# PREDCARE: BEARINGS FAULT PREDICTION

A PROJECT REPORT

*Submitted by*

**JOYAL JOSEPH (TKM21MCA-2024)**

**to**

**The APJ Abdul Kalam Technological University**

*In partial fulfillment of the requirements for the award of the degree of*

**MASTER OF COMPUTER APPLICATION**

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2023

# DECLARATION

I undersigned hereby declare that the project report on **PREDCARE: BEARINGS FAULT PREDICTION** , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Natheera Beevi M . This submission represents my ideas in my own words and where ideas or words of others have been included,I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University..

Joyal

Kollam

19-05-2023

**JOYAL JOSEPH**

**DEPT. OF COMPUTER APPLICATIONS TKM COLLEGE OF ENGINEERING**

**KOLLAM**

**2021 - 23**

## CERTIFICATE

This is to certify that the report entitled **PREDCARE: BEARINGS FAULT PREDICTION** submitted by **JOYAL JOSEPH** (TKM21MCA-2024) to the APJ Abdul Kalam Technological University in partial fulfillment of the Masters degree in Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor | Head of the Department | External Examiner

# Acknowledgement

I express my heartfelt gratitude to God Almighty and my parents for their unwavering support and blessings. I sincerely appreciate the time and knowledge generously shared by those who contributed to the success of my project. **Dr. Fousia M Shamsudeen**, Head of the Department of Computer Applications, provided excellent facilities and guidance throughout. I am also thankful to **Prof. Natheera Beevi M** , my project guide, for their continuous motivation and insights. I extend profound thanks to **Prof. Vaheetha Salam**, the project coordinator, for their exceptional support and coordination. My advisor **Prof. Natheera Beevi M** offered expert advice and valuable suggestions, enhancing the quality of my work. I am grateful to the faculty members of the Department of Computer Applications and TKM College of Engineering for their guidance. Lastly, I thank my friends and all others who directly or indirectly contributed to the completion of this project. Their support and assistance have been invaluable. I deeply appreciate the contributions of all individuals involved.

**JOYAL JOSEPH**

# ABSTRACT

Bearing fault prediction is a critical task in ensuring the reliability and performance of rotating machinery. The aim is to identify any abnormalities in bearing behavior before serious damage occurs. In recent years, machine learning techniques, including artificial neural networks and support vector machines, have been utilized to develop precise bearing fault prediction models. These models use different sensor signals, such as vibration and acoustic signals, to detect and categorize various types of bearing faults. Accurate and dependable bearing fault prediction models are necessary to avoid unexpected machinery breakdowns, reduce downtime, and minimize maintenance expenses.

One of the key ideas in bearing fault prediction is to use various sensor signals, such as vibration and speed signals, to detect and predict faults in the bearings of rotating machinery. Vibration signals are widely used because they provide valuable information about the condition of the bearings. They can be used to measure the magnitude, frequency, and waveform of the vibrations caused by the bearings' movement. Speed signals, on the other hand, provide information about the rotational speed of the machinery, which can also be used to identify the existence of bearing faults. By analyzing these signals, machine learning models can identify patterns and trends that indicate the presence of faults, such as cracks, wear, and misalignment, and predict when the bearings may fail. The use of these signals in bearing fault prediction can help prevent unexpected downtime, reduce maintenance costs, and improve the overall efficiency of rotating machinery..

# Contents

# List of Figures

# Chapter 1

# Introduction

**PREDCARE: BEARINGS FAULT PREDICTION** ,Machines and equipment play a crucial role in the success of any industry, regardless of its size. They are responsible for producing the goods and services that the industry offers, and they need to be maintained properly to ensure their optimal performance and longevity. However, maintenance costs can be a significant burden on companies, both economically and physically. Traditional maintenance approaches, such as corrective maintenance, can be expensive, time-consuming, and can lead to unexpected downtime and losses.

The idea behind predictive maintenance is to use data to predict when a machine or equipment is likely to fail, and to perform maintenance before the failure occurs. In particular, predictive maintenance models can be used to predict bearing failures, which are one of the most common causes of machinery breakdowns. By analyzing data on each bearing, such as vibration and temperature data, machine learning algorithms can identify patterns and anomalies that indicate when a bearing is likely to fail.

Early maintenance based on predictive maintenance models can reduce costs by minimizing the need for unplanned downtime and emergency repairs. It can also improve safety by preventing accidents caused by machine failures. Furthermore, it can improve the overall performance of machines and equipment by ensuring that they are running optimally. In summary, predictive maintenance models, and in particular, bearing fault prediction models, are an essential tool for maintaining the reliability and performance of machines and equipment while minimizing costs and reducing the risk of failures.

## 1.1 Problem Statement

Predictive maintenance of rolling parts has become increasingly important in modern production processes, particularly in industries that rely heavily on machines and equipment. Rolling parts, such as bearings, are subjected to constant wear and tear due to friction, temperature fluctuations, and other factors. If not adequately maintained, rolling parts can fail, leading to unexpected machine breakdowns and costly downtime.

Traditional maintenance approaches often involve periodic inspections and routine maintenance based on predetermined schedules, regardless of the equipment's actual condition. This can be expensive and inefficient, as maintenance may be performed when it is not necessary, or it may not be performed when it is needed. Predictive maintenance models offer a more efficient and cost-effective approach to maintenance by using sensory data to monitor the equipment's actual condition and identify potential issues.

Predictive maintenance models rely on advanced data analytics techniques to process large amounts of sensory data collected from rolling parts, such as vibration, temperature, and noise readings. By analyzing this data, predictive maintenance models can identify patterns and trends in the equipment's performance, providing early warning signals of potential issues. This enables operators to take proactive measures, such as performing maintenance before a failure occurs, reducing downtime costs and improving overall productivity.

The development of robust and accurate predictive maintenance models is essential to realizing the benefits of this approach fully. The models must be able to process vast amounts of data accurately, identify potential issues with high precision, and make reliable predictions about equipment failures. Addressing this challenge requires a multidisciplinary approach that combines advanced data analytical techniques, machine learning algorithms, and domain expertise. By developing more effective predictive maintenance models, companies can improve the reliability and efficiency of their equipment, reduce maintenance costs, and increase overall productivity.

## 1.2 Existing System

- 1. The existing system for predictive maintenance of rolling parts often relies on rule-based algorithms or simple statistical models to determine when maintenance is necessary.
- 2. These approaches may be based on pre-defined rules, thresholds, or limits that trigger maintenance actions.
- 3. Rule-based algorithms may be simple to implement but may not be accurate or flexible enough to adapt to changing operating conditions.
- 4. Simple statistical models may use basic statistical methods to identify trends or patterns in the data, but they may not be able to capture complex interactions between different variables.
- 5. A more advanced approach to predictive maintenance involves graphical analysis using data analyzers to monitor fluctuations in sensory data.
- 6. This approach involves visualizing the data in different ways to identify patterns and anomalies that may indicate potential issues.
- 7. Graphical analysis can help operators to understand the equipment's performance better and identify areas that require attention.
- 8. Data analyzers may use techniques such as trend analysis, scatter plots, box plots, and histograms to visualize the data.
- 9. These techniques can help to identify patterns such as sudden changes in the data, periodic fluctuations, or unusual events that may indicate potential issues.
- 10. By using graphical analysis and data analyzers, operators can make more informed decisions about maintenance schedules and repairs, reducing downtime costs, and increasing equipment reliability and efficiency.

## 1.3 Proposed System

1 .An upgrade to the existing system for predictive maintenance of rolling parts involves incorporating machine learning algorithms to analyze the speed, vibration, audio, or their combinations.

2 .Machine learning algorithms can help to improve the accuracy of predictions about when maintenance is needed, leading to better maintenance scheduling and potentially reducing downtime costs.

3 .One approach to machine learning for predictive maintenance involves using Long Short-Term Memory (LSTM) networks, which are a type of recurrent neural network (RNN).

4 .LSTM networks are particularly well-suited to analyzing time-series data, such as speed and vibration readings, and can learn complex patterns and relationships in the data.

5 .Another approach to machine learning for predictive maintenance involves using anomaly detection algorithms, such as one-class SVM or isolation forest.

6 .Anomaly detection algorithms can identify unusual patterns or outliers in the data that may indicate potential issues with the equipment.

## 1.4 Objectives

The goal is to accomplish the following:

- Unified prediction: The primary objective is to predict accurate results on provided dataset of faulty and healthy bearings.
- Characteristic feature extraction: Fine tuning the data for smooth and characteristic usage.
- Increase efficiency: Objectives also include increase efficiency by ensuring that fault prediction is anomaly free. processes.

# Chapter 2

# Literature Survey

Literature review is that the comprehensive study and interpretation of literature that relates to a selected topic.When doing a literature review, research questions are defined, and then relevant literature is sought for and analysed to address these issues. By reanalyzing the study's data, it is possible to acquire fresh insights, which is an advantage of literature reviews. A literature review is both a summary and an explanation of the complete and current state of information on a topic as contained in academic books and journal articles. There are two types of literature reviews you may be required to write in college: one is written as a stand-alone assignment in a course, while the other is done as an introduction to or preparation for a longer piece of writing, typically a thesis or research report. The primary objective and perspective of your review, as well as the hypothesis or thesis argument you develop, depend on the type of review you are writing. You can learn the distinctions between these two types by reading published literature reviews or the introductory chapters of theses and dissertations in your subject area. Note the framework of their arguments and the manner in which they approach the issues.

## 2.1 Purpose of the Literature Review

1. It chooses top-notch research papers or studies that are pertinent, significant, important, and valid and summarises them into a single comprehensive report to provide readers with quick access to information on a certain issue.

2. By requiring them to describe, assess, and compare original research in this particular field, it gives researchers who are starting their research in a new area a great place to start.

3. It makes sure that researchers don't repeat already completed studies.

4. It can indicate potential directions for future research or suggest topics to concentrate on.

5. It emphasises the important findings.

6. It points up gaps, discrepancies, and inconsistencies in the literature.

7. It offers a helpful critique of the methods and strategies used by other researchers.

## 2.2 Related Works

### 2.2.1 Bearing fault Diagnosis

. Zhang, S. et al. [2] The paper systematically summarize existing literature on bearing fault diagnostics with deep learning (DL) algorithms. While conventional machine learning (ML) methods, including artificial neural network, principal component analysis, support vector machines, etc., have been successfully applied to the detection and categorization of bearing faults for decades, recent developments in DL algorithms in the last five years have sparked renewed interest in both industry and academia for intelligent machine health monitoring. In this paper, we first provide a brief review of conventional ML methods, before taking a deep dive into the state-of-the-art DL algorithms for bearing fault applications. Specifically, the superiority of DL based methods are analyzed in terms of fault feature extraction and classification performances; many new functionalities enabled by DL techniques are also summarized. In addition, to obtain a more intuitive insight, a comparative study is conducted on the classification accuracy of different algorithms utilizing the open source Case Western

Reserve University (CWRU) bearing dataset. Finally, to facilitate the transition on applying various DL algorithms to bearing fault diagnostics, detailed recommendations and suggestions are provided for specific application conditions. Future research directions to further enhance the performance of DL algorithms on health monitoring are also discussed. C. -C. Wang, et al. [8] propose a machine-learning-based fault diagnosis approach for condition monitoring on the constant-speed rotating machines via vibration signals. There are mainly five phases in our approach, i.e., vibration signal measurement, discrete-wavelet-transformation-based preprocessing, feature extraction, base-line encoding, and fuzzy neural network. The advantage of this approach can identify the condition and faults of machine without sufficient diagnosis knowledge. Experimental results have demonstrated this approach is a useful tool for condition monitoring application. Hakim, Mohammed, et al. [10] Rolling bearing fault detection is critical for improving production efficiency and lowering accident rates in complicated mechanical systems, as well as huge monitoring data, posing significant challenges to present fault diagnostic technology. Deep Learning is now an extraordinarily popular research topic in the field and a promising approach for detecting intelligent bearing faults. This paper aims to give a comprehensive overview of Deep Learning (DL) based on bearing fault diagnosis. The most widely used DL algorithms for detecting bearing faults include Convolutional Neural Network, Recurrent neural network, Autoencoder, and Generative Adversarial Network. It discusses a variety of transfer learning architectures and relevant theories while summarises, classifies, and explains several publications on the subject. The research area's applications and problems are also addressed.

### 2.2.2 Predictive maintenance

Zhang, Weiting, et al. [3]With the tremendous revival of artificial intelligence, predictive maintenance (PdM) based on data-driven methods has become the most effective solution to address smart manufacturing and industrial big data, especially for performing health perception (e.g., fault diagnosis and remaining life assessment). Moreover, because the existing PdM research is still in primary experimental stage, most works are conducted utilizing several open-datasets, and the combination with specific applications such as rotating machinery is especially rare. Hence, in this paper, we focus on data-driven methods for PdM, present a comprehensive survey on its applications, and attempt to provide graduate students, companies, and institutions with the preliminary understanding of the existing works recently published.

Specifically, we first briefly introduce the PdM approach, illustrate our PdM scheme for automatic washing equipment, and demonstrate the challenges encountered when we conduct a PdM research. Second, we classify the specific industrial applications based on six algorithms of machine learning and deep learning (DL), and compare five performance metrics for each classification. Furthermore, the accuracy (a metric to evaluate the algorithm performance) of these PdM applications is analyzed in detail. There are some important conclusions: 1) the data used in the summarized literatures are mostly from public datasets, such as case western reserve university (CWRU)/intelligent maintenance systems (IMS); and 2) in recent years, researchers seem to focus more on DL algorithms for PdM research. Finally, we summarize the common features regarding our surveyed PdM applications and discuss several potential directionB. Ahmad et al. [9] proposed a solution to tackle the challenges of machine failure caused by vibrations and misalignment in rotating machines. By considering the parameters of speed and vibration, they aimed to detect faults in bearings. They employed three machine learning algorithms, namely Support Vector Machine (SVM), Naïve Bayes, and Random Forest, to analyze the relationship between machine failure, speed, and vibration. Their study revealed that SVM outperformed the other algorithms, achieving an accuracy of 78%. By integrating speed and vibration parameters and leveraging machine learning techniques, their research contributes to enhancing fault detection and maintenance practices in modern industries, particularly in the context of industry 4.0.

### 2.2.3 Fault detection

Neupane et al. [1] conducted a comprehensive study on machinery fault detection and diagnosis using the Case Western Reserve University (CWRU) bearing dataset. They reviewed recent works that employed deep learning algorithms on the dataset, summarizing the working algorithms, results, and other necessary details. Their paper serves as a valuable resource for future researchers in the field, providing insights and guidance for utilizing the CWRU dataset in machinery fault detection and diagnosis. The use of deep learning algorithms in this area has gained significance with the growth of smart factories and the need for effective fault detection in machinery equipment also it summarizes the recent works which use the CWRU bearing dataset in machinery fault detection and diagnosis employing deep learning algorithms. We have reviewed the published works and presented the working algorithm, result, and other necessary detailsMcInerny et al.[7]states this paper presents a laboratory module on fault detection in rolling element bearings as part of a multidisciplinary course in applied spectral analysis. The module provides background information on bearing characteristics, explores the limitations of conventional vibration spectral analysis, explains envelope analysis and its connection to bearing fault signatures, and introduces a MATLAB-based software utility for exploring envelope analysis. This instructional module can be used as a standalone tutorial or integrated into a course.Samanta, B. I. S. W. A. J. I. T., K. R et al[4]. study is presented to compare the performance of bearing fault detection using two different classifiers, namely, artificial neural networks (ANNs) and support vector machines (SMVs). The time-domain vibration signals of a rotating machine with normal and defective bearings are processed for feature extraction. The extracted features from original and preprocessed signals are used as inputs to the classifiers for two-class (normal or fault) recognition. The classifier parameters, e.g., the number of nodes in the hidden layer in case of ANNs and the radial basis function kernel parameter (width) in case of SVMs along with the selection of input features are optimized using genetic algorithms. The classifiers are trained with a subset of the experimental data for known machine conditions and are tested using the remaining set of data. The procedure is illustrated using the experimental vibration data of a rotating machine. The roles of different vibration signals and signal preprocessing techniques are investigated. The results show the effectiveness of the features and the classifiers in detection of machine condition.

# Chapter 3

# Methodology

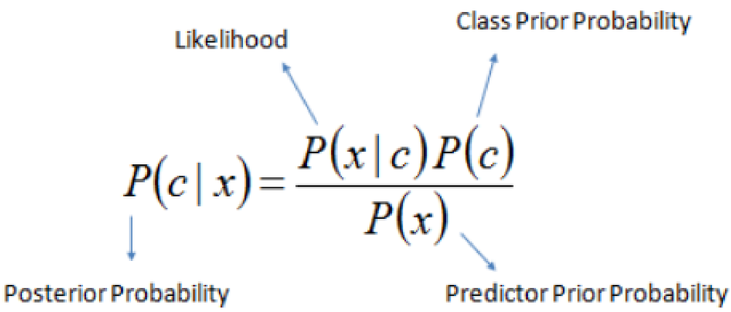
**PREDCARE: BEARINGS FAULT PREDICTION** aims to identify faults in bearings to reduce downtime cost and enable effective maintenance of rolling components. The methodology involves several steps. Firstly, data collected from sensors is stored and organized using Python libraries. The collected data is then preprocessed to clean it, remove noise and outliers, and apply necessary transformations and normalization. After preprocessing, relevant features are extracted from the data, such as statistical measures, frequency domain analysis, time-frequency analysis, or wavelet transforms. The next step involves selecting the most informative features through statistical or correlation analysis, and machine learning techniques.

With the selected features, predictive models are developed using the algorithms Naive Bayes, Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). These models are implemented using appropriate Python libraries like scikit-learn for Naive Bayes and SVM, and TensorFlow or Keras for LSTM. The models are then trained using the training data and evaluated using suitable metrics. Cross-validation techniques may also be employed to ensure robust evaluation.

# 3.1 ALGORITHMS

## 3.1.1 Naive Bayes Algorithm

**Naive Bayes classifiers** is probabilistic classifiers based on Bayes theorem with strong (naive) independence assumptions between the features. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful in the field of medical science for diagnosing heart patients. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Bayes theorem provides a way of calculating the posterior probability, P(c—x), from P(c), P(x), and P(x—c). Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence




$$P(c \mid x) = \frac{P(x \mid c) P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 3.1: Naive bayes equation

- P(c—x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x—c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor Where C and X are two events (e.g. the probability that the train will arrive on time given that the weather is rainy). Such Naïve Bayes classifiers use the probability theory to find the most likely classification of an unseen (unclassified) instance . The algorithm performs positively with categorical data but poorly if we have numerical data in the training set.

There are 3 types of naive bayes classifiers

- **Gaussian Naïve Bayes** (GaussianNB): This is a variant of the Naïve Bayes classifier, which is used with Gaussian distributions—i.e. normal distributions—and continuous variables. This model is fitted by finding the mean and standard deviation of each class.
- **Multinomial Naïve Bayes** (MultinomialNB): This type of Naïve Bayes classifier assumes that the features are from multinomial distributions. This variant is useful when using discrete data, such as frequency counts, and it is typically applied within natural language processing use cases, like spam classification.
- **Bernoulli Naïve Bayes** (BernoulliNB): This is another variant of the Naïve Bayes classifier, which is used with Boolean variables—that is, variables with two values, such as True and False or 1 and 0.

### 3.1.2 Support Vector Machine

**Support Vector Machine or SVM** is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

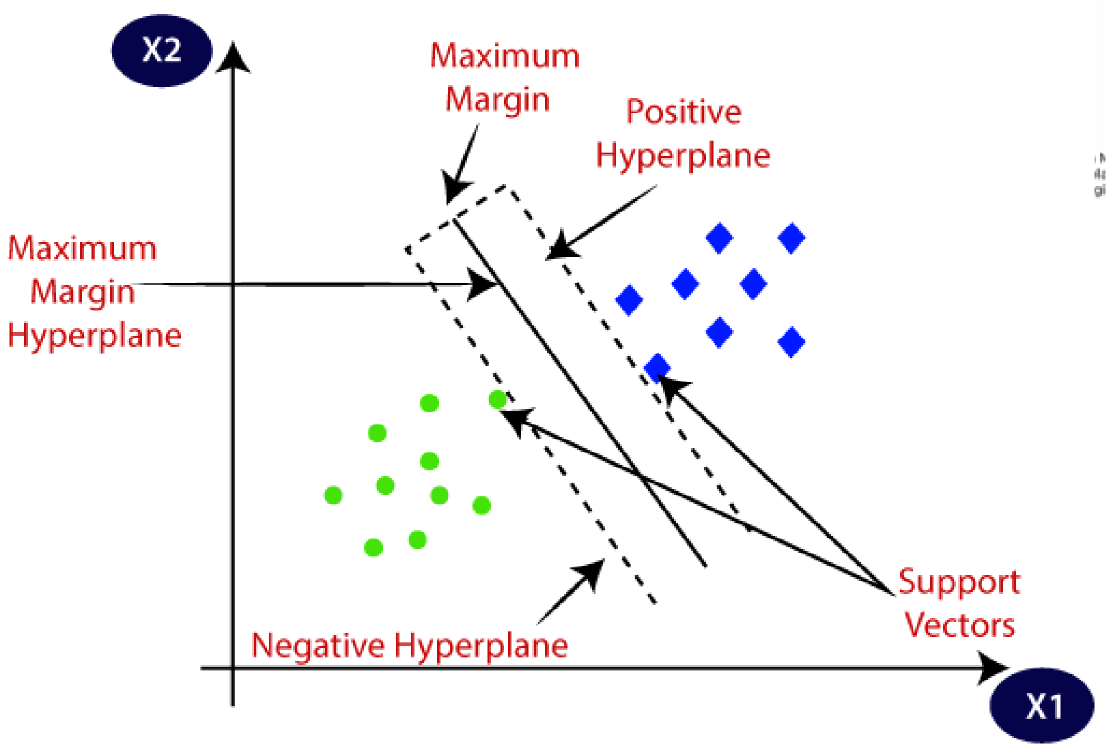



Figure 3.2: Support vector Machine

Types of SVM SVM can be of two types:

- **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier
- **Non-linear SVM**: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier

Hyperplane and Support Vectors in the SVM algorithm:
Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.
We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.
Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

### 3.1.3 Long Short Term Memory(LSTM)

**Long Short Term Memory** Long Short-Term Memory, an evolution of RNN, was introduced by Hochreiter and Schmidhuber [37] to address problems of the aforementioned drawbacks of the RNN by adding additional interactions per module (or cell). LSTMs are a special kind of RNN, capable of learning long-term dependencies and remembering information for prolonged periods of time as a default. According to Olah [33], the LSTM model is organized in the form of a chain structure. However, the repeating module has a different structure. Instead of a single neural network like a standard RNN, it has four interacting layers with a unique method of communication. The structure of the LSTM neural network is shown in Figure 4.

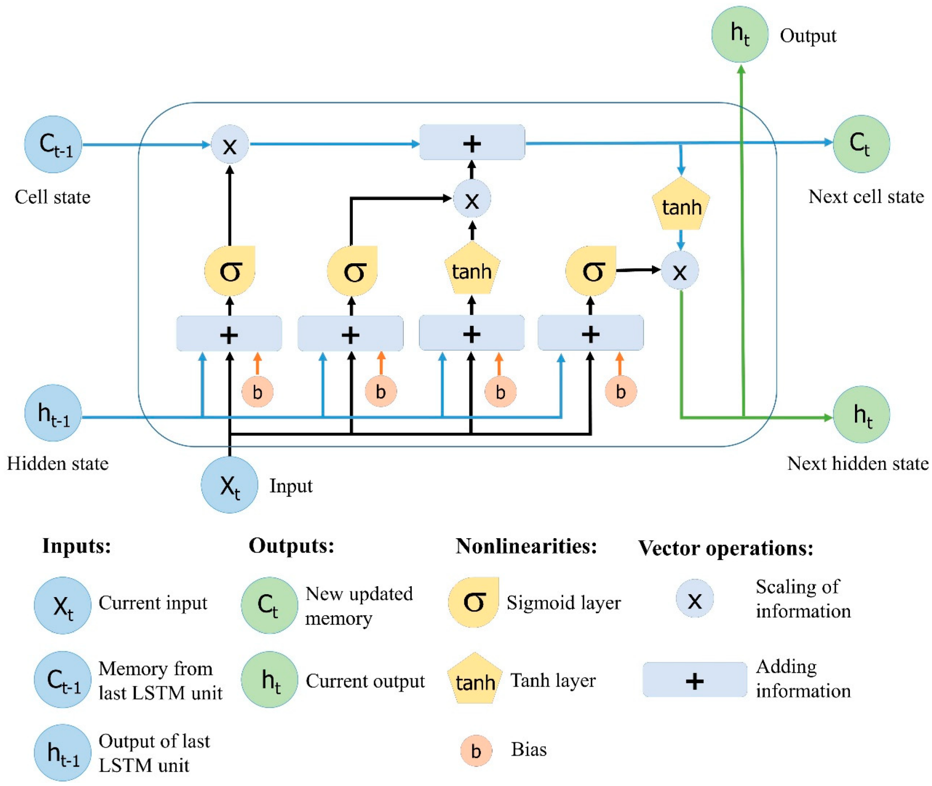



Figure 3.3: LSTM -Long Short Term Memory

### 3.1.4 Isolation Forest

**Isolation Forest** algorithm is that anomalous data points are easier to separate from the rest of the sample. In order to isolate a data point, the algorithm recursively generates partitions on the sample by randomly selecting an attribute and then randomly selecting a split value between the minimum and maximum values allowed for that attribute.

Isolating an Anomalous Point Fig. 3 - an example of isolating an anomalous point in a 2D Gaussian distribution. An example of random partitioning in a 2D dataset of normally distributed points is given in Fig. 2 for a non-anomalous point and Fig. 3 for a point that's more likely to be an anomaly. It is apparent from the pictures how anomalies require fewer random partitions to be isolated, compared to normal points.

Recursive partitioning can be represented by a tree structure named Isolation Tree, while the number of partitions required to isolate a point can be interpreted as the length of the path, within the tree, to reach a terminating node starting from the root

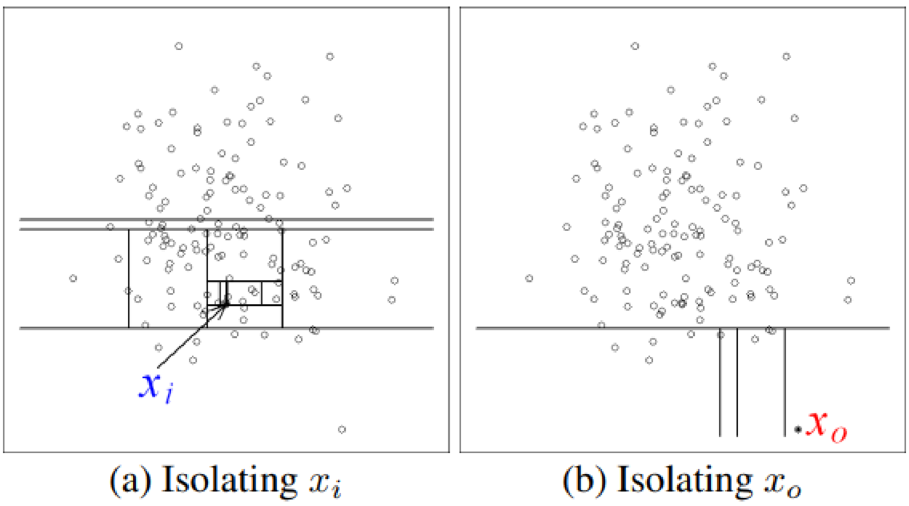



Figure 3.4: Isolation Forest

## 3.2 System Architecture

The application development architecture recognized for this project is specified in this section on the basis of requirements.

### 3.2.1 Dataset

The dataset attributes:

- Speed (Feature name)
- Vibration(Feature name)
- Condition(Target name)

Speed :Rotational Speed in the units RPM(Rotations Per Minute)
Vibration :Vibrational data measured in Hertz(Hz)

**Description** The data contain vibration signals collected from bearings of different health conditions under time-varying rotational speed conditions. There are 60 datasets in total. For each dataset, there are two experimental settings: bearing health condition and varying speed condition. The health conditions of the bearing include :

- (i) healthy
- (ii) faulty with an inner race defect
- (iii) faulty with an outer race defect
- (iv) faulty with a ball defect
- (v) faulty with combined defects on the inner race, the outer race and a ball.

The operating rotational speed conditions are (i) increasing speed (ii) decreasing speed (iii) increasing then decreasing speed (iv) decreasing then increasing speed.

Therefore,here are 20 different cases for the setting. To ensure the authenticity of the data, 3 trials are collected for each experimental setting which results in 60 datasets in total. Each dataset contains two channels: 'Channel 1' is vibration data measured by the accelerometer and 'Channel 2' is the rotational speed data measured by the encoder. All these data are sampled at 200,000Hz and the sampling duration is 10 seconds. The CPR Cycles Per Revolution of the encoder is 1024.

- 1 Data collected from a healthy bearing
- 2 Data collected from a bearing with inner race fault
- 3 Data collected from a bearing with outer race fault
- 4 Data collected from a bearing with ball fault
- 5 Data collected from a bearing with a combination of faults

Institutions University of Ottawa Categories Mechanical Vibration

The dataset attributes:

- Vibration(Feature name)
- Condition(Target name)

Vibration :Vibrational data measured in Hertz(Hz)

CASE WESTERN RESERVE UNIVERSITY EST. 1826

**Description**

The test bench for motor performance assesment consists of:

- Motor with 2 HP power
- Torque transducer
- Dynamometer
- Control electronics

The test bearings support the motor shaft. Defects were introduced at a single point by EDM machining. The diameters of defects in inches (millimeters):

0.007 inches (0.178 millimeters) 0.014 (0.356) 0.021 (0.533) There is a time series for each defect located in 1 of 3 parts of the bearing: **ball, inner race, outer race.** Telemetry measurements are come from 3 acceleroments installed on 3 positions in the system: Drive end (DE) Fan end (FE) Base (BA)

This dataset corresponds to the following conditions: 1 HP load applied to the motor Shaft rotating speed of 1772 rpm 48 kHz sampling frequency of the accelerometers

Acknowledgements This dataset is publicy available thanks to Case Western Reserve University The experiments were initiated in order to characterize the performance of IQ PreAlert, a motor bearing condition assessment system developed at Rockwell. From that point on, the experimental program has expanded to provide a motor performance database which can be used to validate and/or improve a host of motor condition assessment techniques.

Inspiration This dataset serves to the purpose of applying Machine Learning to Predictive Maintenance of industrial machinery. Iis scope is the data-driven fault diagnosis. A common task in this field is the fault detection and classification.

### 3.2.2 Data preprocessing

Data preprocessing is the process of cleaning, transforming, and organizing raw data to make it suitable for analysis or machine learning. This involves several steps like removing missing values, correcting errors, transforming data into a more usable format, selecting relevant features, and normalizing data. The goal of data preprocessing is to ensure that the data is of high quality, consistent, and relevant to the specific task at hand.

Data preprocessing is a crucial phase in the machine learning process since the caliber of the data and the information that can be extracted from it directly influence how well the model can learn. As a result, it is crucial that we preprocess our data before feeding it to the model. Preparing raw data to be used with a deep learning model is known as data preparation. It is both the first and most important step in developing a deep learning model. The format of the data in Deep Learning projects must be correct in order to get better results from the applied model.

- The data is initially in mat file format which is multidimensional dictionary which is converted to a dimensional data frame.
- Dataset form factor involves its dimension to be 2967X2 removing missing values using dropna function on missing value rows reduced the dimensions to 2300X2.
- The rows where dropped according to applied threshold function to decrease data reduction.
- The data is normalised using standard scaler

$$z = \frac{x - \mu}{\sigma}$$

In this equation:

- x represents the original value of the variable.
- $\mu$ represents the mean average of the variable.
- $\rho$ represents the standard deviation of the variable.

### 3.2.3 Building the model

In machine learning, building a model for a given problem typically involves multiple steps, which can be broadly divided into two main stages: building and training the feature extractor, and building and training the classifier. Building and training the feature extractor involves selecting or engineering features from the input data that can capture the relevant patterns and relationships between the input and output variables. After building and training the feature extractor, the next stage is to build and train the classifier, which involves selecting an appropriate machine learning algorithm or model and training it on the features extracted by the feature extractor.

#### 3.2.3.1 Building and training the Feature Extractor

- The dataset initially contains vibrational data from which we extact features.
- The signal and rpm data can synthesised to relational amount of features
- features include maximum,minimum,mean,
  standard deviation,root mean square,skewness,kurtosis,crest and form.
- Along with the extracted features we append the fault type to the dataset.

#### 3.2.3.2 Building and training the Classifier

The dataset is further divided into training and testing set using model selection package
train data and test data are split 70% and 30% test set.
for svm model parameters used:

- C=1.0: The penalty parameter of the error term.
- kernel='rbf': The kernel function used radial basis function.
- gamma='scale': The kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

### 3.2.4 Testing the models

Testing Accuracy : Testing accuracy measures the performance of a model on a separate dataset, called the testing or validation dataset, which the model has not been exposed to during training. It is calculated by comparing the model's predictions on the testing data to the true labels of that data. Testing accuracy provides an estimate of how well the model can generalize its predictions to unseen data. The testing accuracy helps assess how well the model is likely to perform on new, unseen data in real-world scenarios. A high testing accuracy indicates that the model is capable of making accurate predictions on new, unseen data, which suggests good generalization. There an 3 models each tested on 2 separate datasets.
For **crwu dataset**

- The testing was done using the same fraction of dataset 30% of data from crwu data was chosen.
- On the test data standard scalar normalization was applied .
- The crwu dataset obtained a 93.7% test accuracy on svm.
- On LSTM it obtained 98.89% test accuracy.

## 3.3 Software Requirement and Specifications

The software used for the project includes:

- Python
- Google Colaboratory
- Flask development environment

### 3.3.1 Software Description

- Python

Python is an object-oriented programming language that was developed in 1989 by Guido Rossum. It is excellent for quick prototyping of complex applications. It supports a number of operating system functions and libraries and can be converted to C or C ++. Python is a computer language that is used by numerous organizations, including NASA, Google, YouTube, and Bit Torrent. In cutting-edge fields like artificial intelligence, natural language processing, neural networks, and other computer sciences, Python programming is widely employed. The Python Software Foundation currently has authority over Guido van Rossum's intricate artificial language, which he created in the late 1980s. It comes from his ABC language, which he co-created at the beginning of his professional life. Games, graphical user interfaces (GUIs), and other types of software can all be made using the complicated programming language Python. Python scripts can be read and written like how normal English statements are read and written. They must therefore be processed because they are not written in a computer language. before being run by a system, by Python code. A basic language is Python. This suggests that the interpreter evaluates the code and transforms it into machine-readable bytecode when the program is executed. Python is an object-oriented programming language that shows users how to take care of and work with objects or data structures so that they can create and run programs. Python has everything. When they fall short of expectations and are replaced by more capable languages, languages die and become extinct. Python is a dependable and popular programming language.

- **Google Coloboratory**

Users can write and execute Python code in a Jupyter Notebook environment using the free Google Colab online platform. Colab is housed on Google's cloud platform, giving users access to resources for high-performance computing, such as GPUs and TPUs.Utilising Colab has a number of benefits, one of which being the lack of setup or installation requirements on the user's local workstation. Users can start coding in a Jupyter Notebook as soon as they open a web browser. Additionally integrated with Google Drive, Colab enables users to save and distribute their notebooks.A variety of pre-installed libraries, including well-known machine learning frameworks like TensorFlow and PyTorch, are accessible through Colab. Additional libraries can be installed by users using conda or pip. Colab notebooks can be downloaded as a Jupyter notebook (.ipynb) or Python script (.py) or saved to GitHub, Google Drive, or another service. Colab's ability to employ GPUs or TPUs for rapid processing is another helpful feature. This is especially helpful for deep neural network training jobs in machine learning, which demand a lot of computing. The interesting features that each contemporary IDE offers are abundant in Google Colab, in addition to many others. Below is a list of some of the more fascinating aspects.

- **Flask**

Flask is a popular web framework for building web applications with Python. It is a lightweight and flexible framework that is easy to learn and use, making it a popular choice for web developers.

Flask is based on the WSGI (Web Server Gateway Interface) protocol, which is a standard interface between web servers and Python web applications. Flask provides a number of features and tools for building web applications, including:

1. Routing: Flask provides a simple and flexible mechanism for defining URL routes for your application. This allows you to map HTTP requests to specific functions in your Python code.

2. Templates: Flask includes a templating engine that allows you to generate HTML pages dynamically using Python code. This makes it easy to build dynamic web pages that can respond to user input and display data from your application.

3. Sessions: Flask provides a simple way to manage user sessions in your web application. This allows you to store data for a user across multiple requests, such as user authentication information.

4. Forms: Flask includes a form handling library that makes it easy to process user input from HTML forms. This allows you to create forms for your users to input data into your application.

5. Database integration: Flask includes support for integrating with a variety of databases, including SQL databases like MySQL, SQLite, and PostgreSQL.

### 3.3.2 Hardware and experimental environment

- Processor: A multi-core processor is recommended for running Flask applications. The more cores, the better the performance. At a minimum, a dual-core processor should be sufficient.

- Memory: The amount of memory required will depend on the size of the application and the number of users. For small-scale applications, 2-4 GB of memory should be sufficient.

- Storage: The amount of storage required will depend on the size of the application and the amount of data that needs to be stored. For small-scale applications, 10-20 GB of storage should be sufficient.

- Operating System: Minimum requirement involves i3 processor 2 cores ,2.5GHz and above

- Minimum versions required for each browser:
  Google Chrome: Version 80 or newer
  Mozilla Firefox: Version 76 or newer
  Safari: Version 13.1 or newer
  Microsoft Edge: Version 80 or newer

## 3.4 System Design

Designing a system for bearing fault prediction involves several components and considerations.Starting with collection of data its preprocessing , model section and deployment

### 3.4.1 Architectural Design

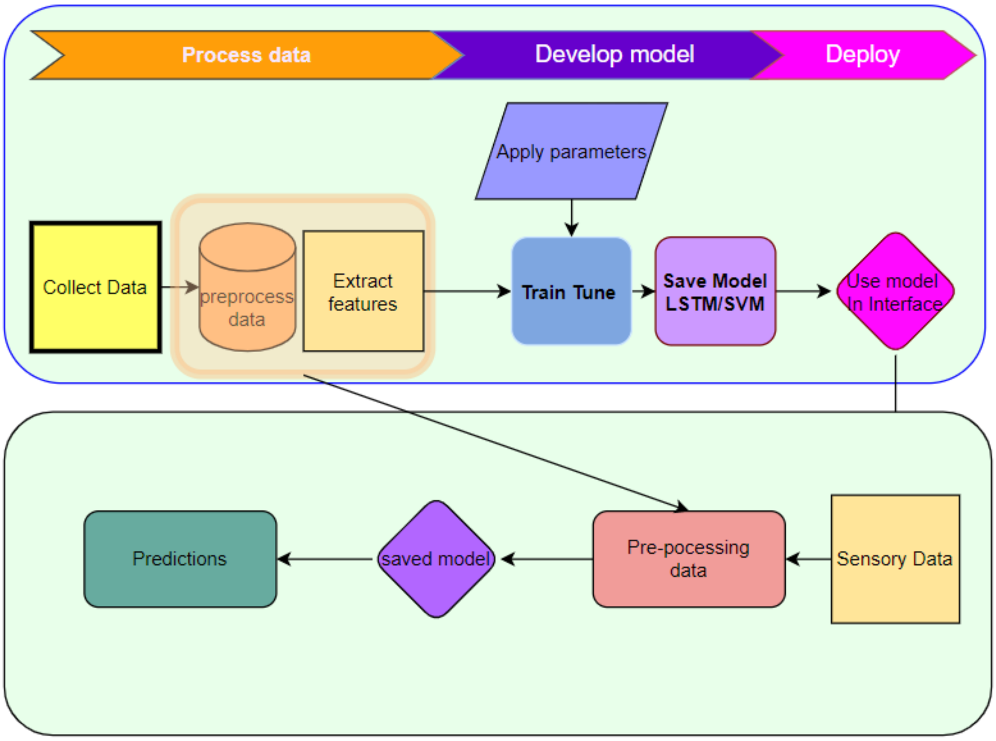



Figure 3.5: Architectural Design

Firstly, the data collection phase involves identifying the sensors and data sources for bearing monitoring, such as vibration sensors or acoustic emission sensors. The sampling frequency and duration of data collection should be determined, and a data acquisition system should be set up to capture and store the sensor readings.

Once the raw sensor data is collected, the next step is data preprocessing. This involves cleaning the raw sensor data by removing noise and outliers. The data is then segmented into smaller time windows or chunks for analysis. Relevant features are extracted from the segmented data, such as statistical measures, frequency domain features, or time-frequency representations.

Feature engineering is an important step in transforming the raw data into meaningful features. Domain-specific feature engineering techniques can be applied, such as time-domain features (e.g., mean, standard deviation), frequency-domain features (e.g., spectral entropy, peak frequencies), or statistical features (e.g., kurtosis, skewness). Advanced signal processing techniques, such as wavelet transforms, Fourier transforms, or empirical mode decomposition, can also be utilized to extract informative features.

The model selection phase involves exploring and evaluating various machine learning models suitable for fault prediction, such as classification algorithms. Ensemble techniques that combine multiple models can also be considered to improve performance. The models are trained using labeled data, where each sample is associated with a specific bearing condition (normal or faulty).

Once the models are trained, the next step is model training and evaluation. The labeled data is split into training and validation sets. The selected model(s) are trained using the training set and evaluated using the validation set and performance metrics, such as accuracy, precision, recall, and F1 score. Fine-tuning of the model(s) can be done by adjusting hyperparameters and conducting cross-validation.

# Chapter 4

# RESULT AND DISCUSSION

Bearing fault prediction is an important task in the field of predictive maintenance and condition monitoring. Recently, support vector machines (SVMs) and long short-term memory (LSTM) models have been used to detect bearing faults with high accuracy.

- A programme is tested by being run with the goal of identifying any errors.
- A excellent test case is one that has the highest chance of spotting an error that hasn't been identified yet.
- A test that finds an error that hasn't been found yet is successful.

Our objective is to develop tests that systematically uncover many sorts of issues with minimal time and effort. Testing indicates that software functionalities appear to operate as expected and that performance criteria appear to have been met. The information acquired during testing is an excellent predictor of programme reliability and a partial indicator of software quality as a whole. Testing has one drawback, however: it can only demonstrate the presence of software defects, not their absence.

## 4.1 Training and Validation Results

The testing and validation results involve model accuracy on each testing set from multiple datasets The first medely dataset in which the data is divided into 80% training data resulted as:

| Model | Parameters and Attributes | Accuracy |
|---|---|---|
| Naive Bayes | model = GaussianNB() | 74.02% |
| SVM | svc_model = SVC(C=1.0, kernel='rbf', degree=3) | 75.61% |
| LSTM | Input Layer:LSTM(32, activation)<br>Dropout(0.2)<br>activation:"relu" | 95.91% |

Figure 4.1: Traiing accuracies on cwru data

The cwru dataset split into 70% and 30% training results are:

| Model | Parameters and Attributes | Accuracy |
|---|---|---|
| Naive Bayes | model = GaussianNB() | 74.02% |
| SVM | svc_model = SVC(C=1.0, kernel='rbf', degree=3) | 93.73% |
| LSTM | Input Layer :64<br>Dropout Layer :0.2<br>Dense Layer:(1,activation function: 'softmax)<br>Epoch 50/50<br>Test loss: 0.030, Test accuracy: 0.989 | 98.43% |

Figure 4.2: Traiing accuracies on cwru data

## 4.2 Performance Metrics for Validation Phase

Performance metrics are used to evaluate the performance of machine learning models. There are several commonly used performance metrics depending on the type of problem you're working on. Here are some widely used performance metrics for classification and regression problems:

**Classification Metrics:**

- Accuracy: Measures the proportion of correctly classified instances.
- Precision: Calculates the proportion of correctly predicted positive instances out of all instances predicted as positive.
- Recall (Sensitivity or True Positive Rate): Measures the proportion of correctly predicted positive instances out of all actual positive instances.
- F1 Score: The harmonic mean of precision and recall, providing a balanced measure between the two.
- Specificity (True Negative Rate): Measures the proportion of correctly predicted negative instances out of all actual negative instances.

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 1.00 | 0.99 | 75 |
| 1 | 1.00 | 0.97 | 0.99 | 75 |
| 3 | 0.99 | 1.00 | 0.99 | 75 |
| 4 | 1.00 | 0.99 | 0.99 | 75 |
| 5 | 1.00 | 1.00 | 1.00 | 75 |
| 6 | 1.00 | 1.00 | 1.00 | 75 |
| 7 | 1.00 | 1.00 | 1.00 | 75 |
| 8 | 1.00 | 0.97 | 0.99 | 75 |
| 9 | 0.99 | 1.00 | 0.99 | 75 |
| 10 | 0.99 | 1.00 | 0.99 | 75 |
| accuracy | | | 0.99 | 750 |
| macro avg | 0.99 | 0.99 | 0.99 | 750 |
| weighted avg | 0.99 | 0.99 | 0.99 | 750 |

Figure 4.3: Performance metrics

**Regression Metrics:**

- Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values.
- Mean Squared Error (MSE): Calculates the average of the squared differences between predicted and actual values.
- Root Mean Squared Error (RMSE): The square root of MSE, providing the error in the same units as the target variable.

Mean Squared Error (MSE): 0.010666666666666666 Root Mean Squared Error (RMSE): 0.10327955589886445 Mean Absolute Error (MAE): 0.008

### 4.2.1 Confusion Matrix

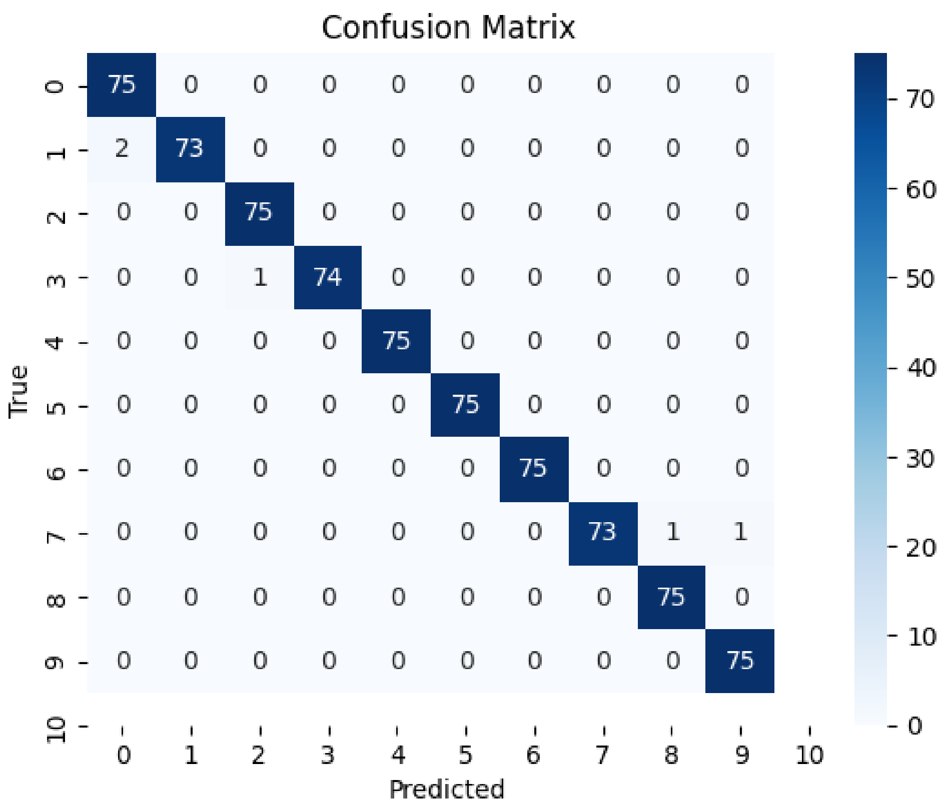



Figure 4.3: Confusion matrix

## 4.3 Output Screens and Results

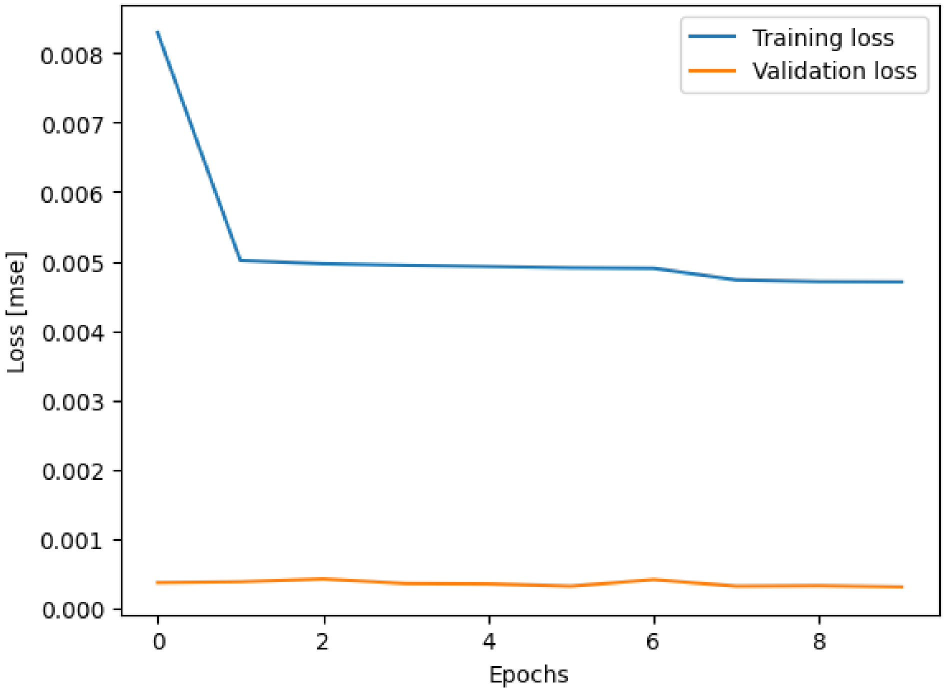



Figure 4.4: Training and validation on medely data

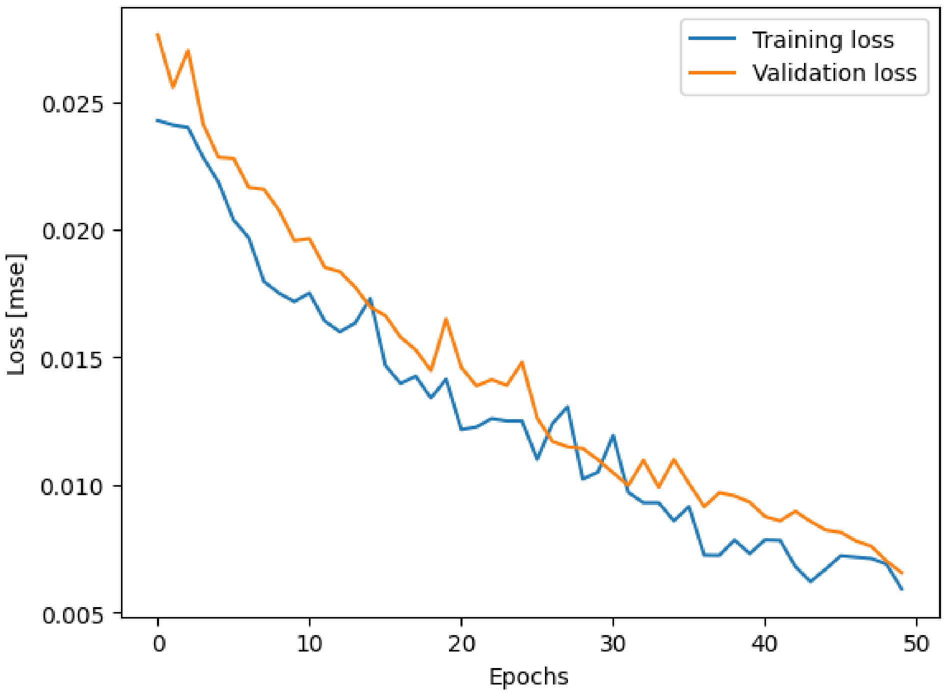



Figure 4.5: Training and validation on cwru data

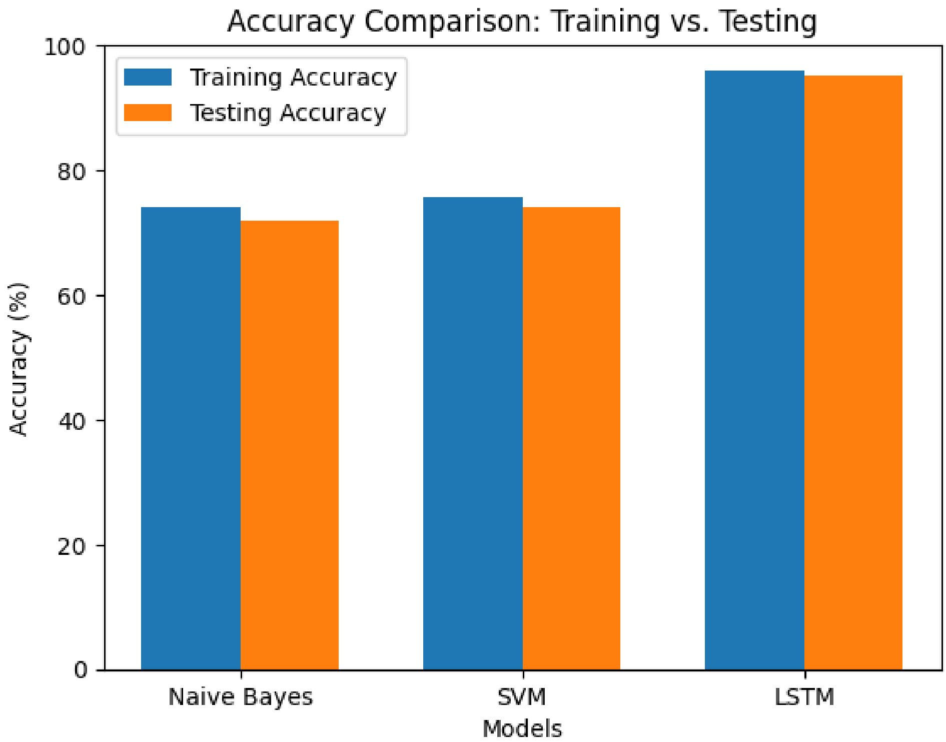



Figure 4.6: Model comparison on Medely

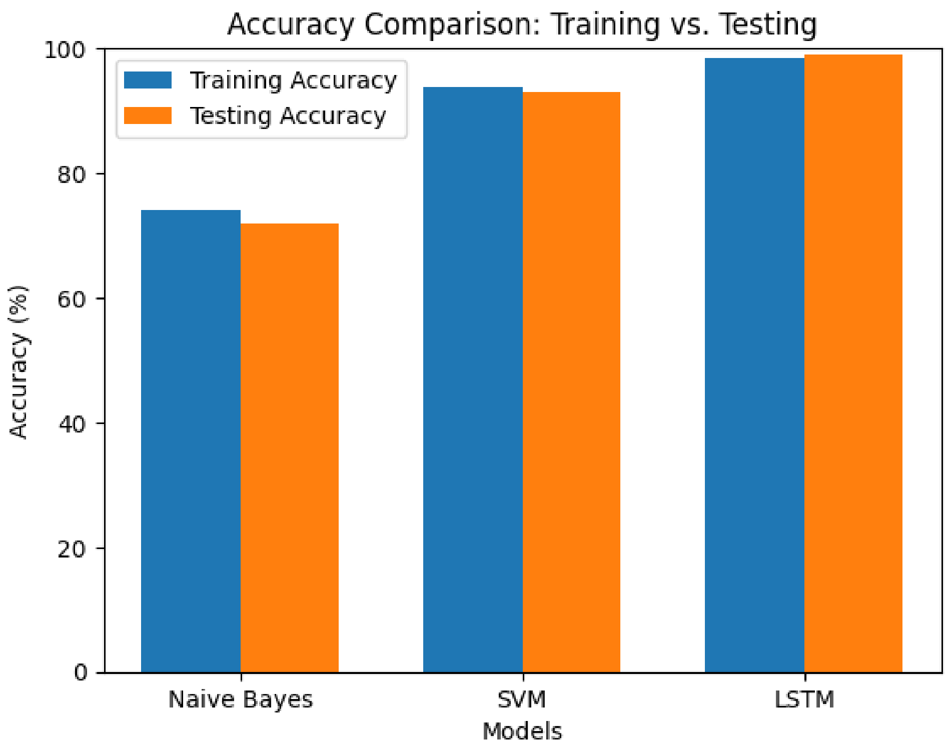



Figure 4.7: Model comparison on cwru data

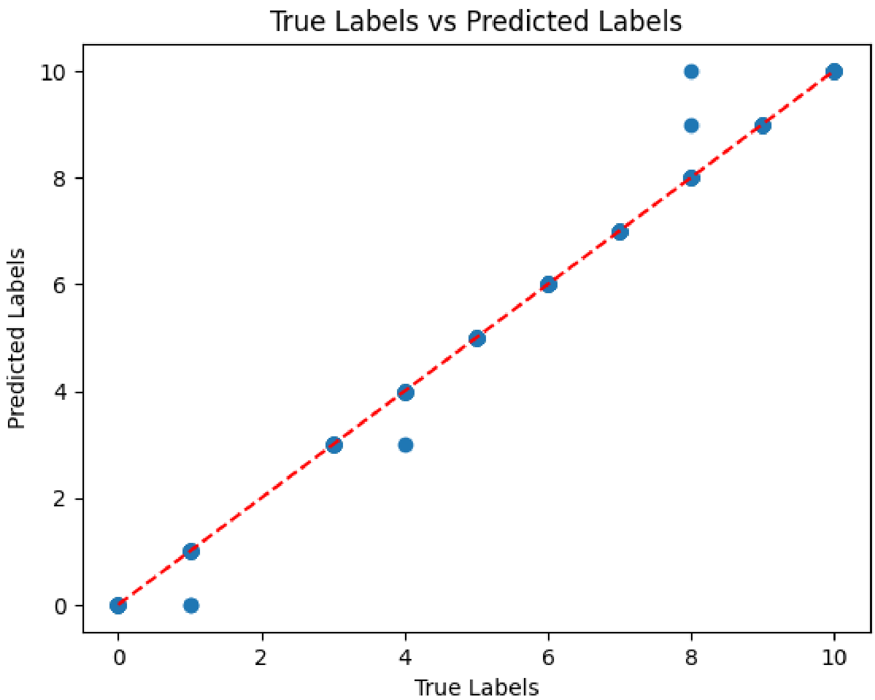



Figure 4.8: Predicted vs True Labels

# Chapter 5

# CONCLUSION

## 5.1 Conclusion

In conclusion, predictive maintenance of rolling parts is a critical aspect of modern industrial operations. By using sensory data and machine learning algorithms, it is possible to accurately predict when maintenance is required, thereby reducing downtime costs and increasing efficiency. Bearing fault detection is a key part of predictive maintenance, as bearing failure can lead to significant downtime and repair costs.

In this context, the use of machine learning algorithms such as SVM, Isolation Forest, and LSTM can greatly enhance the accuracy of bearing fault detection. Overall, bearing fault detection using machine learning algorithms is a promising area of research and has the potential to significantly improve the reliability and efficiency of industrial operations. As machine learning techniques continue to evolve, we can expect further advancements in this field, ultimately leading to safer and more efficient industrial operations.

## 5.2 Future Enhancement

- **Real-time data** can be used as input to the current system which can synthesis real-time faults
- **Improved algorithm** will help in making the predictions accurate.
- **Efficient mapping** can be implemented reducing the error points and anomily values.

# REFERENCES


[1] Neupane, Dhiraj, and Jongwon Seok. ”Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review.” IEEE Access 8 (2020): 93155-93178.

[2] S. Zhang, S. Zhang, B. Wang and T. G. Habetler, ”Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review,” in IEEE Access, vol. 8, pp. 29857-29881, 2020, doi: 10.1109/ACCESS.2020.2972859..

[3] Zhang, Weiting, Dong Yang, and Hongchao Wang. ”Data-driven methods for predictive maintenance of industrial equipment: A survey.” IEEE Systems Journal 13.3 (2019): 2213-2227.

[4] Samanta, B. I. S. W. A. J. I. T., K. R. Al-Balushi, and S. A. Al-Araimi. ”Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection.” Engineering applications of artificial intelligence 16.7-8 (2003): 657-665.

[5] Mendez, Kevin M., et al. ”Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing.” Metabolomics 15 (2019): 1-16.

[6] Grinberg, Miguel. Flask web development: developing web applications with python. ” O’Reilly Media, Inc.”, 2018.

[7] McInerny, Sally A., and Yi Dai. ”Basic vibration signal processing for bearing fault detection.” IEEE Transactions on education 46.1 (2003): 149-156.

[8] C. -C. Wang, C. -W. Lee and C. -S. Ouyang, ”A machine-learning-based fault diagnosis approach for intelligent condition monitoring,” 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 2010, pp. 2921-2926, doi: 10.1109/ICMLC.2010.5580753.

[9] B. Ahmad, B. K. Mishra, M. Ghufran, Z. Pervez and N. Ramzan, ”Intelligent Predictive Maintenance Model for Rolling Components of a Machine based on Speed and Vibration,” 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea (South), 2021, pp. 459-464, doi: 10.1109/ICAIIC51459.2021.9415249..

[10] Hakim, Mohammed, et al. ”A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations.” Ain Shams Engineering Journal (2022): 101945.

# APPENDIX

## Graphs

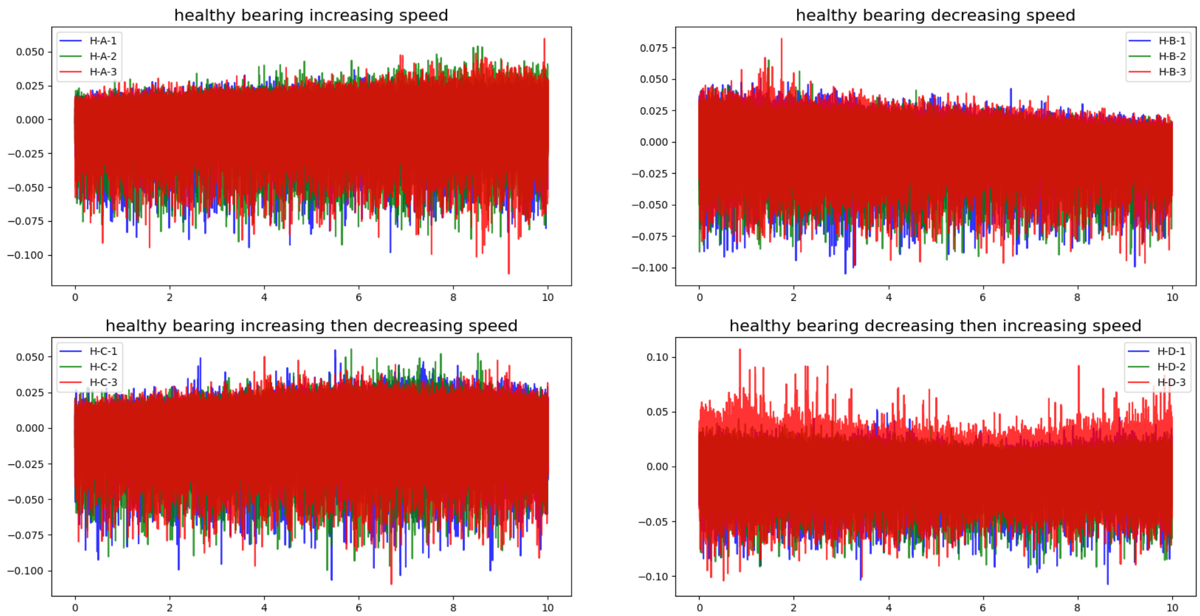



Figure 5.0: Raw data graph medely

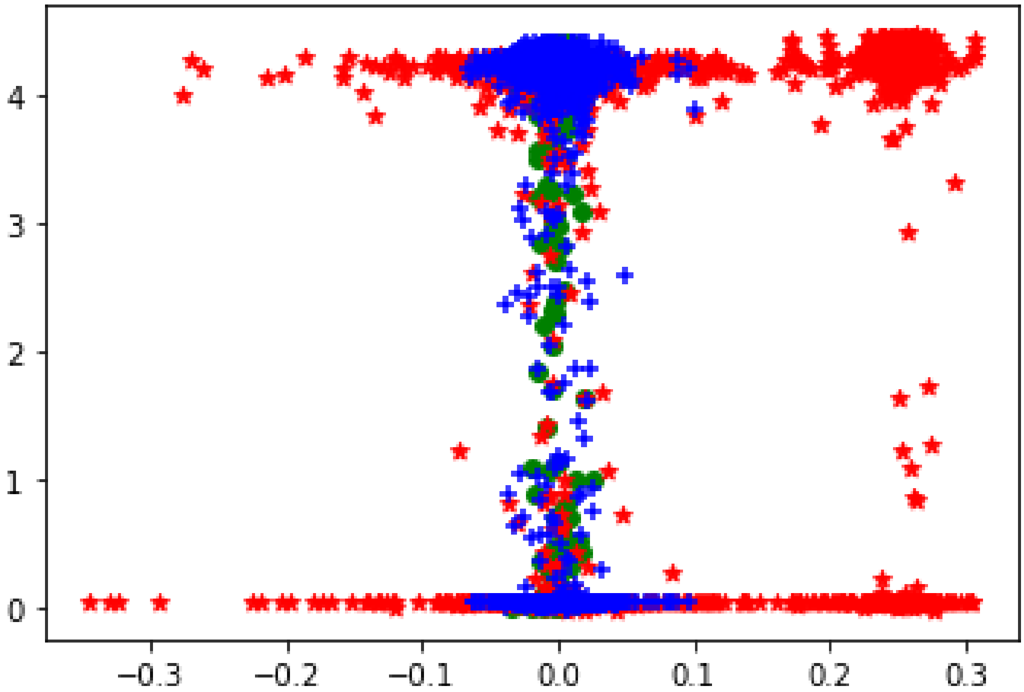



Figure 5.1: Raw data scatterplot

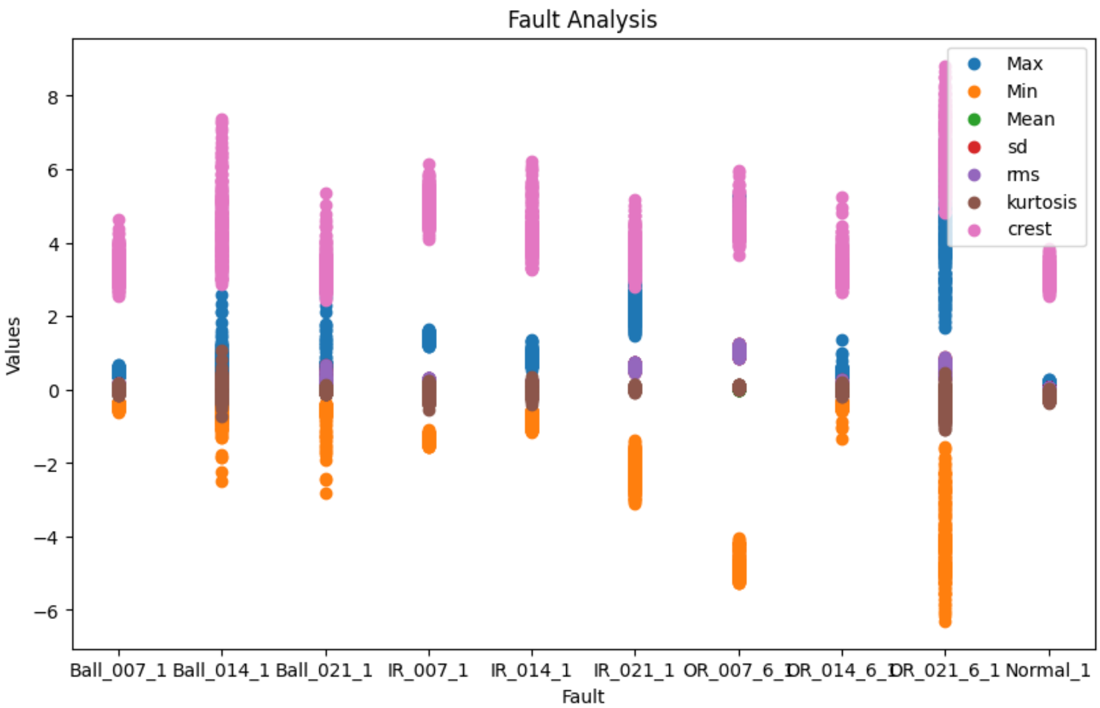



Figure 5.2: Feature scatterplot

## Screenshots

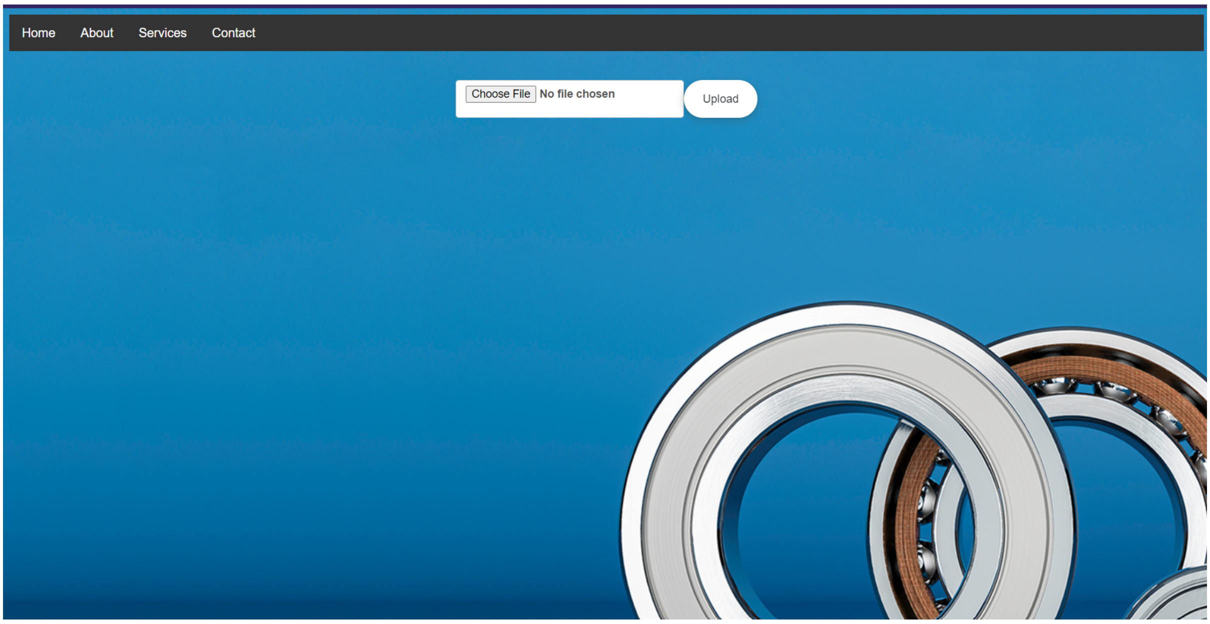
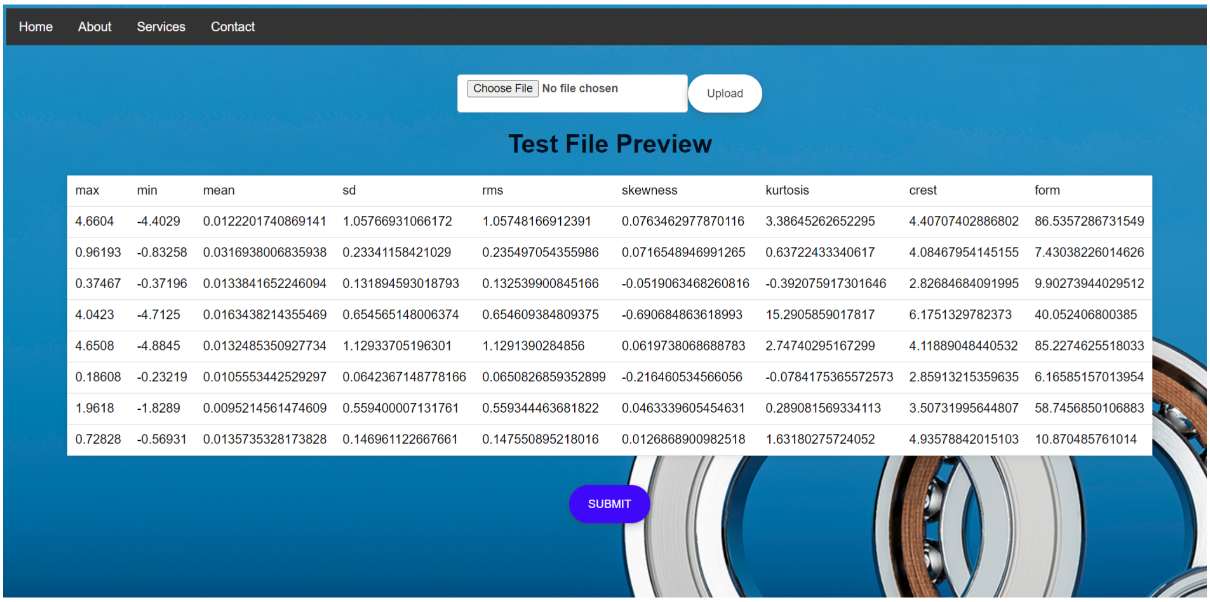



Figure A.1: Index Page




| max | min | mean | sd | rms | skewness | kurtosis | crest | form |
|---|---|---|---|---|---|---|---|---|
| 4.6604 | -4.4029 | 0.0122201740869141 | 1.05766931066172 | 1.05748166912391 | 0.0763462977870116 | 3.38645262652295 | 4.40707402886802 | 86.5357286731549 |
| 0.96193 | -0.83258 | 0.0316938006835938 | 0.23341158421029 | 0.235497054355986 | 0.0716548946991265 | 0.63722433340617 | 4.08467954145155 | 7.43038226014626 |
| 0.37467 | -0.37196 | 0.0133841652246094 | 0.131894593018793 | 0.132539900845166 | -0.0519063468260816 | -0.392075917301646 | 2.82684684091995 | 9.90273944029512 |
| 4.0423 | -4.7125 | 0.0163438214355469 | 0.654565148006374 | 0.654609384809375 | -0.690684863618993 | 15.2905859017817 | 6.1751329782373 | 40.052406800385 |
| 4.6508 | -4.8845 | 0.0132485350927734 | 1.12933705196301 | 1.1291390284856 | 0.0619738068688783 | 2.74740295167299 | 4.11889048440532 | 85.2274625518033 |
| 0.18608 | -0.23219 | 0.0105553442529297 | 0.0642367148778166 | 0.0650826859352899 | -0.216460534566056 | -0.0784175365572573 | 2.85913215359635 | 6.16585157013954 |
| 1.9618 | -1.8289 | 0.0095214561474609 | 0.559400007131761 | 0.559344463681822 | 0.0463339605454631 | 0.289081569334113 | 3.50731995644807 | 58.7456850106883 |
| 0.72828 | -0.56931 | 0.0135735328173828 | 0.146961122667661 | 0.147550895218016 | 0.0126868900982518 | 1.63180275724052 | 4.93578842015103 | 10.870485761014 |

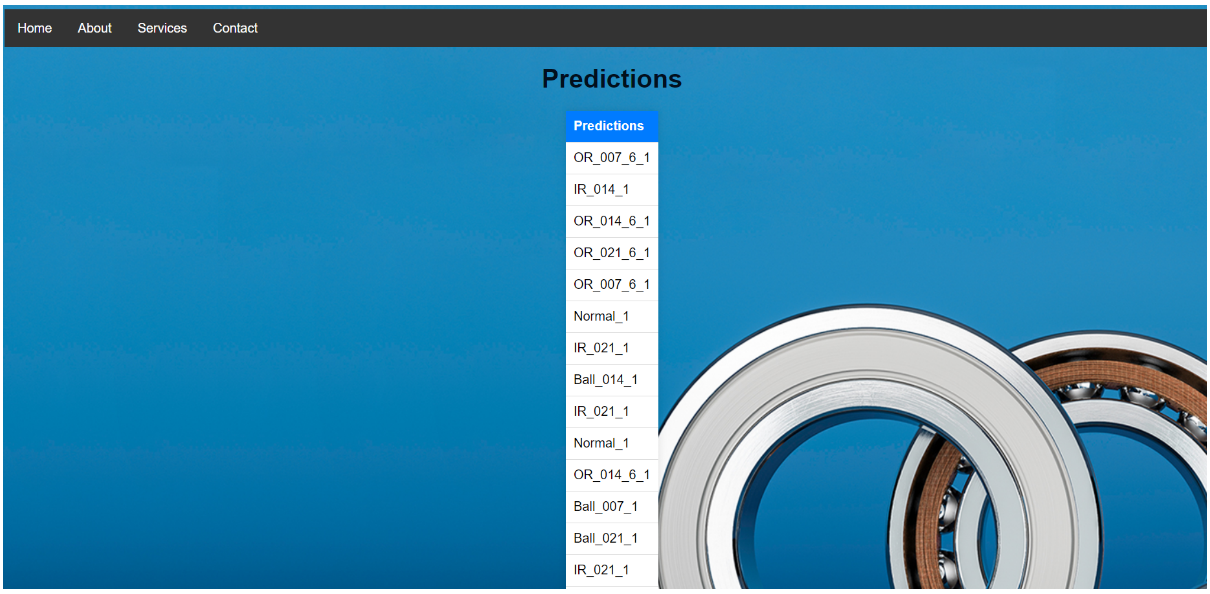



Figure A.2: Prediction