, such as one tail being long but thin, and the other being short but fat
4. Kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable. In a similar way to the concept of skewness, kurtosis is a descriptor of the shape of a probability distribution and, just as for skewness, there are different ways of quantifying it for a theoretical distribution and corresponding ways of estimating it from a sample from a population. Depending on the particular measure of kurtosis that is used, there are various interpretations of kurtosis, and of how particular measures should be interpreted.
5. Root mean square (RMS) is defined as the square root of the mean square (the arithmetic mean of the squares of a set of numbers). The RMS is also known as the quadratic mean and is a particular case of the generalized mean with an exponent. RMS can also be defined for a continuously varying function in terms of an integral of the squares of the instantaneous values during a cycle.
6. Peak to peak is the difference between the maximum and minimum values in time signal.

**Frequency domain features:**

The Fourier transform takes a time-based pattern, measures every possible cycle and returns the overall amplitude, offset and rotation speed for every cycle. The Fourier transform is a mathematical [function](https://simple.wikipedia.org/wiki/Function_(mathematics)) that can be used to show the different parts of a continuous signal. It is mostly used to convert from the [time domain](https://simple.wikipedia.org/wiki/Time_domain) to frequency domain. Fourier transforms are often used to calculate the [frequency](https://simple.wikipedia.org/wiki/Frequency) [spectrum](https://simple.wikipedia.org/wiki/Spectrum) of a [signal](https://simple.wikipedia.org/wiki/Signal) that changes over time.

A Fourier transform shows what frequencies are in a signal. For example, consider a [sound](https://simple.wikipedia.org/wiki/Sound) wave which contains three different musical [notes](https://simple.wikipedia.org/wiki/Note): A, B, and C. Making a [graph](https://simple.wikipedia.org/wiki/Graph) of the Fourier transform of this sound wave (with the frequency on the x-axis and the intensity on the y-axis) will show a peak at each frequency which corresponds with one of the musical notes.

Many signals can be created by adding together cosines and sines with varying [amplitudes](https://simple.wikipedia.org/wiki/Amplitude) and frequencies. The Fourier transform plots the amplitudes and [phases](https://simple.wikipedia.org/wiki/Phase) of these cosines and sines against their respective frequencies.

Fourier transforms are important because many signals make more sense when their frequencies are separated. In the audio example above, looking at the signal with respect to time does not make it obvious that the notes A, B, and C are in the signal. Many systems do different things at different frequencies, so these kinds of systems can be described by what they do to each frequency.

## **Envelope Analysis:**

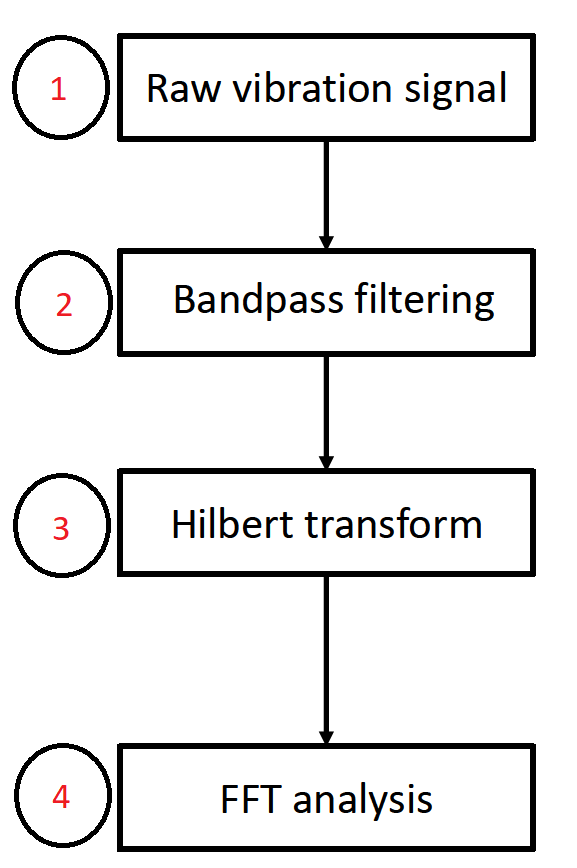
Envelope-analysis of a signal is an established technique for faults diagnosis in rolling-element bearings. Figure 3-1 shows all necessary steps required to perform an envelope analysis on the signal. These steps are summarized as follows:

* 1- Preprocessing data
* 2- Choosing a frequency range for filtering data- based on FFT or Spectral Kurtosis analysis
* 2- Bandpass data around chosen frequency
* 3- Use Hilbert transform for finding the upper edge of the signal
* 4- Take FFT of Hilbert transform of the signal

The most important step in Envelope analysis of a signal is about choosing a suitable bandpass frequency. Selection of a poor window can lead to failure in detecting faults.

Here we will investigate two different methods for finding bandpass frequency of a signal.

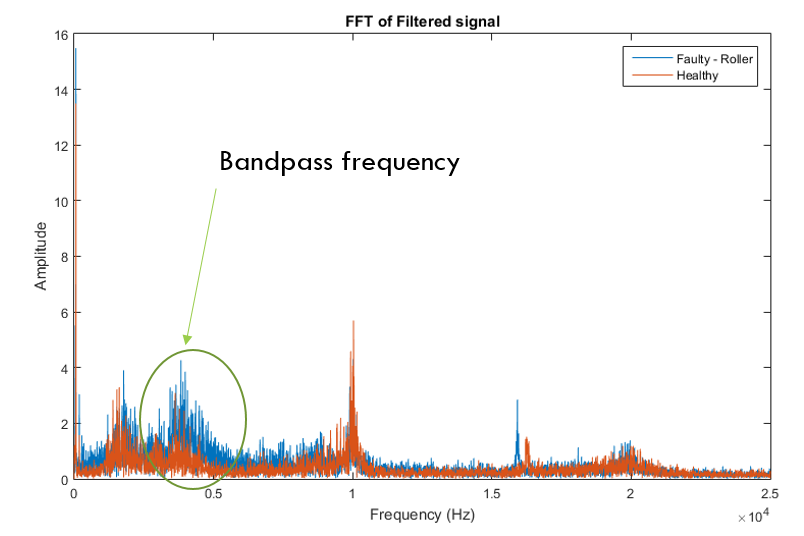
**Figure 16: Envelope analysis process on a vibration signal**



## **Using FFT analysis**

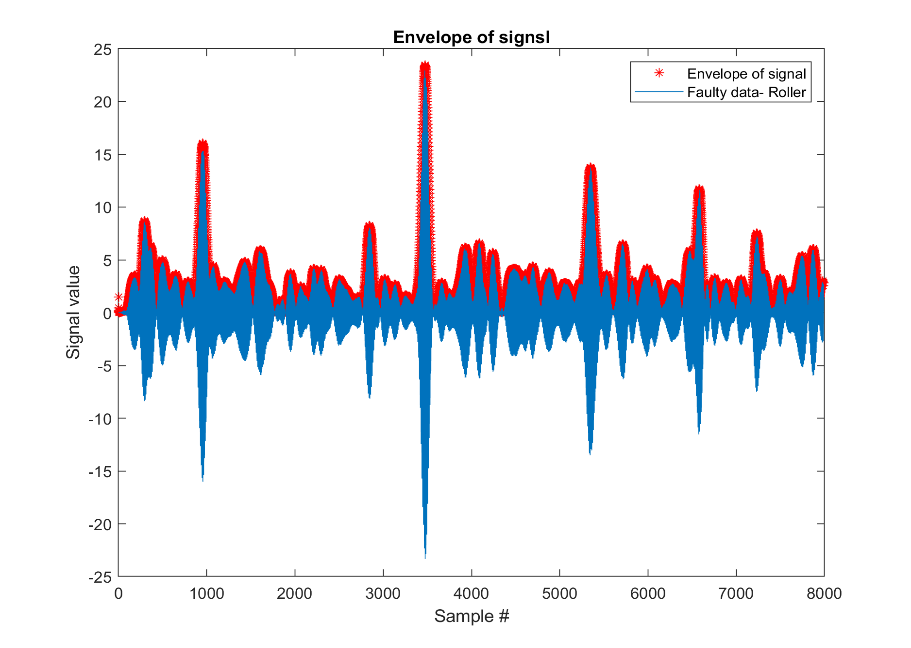
In this method, FFT analysis is performed on both healthy and faulty signal and results are compared together. Those frequencies that show a significant difference between the amplitude of healthy and faulty signals must be used as bandpass frequency. Figure 3-2 shows FFT analysis on healthy and faulty (Roller fault) signals. As is shown, at frequencies around 5000 Hz (the area inside the green circle) a significant difference is observed between two signals and this area must be used as bandpass frequency.

**Figure 17: FFT analysis for finding bandpass frequency**



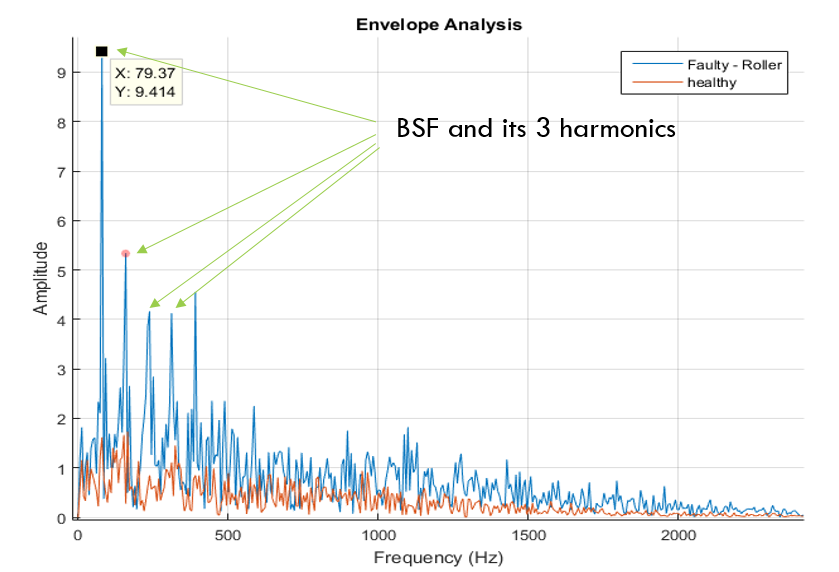
After choosing a proper bandpass frequency and filtering the signal around that frequency, Hilbert transform of the signal is obtained. Figure 3-3 shows absolute value of Hilbert transform of the signal which is called Envelope of that signal.

**Figure 18: Envelope analysis on faulty data**



After finding Envelope of a signal, FFT of that signal is obtained. Figure 3-4 shows FFT results on a sample faulty data (Roller fault). As was observed, BSF and its 3 harmonics are clearly distinguishable. We later use these amplitudes as features to detect roller faults.

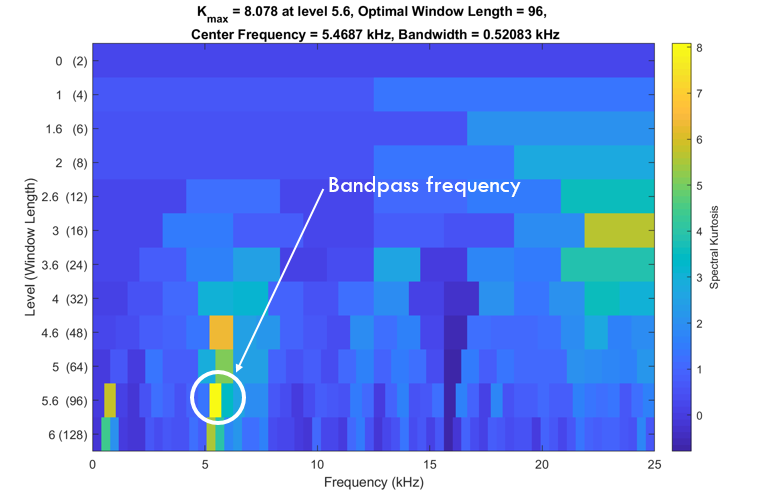
**Figure 19: Envelope analysis on faulty data – using FFT for finding bandpass frequency**



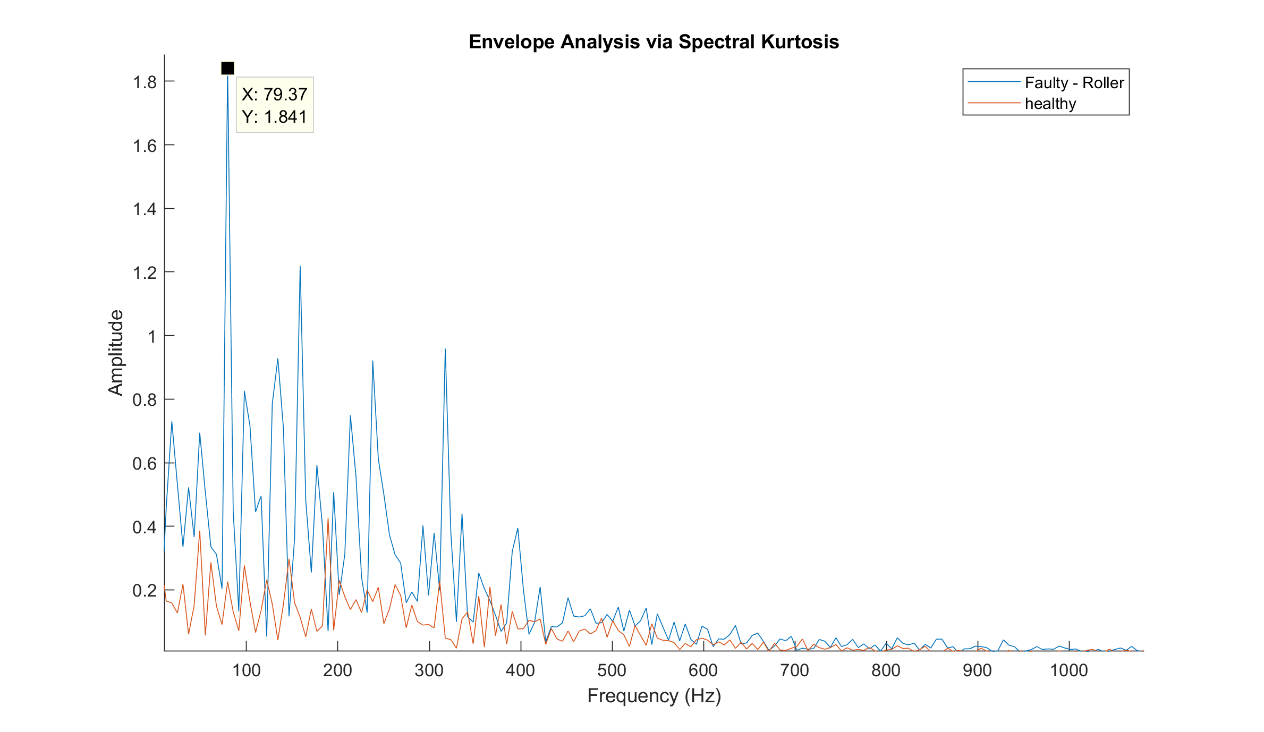
## **Spectral Kurtosis**

Spectral Kurtosis is a statistical tool which can indicate the presence of series of transients and their locations in the frequency domain. Here we use this technique to find bandpass frequencies for filtering signal prior to using Envelop analysis. Figure 3-5 shows Spectral Kurtosis of a sample faulty signal (Roller fault). In this analysis, transient frequency is shown in yellow color (inside the white circle). Signal is filtered around this bandpass frequency and after Envelope analysis the results are given in figure 3-6. As is observed, BSF (77.65) and its harmonics are observed in the signal which indicate 100 % fault diagnosis.

**Figure 20: Spectral Kurtosis analysis for finding bandpass frequency**

****

**Figure 21: Envelope analysis on faulty data - using Spectral Kurtosis for finding bandpass frequency**

****

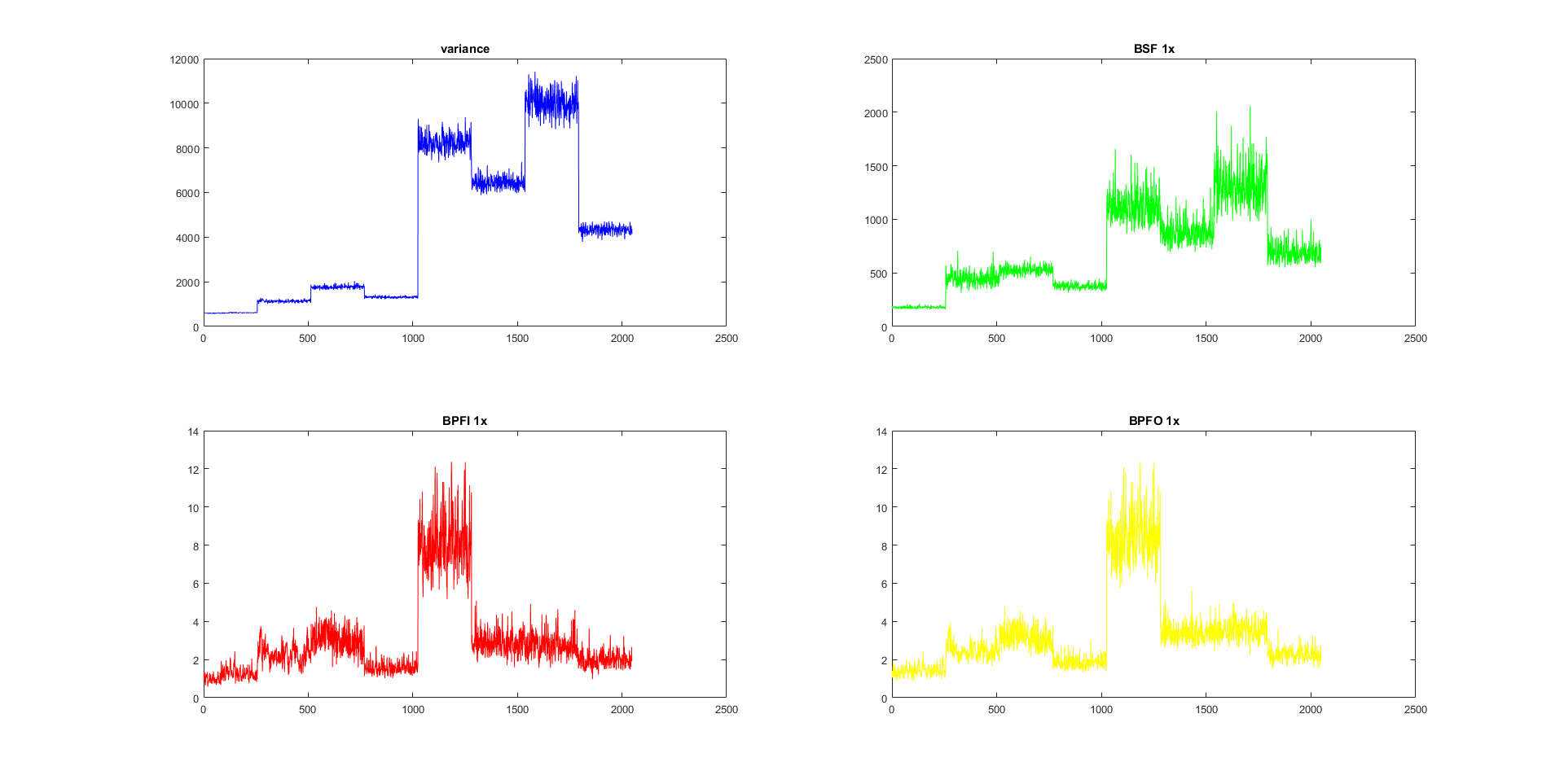
# Feature selection

Once the features are extracted, we need to select the features which plays a major role in classification. By choosing bad feature, it may result in poor results of the classification. We have used the Fisher Criterion and Principle component analysis.

## **Multi Class Fisher Criterion:**

Fisher criterion is used to get qualified features. It maximizes the distance between the means of the two classes while minimizing the variance within each class. In selection step, fisher discriminant score for each variable is computed and the variables with high scores are selected. Finally, Variance, 1x BSF, 1x BPFI and 1x BPFO are selected for training.

**Figure 22: Multi Class Fisher Criterion**

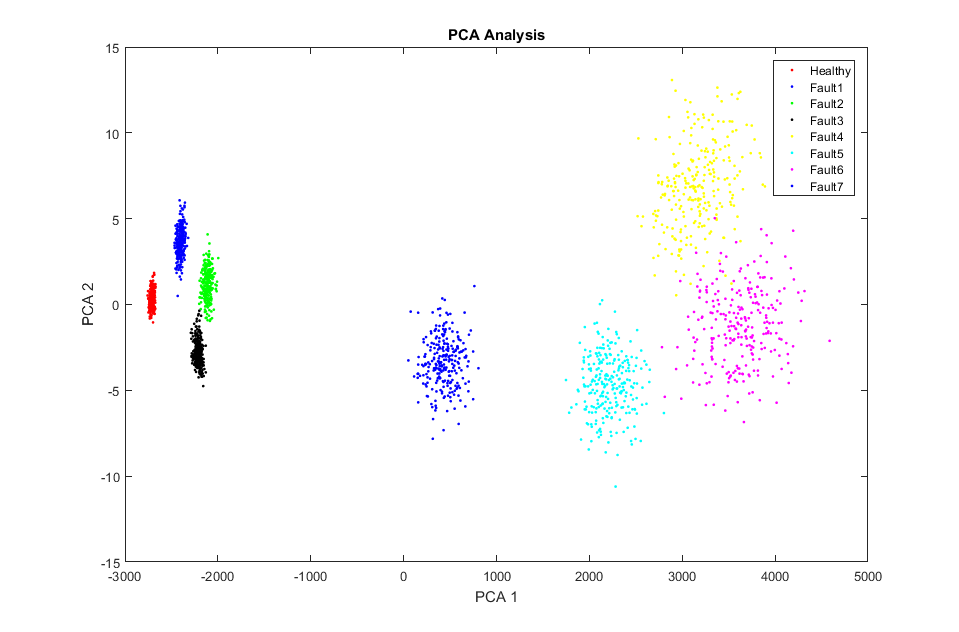


**Principle Component Analysis:**

Principal component analysis is a statistical procedure that uses an [orthogonal transformation](https://en.wikipedia.org/wiki/Orthogonal_transformation) to convert a set of observations of possibly correlated variables into a set of values of [linearly uncorrelated](https://en.wikipedia.org/wiki/Correlation_and_dependence) variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible [variance](https://en.wikipedia.org/wiki/Variance) (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is [orthogonal](https://en.wikipedia.org/wiki/Orthogonal) to the preceding components. The resulting vectors are an uncorrelated [orthogonal basis set](https://en.wikipedia.org/wiki/Orthogonal_basis_set). PCA is sensitive to the relative scaling of the original variables.

In this part totally 12 principle components are formed and top 2 components are selected.

**Figure 23: Principal Component Analysis**



In this chapter, different classification algorithms discussed in the previous section will be used on training and test datasets. Performance of these methods are discussed on both datasets and effectiveness of feature selection on analyzing data will be investigated.

# Learning Models (Classification Techniques)

Support vector machines are supervised learning models associated with learning algorithms that analyze data used for classification and regression analysis. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector machine applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

As we discussed earlier, in this method choosing a suitable Kernel function and Kernel scale are crucial to get accurate results. We have investigated effectiveness of Medium Gaussian, Cubic, linear, and Quadratic as kernel functions and scale factor was changed to get the best performance in each case. In addition, effectiveness of feature selection is studied for all cases. (Wikipedia, 2018)

**K-Fold Cross Validation:** Cross validation is a technique to evaluate the models by splitting the data into parts, of which 25% of the data is used to test the model and 75% of the data is used to rain the model. In K-fold cross validation the training data set is split into k parts, of which k-1 parts are used to train the model and the remaining part is used to test the model. This procedure is repeated for every k parts and the best model is used to train the model finally. We can use a MATLAB K-fold loss function that gives us the classification loss for in-fold observations using a model trained on out of -fold observations. (Open MGL, 2018)

**Figure 24: K-fold Cross Validation**

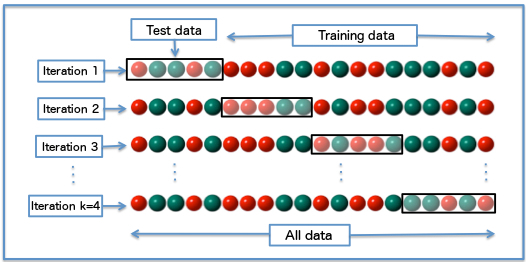


Figure 25 shows confusion matrix for a SVM model trained with Medium Gaussian Kernel function with 10-fold cross validation and consideration of all features. Accordingly, 100 % accuracy is achieved on training data sect which means, training dataset is classified correctly. In this figure labels are shown with numbers 1-8, which are clarified as follows:

Label number 1: Healthy

Label number 2: Roller Fault

Label number 3: Inner Fault

Label number 4: Outer Fault

Label number 5: Inner +Roller Fault

Label number 6: Inner +Outer Fault

Label number 7: Outer+ Inner+ Roller Fault

Label number 8: Outer + Roller Fault

Because we have labels of test data, the same analysis is performed on the test dataset and results are shown in Figure 26. In this figure faults are presented with IDs given here:

Fault 1: Roller Fault

Fault 2: Inner Fault

Fault 3: Outer Fault

Fault 4: Inner+ Roller Fault

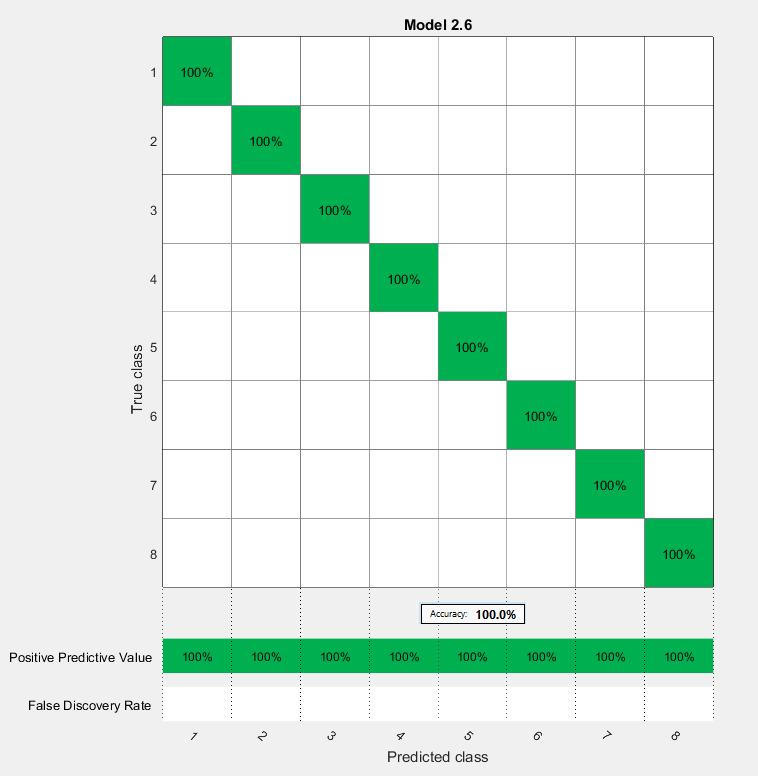
Fault 5: Inner+ Outer Fault

Fault 6: Outer+ Inner+ Roller Fault

Fault 7: Outer+ Roller Fault

Accordingly, accuracy of fault diagnosis is 94.73 % on test data. As is observed, misclassification happens between faults 5,6, and 7.

**Figure 25: SVM results on training dataset – Using all features**



**Figure 26: SVM results on test dataset– Using all features**

A bunch of different colors

Description generated with high confidence

In order to improve accuracy of the model, effectiveness of features that are able to distinguish faults 5,6, and 7 in the dataset must be increased. For this purpose, feature selection is performed. Finally, in addition to four dominant features (Var, BSF, BPFO, BPFI), two other features (second and third harmonics of BPFO) which are stronger for distinguishing faults 5,6, and 7 are also considered in our analysis.

Figure 27 shows results of SVM model on training dataset with 10-fold cross validation. As is shown, accuracy on tanning dataset decreased from 100 % in figure 5-1 to 99.87 % in figure 27. As before, the trained model was used on the test dataset and results are shown in Figure 28. As is shown, accuracy of the model is 99.81 % which is higher than the accuracy - 94.73 % - given in figure 28 which means selecting features carefully can lead to higher accuracy.

**Figure 27: SVM results on training dataset with selected features**

A screenshot of a cell phone

Description generated with very high confidence

**Figure 28: SVM results on the test dataset with selected features**

A bunch of different colors

Description generated with high confidence

Results of all SVM models are given in table 5-1 for both scenarios, 1- considering all features, 2- considering only strong features. In both cases; 10-fold cross validation was used.

Table 2: SVM results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset  Kernel Model | Accuracy – Using All Features | | Accuracy – Using Selected Features | |
| Training Data | Test Data | Training Data | Test Data |
| Medium Gaussian | 100 % | 94.73 % | 99.87 % | 99.81 % |
| Cubic | 100 % | 54 % | 99.6 % | 99.61 % |
| Quadratic | 100 % | 63 % | 99.8 % | 99.81 % |
| linear | 100 % | 61.45 % | 99.8 % | 99.81 % |

Based on the results presented in table 5-1, feature selection has a great impact on the accuracy of the results. Between different Kernel functions, Cubic has the worst accuracy on training and test datasets and no significant difference was observed between results of different Kernel functions when features were selected carefully.

## **KNN Method**

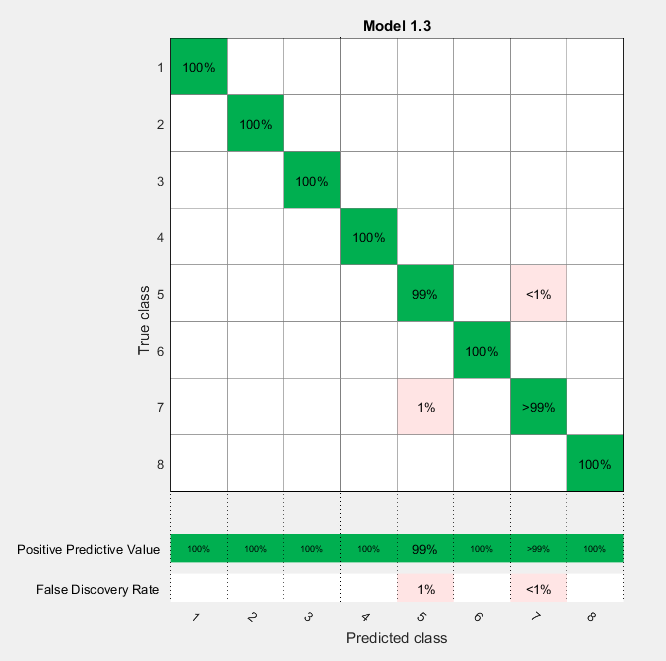
The k-nearest neighbor algorithm is a non-parametric method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). In both cases, the input consists of the k closest training examples in the [feature space](https://en.wikipedia.org/wiki/Feature_space). In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

KNN Method also can be tuned in different ways. Here, we present the best results achieved by using Cosine KNN method for both scenarios; 1- considering all features and 2- considering only useful features.

Figure 29 shows KNN results on training dataset with 10-fold cross validation and considering all features. Accordingly, 99.9 % is achieved in this case. Figure 30 shows results of the trained model on the test dataset. It is shown that 99.23 % accuracy is obtained in this case and misclassification occurs between faults 6 and 4.

As before, the same analysis is performed on a training and test datasets with selected features. Figure 31 and 32 show obtained results on training and test datasets separately and 98.8 % and 90.03 % accuracy are achieved accordingly. This result shows that for this dataset considering all features for KNN method leads to a better accuracy on both training and test datasets.

**Figure 29: KNN results on training dataset – Using all features**

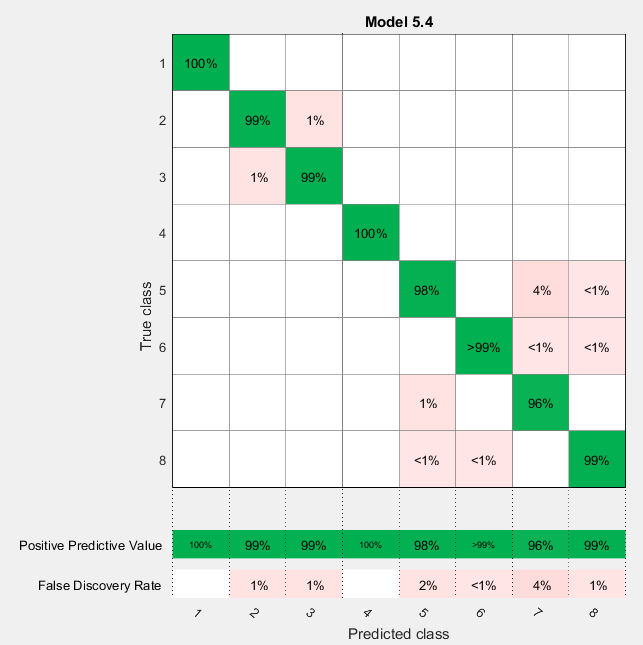


**Figure 30: KNN results on test dataset- Using all features**

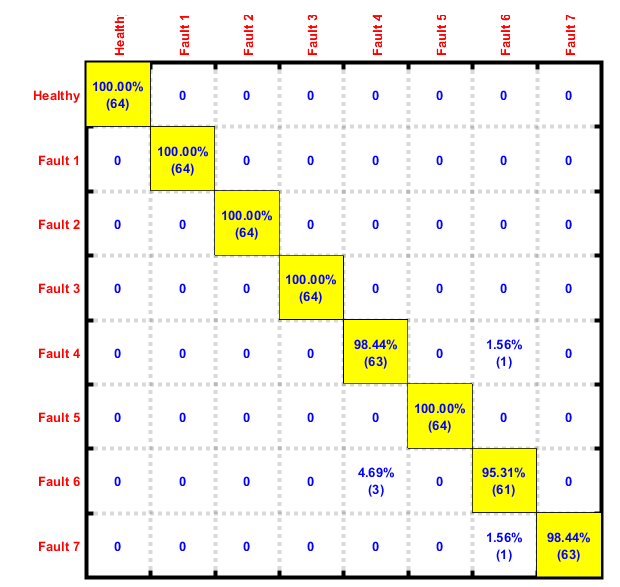
A bunch of different colors

Description generated with high confidence

**Figure 31: KNN results on training dataset – Using selected features**



**Figure 32: KNN results on test dataset- Using selected features**



## **Decision Tree Method**

Decision tree learning uses a [decision tree](https://en.wikipedia.org/wiki/Decision_tree) (as a [predictive model](https://en.wikipedia.org/wiki/Predictive_modelling)) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in [statistics](https://en.wikipedia.org/wiki/Statistics), [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, [leaves](https://en.wikipedia.org/wiki/Leaf_node) represent class labels and branches represent [conjunctions](https://en.wikipedia.org/wiki/Logical_conjunction) of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically [real numbers](https://en.wikipedia.org/wiki/Real_numbers)) are called regression trees.

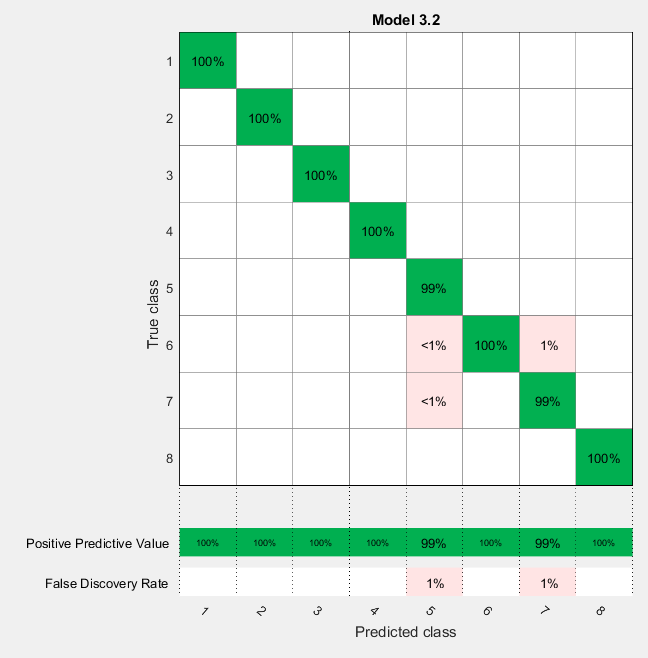
A decision tree is a simple representation for classifying examples. For this section, assume that all of the input [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

Decision tree Method (with a medium size tree) was used for analyzing both training and test datasets for two different scenarios; 1- considering all features and 2- considering only useful features.

Figure 33 shows results on training dataset with 10-fold cross validation and considering all features. 99.8 % accuracy is achieved in this case. Figure 34 shows results of the trained model on the test dataset. It is shown that 88.9 % accuracy is obtained in this case and misclassification occurs between faults 5, 6 and 7.

We also repeated the same analysis on training and test datasets with selected features. Figure 35 and 36 show obtained results on training and test datasets. Accordingly, 98.9 % and 98.83 % accuracy are achieved in this study. These results show that feature selection improves accuracy of classification significantly.

**Figure 33: Decision Tree results on training dataset – Using all features**

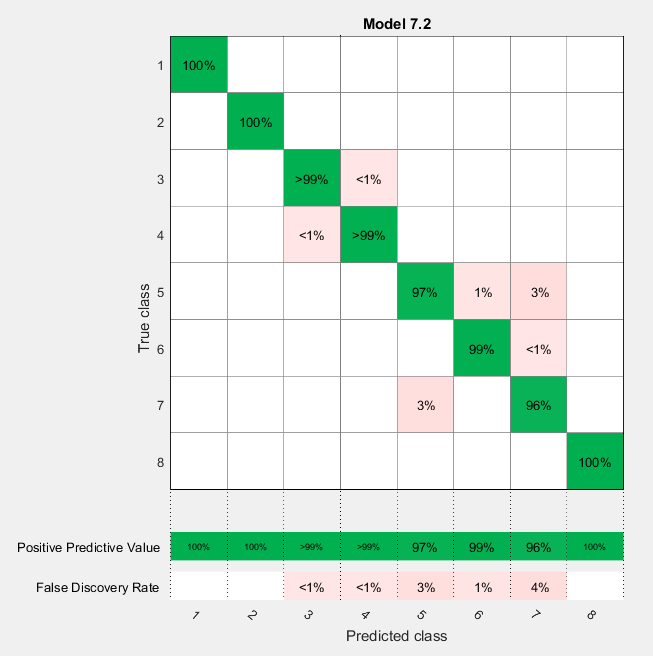


**Figure 34: Decision Tree results on test dataset- Using all features**

**A bunch of different colors

Description generated with high confidence**

**Figure 35: Decision Tree results on training dataset – Using selected features**



**Figure 36: Decision Tree results on test dataset- Using selected features**

A bunch of different colors

Description generated with high confidence

## **SOM Map tool**

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANN) that is trained using [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning) to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do [dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction). There are two ways to interpret a SOM. Because in the training phase weights of the whole neighborhood are moved in the same direction, similar items tend to excite adjacent neurons. Therefore, SOM forms a semantic map where similar samples are mapped close together and dissimilar ones apart. This may be visualized by a [U-Matrix](https://en.wikipedia.org/wiki/U-Matrix) (Euclidean distance between weight vectors of neighboring cells) of the SOM. The other way is to think of neuronal weights as pointers to the input space. They form a discrete approximation of the distribution of training samples. More neurons point to regions with high training sample concentration and fewer where the samples are scarce.

Here we used selected features and classified data by using SOM Map tool. Figure 37 shows SOM Map for training data set and Figure 38 shows the results on test data. Labels of different classes are shown besides each map for a better understanding. As is observed, all classes in training and test datasets are separated from each other and accuracy of the tool for both datasets is 100 %. However, boundary between different classes like 5 and 7 is not very clear. In the next section we use SOM-MQE to see the effectiveness of this tool more clearly.

**Figure 37: SOM Map results on train dataset- Using selected features**

A close up of a logo

Description generated with high confidence

**Figure 38: SOM Map results on test dataset- Using selected features**

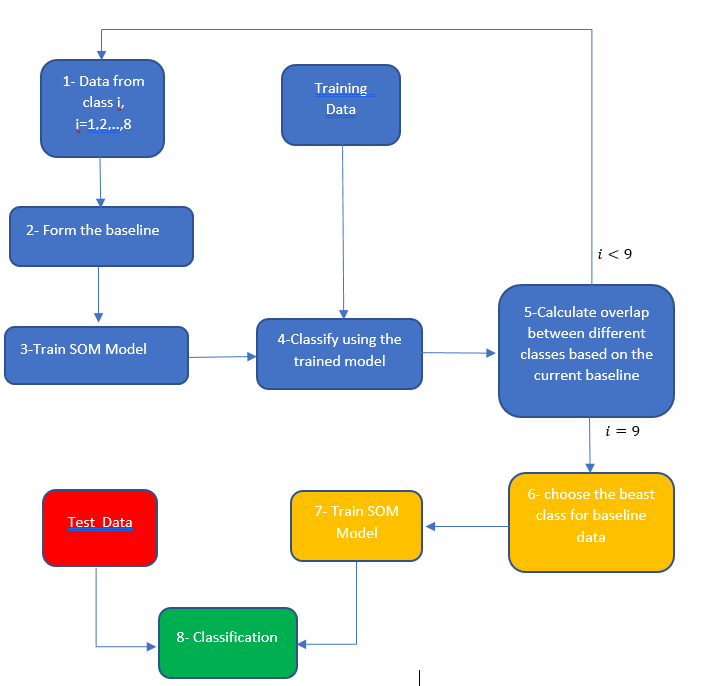
A close up of a logo

Description generated with high confidence

## **SOM-MQE Study**

As was discussed in the previous section, SOM-MQE is one of the most important tools used for health assessment. In this section we use the same method for classification on training and test dataset. The proposed classification algorithm is presented in figure 39.

**Figure 39: The proposed classification method based on SOM-MQE**



In the proposed algorithm the first step is to form a baseline by using only one class of data. In step 3 SOM-MQE model is trained. Then in step 4 the trained model is used to classify all training dataset. Results of step 4 when healthy data is used as the baseline is shown is figure 40. As is shown, all classes are separated very well and accuracy 100 % is obtained on training dataset. However, classes 4 and 6 have overlap (some samples have same MQE values); then we need to try another baseline that minimizes overlap between data. After using all classes as the baseline, we finally chose class 6 as the final baseline for training model in 7. MQE plot for using class 6 as the baseline is shown in figure 40.

In the step 8 we used the trained model to find labels of test data; confusion matrix for these results is shown in figure 42. Accordingly, accuracy 97 % is achieved in this method. However, this method is very simple, new and still lots of improvements can be done to stablish it as a high-performance classification method.

**Figure 40: SOM MQE results on training dataset – Healthy data as baseline - Using selected features**

A screenshot of a video game

Description generated with high confidence

**Figure 41: SOM MQE results on training dataset – Class 6 as baseline - Using selected features**



**Figure 42: SOM-MQE results on test dataset- Class 6 as baseline- Using selected features**

A bunch of different colors

Description generated with high confidence

**Table 3: Results of all methods (moving window was used for removing outliers)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset  methods | | Accuracy – Using All Features | | Accuracy – Using Selected Features | |
| Training Data | Test Data | Training Data | Test Data |
| SVM | Medium  Gaussian | 100 % | 94.73 % | 99.87 % | 99.81 % |
| Cubic | 100 % | 54 % | 99.6 % | 99.61 % |
| Quadratic | 100 % | 63 % | 99.8 % | 99.81 % |
| linear | 100 % | 61.45 % | 99.8 % | 99.81 % |
| KNN | Cosine | 99.9 % | 99.23 % | 98.8 % | 90.03 % |
| Decision Tree | Medium size | 99.8 % | 88.9 % | 98.9 % | 98.83 % |
| SOM - MQE | Proposed | - | - | 100 % | 97 % |

# Conclusions

In this study, Grubbs' test, Kalman filter, and moving window were used to remove outliers. Our study showed that using moving window leads to the best accuracy for classifications. Kalman Filter leads to the worst classification results because it takes average of signal rather than removing outliers. Grubbs’ test also gives good performance in removing outliers.

Different set of features were used to improve classification accuracy. Finally, frequency domain features were selected by using Fisher test. Choosing frequency domain features increases model’s performance on classification significantly.

Results of different methods discussed in this chapter are given in table 3. As was discussed earlier, using selected features increases classification performance significantly. Between different methods that were tried, SVM- linear has the best accuracy - 99.81 % - on the test dataset. Also, no significant difference was observed between different tried Kernel methods. KNN gives the worst accuracy- 90.03- for selected features and its performance is better when a large number of features are used (accuracy 98.8 %). Our study also shows that Decision tree also gives very good results 98.83 % accuracy on test dataset. We also studied performance of MQE for classification and developed a new algorithm for this purpose; the accuracy of this method on test dataset is around 97 % which is considered as a good performance. However, cause our algorithm is new, it can still be improved.

# Future Work

In this study we tried different methods for preprocessing data, features selection, and finally for classification; however, some of these methods can be investigated more. For example, tuning parameters in SVM model is very important and must be investigated deeply. The proposed SOM-MQE method has shown a great potential for classification as well as for health assessment and must be improved and tested on other challenging datasets.

# References

1. a*electronicsforu*. (2018, 4 26). Retrieved from https://electronicsforu.com/technology-trends/frequency-spectrum-analysis-signals
2. Lacey, T. (2018, 4 25). Retrieved from http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf
3. machinerylubrication. (2018, 4 26). *machinerylubrication*. Retrieved from http://www.machinerylubrication.com/Read/501/stress-wave-analysis
4. *nist*. (2018, 4 26). Retrieved from https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h1.htm
5. *Open MGL*. (2018, 4 26). Retrieved from https://www.openml.org/a/estimation-procedures/1
6. SKF. (2018, 4 26). *SKF*. Retrieved from http://www.skf.com/group/services/services-and-solutions/introduction-to-condition-monitoring/time-waveform-analysis-and-fft-spectrum-analysis.html
7. Wikipedia. (2018, 4 26). *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Support\_vector\_machine
8. Wikipidea. (2018, 4 26). Retrieved from https://en.wikipedia.org/wiki/Moving\_average
9. Bechhoefer, E., Menon, P., & Kingsley, M. (2011). Bearing envelope analysis window selection Using spectral kurtosis techniq\4ues. IEEE.
10. Limited, L. S. (2018, April 01). Lubrication System Limited. Retrieved from Lubrications: <http://www.lubricationsystems.ltd.uk/why-bearings-fail.htm>
11. Siegel, D., Al-Atat, H., Shauche, V., Liao, L., & John Snyder, J. Lee (2012). Novel Method for Rolling Element Bearing Health Assessment. Elsevier.
12. Randall, R. B., & Antoni, J. (2010). Roller element bearing diagnostics - A tutorial. ELSEVIER.
13. Tse, P. W., Peng, Y. H., & Yam, R. (2001). Wavelet Analysis and Envelope Detection for Rolling Element Bearing Fault Diagnosis - Their Effectiveness and Flexibilities. ASME - Journal of Vibration and Acoustics.