Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project

# ANOMALY DETECTION IN TIME SERIES DATA

*Kaushik Manjunatha*

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Paul G. Plöger
Dr. Anastassia Küstenmacher

September 2021

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date

Kaushik Manjunatha

# Abstract

In this world of digitization, the amount of data transferred exceeds the ability of humans to study it manually. So, data analysis will become a necessity. The detection of anomalies in data is among the most vital tasks of data analysis. Anomalies are data points that deviate from the conventional distribution of the entire dataset, and anomaly detection is the technique to find them. An anomaly can indicate an intrusion attack when reviewing a dataset of network activities. Anomalies are not limited to the detection of intrusion attacks, an anomaly in a financial transaction hints at a financial fraud, and that in a medical image hints at a disease. Other applications of anomaly detection include detecting industrial damages, preventing data leaks, identifying security vulnerabilities, military surveillance, etc. The detection of the abnormal instances within the datasets is of much interest in machine learning, this process is called anomaly detection or intrusion detection. Anomaly detection methods are specific to the type of data. Algorithms suitable for detecting an anomaly in images are quite different from those for detecting anomalies in data streams. In this project, an attempt is made to identify methods for anomaly detection in time-series data. To identify and understand the different state-of-art techniques used to detect the anomalies in time series data. The existing methods are mostly applied for image data and text data. Relevant work is not carried out for measurement data. In this, we focus on selecting a state-of-the-art method for detecting anomalies from time-series data (measurement data) using Autoencoders. An alarm can be raised when a deviation from the normal behavior of the system is detected by condition monitoring, failure diagnostics, and predictive maintenance.

# Acknowledgements

It is my great pleasure and sincere gratitude that I thank my first supervisor, Prof. Dr. Paul G. Plöger, for giving me the opportunity to work on this Research and Development project. I would also like to thank my second supervisor, Dr. Anastassia Küstenmacher, who guided me throughout the project and kept me motivated at challenging times.

I extend my sincere gratitude to Mrs. Iman Awaad for the strong support and guidance throughout my journey in this master's course. I want to express my deep gratitude to Mrs. Iman Awaad for her strong support and insightful guidance throughout my master's study.

I appreciate my colleagues, Suhasini Venkatesh, Prabhudev Bengaluru Kumar, Alex Jude for their continuous support, constructive criticism, guidance and motivation. I thank my friends Nandini Krupa Krishnamurthy, Shalini Shankar, and my sister Keerthana Manjunatha for their mental support and motivation which ensured me to push my limits.

As a final note, I would like to express my thanks, respect, and love for my parents Mrs. Jayalakshmi Manjunatha Upadhya and Mr. Manjunatha Upadhya through this work.

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

The term time series refers to a series of measurements taken over an extended period at equal intervals. Time series is an ordered collection of data points that are recorded at equal time intervals discretely or continuously. The majority of the data present in the real world contains a temporal component that depends on or varies with time. For example, weather, sound waves, a person's heart rate, etc.These time-series data are hard to interpret due to their varied features such as their patterns, internal noises, larger dimensions, etc.

Time-series can be used for predicting the future based on past data or understanding the past and observing the pattern of the data. Time series forecasting is an approach to predicting future values by using past data values and analyzing the pattern. Based on the characteristics and the trends of the time series approaches, the forecasting techniques are chosen to predict the future time constraint values from the historical data.

Time series analysis is a must to understand seasonality, cyclicality, trend, and randomness in the series.In many different statistical models, anomaly detection is considered to be an unsupervised task of pattern recognition. In the anomaly detection task, the goal is to learn a representation of the features of the training sample set that captures the appearances that are normal to the training sample set.

Anomaly detection is an effective approach to dealing with problems in the area of network security. In order to detect novel and advanced intrusions, machine learning detection systems have become increasingly important due to the rapid advancement of technology. As there is an increase in the amount of data that is being transmitted every day from one network to another, there is an increased need to identify intrusion in such large datasets effectively and periodically. Data mining and machine learning approaches could prove effective in this regard. Anomaly or Outlier detection is an especially tricky problem in networks, financial transactions, and real-world data analysis because the statistical properties of anomalies are difficult to be assessed with a low training dataset.

A neural network which is designed to reconstruct the original input using the learnt identity function by commpressing the data into a more adequate compressed way of representation is done using autoencoders. An autoencoder will reconstruct the input vectors of high dimensions by converting the high dimensional data into codes of low-dimension by training a neural network with multilayers. The weights are initialized to the deep autoencoder networks to make it more effectively learn the codes of low dimensionality.

The core concept is to design an autoencoder network to make it learn the data representation of nonmalicious normal data sequences. In the case of data that is typically recorded on a regular basis, encoding will be uniform. In order to minimize the loss in reconstruction, the decoder will try to regenerate encodings. A key point to note is that anomalous data produces greater reconstruction errors compared with the regular datasets.

## 1.1 Motivation

The detection of the abnormal instances within the datasets is of much interest in machine learning, this process is called anomaly detection or intrusion detection.
The first definition of anomaly detection is given by Grubbs in 1969 : *"An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs."* [21]
The main reason for the detection of these anomalies was to remove the outliers from the training data as there were sensitive algorithms used in pattern recognition that had an impact on the outliers in the data. This procedure is usually called the cleansing of the data. By using classifiers that are more robust the anomalies would decrease, many new algorithms were used to detect the anomalies and evaluate the performances.
The anomalies will have two main characteristics [18].

- Based on the features of the anomaly detection, it is completely different from the norm [18].
- The rarity is more in a dataset than compared with the normal instances [18].

Anomaly detection algorithms are mainly used to enhances the traditional rule-based detection systems and their applications in intrusion detection [17]. Monitoring the network traffic, server applications are the applications in which intrusion detection is used. Context behavioral analysis, Intrusion Detection Systems (IDS) are used to process huge amounts of data in real-time using fast anomaly detection algorithms [65]. The Intrusion detection systems will usually complement the traditional rule-based detection systems by using the simple concepts of pattern matching for fast, reliable detection of the anomalies when there is any kind of suspicious activity. Fraud Detection, the log data are analyzed for the suspicious events of fraud in the system due to misuse of the information, etc [60]. Financial transactions for fraudulent accounting, credit card payments logs, etc are the applications that use fraud detection techniques [54].As there is a large increase in digital goods and digitalized transactions in the internet payment systems, fraud detection is a more important area that needs to identify to put scammers behind bars. Data Leakage Protection (DLP) is an important aspect where sensitive information will be protected at early stages based on the detection of data loss which is identified in initial data analysis. Fraud detection precaution methods are undergone when there is access to databases or file servers to check for the logs and to analyze for any detection of uncommon access patterns. In medical and life sciences applications, anomaly detection is used for patient monitoring, electrocardiography (ECG) signals, analyzing medical images, computer tomography (CT) for detecting abnormal cells or any tumors. Anomaly detection algorithms rely on the complex image processing methods for their pre-processing step, this is usually used to find the mutants and pathologies in the anomaly detection.

There are numerous other applications associated with anomaly detection apart from the above main areas such as surveillance camera data for suspicious moments, smart energy consumption anomalies, mobile network manager for communication discrepancies, investigation of forged documents in forensic applications, etc.
Time series data is not so representative in the community in the technique mentioned above. In the computer vision area, image data can be used by autoencoders to detect anomalies. This project goes a step further and uses time-series data for anomaly detection on measurement data.
The autoencoders learn better due to their low dimensionality representation and hence are expected to give a better outcome than machine learning approaches. The aim of this project is to build a robust model for time series anomaly detection for measurement data. Using different feature extraction techniques, the extraction of features such as sensor values, frequency, amplitude, phase and time stamp, etc., along with some statistical features such as mean and standard deviation is done on measurement data. Machine learning approaches will be used to compute the accuracy of the anomaly, based on the input to the anomaly detection methods.
These results are compared with the baseline results which give an overview of detected results, and goodness along with the performance of the state-of-art approaches which is very important for machine learning models to learn and to provide better results.

## 1.2 Challenges and Difficulties

In the progress of this research work, the following challenges and difficulties have been encountered:

- Finding a time series dataset with measurement data. The major hurdle was to find a proper large dataset which had measurement readings to perform the experimentation.
- Identification of methodologies which can be used to train and analyse the measurement data.
- Preprocessing of the large dataset and handling the missing data to make it more suitable and usable in the machine learning algorithms.
- Tuning of the Deep Neural Netowrks to obation the accuracy. This was a major challenge to tune the different hyper parameters to get the desired accuracy for the dataset used in the experimetaion.
- Resources for computation. As the datasets are very huge, it requires large RAM and high end system with graphics card for smoother computation.

## 1.3 Problem Statement

The autoencoders technique will majorly use image data for the anomaly detection in the computer vision area, this approach is an attempt to use time series data on autoencoders for anomaly detection on measurement. Time series data is the data that is not so representative in the community in this technique. To identify and understand the different state-of-art techniques used to detect the anomalies in time series data.

What is the problem?

Measurement data is a type of time series data which will have different sensor readings to measure the various behavior of the system. Time series data is a sequence of historical measurements of an observable variable at equal time intervals. The prediction of the abnormal patterns and internal noises in the sensor data is hard. Anomaly detection is to identify data points, events, or any observations that deviate from the dataset's normal behavior [14]. The red marks will represent the

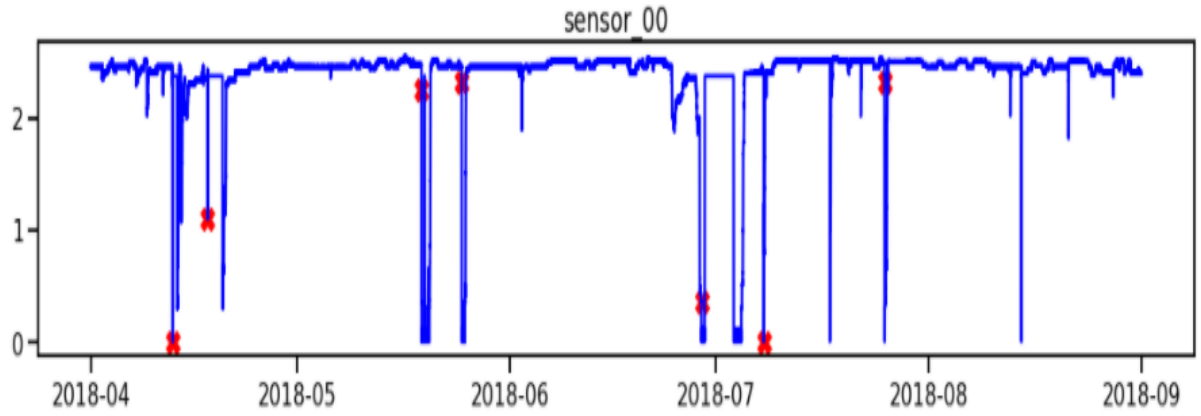



Figure 1.1: *The red marks will represent the broken states which are observed disturbances of the sensor reading in a time series data, reproduced from [31]*

broken states which are observed disturbances of the sensor reading in time-series data as shown in figure 1.1.

Why is it relevant?

This work will concentrate on the anomaly detection of sensor data using deep learning approaches. Sensor data is used across different disciplines such as

- Robotics – fault detection and analysis, adaptive learning [30].
- Manufacturing – production management.
- IoT – temperature sensors, proximity sensors.
- Medical – sleep and heart monitoring, body temperature sensors.

This work will benefit the

- Symptoms tracking and health monitoring in Healthcare.
- Speed and motion tracking in Automobile.
- Fault detection and analysis in robotics and Manufacturing.

Why is this not sufficient?

- Time constraint: It takes a lot of time for the preprocessing of the data and label the data and train the model.
- Dimensionality: The obtained outcome will have high dimensionality.

Why this approach?

Autoencoders is an unsupervised machine learning approach, and it will encode the data which has less scope for pre-processing [64]. Autoencoders will reduce the dimensionality into a latent space [33]. Accuracy and performance are better when compared to the classical machine learning approaches. The computational cost reduces, the amount of training data that needs to be learned can be reduced. Various methods or approaches are available for explaining and interpreting deep learning models [25][33][64]. The existing methods are mostly applied for image data and text data. There is no much relevant work done in anomaly detection for the measurement data[25][33][64]. This work focuses on selecting a state-of-the-art method for detecting anomalies from time-series data (measurement data) using Autoencoders. An attempt to do an analysis on different state-of-art techniques between the classical machine learning approaches and deep learning techniques like autoencoders. In this R&D project, the aim is to find, understand, and do analysis for the different state-of-art techniques used for anomaly detection in time series measurement data.

## 1.4 Report Outline

This report is composed of six chapters; Chapter 1 provides the introduction followed by the motivation for this research work, an overview of the research proceedings, challenges and difficulties and a report outline. Chapter 2 enlightens the necessary knowledge about the concepts used throughout this research work. Chapter 3 provides various State-Of-The-Art methodology used for time-series anomaly detection, data set description and machine configuration required for the research work. Chapter 4 states the proposed method for the anomaly detection for the sensor data, and the results and working of the proposed methodology.. Chapter 5 presents the contributions, future direction, and lessons learned from this research work.

# 2

# Background

## 2.1 Anomalies and outliers

Anomalies and outliers do not have a consensus distinction. Most of the following references aim at proving that anomaly and outlier are equal.

*"Outliers are also referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature."*[1]
With this regard to the outliers there can be addition of the noise to the anomalies [50]. This might be considered as the corruption of data or specific anomalies with irregular point patterns [22]. As the usage of the outlier and the anomaly is conversely used in the evaluation of any time series. There are different anomaly detection methods to detect the anomalies. The anomalies are the patters which do not follow the approach of a normal behaviour in the data. The following are the most common definitions for anomalies.
Chandola has defined it as *"Anomalies are patterns in data that do not conform to a well defined notion of normal behavior."*[10]
Ord has used as *" An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data."*[43]
Hawkins uses it as *" An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."*[24]

From the above definitions we can infer the anomaly characteristics as

- Based on the distribution of the data the anomalies can deviate strangely.
- Data from normal samples patterns make up a large majority of the dataset, then only a very small portion of the dataset contains anomalies.

In order to identify anomalies, it is essential that these two aspects are taken into account. A system like auto-encoders can therefore serve as a semi supervised method for detecting anomalies, while even traditional classification approaches cannot be used due to their reliance on balanced data sets. Distinguishing anomalies from noise is important as well. Data can have errors in the attribute values or

be mislabeled as noise [50]. When it comes to medical images, for instance, an anomaly may represent a tumor, but the odd noise is just a random variation of color, glow, illumination, and undesirable information.

The following figure 2.1 will depict the differences in data with noise and anomalies.

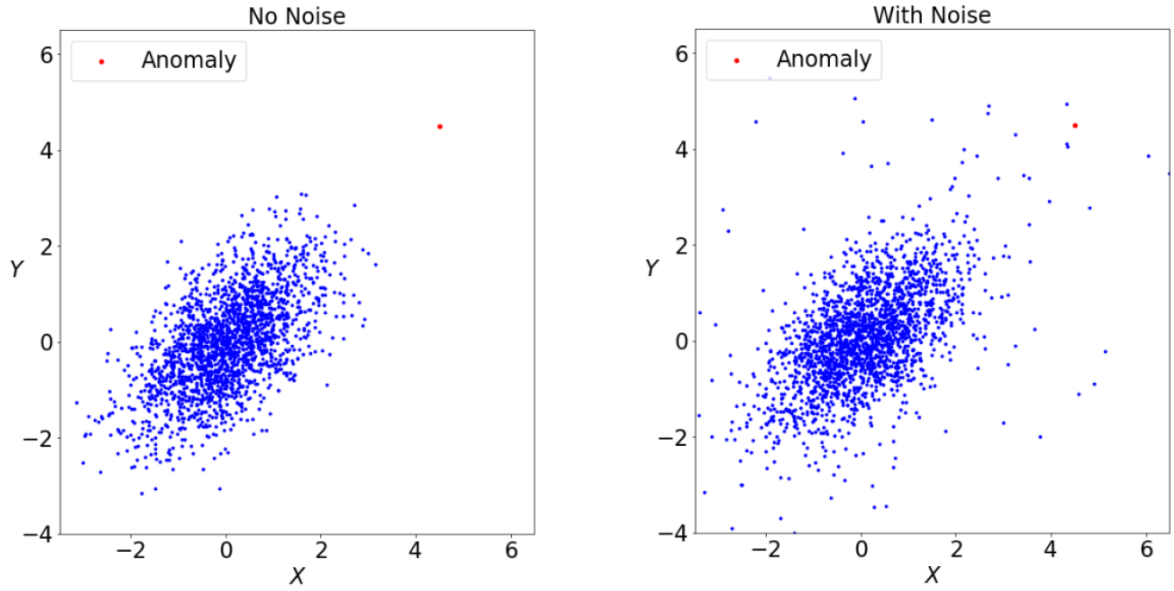



Figure 2.1: *Data without noise and Data including noise, reproduced from [8]*

The Figure 2.1 show identical data distributions. There is a clear anomaly in left subplot of Figure 2.1 , as the red point marked has a noticeable deviation from the rest of the plot. Wheras in right subplot of Figure 2.1 is difficult to decipher, because it is difficult for the anomaly point to be marked off from the other points in sparse space. It can be infered that the datasets with irregularities or noise create a challenge for detecting anomalies. It is therefore essential that you have a comphrehensive understanding of the dataset. The anomalies must be differenciated from the novelty data point patterns [10][46] that are not recognised in the data. These novelities might be treated as normal in the detection process of the anomalies.

## 2.2 Time Series Analysis

### 2.2.1 Data Preprocessing

The data must be chronologically ordered and the timestamps should be equidistant during subsequent periods of time. By sorting the data frame in accordance with the timestamps, the chronological order can be achieved for the respective data input. Timestamps that are equidistant indicate a constant interval of time. In order to verify whether the timestamps are equidistant, the difference between each timestamp is

taken into account. We can re-sample the data by determining a constant time interval if the timestamps are not equidistant.

### 2.2.2 Handle the Missing data

The dataset may contain null values. If we plot the time series data, it might show us the zero and null values. These values are to be cleaned by replacing them with NaN values. Figure 2.2 will portray the NaN values that can be filled later with an outlier value or zero, mean value, with the last value or with a linearly interpolated value.

1. To fill out NaN with outlier value or zero.
   The NaN value can be filled with a zero or infinite value using np.inf or even a high negative value like -999.
2. To fill out NaN with Mean Value.
   NaN values can also be filled with the mean value, but it does not seem to be the best option.
3. To fill out NaN with Last Value.
   Using the .ffill() last values can be populated to NaN values.
4. To fill out NaN with a linearly interpolated value.
   Using the knowledge of the neighboring values, the NaN values can be populated using the .interpolate() method. This is the best option to handle the missing values.

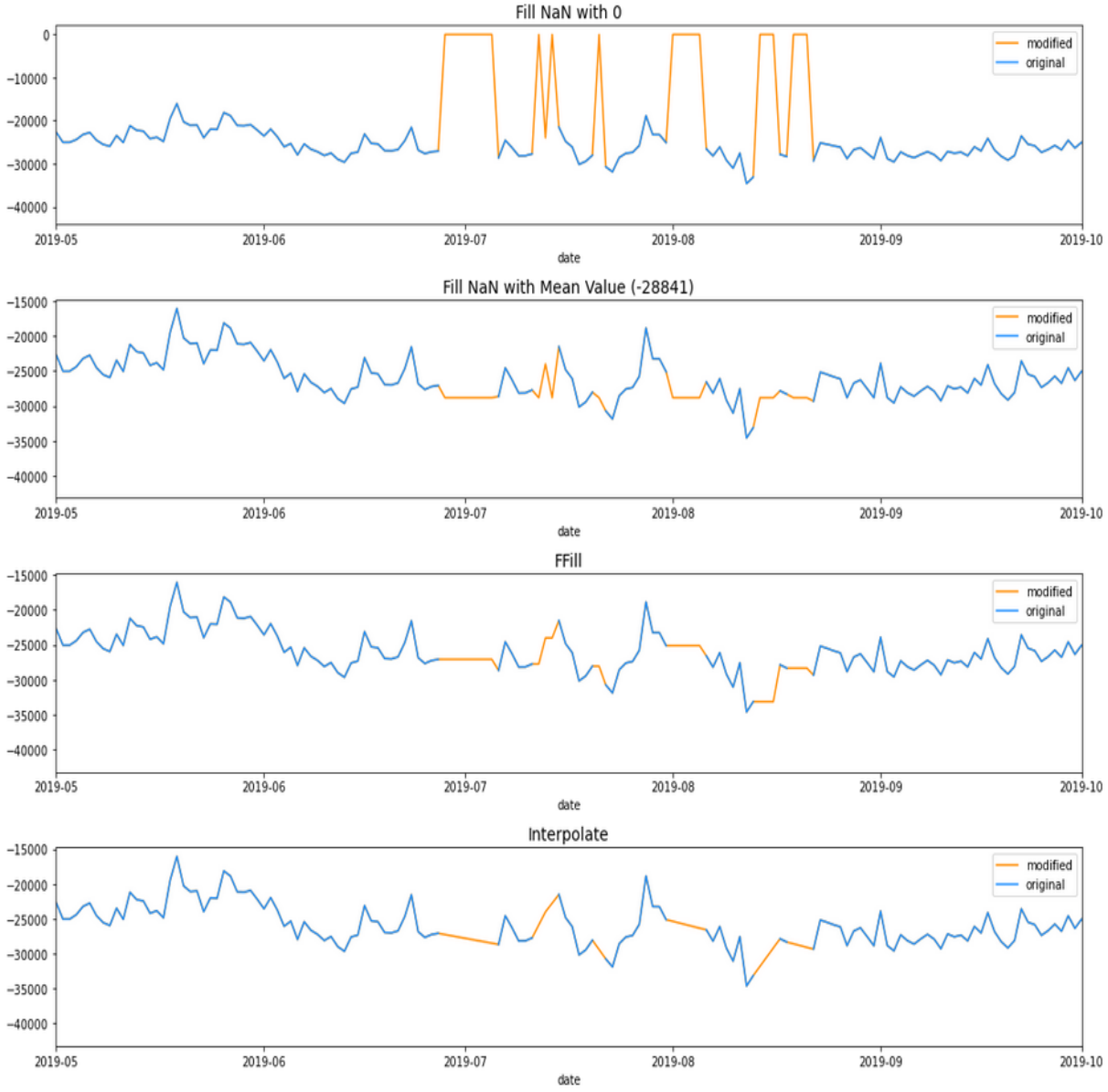



Figure 2.2: *Handling the missing data in time series namely Fill NaN with 0, Fill NaN with Mean value, Fill with Last value and Linear Interpolation, reproduced from [4]*

### 2.2.3 Stationarity

During each interval of time, a stationary time series shows the same characteristics and is constant at any time interval.
Basically, *"$X_t$ is a stationary time-series, if $\forall$ $s \epsilon R$ : the distribution of ($x_t$, . . . , $x_t + s$) is equal."* [26]

From the above statement we can entail that in a stationary time series $x_1$,...,$x_T$ will consist the characteristics as follows:

1. There is no trend in the time-series due to the constant mean.
2. A constant variance characterizes the time series [7].
3. Over specific period of time the autocorrelation is constant.
4. Seasonality is absent, i.e., no periodicity is present in the series[41].

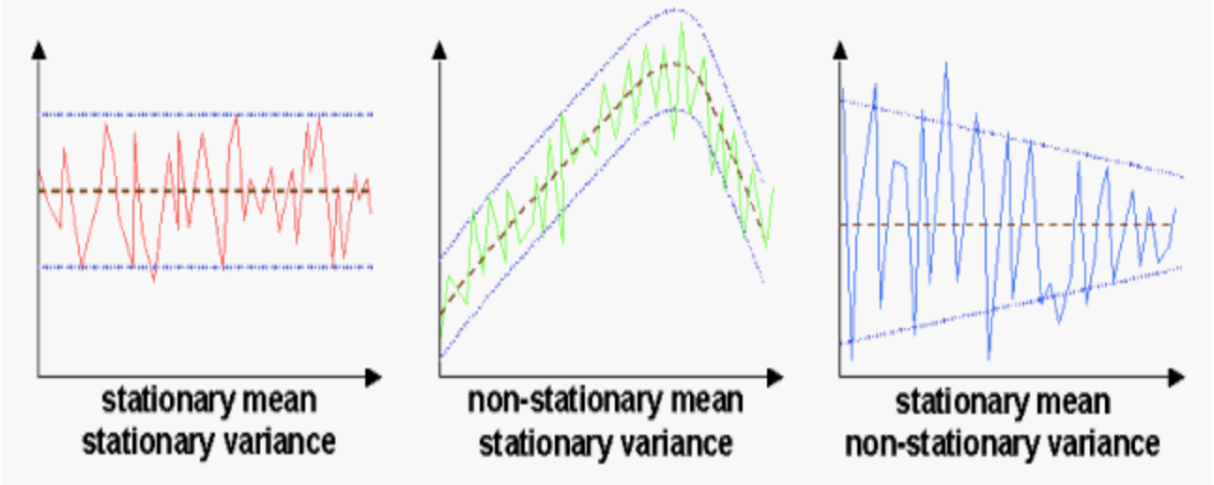



Figure 2.3: *Constancy in mean and variance from [44]*

There are three methods for checking stationarity.

1. Analyze trends and seasonality by plotting time series.
2. Analyze each partition's mean and variance by splitting the time series into parts.
3. Augmented Dickey Fuller test[40]

### 2.2.4 Data Smoothening / Resampling

It is possible to resample the data to obtain additional information[39]. There are two types of resampling [62]:

- In the case of increasing the frequency of sampling, it is Upsampling. An Upsampling will increase the number of samples in signals using a "Zero-Padding Procedure" [5]. In particular, during the upsampling, a signal is sampled multiple times and zeros are added in between.

- In the case of decreasing the frequency of sampling, it is Downsampling. The downsampling the sample size is decreased. A new signal is then formed by taking a point from each segment of an input signal, or by dividing the signal into segments [32].
  The data will be downsampled with a specific time-frequency and normalized using the Statistical or Z-score Normalisation:
  $$z = \frac{(x–u)}{s} \tag{2.1}$$
  where $x$ is data to normalize, $u$ is the mean of training samples and $s$ is the standard deviation of training samples [66].

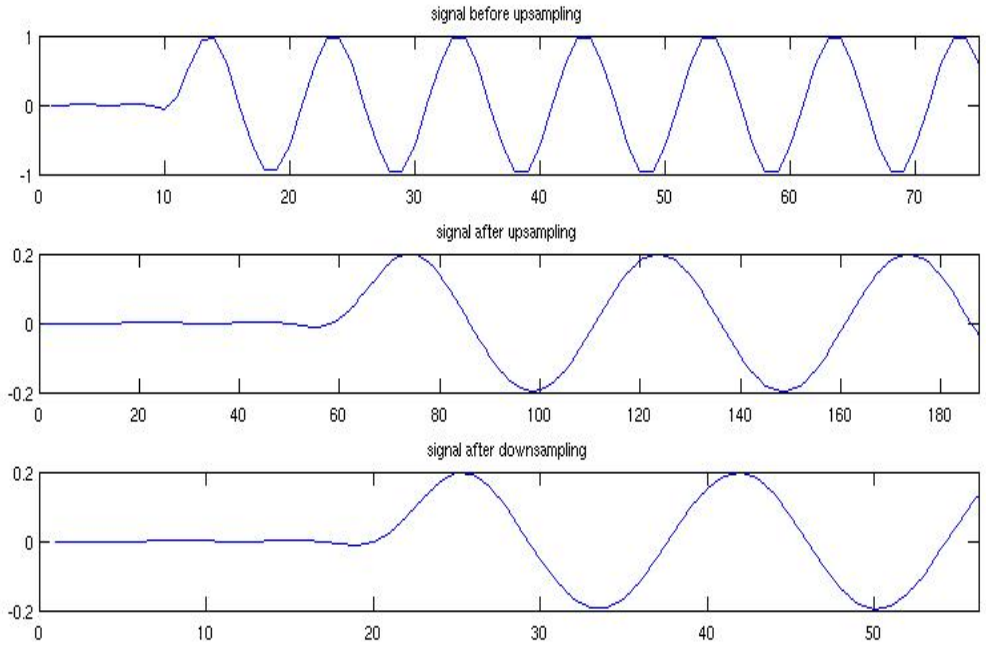



Figure 2.4: *A sine wave function which is upsampled by a factor of* 5 *and downsampled by a factor of* 3 *by using MATLAB embedded function with order* $= 20$*, cut of frequency* $= \frac{1}{5}$ *from [67]*

## 2.3 Anomaly Detection

Deviations are typically deemed as measures of the strength of an anomaly, or also as a measure of its probability of being one. This is called an anomaly score [1]. Those with a significant anomaly score are considered anomalous, otherwise normal patterns are considered regular.

Based on the properties of the data the anomalous detection approaches will vary. These are the important ones.

- Temporal or Non temporal data: Temporal data are the data with timestamps but in non equal intervals, whereas the non temporal data will be protein sequences or the medical images.
- Uni variate or Multivariate data: Uni variate will be in a single dimension, whereas the multivariate has multiple dimensions like the images.
- Labeled or unlabeled data: If there exists some annotation of the data set then it is labeled data else it is unlabeled data.
- Different anomaly types in the data set. These can be point anomalies, collective anomalies and contextual anomalies.

### 2.3.1 Statistical approaches based anomaly detection

We have many regressive models like AR, MA, AR MA, ARIMA which works in a statistical way of approaching the anomaly detection.

#### Auto regressive Model (AM)

AR is a linear model that uses a set of prior values of length $p$ and an error value to predict the current value of the stochastic process $X_t$

$$X_t = \sum_{i=1}^{p} a_i.X_{t-1} + c + \varepsilon_t \tag{2.2}$$

Uni variate time-series stochastic models are among the most basic models which are Auto regressive in nature, these models assume the data to be as stationary [11].

#### Moving Average Model (MA)

The Moving average model will consider the latest p values of the combination along with the linear transformation [49].

$$X_t = \sum_{i=1}^{q} a_i.\varepsilon_{t-1} + \mu + \varepsilon_t \tag{2.3}$$

#### Auto regressive Moving Average Model (ARMA)

This model is the combination of the MA and AR., usually used for uni variate time series data.

$$X_t = \sum_{i=1}^{p} a_i.X_{t-1} + \sum_{i=1}^{q} b_i + \varepsilon_t \tag{2.4}$$

The Box Jenkins methods, correlograms, Leave One Out cross validation [1] are the methods used to fit the model for different values of p and q.

#### ARIMA Model

The major problem in ARIMA model is that it can be non stationary. This ARIMA model [38] is usually called as the generalised model of ARMA. An additional d factor called as differencing also adds up along with p and q.
$X'_t = X_i - X_{i-1}$

### 2.3.2 Classical machine learning approaches based anomaly detection techniques

The algorithms employed by machine learning algorithms search for anomalous behavior in time series data without making assumptions about a particular generative model. To make time-series prediction and identify time-series anomalies, one does not have to know the underlying processes of the data [6]. Several techniques for uni variate anomaly detection based on classical machine learning are discussed in this section.

#### K-Means Clustering

K means is a clustering algorithm also know as *Subsequence time-series Clustering (STSC)*[37]. The sliding window concept[63] is used in the K-Means clustering[16] where the w is window length and slide length is $\gamma$. The euclidean distance will be used in case of the uni variate time series data. By defining the threshold on a window, the anomaly is determined if the error value is comparatively high.

#### Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

The DBSCAN [13] will analyses the density of the data and is a clustering algorithm. Three different categories will interact with the model with different data points are Core points, Border points, and the anomalies. For univariate dataset [9] anomaly detection is done using DBSCN.

#### One-Class Support Vector Machines (OC-SVM)

The first SVM was built in a linear supervised approach[58] in 1953. An OC-SVM is a semi-supervised approach in which only a single class of data is used for training [51]. Anomalies were initially detected

by OC-SVM using a set of vectors rather than time series. The normalized data must be provided to SVM to get a non biased results.

### 2.3.3 Neural Network approaches for Anomaly detection

Forecasting and analysis of the time series and major tasks related to the detection of objects , segmentation and classification the results are good when using newural networks for these tasks. The most frequently used approaches are

#### Multiple Layer Perceptron (MLP)

The Multiple Layer perceptron[8] uses the artificial neural network as a fundamental part and it is a feed forward network which is fully connected. The MLP[8] has more number of hyperparameter when compared to the other neural network architechtures. They are the Length of window, Depth and width of the MLP, learning rate and the function optimizer. Figure 2.5 shows the Multilayer perceptron

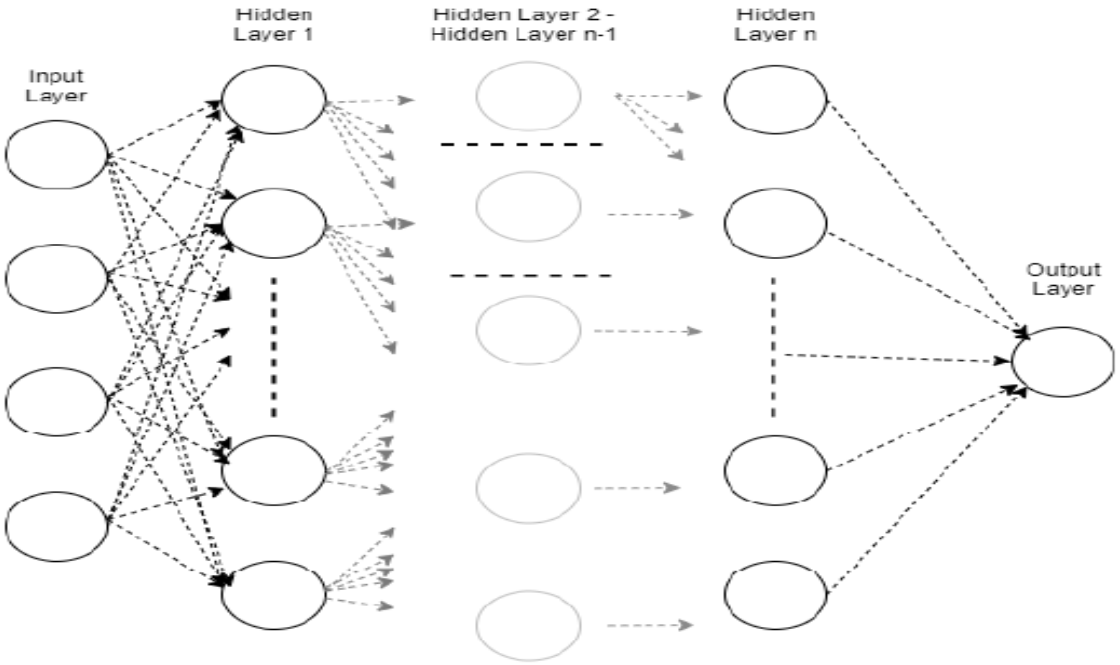



Figure 2.5: *Multilayer Perceptron from [8]*

### Convolutional Neural Networks CNN

Deep Convolutional Neural Networks are another neural networks which is commonly used in time series analysis for anomaly detection. The training time is faster as it can be deep or partial neural network, and it focuses on the data patterns by pooling the results[34].

The Figure 2.6 shows the time series prediction in an Deep convolutional neural network [34].

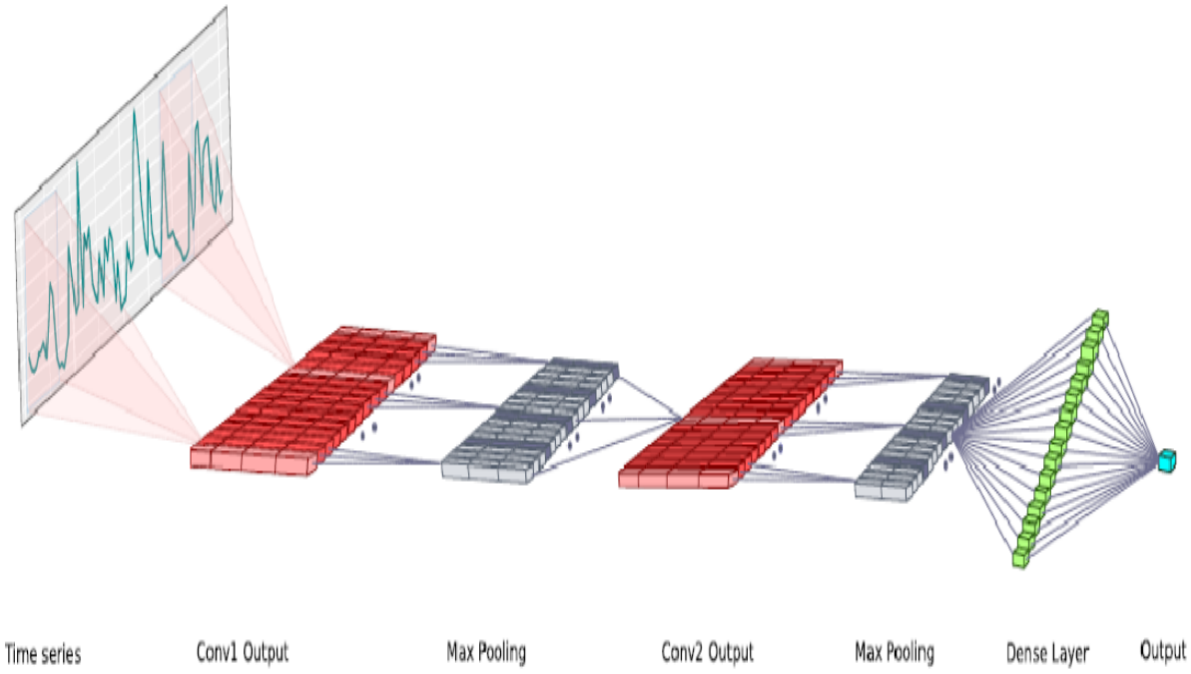



Figure 2.6: *Deep Convolutional neural network reproduced from [8] [34]*

# 3

# Methodology

This section discusses about the datasets used, different architectures applied on these datasets for experimentation, anomaly detection system. The section illustrates the datasets used for this experiment, different architectures applied to these datasets for detection of anomalies.

## 3.1 Dataset description

The following section describes about the two time-series datasets (NASA bearings datasets and NAB dataset) that were used to train the models for anomaly detection.

## 3.2 Numenta Anomaly Benchmark (NAB) dataset

The Numenta Anomaly Benchmark (NAB) [35] is a novel benchmark that evaluates techniques for detecting anomalies in streaming, online applications. A novel scoring mechanism is designed for real-time applications that integrates over 50 time series files labeled with real-world and artificial data.

### 3.2.1 Data corpus of NAB

This corpus of 58 timeseries files consists of timeseries data to assist in the study of stream anomaly detection. The dataset [35] consists both of real-world data and of artificial data that contains labeled periods of anomalous behavior. Using timestamps, sorting, and a single value, the data metrics are compiled using variety of real world applications The Artificially generated data with and without anomalies were used for the initial stages of the experimentaion for understanding the concept behind similar time series data. This small dataset containing day to day activities as a time series data was used as a experimentation purpose before using a larger dataset. Each dataset contained a timestamp along with the value associated with it [35].

## 3.3 IMS NASA Bearing dataset

In many of the bearing diagnostics [42] research generally it is to study defective bearings extracted from the field, which have mature faults, or to study bearings that have been damaged in a simulated

environment. Usually, damage simulations are achieved by adding debris into the lubricant, or by running machines with electrical discharges or scratches[47]. Natural defect propagation is harder to detect in early stages of an experiment using defective bearings. This is a *"Run to failure"*[28] dataset.

### 3.3.1 The test rig

In 2014, IMS bearings, tested under endurance test rigs, have been released by the University of Cincinnati [28]. All the reading recorded are in ASCII format. The test rig composing of the following features has been illustrated in the Figure 3.1:

- Rexnord ZA-2115 type 4 double row bearings has the charactertics as depicted in the Table 3.1,
- Stationary speed of 2000 rpm,
- A spring mechanism supports the bearing and the shaft is subjected to a load of 6000 lbs,
- High sensitivity Quart ICP® accelerometers PCB 253B33.

AC motors are responsible for maintaining the motor's rotation speed, while belt drives keep this speed constant. There is a circulation system between the four bearings on one shaft that controls both the flow and temperature of forced lubrication. This information is available in the "Readme Document for IMS Bearing Data" provided with the file download [47], tt was noted that debris accumulated above a certain level in the magnetic plug during testing, which indicated an impending failure.

Four double row Rexnord ZA-2115 bearings were installed on one shaft as illustrated in Fig 3.1. 16 rollers are placed in each row in the bearing, and the diameter of the roller is 0.331 inches. The bearing pitch is 2.815 inches with a contact angle tappered at 15.171. The bearing housings are each equipped with a *CB 353B33 High Sensitivity Quartz ICPs Accelerometer*[45]. Each bearing was fitted with four thermocouples so that the bearing temperature could be recorded for purposes of lubrication monitoring. *DAQCard-6062E data acquisition card of National Instruments* [27] was used to collect vibration data every 20 minutes. There are 20480 points in each data set at a sampling rate of 20 kHz. The LabVIEW program [12] from the National Instruments was used for the data collection process.

All the four bearing are identical in nature. The rolling elements bearing consist of two ranges of pillow blocks. Table 3.1 provides information on their characteristics.

The Table 3.2 shows the first dataset in the IMS NASA repository. It consisted of 2,156 files having 8 channels. The channels were arranged for each of the bearings, two channels for one bearing respectively. At every 10 minutes the data was recorded except the first 43 files, which were recorded for 5 minutes. The duration was of thirty four days.

| **Characteristics of the Rexnord ZA-2115** | | |
|---|---|---|
| Diameter Pitch | 2.815 inch | 71.5mm |
| Diameter of the rolling element | 0.331 inch | 8.4mm |
| Number of rolling element per row | 16 | 16 |
| Contact angle | 15.17° | 15.17° |
| Static load | 6000 lbs | 26690 N |

Table 3.1: Rexnord ZA-2115 Characteristics tabulated using [19][47]

| **Set No. 1:** | |
|---|---|
| Recording Duration: | October 22, 2003 12:06:24 to November 25, 2003 23:39:56 |
| No. of Files: | 2,156 |
| No. of Channels: | 8 |
| Channel Arrangement: | Bearing 1 – Ch 1&2; Bearing 2 – Ch 3&4;<br>Bearing 3 – Ch 5&6; Bearing 4 – Ch 7&8. |
| File Recording Interval: | Every 10 minutes (Exception is that for every 5 mins first 43 files were taken away ) |
| File Format: | ASCII |
| Description: | At the end of the test-to-failure experiment, inner race defect occurred in<br>bearing 3 and roller element defect in bearing 4. |

Table 3.2: Dataset description of Set 1 of NASA IMS Bearing Dataset, tabulated using [19][47] [28]

The Table 3.3 shows the second dataset in the IMS NASA repository. It consisted of 984 files having 4 channels. The channels were assigned for each of the bearings At every 10 minutes the data was recorded. This data acquisition was done for a period of six days.

| **Set No. 2:** | |
|---|---|
| Recording Duration: | February 12, 2004 10:32:39 to February 19, 2004 06:22:39 |
| No. of Files: | 984 |
| No. of Channels: | 4 |
| Channel Arrangement: | Bearing 1 – Ch 1; Bearing2 – Ch 2;<br>Bearing3 – Ch3; Bearing 4 – Ch 4. |
| File Recording Interval: | Every 10 minutes |
| File Format: | ASCII |
| Description: | Outer race failure occurrs in bearing 1,<br>at the end of the test-to-failure experiment, |

Table 3.3: Dataset description of Set 2 of NASA IMS Bearing Dataset, tabulated using [19][47] [28]

The Table 3.4 shows the second dataset in the IMS NASA repository. It consisted of 4448 files having 4 channels. The channels were assigned for each of the bearings At every 10 minutes the data was recorded. This data acquisition was done for a period of thirty one days.

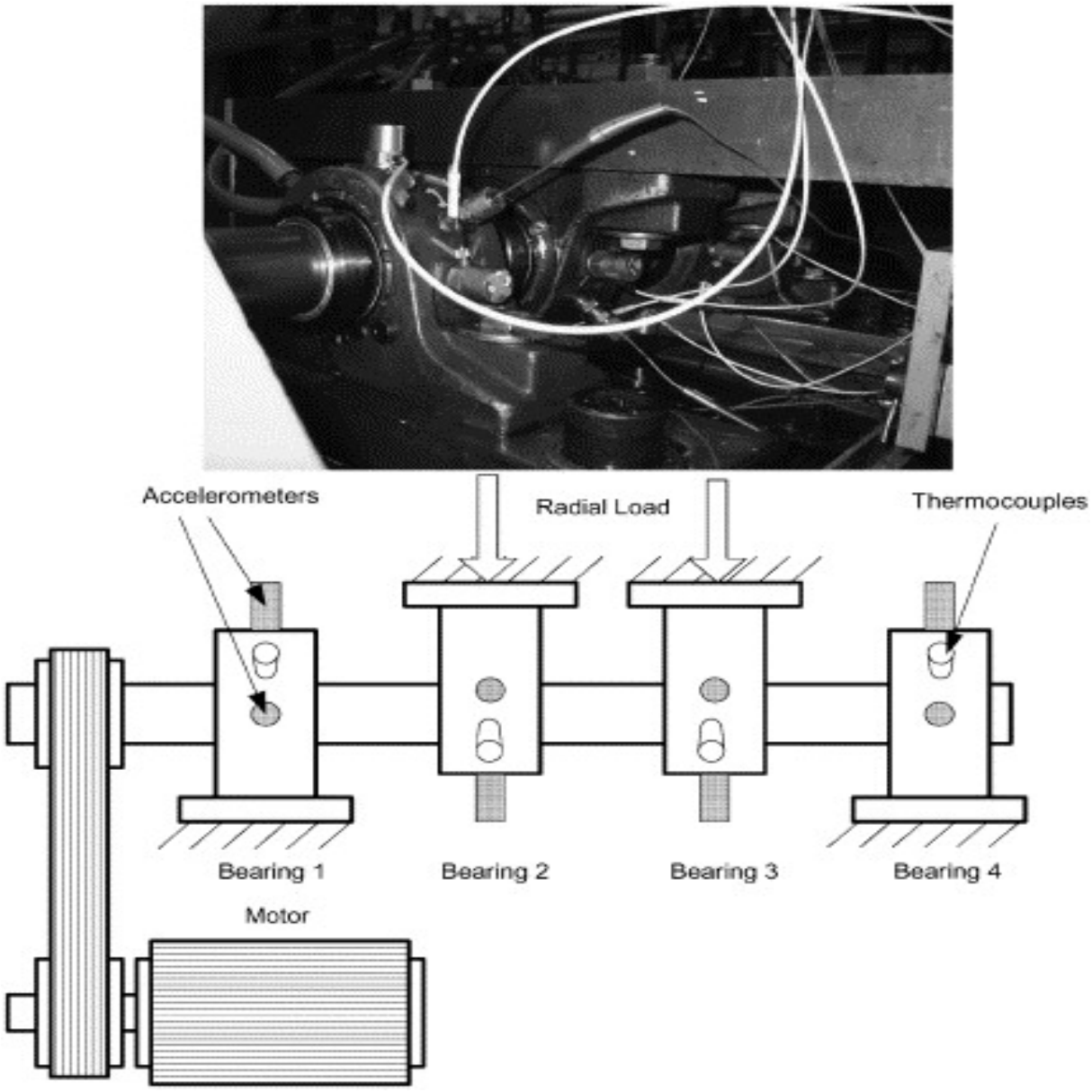



Figure 3.1: *Illustration showing bearing assembly and sensor placement, reproduced from [47]*

### 3.3.2 Data description

The downloaded file from the NASA repository includes three datasets, each consisting of several files of one second. During the recording process, each file is represented by 20,480 samples with a sampling frequency of 20 kHz. The data acquisitions take place every ten minutes at a rate of one second. In the

| **Set No. 3:** | |
|---|---|
| Recording Duration: | March 4, 2004 09:27:46 to April 4, 2004 19:01:57 |
| No. of Files: | 4,448 |
| No. of Channels: | 4 |
| Channel Arrangement: | Bearing 1 – Ch 1; Bearing2 – Ch 2;<br>Bearing3 – Ch 3; Bearing 4 – Ch 4. |
| File Recording Interval: | Every 10 minutes |
| File Format: | ASCII |
| Description: | outer race failure occurred in bearing 3,<br>at the end of the test-to-failure experiment. |

Table 3.4: Dataset description of Set 3 of NASA IMS Bearing Dataset, tabulated using [19][47] [28]

first dataset for every 5 minutes, the first 43 files are acquired,

The description of the datasets is given the Table 3.5 concerning the dataset 1, there are two accelerometers on each bearing (x and y positions). The two other datasets only have one accelerometer on each bearing.

| | Number of files | Number of channels | Endurance duration | Duration of recorded signal | Announced damages at the end of the endurance |
|---|---|---|---|---|---|
| Dataset 1 | 2156 | 8 | 49680 min<br>34 days 12h | 36 min | Bearing 3: inner race<br>Bearing 4: rolling element |
| Dataset 2 | 984 | 4 | 9840 min<br>6 days 20h | 16 min | Bearing 1: outer race |
| Dataset 3 | 4448 | 4 | 44480 min<br>31 days 10h | 74 min | Bearing 3: outer race |

Table 3.5: Summarized dataset description of NASA IMS Bearing Dataset [47]
, tabulated using [19][47]

Calculations based on bearing characteristics were made to determine the fault frequencies for diagnosis of rolling element bearings. The fault frequencies required for the diagnosis of the rolling element bearing have been calculated from the bearing characteristics. The following Table 3.6 summarizes them .

| **Characteristic frequencies** | |
|---|---|
| Frequency of the shaft | 33.3 Hz |
| Ball Pass Frequency Outer race (BPFO) | 236 Hz |
| Ball Pass Frequency Inner race (BPFI) | 297 Hz |
| Ball Spin Frequency (BSF) | 278Hz (139x2 Hz) |
| Fundamental Train Frequency (FTF) | 15 Hz |

Table 3.6: Test rig frequency characteristics tabulated using [19][47]

### 3.3.3 Experimental results of bearing tests

The test bearing was subjected to 35 days of testing and was eventually found to possess significant metal debris on the magnetic plug. According to Figure 3.2, test bearing 3 has been found to have failed its inner race while bearing 4 has suffered a failure of its outer race. .

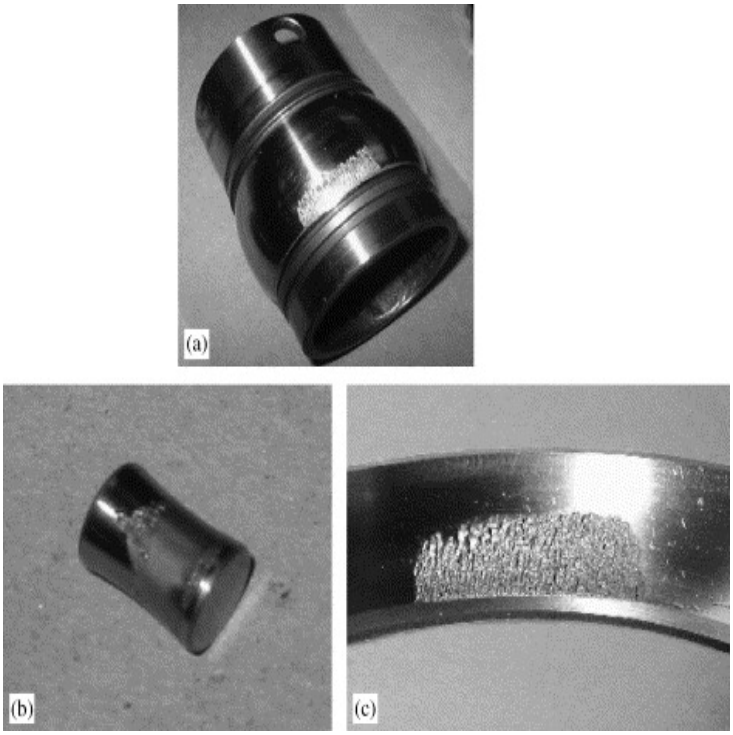



Figure 3.2: *A picture of a bearing component after it has been tested: (a) bearing 3 inner race defect , (b) bearing 4 roller element defect and (c) bearing 4 outer race defect, reproduced from [47]*

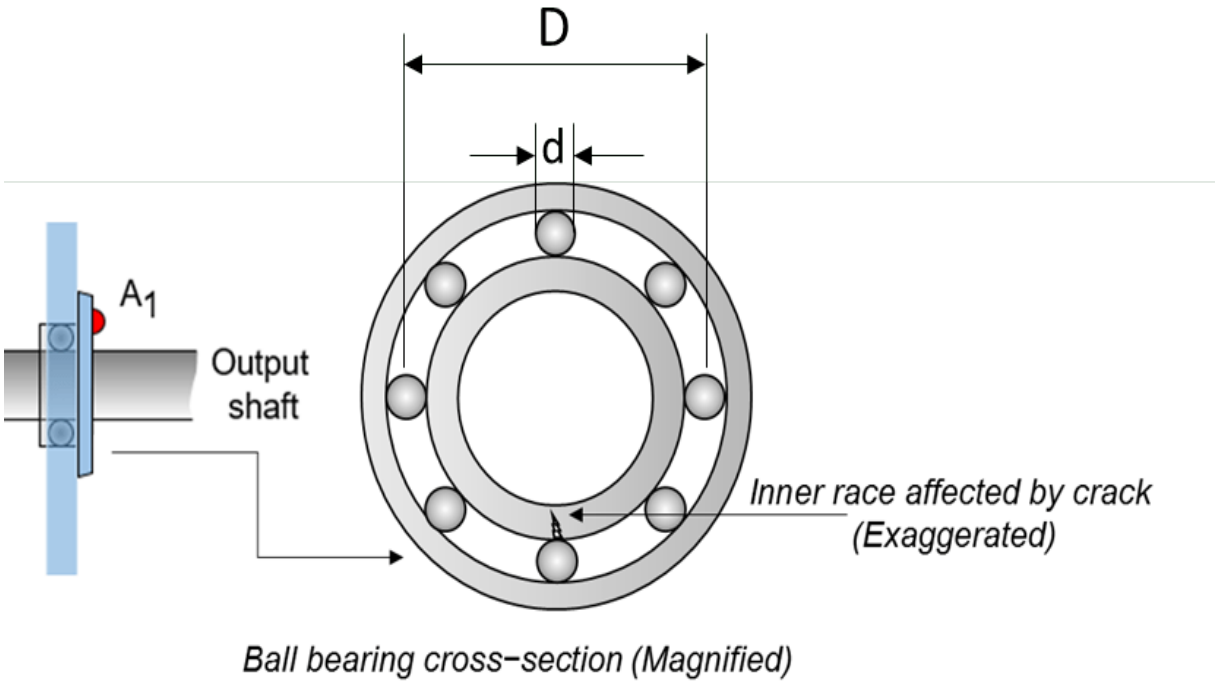



Figure 3.3: *Magnified image of a Ball bearing cross-section, reproduced from [61]*

The outer race, inner race, cage, or rolling element of a bearing equipped with rolling elements can develop localized faults during operation. Bearing response transducers are subjected to high-frequency resonant frequencies as the rolling elements strike any local faults on the outer or inner races or if any internal faults are encountered between the bearing and the rolling elements [48]. An inner race fault on a rolling element is illustrated in the following Figure 3.3.

## 3.4 Machine Configuration

The datasets which are described in the previous section is large. To process these large datasets, high memory capacity and good computation power is required. The experimentation is conducted on the cluster provided by the Platform for Scientific Computing at Bonn-Rhein-Sieg University [23]. It is built of 5,300 hardware threads that interface 48,000 graphics processing units (GPUs), each with 460 TeraBytes (TBs) of collective memory. Students' research work is enabled by a subset of this configuration. The details are listed in the following table 3.7

| **Components** | **Description** |
|---|---|
| Operating system | Nitrogen, Linux 7.8 |
| Central Processing Unit (CPU) | 50 nodes with 2x Xeon processors having 16 cores |
| Graphical Processing Unit (GPU) | Nvidia Tesla V100 |

Table 3.7: Machine configuration at Bonn-Rhein-Sieg University [23]

In Appendix B, the software prerequisities are explained. Before executing the code on the University Computing Platform [23], it is recommended to test a small portion or subset for errors.

## 3.5 Methods

There have been many studies conducted on time series anomaly detection and fault/anomaly detection [1]. The generation of features is a major part in the field of anomaly detection and analysis of data. The time and frequency domain features are used for condition monitoring and remaining user life estimation [47] [57]. Deep neural networks have been demonstrated to be efficient at classifying faults by using features extracted from time and frequency domains to develop autoencoders. [29].An anomaly score was computed based on the reconstruction errors on sliding windows. The best way to model time series data, such as sensor data, is to see each time step as being linked to the previous one.

### 3.5.1 LSTM autoencoders

With LSTM-based autoencoders, time-series sequences can be encoded as well as inputs of varied length. The input for an autoencoder will be the measurement vibration signal data with which it will be trained so as to get the least reconstruction error in between the reconstruction and the original data signal. The mapping of the input vector $x$ into the hidden representation of low dimension $h'$ is done using the encoder part, from the input vector $x$, estimation of $x'$ is performed by the decoder. LSTM are recurrent neural networks which will maintain the hidden state information based on the previous memory along with the addition of the temporal information [20]. The LSTM consists of majorly two components: the encoder and the decoder. The equations represented below are reproduced from [2] research paper.

### 3.5.2 Training Phase

- Encoder: The input sequence is

$$X = (x^1, x^2, ...., x^N) \tag{3.1}$$

For every input sequence

$$C'_t = \tanh\left(W_c.x_t + R_c.h_{t-1} + b_c\right) \tag{3.2}$$

The vector output of the decoder will be defined as

$$h_t = \sigma^e_\phi(x_t, h_{t-1}) \tag{3.3}$$

where

- $h_t$ is the encoder ouput,
- $C_t, x_t, h_{t-1}$ will be the memory state , input and output respectively from the previous step.
- $\sigma$ is the ReLU function.
- $\phi$ and $x'$ will be the reconstructed errors.

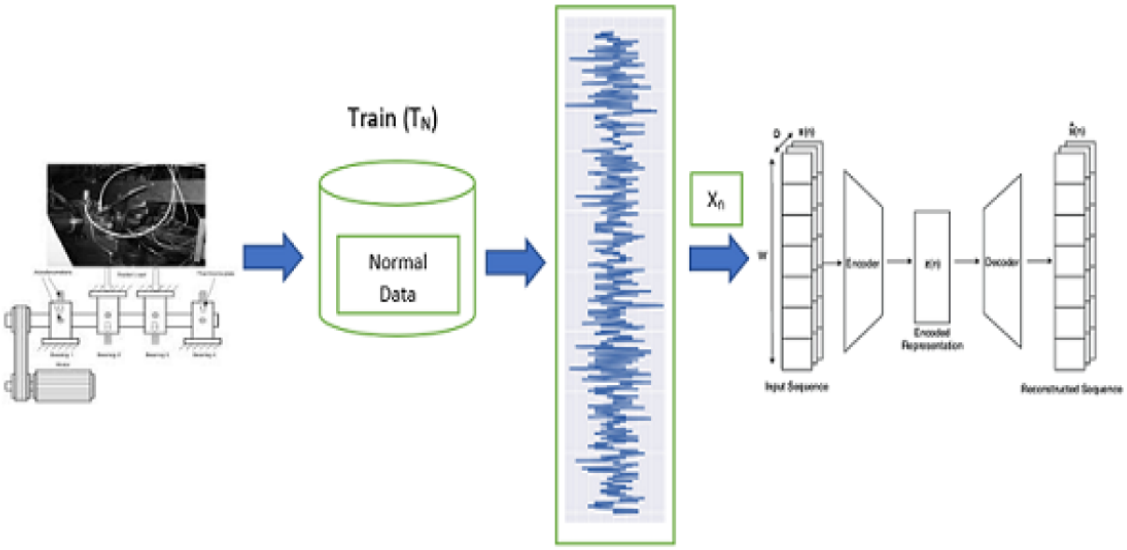



Figure 3.4: *The LSTM-AE involves diagnosing anomalies through condition monitoring for the training phase, visualized using [3]*

As shown in the Figure 3.4, it will illustrate the Training phase of the condition monitoring using LSTM-AEs for diagnosis of the anomalies. The input raw data is usually normalised and then divided into four

sets:

- $T_N$ defining a training set,
- $V_N$ & $V_A$ defining the two validation sets, and the
- $T_A$ will define a test set.

In the training phase Figure 3.4 the train set $T_N$ will be compressed within the autoencoder. The specific weights are assigned using the adam optimizer and the gradient descent with small mini batches. For the training to be stable and fast , normalisation of the batches are required. The drop out rate is defined for the restriction of the model from overfitting and for avoiding the clipping of the gradients.

### 3.5.3 Detection phase

As a way to identify anomalies, reconstruction errors are converted into anomaly scores for each input sequence, and the scores are then used to derive a threshold for determining whether the machine has behaved normally. The reconstruction errors are generated after the training phase, these are used for estimation of the normal distribustion parameters. The anomaly score of datapoint $x_t^N$ is defined as below:

$$a_i = (e_i - \mu)^T \sum^{-1} (e_i - \mu) \tag{3.4}$$

where

- $p_i$ is the probability of getting the error of reconstruction.
- $e_i$ is the reconstruction error.
- An anomaly score $\sum$ and a score $\mu$ are assigned to two data points

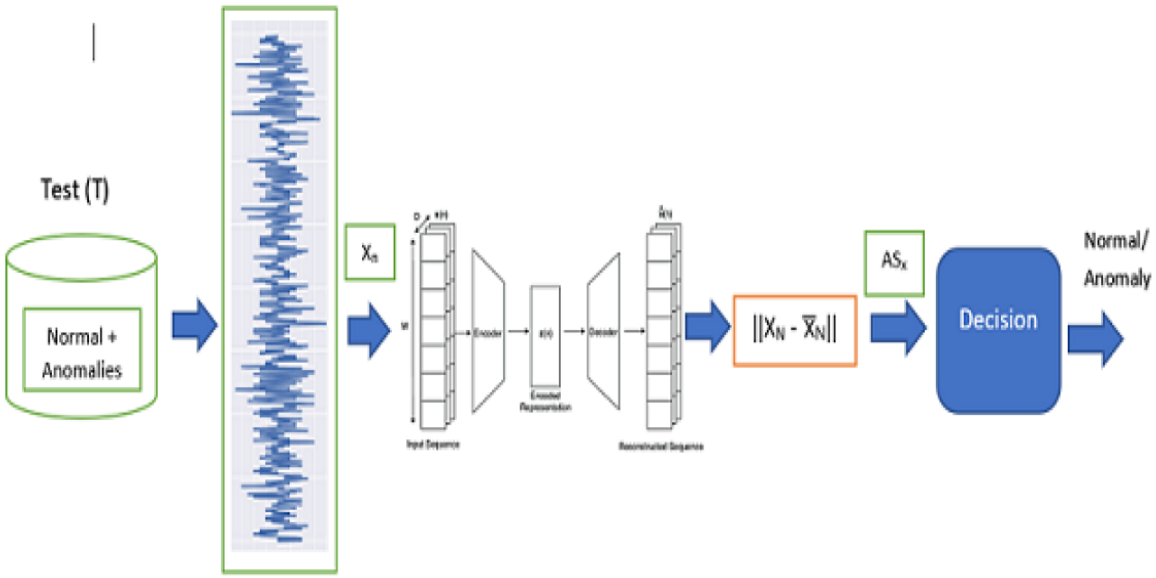



Figure 3.5: *The LSTM-AE involves diagnosing anomalies through condition monitoring for the detection phase, visualized using [3]*

As depicted in the Figure 3.5 the detection phase of the process, the validation set will learn the threshold anomaly score along with the sequences which are normal in nature. If the anomaly score ¿ T then the sequence will mark it as a anomaly, else they are considered as normal patterns.

# 4

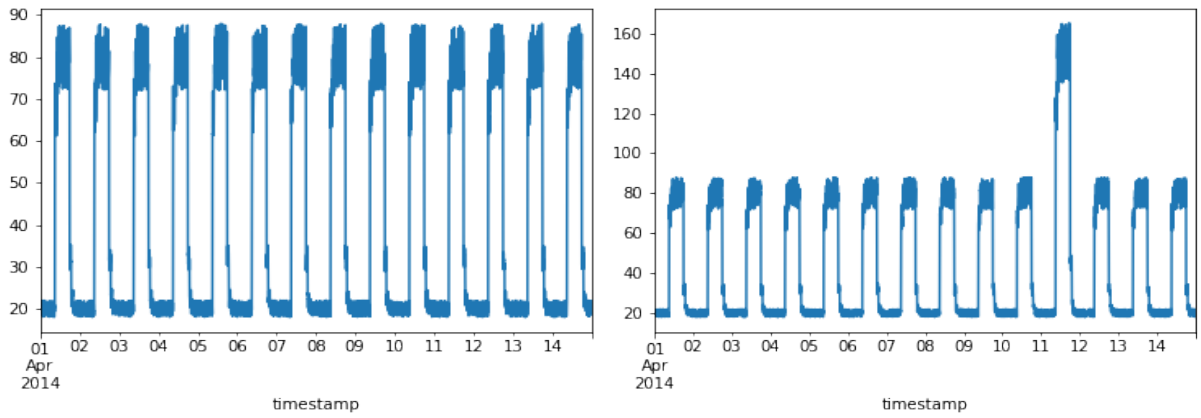
# Results

Analyzing the data obtained through the implementation of the methods is the focus of this chapter. First, we will discuss the preliminary results of the experiment that was conducted on the *Numenta Anomaly Benchmark(NAB) dataset* [35], later part of the chapter discusses on the results obtained by using the IMS NASA Bearing dataset [28]. The dataset description is explained in the previous chapter.

## 4.1 Numenta Anomaly Benchmark(NAB) dataset

The early stage of the implementation was done using a small subset of the Numenta Anomaly Benchmark(NAB) dataset [35]. The description about the dataset is already given in Section 3.2. A reconstruction convolutional autoencoder model was used to detect the anomalies in the time series data. The *art_daily_small_noise.csv* [35] file is being used for the training phase and the *art_daily_jumpsup.csv* [35] file for the testing phase.

The visualization of the training data which is used without the anomalies are as shown in the Figure 4.1 (a). The visualization of the testing data with the anomalies are as shown in the Figure 4.1 (b).




(a) Timeseries data without anomalies

(b) Timeseries data with anomalies

Figure 4.1: The plots are visualized using the dataset from [35], (a) Timeseries data without anomalies and (b) Timeseries data with anomalies

The data values from the timeseries training set is normalised for 5 mins each over a period of fourteen days. So, it will be a total of 28 timesteps per day, and a total of 4032 data points.

A convolutional reconstrtion autoencoder model is created with dropout rate of 0.2 and learning rate as 0.001. The batch size is set as 128 with 50 epochs.

After the training process, the Training Vs Validation loss is plotted as a subplot as the Figure 4.2.

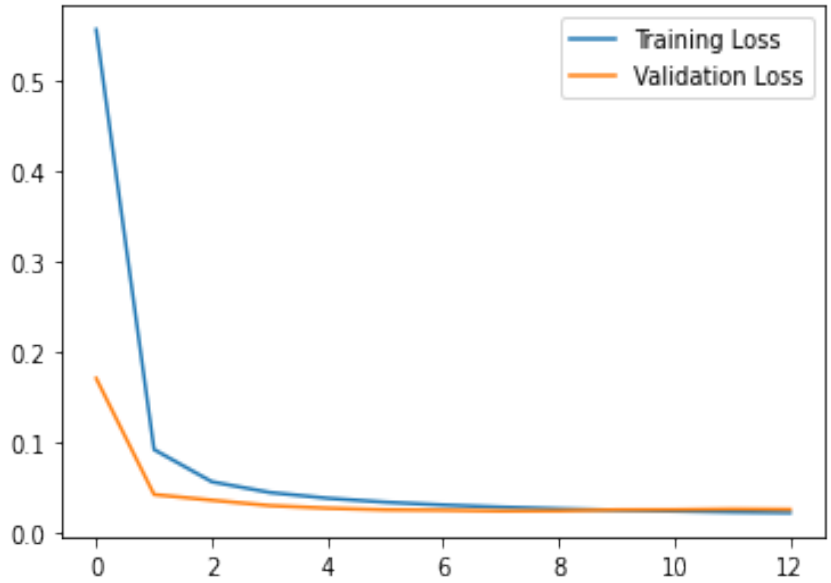



Figure 4.2: *Training Vs Validation loss, visualised using the dataset from [35]*

Analyzing our model's ability to reconstruct anomalies will allow us to detect anomalies. Analyze training samples to determine MAE loss [52]. By determining the max MAE loss value, we can find an anomaly detection threshold. The max MAE value will be worst case performance of the model while trying to reconstruct the input. The Figure 4.3 illustrates the reconstruction error threshold.
The Mean Absolute Error (MAE) is defined by using the equation [52]:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{4.1}$$

The Figure 4.4 will illustrate the first sample after the reconstruction from the model. For this experimentation, the reconstruction error obtained was 0.12938.

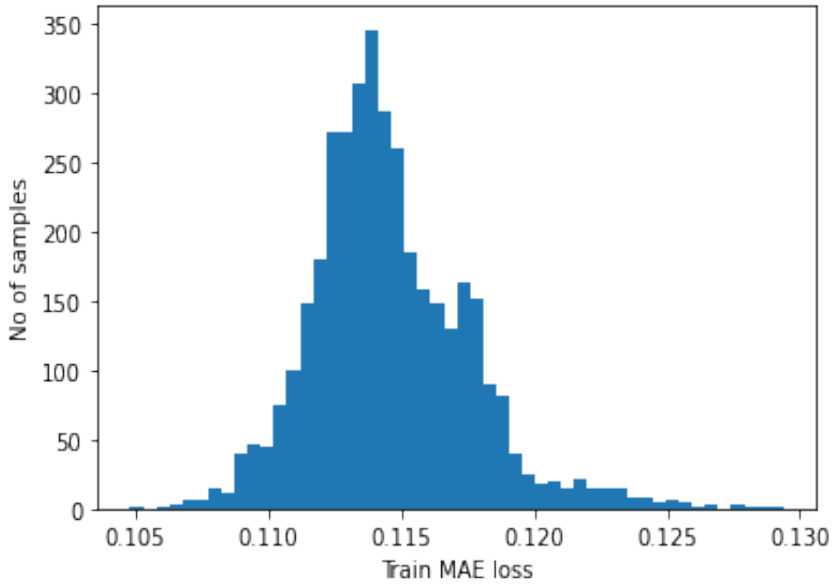



Figure 4.3: *Reconstruction error threshold, visualised using the dataset from [35]*

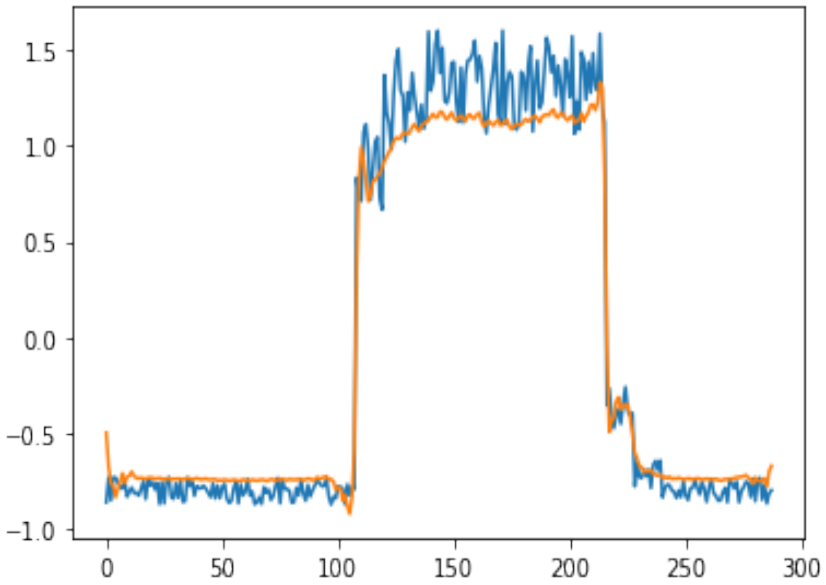



Figure 4.4: *Reconstruction plot for the first sample, visualised using the dataset from [35]*

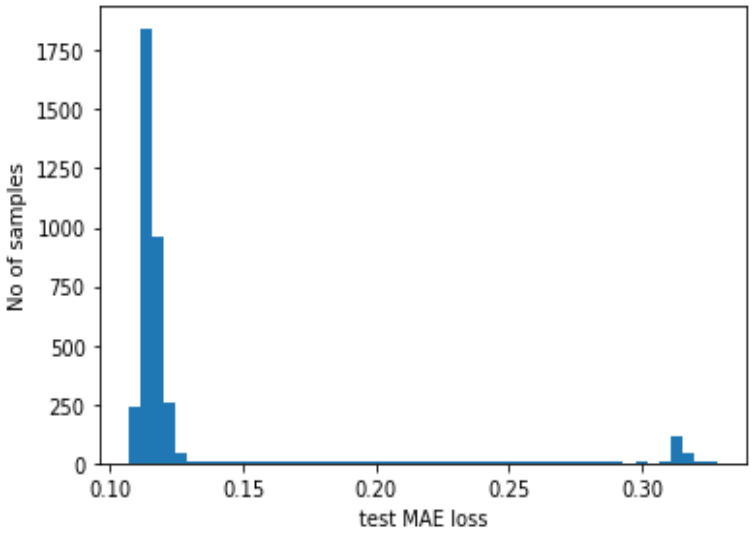



Figure 4.5: *MAE loss of the test data, visualised using the dataset from [35]*

The same procedure is done with the test data, and the resultant MAE loss plot is as illustrated in the Figure 4.5. By using this MAE loss of the testing data, we can determine the anomalies by comparing with the threshold.

$$anomalies = test\_mae\_loss > threshold. \tag{4.2}$$

We can infer the the data values are anomalous if the value of the Mae_loss is much higher than the obtained threshold value. By getting to know the indices of the samples of the data which are anomalous in nature, we can plot them with accordance with the respective timestamps and overlay on the original plot of the test data as illustrated in the Figure 4.6.

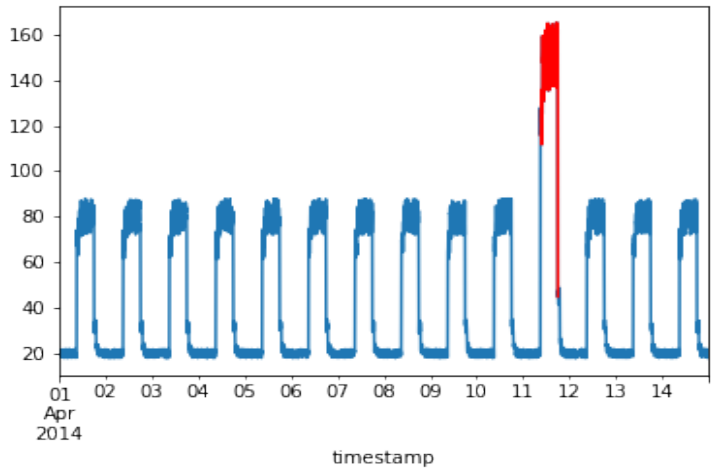



Figure 4.6: *Anomaly overlay on the original plot of test data, visualised using the dataset from [35]*

## 4.2 IMS NASA Bearing dataset

The description of the NASA Bearing dataset [28] is already discussed in the Section 3.3. This section discusses about the anomaly detection mechanism of the NASA bearing dataset. The following are the steps involved in anomaly detection:

- ($\hat{y}$) are generated as model predictions.
- The calculation of reconstruction error is done by finding the difference between the actual value ($y$) and predicted value ($\hat{y}$) i.e., $|y - \hat{y}|$.
- It is necessary to select a threshold value.
- During training, the distribution of loss functions is observed to determine a static threshold value.
- The next step in locating anomalies is to select a threshold. The loss distribution is exhibited in the Figure 4.8. From the figure, we can deduce that the threshold value to be .
- In this procedure, data points originating from the time series with prediction error exceeding the threshold are classified as anomalies, whereas data points with prediction error below the threshold are classified as normal.

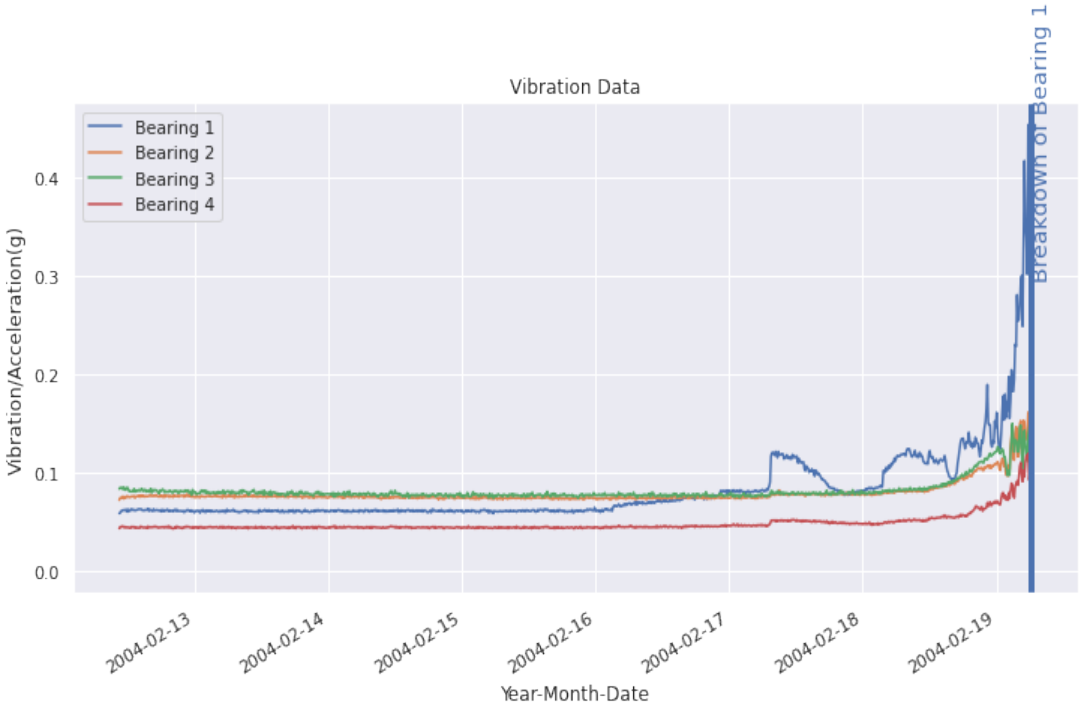



Figure 4.7: *Anomaly overlay on the original plot of test data, visualised using the dataset from [35]*

The visualisation of the bearing data into a single plot is as exhibited in the Figure 4.7

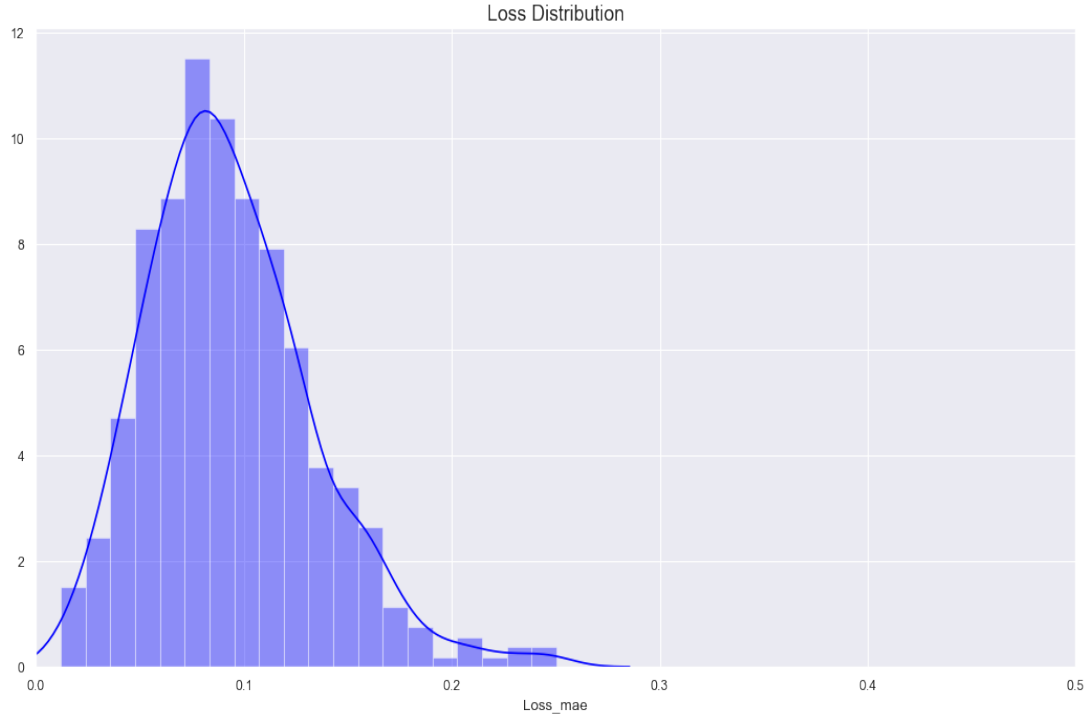



Figure 4.8: *Distribution of loss to determine threshold value, visualised using the dataset from [28]*

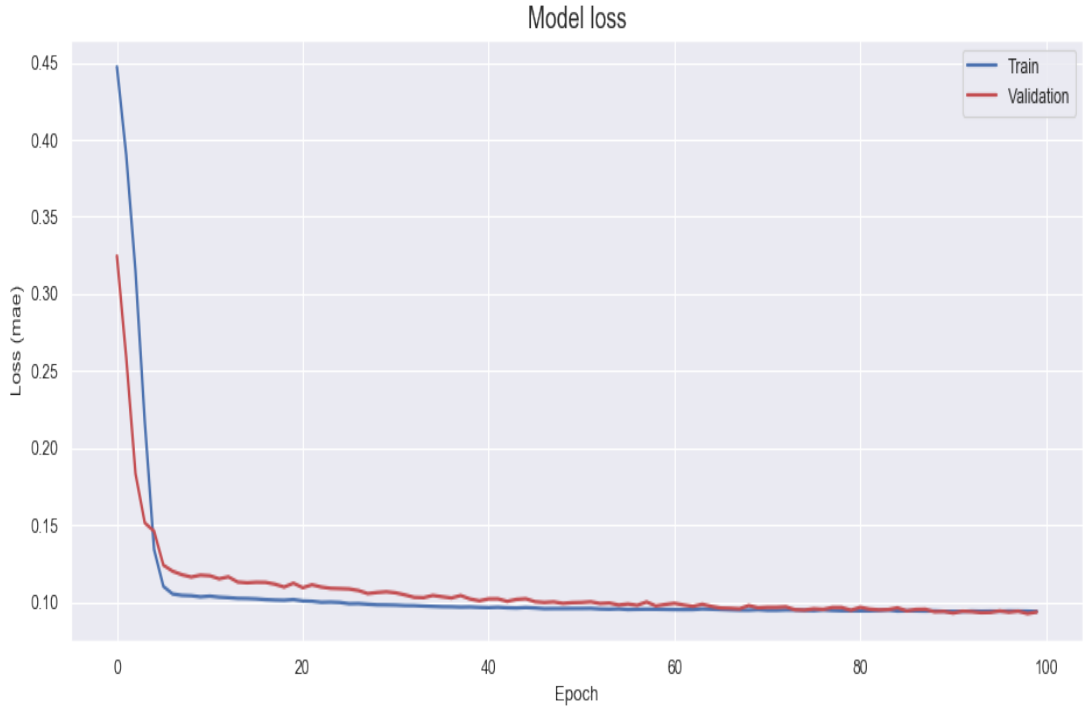



Figure 4.9: *Train Vs Validation loss for the Set 2, visualised using the dataset from [28]*

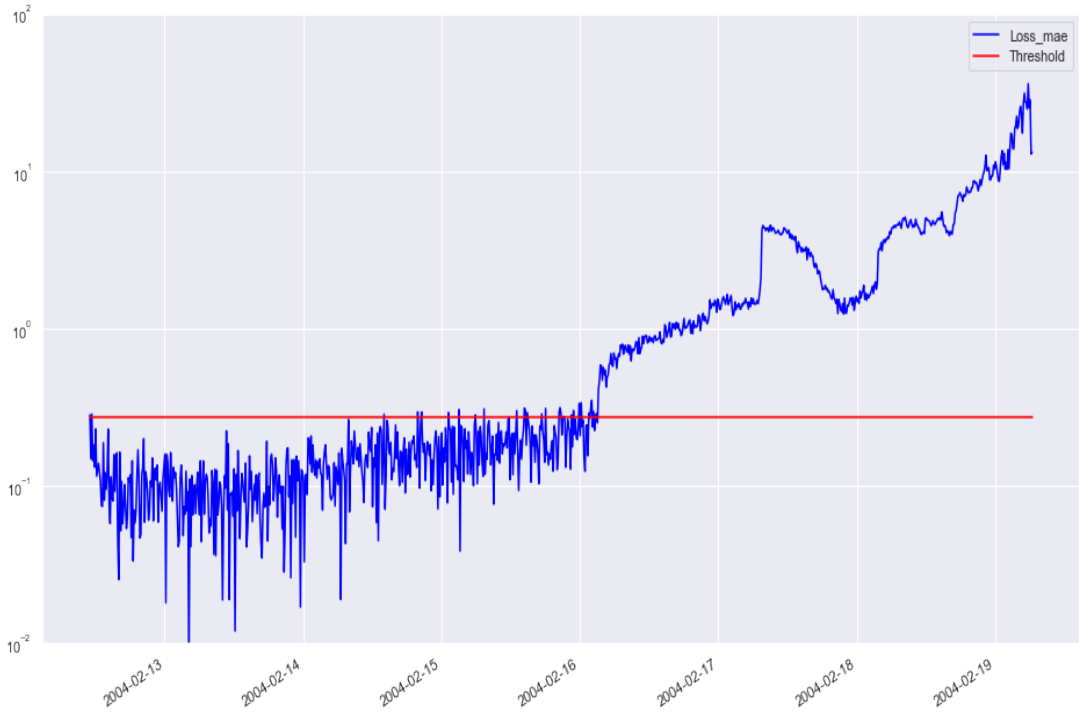



Figure 4.10: *MAE Loss Vs Threshold for the Set 2, visualised using the dataset from [28]*

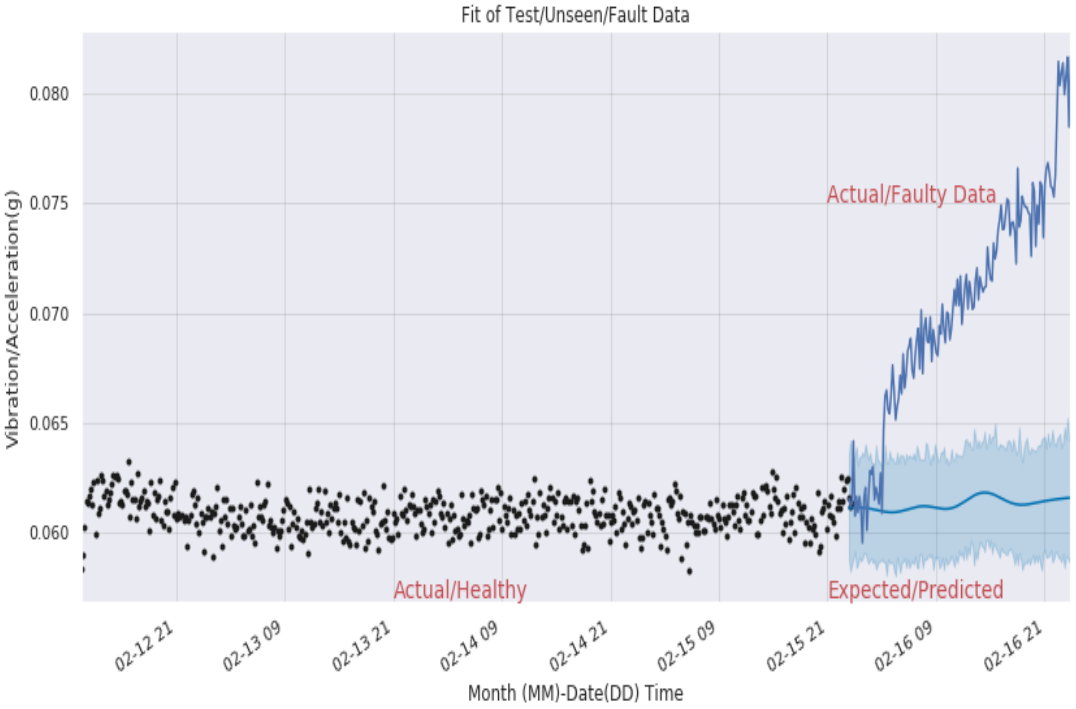



Figure 4.11: *Visualisation of MAE Loss Vs Threshold for the Set 2using Prophet model], visualised using the dataset from [28]*

The model summary of the set 2 is as illustrated, using this model the Fig 4.10 and Fig 4.8 were plotted.

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 1, 4)]            0
_________________________________________________________________
lstm_4 (LSTM)                (None, 1, 16)             1344
_________________________________________________________________
lstm_5 (LSTM)                (None, 4)                 336
_________________________________________________________________
repeat_vector_1 (RepeatVecto (None, 1, 4)              0
_________________________________________________________________
lstm_6 (LSTM)                (None, 1, 4)              144
_________________________________________________________________
lstm_7 (LSTM)                (None, 1, 16)             1344
_________________________________________________________________
time_distributed_1 (TimeDist (None, 1, 4)              68
=================================================================
Total params: 3,236
Trainable params: 3,236
Non-trainable params: 0
```

The dataset description of the above plot is discussed in the Section 3.3. There are three dataset which are called as Set 1, Set 2 and Set 3 respectively in the experimentation. As all the readings were in ASCII format, we had to preprocess the data into .csv format for more ease of handling the data. The vibration data for each 10 minute segment is aggregated based on the average absolute value of the vibration recordings across the 20,480 data points in each file. A single dataframe is created by merging everything together. The train and test dataset is defined before the model setup. For this setting up of the model, the simple split is done to the main dataset, where the normal operating conditions part is considered as the first part of the dataset which is used for the training purposes, the remaining parts which consists of the bearing failure is considered as the test part.

The above experimentation was the early stage of the implementation on the set 2. The same procedure was applied for the other two sets for detecting the anomalies. The model used and their results will be dealt in the next part. The training time of the data was more than 32 hours for each dataset with a batch size of 64 and learning rate 0.001 and 100 epochs. The training was done for different versions based on different running epochs.

The model below is for the Set 2 bearing with 984 files. The data input format is of 3 inputs (data samples,

no of timesteps, no of features). Here we use data consisting of the arbitrary input and the sequence lenght output. There are two parts in the LSTMs namely Encoder and the Decoder.

The multiple LSTM layers consists of two components. The data input sequence is read and encoded into a vector of some length which is already fixed is done using the encoder part. This vector will be represented as a context vector, and it will be decoded into the output sequence by prediction. The reconstruction of the input sequence is done by encoder using the encoded sequence.

```
Number of files: 984
 S2_bearing-1 984
 S2_bearing-2 984
 S2_bearing-3 984
 S2_bearing-4 984
 (688, 4096) (296, 4096)
 (688, 4096) (296, 4096)
 (667, 20, 4096) (275, 20, 4096)
 Model: "model_version_1_set_2"
 _________________________________________________________________
 Layer (type)                 Output Shape              Param #
 =================================================================
 lstm (LSTM)                  (None, 20, 600)           11272800
 _________________________________________________________________
 lstm_1 (LSTM)                (None, 250)               851000
 _________________________________________________________________
 repeat_vector (RepeatVector) (None, 20, 250)           0
 _________________________________________________________________
 lstm_2 (LSTM)                (None, 20, 250)           501000
 _________________________________________________________________
 lstm_3 (LSTM)                (None, 20, 600)           2042400
 _________________________________________________________________
 time_distributed (TimeDistri (None, 20, 4096)          2461696
 =================================================================
 Total params: 17,128,896
 Trainable params: 17,128,896
 Non-trainable params: 0
```

The below model summaries show the trained time steps which is developed using the functional keras API. The input is fed in the shape of (None, 20, 600). There are two LSTM layers with 20 units and 250 units respectively. These two LSTM layers will be the encoder part which will take the input sequence and encode it into an internal representation. Usually, this is called as the feature vector. In Keras, it is depicted as the Repeat vector. The repeat vector will contain the encoded sequence of shape (None,

20, 250) . After the repeat vector, there are another two layers which are of 250 units and 600 units respectively. The choice of these is determined by how many units are present in the dataset. The set 2 contains 984 files, so train and test will be 667 units and 275 units respectively. The two layers after the repeat vector will be the decoder part. From the repeat vector the input is used by the decoder to learn and decode the predicted output. This model will be for a single time step, the same will be retained for the time steps defined in the model while training. The same model is run under different input parameter for checking the feasibility of which gives a better accuracy in detection of the anomalies.

```
Number of files: 984
S2_bearing-1 984
S2_bearing-2 984
S2_bearing-3 984
S2_bearing-4 984
(688, 10240) (296, 10240)
(688, 10240) (296, 10240)
(667, 20, 10240) (275, 20, 10240)
Model: "model_version2_set_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 20, 600)           26018400
_________________________________________________________________
lstm_1 (LSTM)                (None, 250)               851000
_________________________________________________________________
repeat_vector (RepeatVector) (None, 20, 250)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 20, 250)           501000
_________________________________________________________________
lstm_3 (LSTM)                (None, 20, 600)           2042400
_________________________________________________________________
time_distributed (TimeDistri (None, 20, 10240)         6154240
=================================================================
Total params: 35,567,040
Trainable params: 35,567,040
Non-trainable params: 0
```

The model below is for the set 3 which has 6324 files, so the training and test is split into 4428 units and 1896 units respectively. The total number of parameters trained will be of 351,236,096 units.

```
Number of files: 6324
```

```
S3_bearing-1 6324
S3_bearing-2 6324
S3_bearing-3 6324
S3_bearing-4 6324
(4428, 4096) (1896, 4096)
(4428, 4096) (1896, 4096)
(4407, 20, 4096) (1875, 20, 4096)
Model: "model_version1_set3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 20, 4400)          149547200
_________________________________________________________________
lstm_1 (LSTM)                (None, 1850)              46257400
_________________________________________________________________
repeat_vector (RepeatVector) (None, 20, 1850)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 20, 1850)          27387400
_________________________________________________________________
lstm_3 (LSTM)                (None, 20, 4400)          110017600
_________________________________________________________________
time_distributed (TimeDistri (None, 20, 4096)          18026496
=================================================================
Total params: 351,236,096
Trainable params: 351,236,096
Non-trainable params: 0
_________________________
```

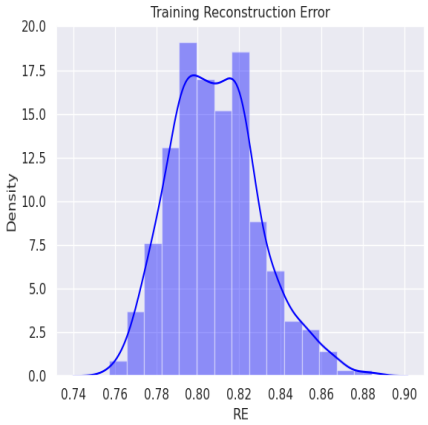
Using the above models on the dataset [28], the training reconstruction error, training loss , training Vs the validation loss and the threshold to detect the anomalies are plotted in the graphs.

The below Figure 4.12, Figure 4.13 and Figure 4.14 are plotted using the Dataset 2 which contained 984 files.

Similarly for the same procedure was followed for the set 1 and set 3 based on the models defined in the above section. From the above graphs we can infer that the bearing has a fault or anomaly at the end of the experiment. Based on the threshold value, we can get to know at point in time the bearing has started to fail. The Remaining life of the bearing after the failure has occurred is discussed in the next section.

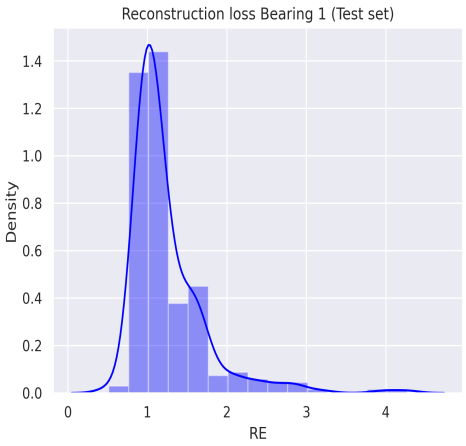
Visualisation of healthy and faulty state using the Prophet model[53]


(a) Reconstruction error of Set 2 - version 1

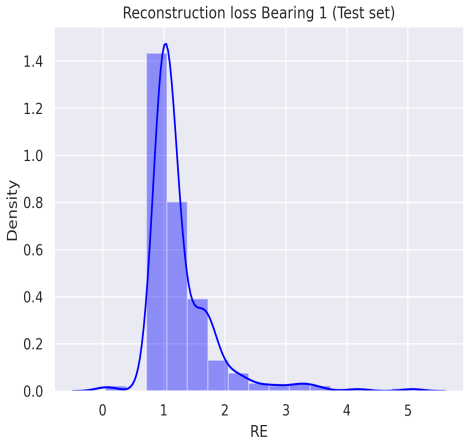



(b) Reconstruction error of Set 2 - version 2

Figure 4.12: Reconstruction error plot of the training of set 2 dataset, visualised using the dataset from [28]

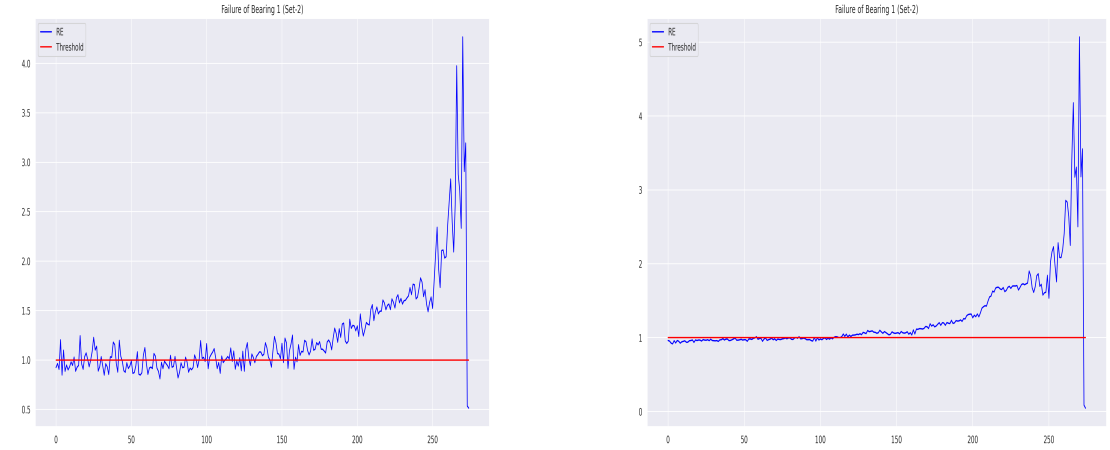



(a) Reconstruction loss of the test dataset of Set 2 - version 1




(b) Reconstruction loss of the test of Set 2 - version 2

Figure 4.13: Set 2 Dataset's reconstruction loss plot of the test datatet, visualised using the dataset from [28]




(a) MAE loss Vs Threshold of Set 2 - version 1

(b) MAE loss Vs Threshold of Set 2 - version 2

Figure 4.14: MAE loss Vs Threshold for the set 2 dataset, visualised using the dataset from [28]

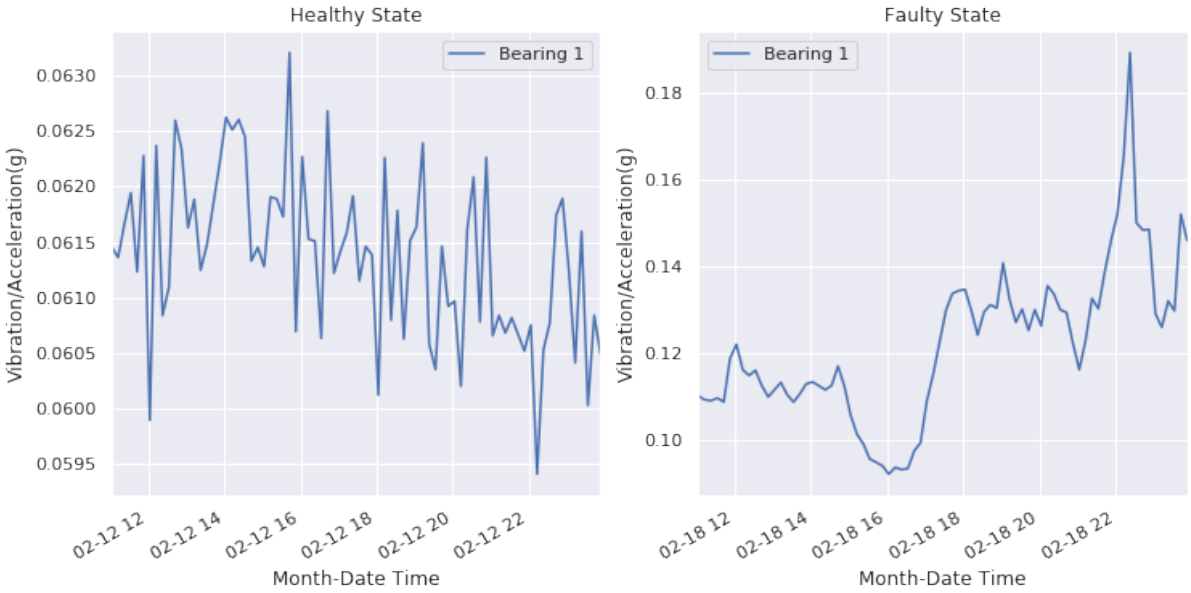



Figure 4.15: *Healthy state Vs Faulty state of bearing 1 for the Set 2, visualised using the dataset from [28]*

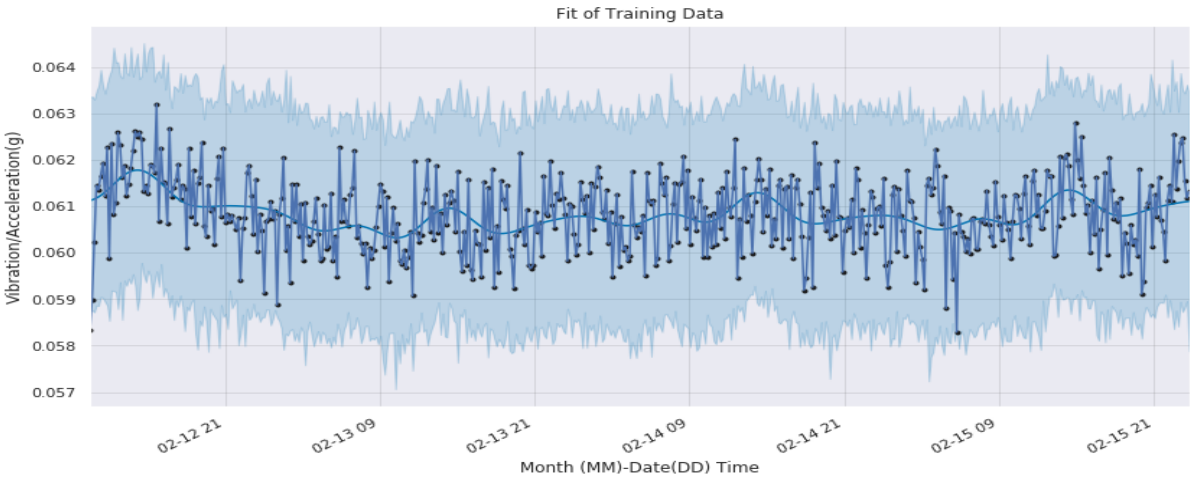



Figure 4.16: *Visualisation of Bearing frequency using Prophet model [53], visualised using the dataset from [28]*

## 4.3 Remaining User Life Estimation

To find the Remaining life of a failure diagnostic experiment, we need to find out the relevant features which are associated with the data.Each file will have a total of 20480 points. Each data point has a sampling rate of 20kHz, Each of the four bearings visualisations of set 2 are as shown in the Figure 4.17, Figure 4.18, Figure 4.19, and Figure 4.20,.

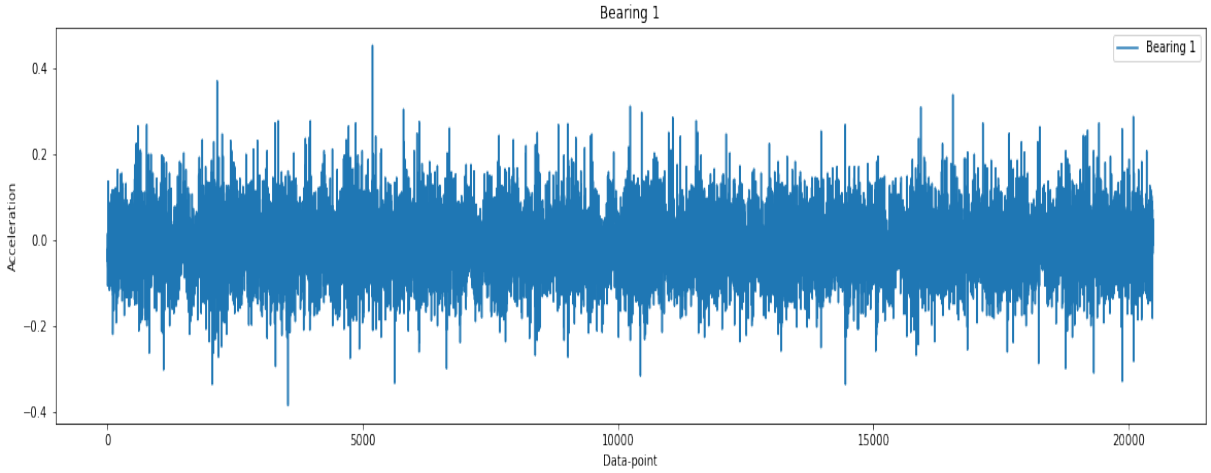



Figure 4.17: *Visualisation of Bearing 1, visualised using the dataset from [28]*

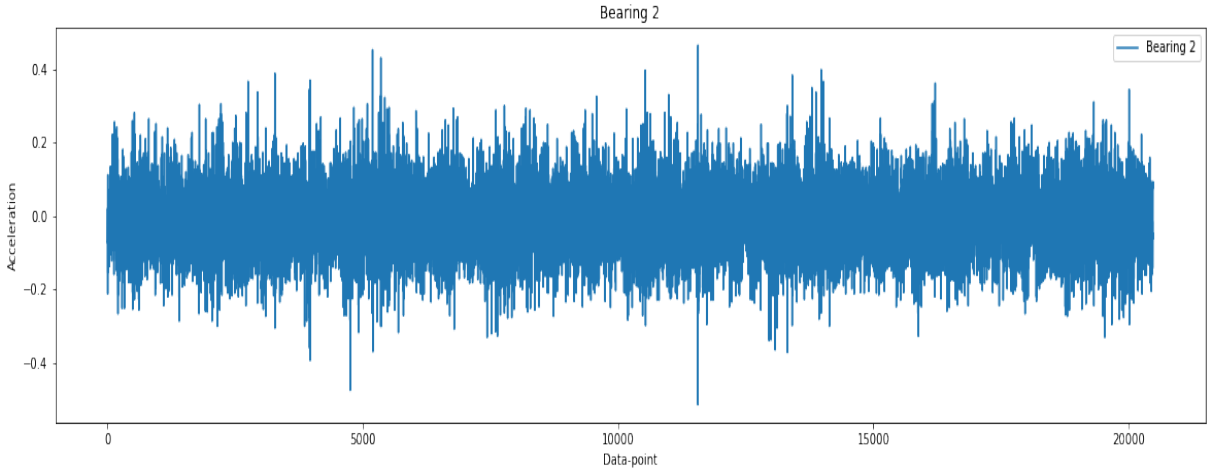



Figure 4.18: *Visualisation of Bearing 2, visualised using the dataset from [28]*

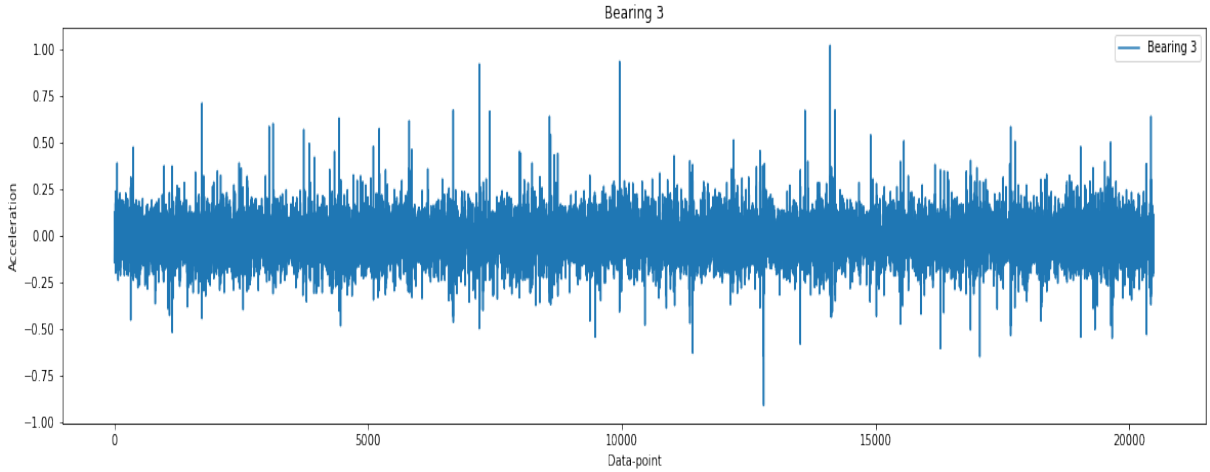



Figure 4.19: *Visualisation of Bearing 3, visualised using the dataset from [28]*

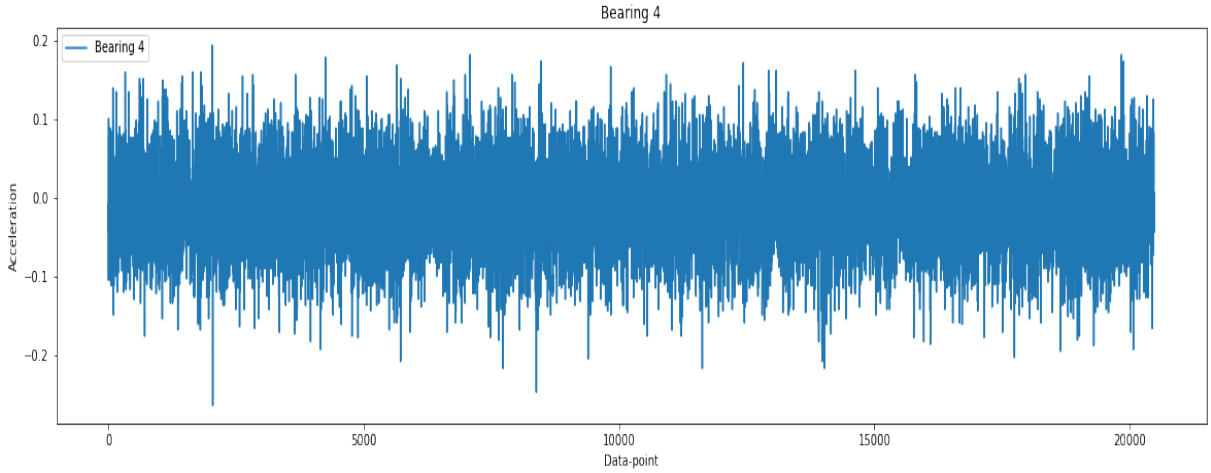



Figure 4.20: *Visualisation of Bearing 4, visualised using the dataset from [28]*

The calculation of each of the time domain features

- Maximum value
- Minimum value
- Mean value
- Standard deviation (Unbiased std)
- Root mean square value (RMS)
- Skewness
- Kurtosis
- Crest factor = $\frac{Max}{RMS}$
- Form factor = $\frac{RMS}{Mean}$

Based on the above time domain features the faults in the sets can be classified. As in the datasheet provided by the NASA[28] has the faults occured for the different bearings.

- The bearing 3 has inner race defect and bearing 4 has roller element defect in set 1.
- The bearing 1 has outer race defect in set 2.
- The bearing 3 has outer race defect in set 3.

These values are populated into an Time feature matrix as shown in the Table 4.1

To calculate mean and standard deviation the formula is [55]:

$$\mu(D(S_i)) = \frac{\sum_{w=1}^{\Omega} D(S_{iw})}{\Omega} \tag{4.3}$$

$$\sigma(D(S_i)) = \frac{\sum_{w=1}^{\Omega} [D(S_{iw}) - \mu(D(S_i))]^2}{\Omega} \tag{4.4}$$

where,

- $D(.)$ is visit duration
- $i$ is the state index
- $w$ is the visit index
- $\Omega$ is the total of visits

| | Max | Min | Mean | Std | RMS | Skewness | Kurtosis | Crest Factor | Form Factor | Fault |
|---|---|---|---|---|---|---|---|---|---|---|
| **2004-02-12 10:32:00** | 0.454 | -0.386 | -0.010196 | 0.073477 | 0.074179 | 0.083987 | 0.628408 | 6.120331 | -7.275307 | Normal |
| **2004-02-12 10:42:00** | 0.369 | -0.388 | -0.002585 | 0.075340 | 0.075382 | 0.052138 | 0.647935 | 4.895038 | -29.161382 | Normal |
| **2004-02-12 10:52:00** | 0.503 | -0.400 | -0.002484 | 0.076191 | 0.076230 | 0.032806 | 0.513132 | 6.598472 | -30.688508 | Normal |
| **2004-02-12 11:02:00** | 0.608 | -0.576 | -0.002277 | 0.078693 | 0.078724 | 0.041483 | 1.157547 | 7.723217 | -34.579322 | Normal |
| **2004-02-12 11:12:00** | 0.391 | -0.391 | -0.002404 | 0.078439 | 0.078474 | 0.028222 | 0.602825 | 4.982524 | -32.645152 | Normal |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2004-02-19 05:42:00** | 4.998 | -5.000 | -0.002752 | 0.725014 | 0.725001 | -0.510519 | 12.576183 | 6.893779 | -263.426419 | Outer Race |
| **2004-02-19 05:52:00** | 2.688 | -2.881 | -0.004487 | 0.462001 | 0.462012 | -0.325344 | 3.759312 | 5.818035 | -102.977656 | Outer Race |
| **2004-02-19 06:02:00** | 3.501 | -3.696 | -0.001703 | 0.483844 | 0.483835 | -0.377068 | 4.890985 | 7.235932 | -284.094983 | Outer Race |
| **2004-02-19 06:12:00** | 0.005 | 0.000 | 0.001857 | 0.000987 | 0.002103 | 0.579656 | 3.636864 | 2.377543 | 1.132578 | Normal |
| **2004-02-19 06:22:00** | 0.002 | -0.005 | -0.001162 | 0.001000 | 0.001533 | 0.317009 | -1.609910 | 1.304860 | -1.319089 | Normal |

Table 4.1: Time feature matrix with faults for the dataset 2, calculated using the dataset from [28]

To calculate RUL with $\eta$ is confidence value [55]:

$$RUL_{upper} = \sum_{i=currentstate}^{N} [\mu(D(S_i)) + \eta.\sigma(D(S_i))] \tag{4.5}$$

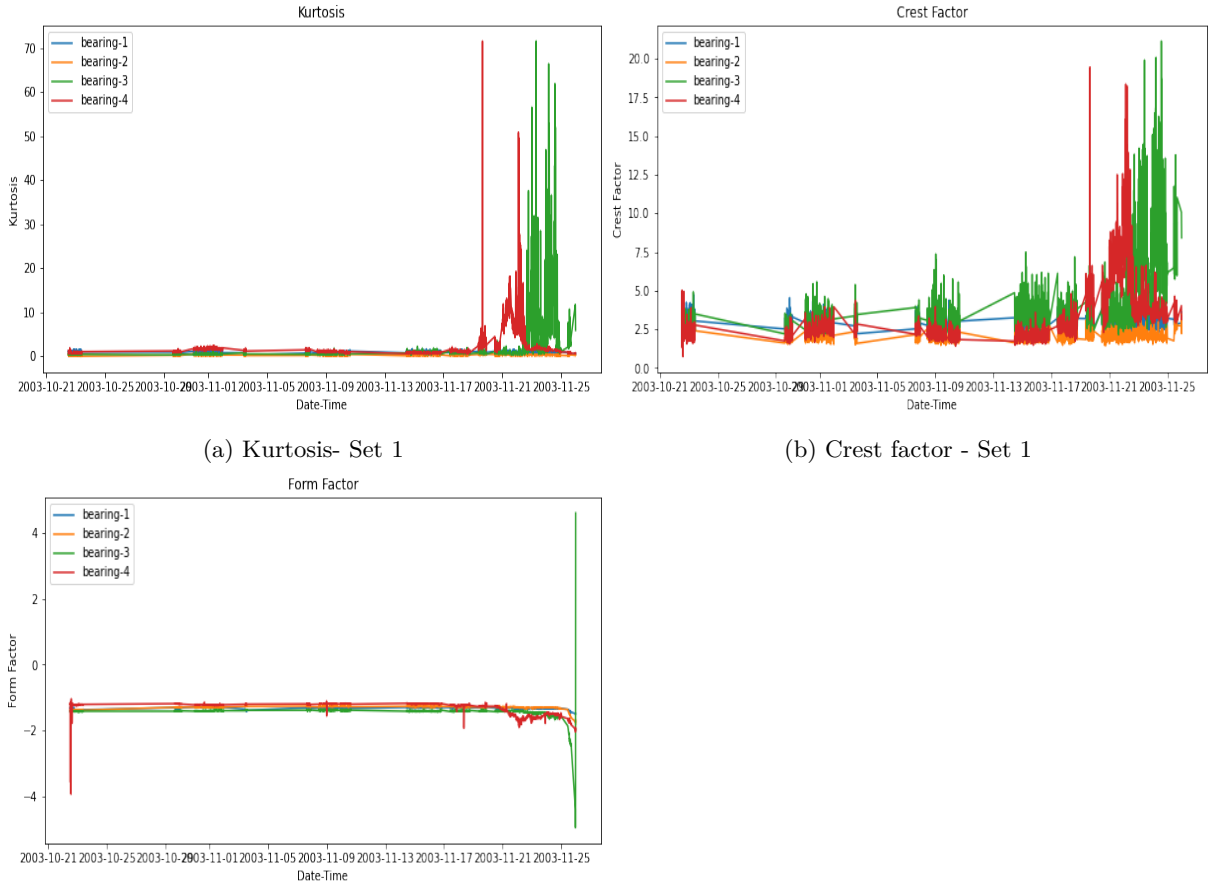
$$RUL_{mean} = \sum_{i=currentstate}^{N} \mu(D(S_i)) \tag{4.6}$$

$$RUL_{lower} = \sum_{i=currentstate}^{N} [\mu(D(S_i)) - \eta.\sigma(D(S_i))] \tag{4.7}$$

In the visualisation of the bearing frequency using the prophet model [53] as depicted in the Figure 4.16, The true data points are portrayed as the black points of the vibrations. Whereas the trend line is presented as the blue line and the portion colored in light blue will be the acceptable variance. If the signals are above the portions, there are usually anomolous in nature. The black points are the true data points of the vibration sensor. The blue line represents the fitted line (trend line) with the light blue portion showing the acceptable variance.

### 4.3.1 Time features of set 1 dataset.

The time features of the set 1 dataset of the IMS NASA Bearing dataset is visualised as shown in Figure 4.21 and Figure 4.22. By these visualisations we can infer that the bearing 3 has inner race defect and bearing 4 has roller element defect in set 1.

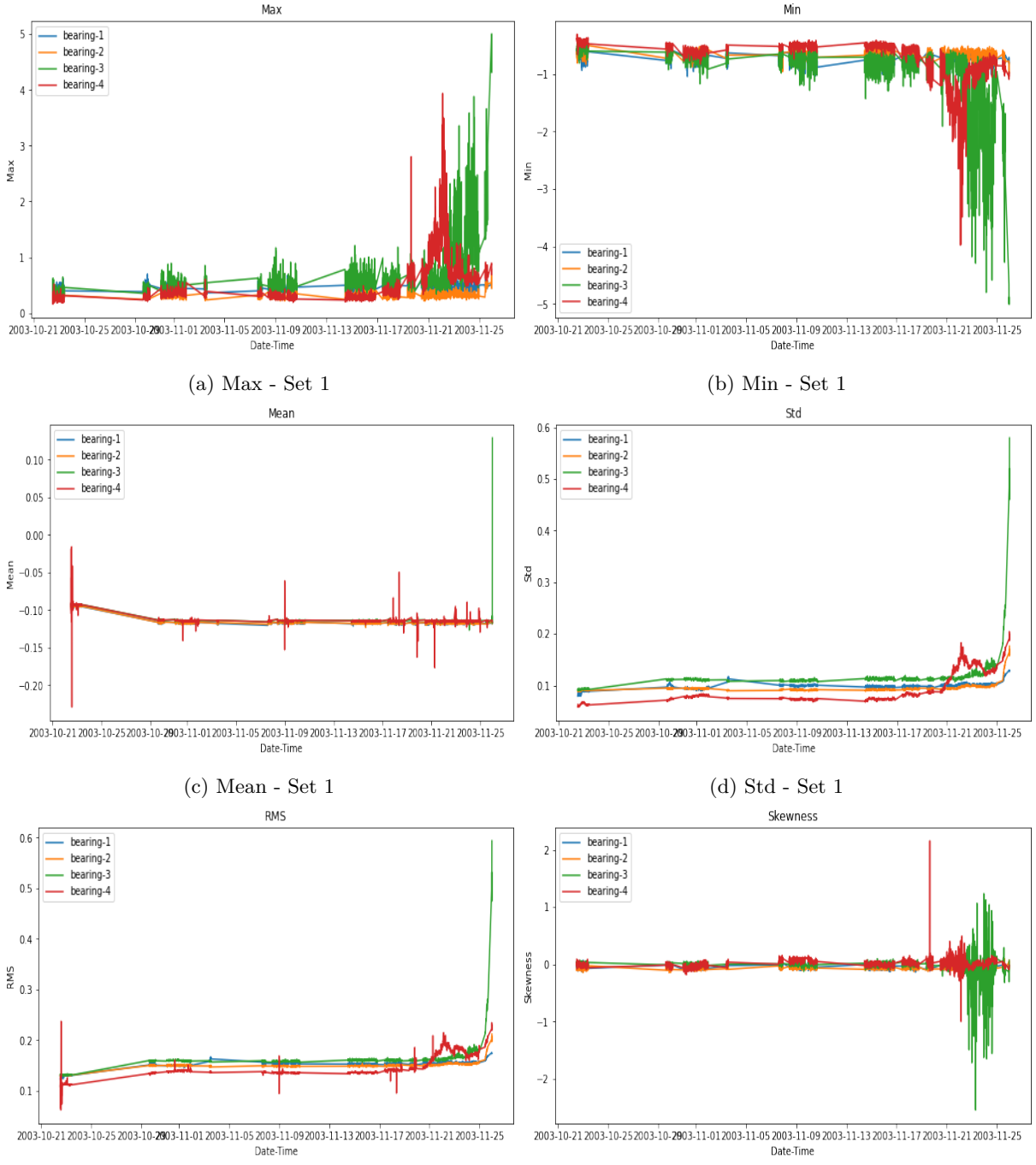



(a) Kurtosis- Set 1

(b) Crest factor - Set 1

(c) Form factor - Set 1

Figure 4.21: First set of data visualisations of time feature matrix in Set 1 of IMS NASA Bearing dataset, visualised using the dataset from [28]




(a) Max - Set 1

(b) Min - Set 1

(c) Mean - Set 1

(d) Std - Set 1

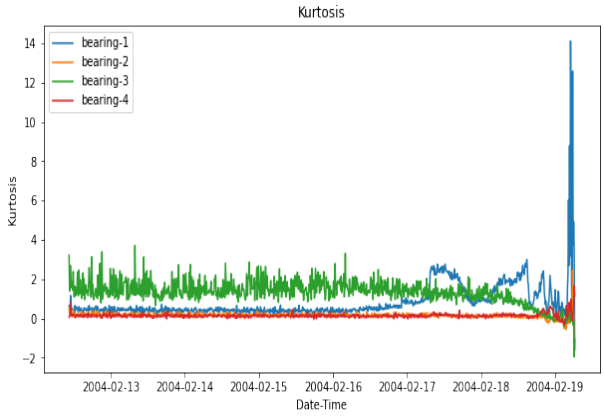
(e) RMS - Set 1

(f) Skewness - Set 1

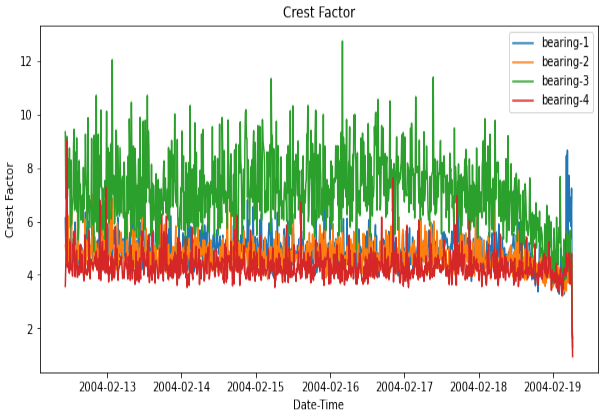
Figure 4.22: Second set of data visualisations of time feature matrix in Set 1 of IMS NASA Bearing dataset, visualised using the dataset from [28]

### 4.3.2 Time features of set 2 dataset.

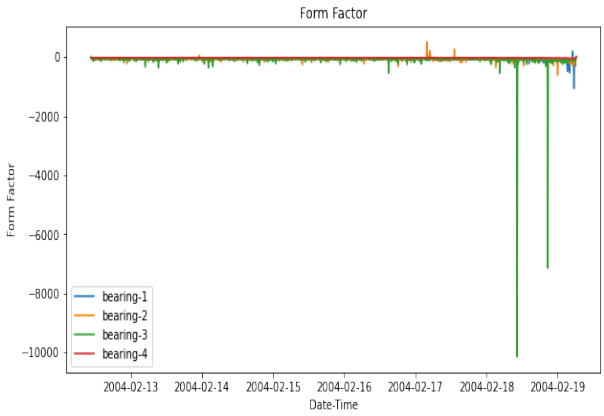
The time features of the set 2 dataset of the IMS NASA Bearing dataset is visualised as shown in Figure 4.23 and Figure 4.24. By these visualisations we can infer that the bearing 1 has outer race defect in set 2.




(a) Kurtosis- Set 2

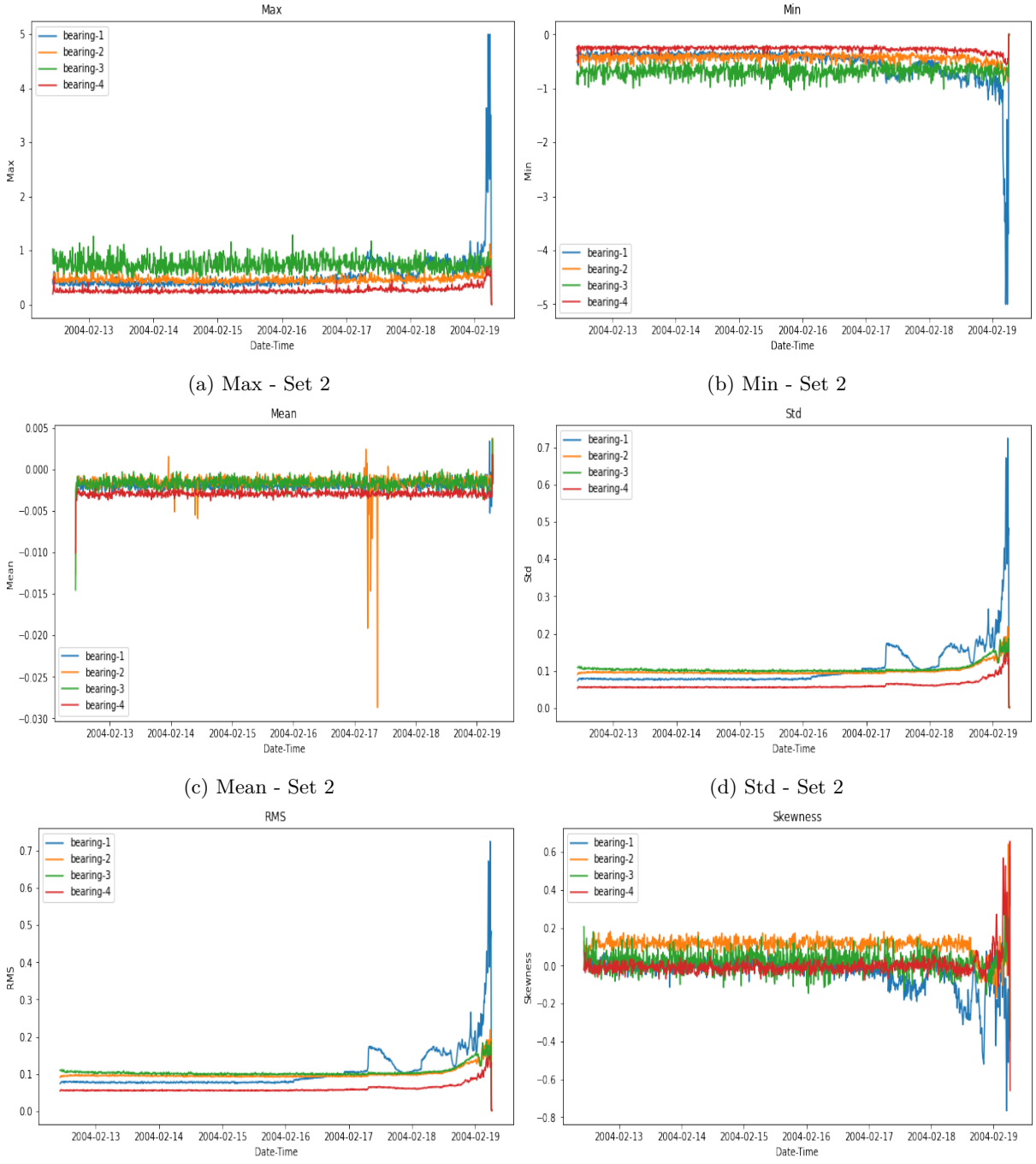



(b) Crest factor - Set 2




(c) Form factor - Set 2

Figure 4.23: First set of data visualisations of time feature matrix in Set 2 of IMS NASA Bearing dataset, visualised using the dataset from [28]




(a) Max - Set 2

(b) Min - Set 2

(c) Mean - Set 2

(d) Std - Set 2

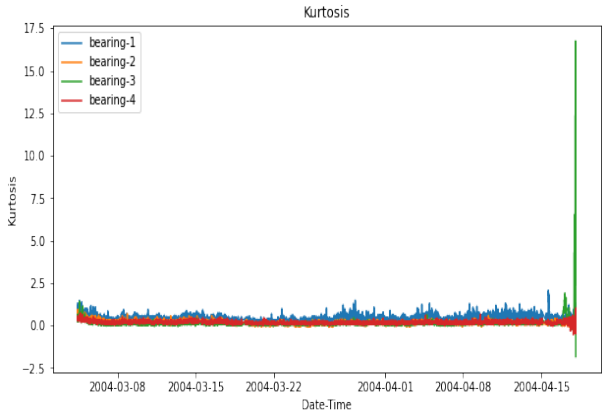
(e) RMS - Set 2

(f) Skewness - Set 2

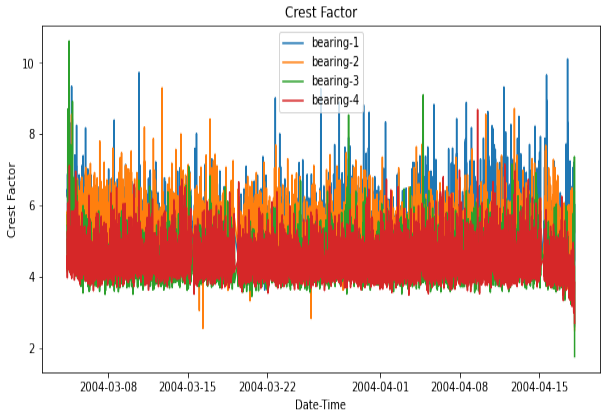
Figure 4.24: Second set of data visualisations of time feature matrix in Set 2 of IMS NASA Bearing dataset, visualised using the dataset from [28]

### 4.3.3 Time features of set 3 dataset.

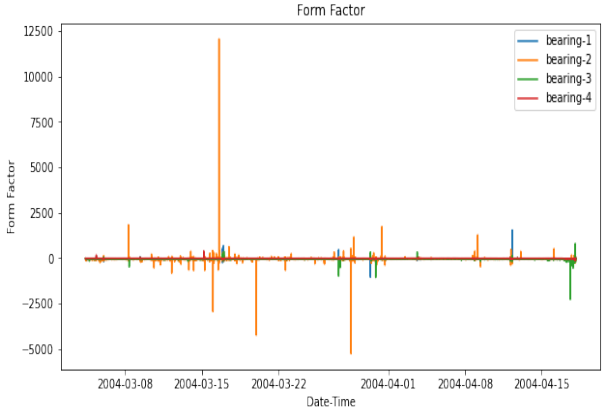
The time features of the set 3 dataset of the IMS NASA Bearing dataset is visualised as shown in Figure 4.25 and Figure 4.26. By these visualisations we can infer that the bearing 3 has outer race defect in set 2.




(a) Kurtosis- Set 3

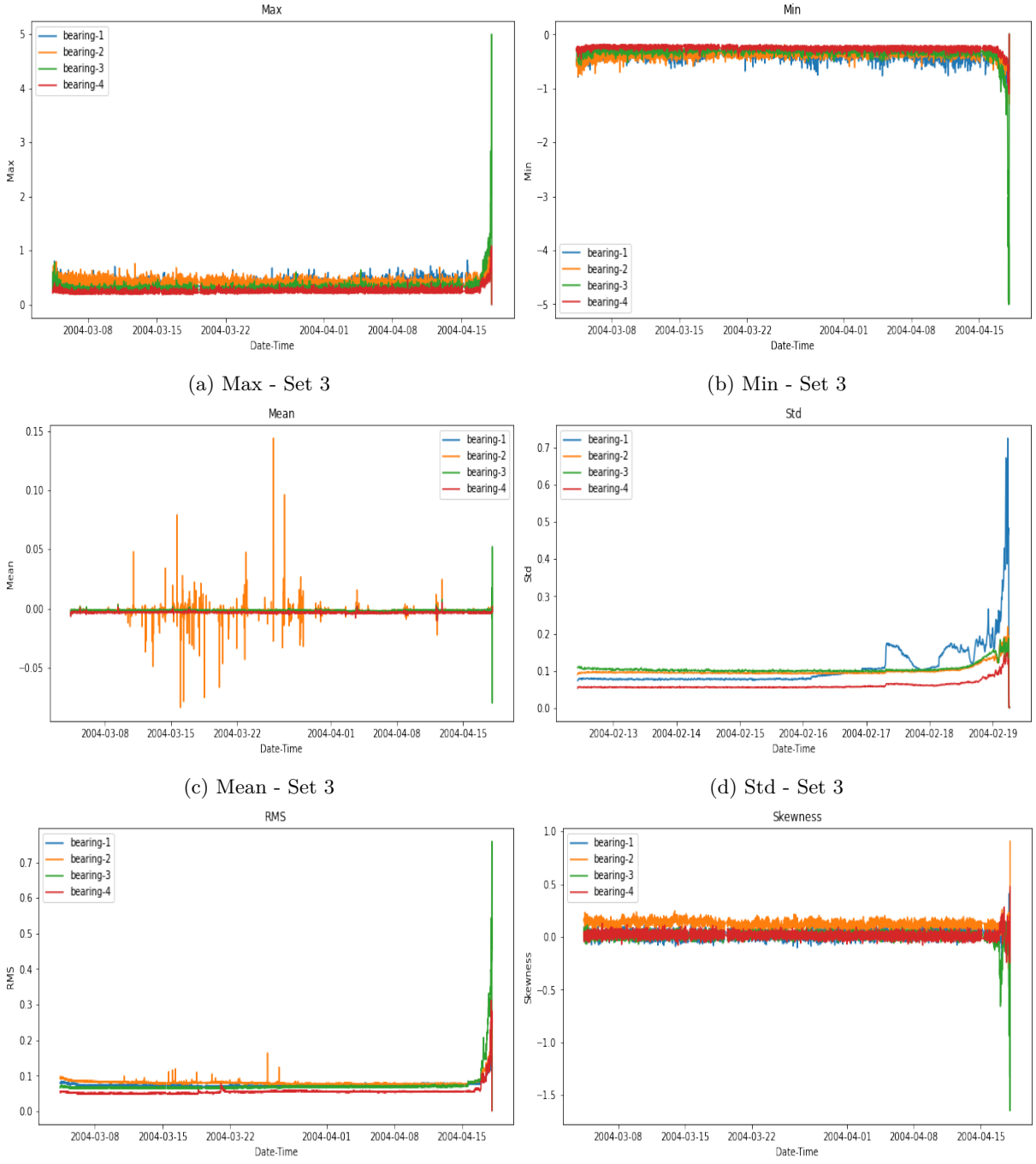



(b) Crest factor - Set 3




(c) Form factor - Set 3

Figure 4.25: First set of data visualisations of time feature matrix in Set 3 of IMS NASA Bearing dataset, visualised using the dataset from [28]




(a) Max - Set 3

(b) Min - Set 3

(c) Mean - Set 3

(d) Std - Set 3

(e) RMS - Set 3

(f) Skewness - Set 3

Figure 4.26: Second set of data visualisations of time feature matrix in Set 3 of IMS NASA Bearing dataset, visualised using the dataset from [28]

### 4.3.4 Classfication of faults

The confusion matrix is a measure to check the performance of a model. With the help of all the fault data, the classification of faults is done using the confusion matrix. From the Figure 4.27 we can see that all the faults have adequately classified.

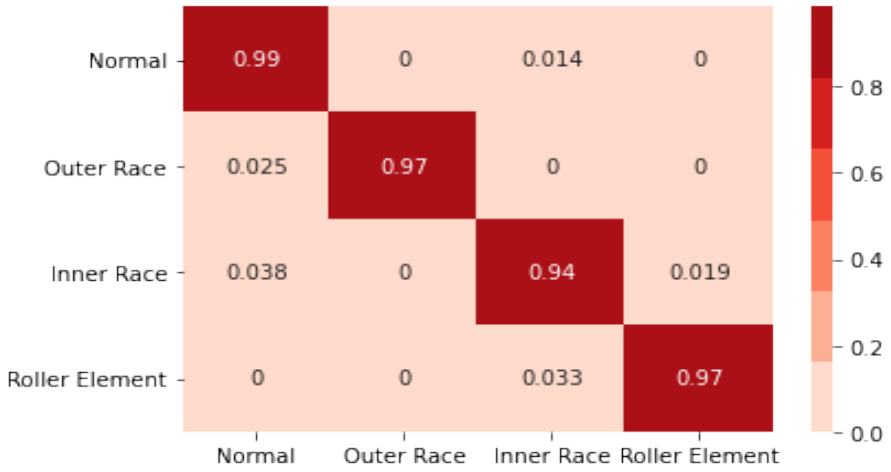



Figure 4.27: *Classification of faults for the IMS NASA Bearing dataset, visualised using the dataset from [28]*

### 4.3.5 Random Forest model

We can infer the faults which has occured in the IMS NASA Bearing dataset by the Figure 4.28. As random forest is a classification algorithm which uses the randomness of the features and bag them accordingly. These decision trees will build individual trees to create a non correlated cluster of trees. These cluster of trees i.e., the forest will have an accurate prediction than individual tree.

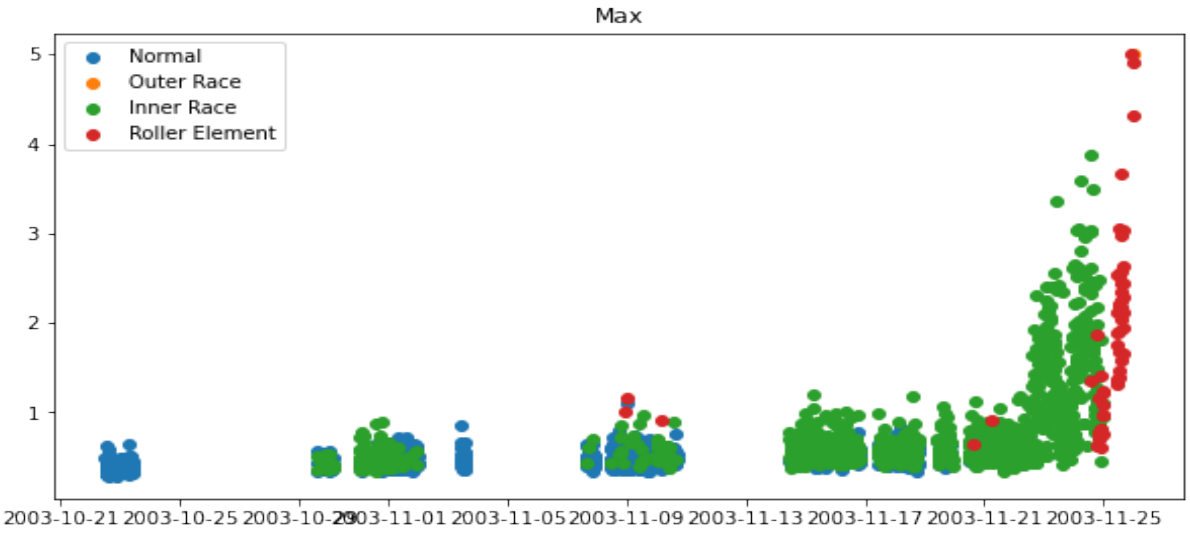



Figure 4.28: *Testing with the Random forest model on the IMS NASA Bearing dataset, visualised using the dataset from [28]*

# 5

# Conclusions

One can reduce operating expenses and identify early stage damages by employing predictive maintenance strategies. Regular maintenance routines can be planned to meet the needs of current schedules, instead of shutting down equipment for preventive maintenance for the healthy condition and reliability improvement of the equipment. Models can learn to predict healthy trends with acceptable variance by analyzing past trends. An alarm can be raised if any deviations are observed in the predicted trends in the model which is trained. The goal of predictive maintenance and failure diagnostics is to detect defects, faults and predict failure by using sensors, machine learning, and advanced algorithms. By using autoencoders for the large measurement datasets, it provides code re usability and dimensionality reduction feature.

This section will further provide the contributions, lessons learned and the future research work which can be done in the same field.

## 5.1 Contributions

1. **Failure diagnostics and predictive maintenance:** This research works provides detailed explanation about failure diagnostics features and anomaly detection methodology which can be used. In addition, various anomalous detection techniques are performed on the NASA Bearing dataset to predict the failure or anomalous behavior of the bearings.
2. **Literature review:** This research will provide a detailed review on the anomaly detection mechanisms, failure diagnostics and a good insight on the time series data.
3. **Importance of predictive maintainance and failure diagnostics:** Initially the research was around anomaly detection in the time series data, later based on the literature surveys the concept and the approach was moved on to the field of predictive maintenance and condition monitoring.

## 5.2 Lessons learned

The major portion of the research was to understand and learn new methodologies and concepts. Some of the major learning's are summarised as follows:

- Learning and understanding about the time series data. To understand the pattern, trend, and challenges to process the time series data.

- Finding proper dataset. This is the major hurdle of my research, to find the dataset which matched our requirements and later to tweak it accordingly for the research process.
- Preprocessing of the data. This is the major part of the research and if the data is not handled properly it might lead to unwanted results. At later stage, it will be harder to fix those mistakes.
- Hyper parameter tuning. One must invest some time during their research for the tuning of the parameters, so that the results will be more good.

## 5.3 Future work

The future developments for this research and developement project is listed below:

- The research can be extended with other deep learning and machine learning approaches and comparitive analysis can be performed.
- Hyper parameter optimization. There is always a scope for improvement, by tuning the hyperparameter, an optimal method can be obtained.
- Finding different alternative to choose the threshold value, so as to increase the perfomance.
- Further understanding in the field of autoencoders and its variants.
- Using new performance metrics for evaluation of the models.

# A

# Prognostics and Diagnostics

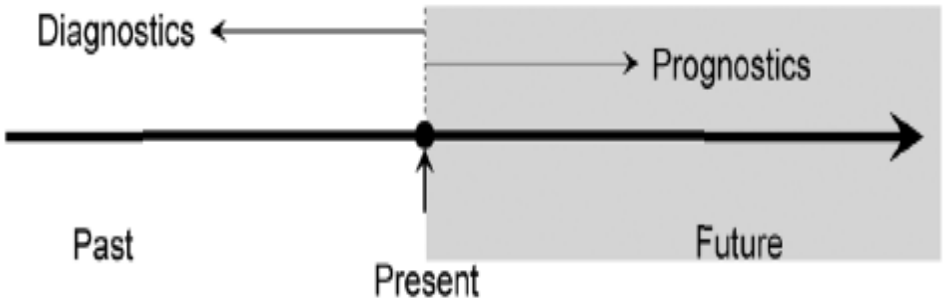



Figure A.1: *Picture of Prognostics Vs Diagnostics , reproduced from [56]*

The Figure A.1 is reproduced from [56] An assessment of the extent to which a product has deviated from its expected state of health under its expected usage conditions is prognostics and the estimation of the remaining useful life (RUL) of the product [36]. So Prognostics will be based on the future outcome what will happen in near future. Based on current conditions, prognostics calculates the remaining useful life (RUL) for an event in the future [59].

The Diagnostics is the one which we have the data of the past to act upon. Fault or failure can be detected and isolated with the help of diagnostics [15].

# B

# List of Abbreviations

ASCII American Standard Code for Information Interchange
AR Auto Regressive
ARIMA Auto Regressive Integrated Moving Average
AWS Amazon Web Services
CNN Convolutional Neural Network
CPU Central Processing Unit
CT Computer Tomography
DLP Data Leakage Protection
ECG Electrocardiography
FFT Fast Fourier Transform
GAN Generative Adversarial Networks
GPU Graphical Processing Unit
ICP Integrated Circuit Piezoelectric
IDS Intrusion Detection System
IMS IBM Information Management System
IoT Internet of Things
LSTM Long Short Term Memory
LSTM-AE LSTM-Autoencoder
MA Moving Average
MAD Mean Absolute Deviation
MAE Mean Absolute Error
ML Machine Learning
NAB Numenta Anomaly Benchmark
NaN Not a Number
NASA The National Aeronautics and Space Administration
PCA Principal Component Analysis
PCB Printed Circuit Board
RAM Random Access Memory

ReLU Rectified Linear Unit
RM Rotation Motor
RF Random Forest
RMSE Root Mean Squared Error
RNN Recurrent Neural Network
SOTA State-Of-The-Art
SVM Support Vector Machine
SVR Support Vector Regression

# C

# Software Prerequisities and Usage

To replicate the research work following software packages are necessary

1. Python3 $\geq$ 3.5.X
2. Numpy $\geq$ 1.19.X
3. Pandas $\geq$ 1.1.5
4. Tensorflow_gpu $\geq$ 2.3.X
5. Keras $\geq$ 2.4.3
6. Matplotlib $\geq$ 3.3.2
7. Scipy $\geq$ 1.5.2
8. Sklearn / Scikit_learn $\geq$ 0.23.2
9. Seaborn $\geq$ 0.11.0
10. Jupyter Notebook $\geq$ 5.4.1
11. Anaconda $\geq$ 3.0.0

# D

# Mind Maps

## D.1 Mind Map of Time Series Analysis

The Figure F.1 shows the mindmap of the Time Series Analysis created using XMind - Trial version application. The mindmap depicts the different approaches to handle the time series data and data mining techniques.

## D.2 MindMap of Anomaly Detection

The Figure F.2 shows the mindmap of the Anomaly detection is created using XMind − Trial version application. The mindmap will entail the different anomaly detection modes, categorization of the different anomalies, different training and test data and along with the applications where the anomaly detection is commonly used.

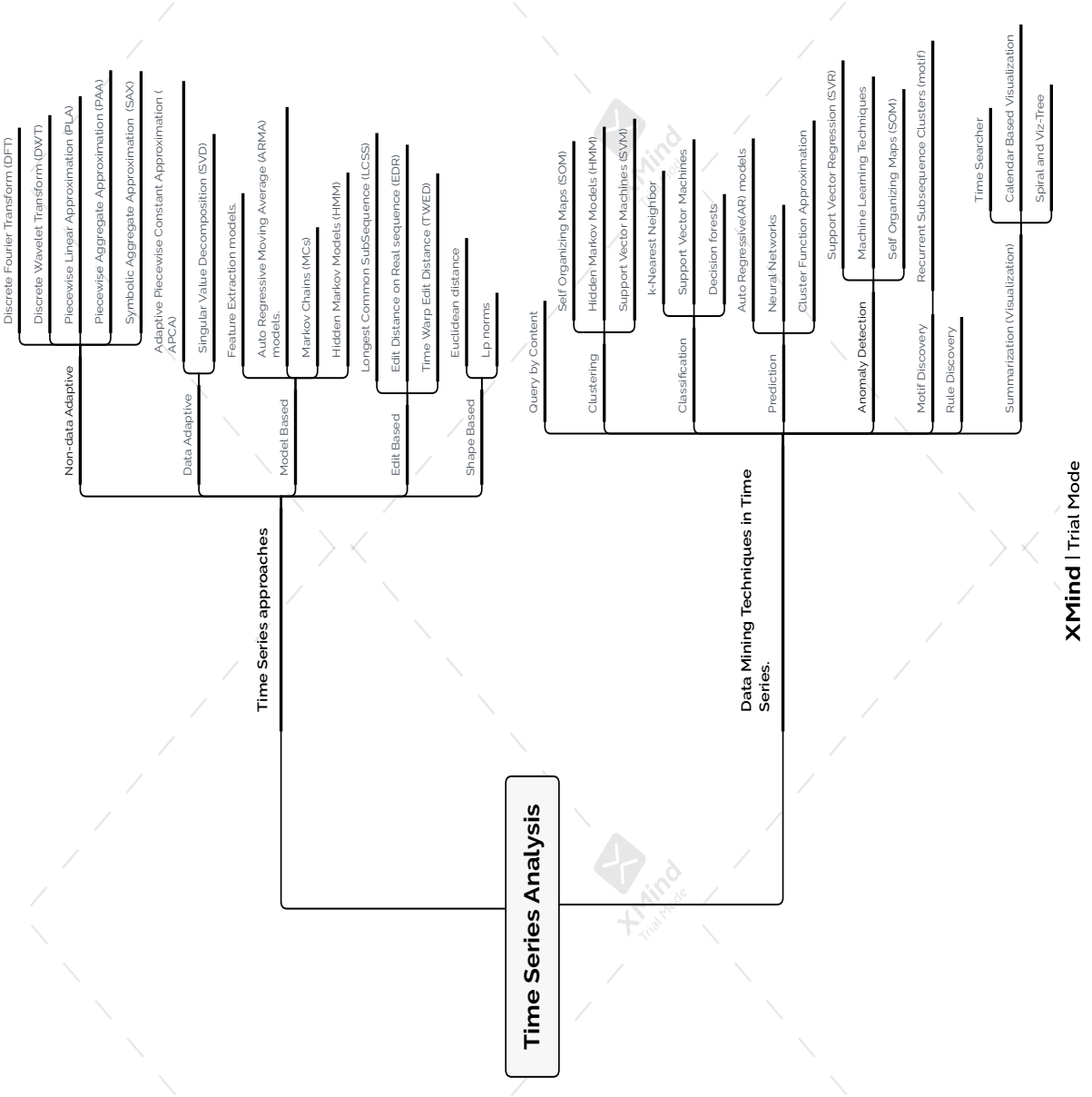



Figure D.1: *Mind Map of Time Series Analysis*

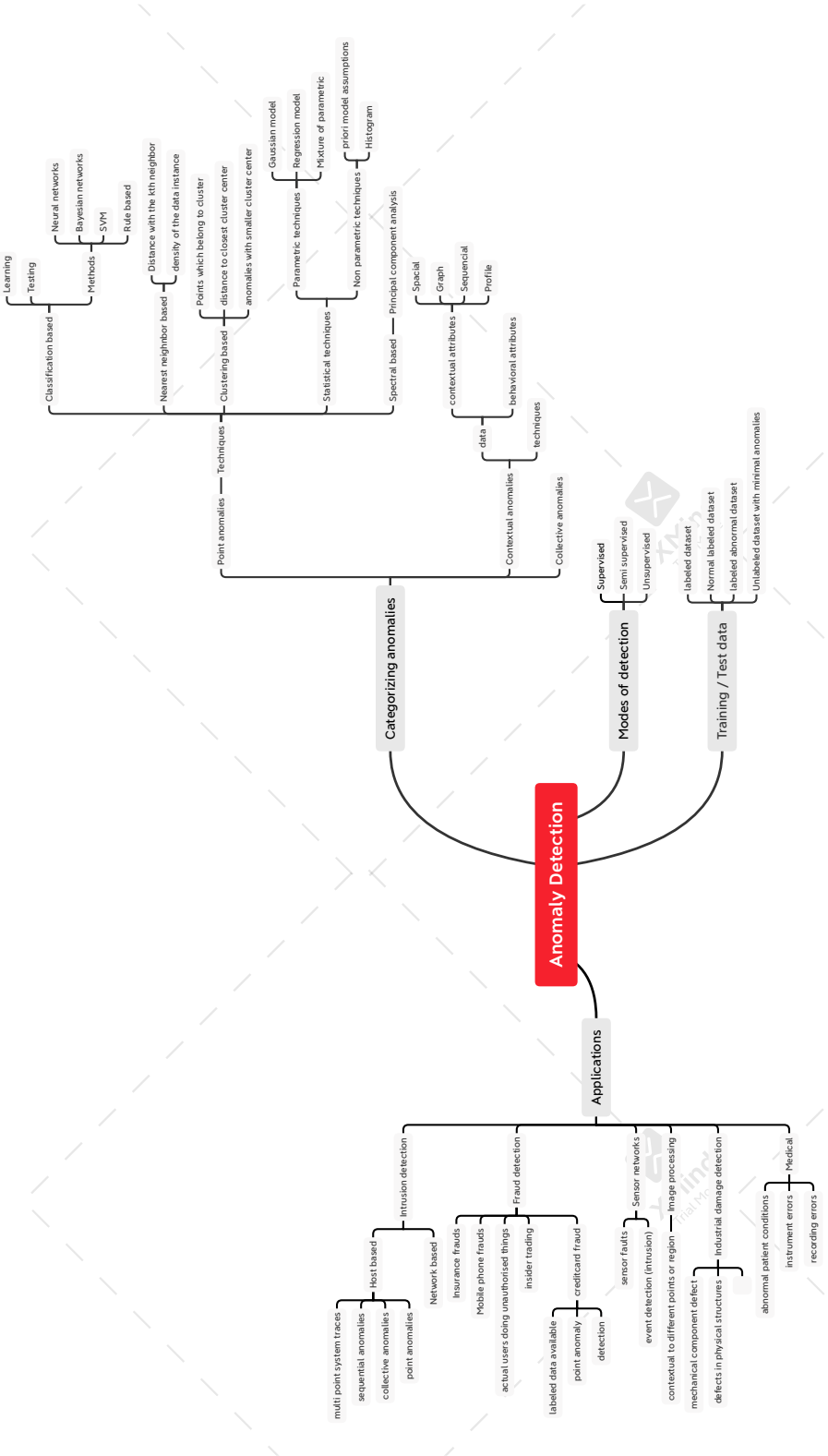



Figure D.2: *Mind Map of Anomaly Detection*

# References


[1] Charu C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2nd edition, 2016. ISBN 3319475770.

[2] Sabtain Ahmad, Kevin Styp-Rekowski, Sasho Nedelkoski, and Odej Kao. Autoencoder-based condition monitoring and anomaly detection method for rotating machines, 2021.

[3] Sabtain Ahmad, Kevin Styp-Rekowski, Sasho Nedelkoski, and Odej Kao. Autoencoder-based condition monitoring and anomaly detection method for rotating machines, 2021.

[4] H G Andres. Time series analysis - a complete guide. `https://www.kaggle.com/andreshg/timeseries-analysis-a-complete-guide`, 2017.

[5] Hiroshi Ashikaga, Heidi Estner, Daniel Herzka, Elliot Mcveigh, and Henry Halperin. Quantitative assessment of single-image super-resolution in myocardial scar imaging. *Translational Engineering in Health and Medicine, IEEE Journal of*, 2:1–12, 01 2014. doi: 10.1109/JTEHM.2014.2303806.

[6] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Bernie Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, Laurent Callot, and Tim Januschowski. Neural forecasting: Introduction and literature overview, 2020.

[7] Georgi N Boshnakov. On first and second order stationarity of random coefficient models. *Linear Algebra and Its Applications*, 2(434):415–423, 2011.

[8] Mohammad Braei and Sebastian Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art, 2020.

[9] M. Celik, Filiz Dadaşer-Çelik, and A. S. Dokuz. Anomaly detection in temperature data using dbscan algorithm. *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 91–95, 2011.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL `https://doi.org/10.1145/1541880.1541882`.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41, 07 2009. doi: 10.1145/1541880.1541882.

[12] Chance Elliott, Vipin Vijayakumar, Wesley Zink, and Richard Hansen. National instruments labview: A programming environment for laboratory automation and measurement. *JALA: Journal of the Association for Laboratory Automation*, 12(1):17–24, 2007. doi: 10.1016/j.jala.2006.07.012. URL `https://doi.org/10.1016/j.jala.2006.07.012`.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[14] Muhammad Fahim and Alberto Sillitti. Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access*, 7:81664–81681, 2019. doi: 10.1109/ACCESS.2019.2921912.

[15] Marek Florkowski and Jian Li. Condition monitoring and diagnostics. *IEEE Transactions on Dielectrics and Electrical Insulation*, 27(6):1769–1769, 2020. doi: 10.1109/TDEI.2020.009340.

[16] Ellen Friedman and Ted Dunning. Practical machine learning: A new look at anomaly detection. 2014.

[17] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.*, 28(1–2):18–28, February 2009. ISSN 0167-4048. doi: 10.1016/j.cose.2008.08.003. URL `https://doi.org/10.1016/j.cose.2008.08.003`.

[18] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[19] William Gousseau, Jérôme ANTONI, François Girardin, and Julien Griffaton. Analysis of the Rolling Element Bearing data set of the Center for Intelligent Maintenance Systems of the University of Cincinnati. In *CM2016*, Charenton, France, October 2016. URL `https://hal.archives-ouvertes.fr/hal-01715193`.

[20] Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28 (10):2222–2232, Oct 2017. ISSN 2162-2388. doi: 10.1109/tnnls.2016.2582924. URL `http://dx.doi.org/10.1109/TNNLS.2016.2582924`.

[21] Frank E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1): 1–21, 1969. doi: 10.1080/00401706.1969.10490657. URL `https://www.tandfonline.com/doi/abs/10.1080/00401706.1969.10490657`.

[22] Nikou Günnemann, Stephan Günnemann, and Christos Faloutsos. Robust multivariate autoregression for anomaly detection in dynamic product ratings. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 361–372, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327442. doi: 10.1145/2566486.2568008. URL `https://doi.org/10.1145/2566486.2568008`.

[23] Hochschule Bonn-Rhein-Sieg (H-BRS). Platform for scientific computing at bonn- rhein-sieg university. `https://wr0.wr.inf.h-brs.de/wr/index.html`, 2017.

[24] Douglas M Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

[25] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, page 387–395, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

[26] R.J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice. OTexts*. 2014. URL https://books.google.de/books?id=gDuRBAAAQBAJ. ISBN: 9780987507105.

[27] National Instruments. DAQCard - 6062E. 11500 N. Mopac Expwy Austin, TX 78759-3504 USA, 2011. URL https://www.ni.com/de-de/support/model.daqcard-6062.html.

[28] G. Yu J. Lin J. Lee, H. Qiu and University of Cincinnati. Bearing Data Set NASA Ames Prognostics Data Repository NASA Ames Research Center Moffett Field CA Rexnord Technical Services, IMS. http://ti.arc.nasa.gov/project/prognostic-data-repository, 2017.

[29] Feng Jia, Yaguo Lei, Jing Lin, Xin Zhou, and Na Lu. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72:303–315, May 2016. doi: 10.1016/j.ymssp.2015.10.025.

[30] Yongjun Jin, Chenlu Qiu, Lei Sun, Xuan Peng, and Jianning Zhou. Anomaly detection in time series via robust pca. In *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pages 352–355, 2017. doi: 10.1109/ICITE.2017.8056937.

[31] Bauyrjan Jyenis. Anomaly detection in time series sensor data. https://towardsdatascience.com/anomaly-detection-in-time-series-sensor-data-86fd52e62538, 2020.

[32] Maurice Kendall, Stuart. Signal smoothing - mathworks, May 2017. URL https://www.mathworks.com/help/signal/ug/signal-smoothing.html.

[33] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2725–2732. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/378. URL https://doi.org/10.24963/ijcai.2019/378.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[35] Alexander Lavin and Subutai Ahmad. Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark. *2015 IEEE 14th International Conference on Machine Learning and*

*Applications (ICMLA)*, Dec 2015. doi: 10.1109/icmla.2015.141. URL http://dx.doi.org/10.1109/ICMLA.2015.141.

[36] Zhi Lv, Jian Wang, Guigang Zhang, and Huang Jiayang. Prognostics health management of condition-based maintenance for aircraft engine systems. In *2015 IEEE Conference on Prognostics and Health Management (PHM)*, pages 1–6, 2015. doi: 10.1109/ICPHM.2015.7245055.

[37] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[38] Soheila Mehrmolaei and Mohammad Reza Keyvanpour. Time series forecasting using improved arima. In *2016 Artificial Intelligence and Robotics (IRANOPEN)*, pages 92–97, 2016. doi: 10.1109/RIOS.2016.7529496.

[39] M. Mehdi Moradi, Ottmar Cronie, Ege Rubak, Raphael Lachieze-Rey, Jorge Mateu, and Adrian Baddeley. Resample-smoothing of voronoi intensity estimators, 2018.

[40] Rizwan Mushtaq. Augmented dickey fuller test. 2011.

[41] Donald E Myers. To be or not to be... stationary? that is the question. *Mathematical Geology*, 21(3): 347–362, 1989.

[42] Dhiraj Neupane and Jongwon Seok. Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access*, 8:93155–93178, 2020. doi: 10.1109/ACCESS.2020.2990528.

[43] Keith Ord. Outliers in statistical data: V. barnett and t. lewis, 1994, 3rd edition, (john wiley sons, chichester), 584 pp., [uk pound]55.00, isbn 0-471-93094-6. *International Journal of Forecasting*, 12 (1):175–176, 1996. URL https://EconPapers.repec.org/RePEc:eee:intfor:v:12:y:1996:i:1:p:175-176.

[44] Shay Palachy. Stationarity in time series analysis - a review of the concept and types of stationarity, April 2019. URL https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322.

[45] An Amphenol Company PCB Piezontronics. PCB 353B33 High Sensitivity Quartz ICPs Accelerometer. 3425 Walden Avenue Depew, NY 14043, 2011. URL https://www.pcb.com/products?model=353b33.

[46] Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. Review: A review of novelty detection. *Signal Process.*, 99:215–249, June 2014. ISSN 0165-1684. doi: 10.1016/j.sigpro.2013.12.026. URL https://doi.org/10.1016/j.sigpro.2013.12.026.

[47] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289

(4):1066–1090, 2006. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2005.03.007. URL https://www.sciencedirect.com/science/article/pii/S0022460X0500221X.

[48] Robert B. Randall and Jérôme Antoni. Rolling element bearing diagnostics—a tutorial. *Mechanical Systems and Signal Processing*, 25(2):485–520, 2011. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2010.07.017. URL https://www.sciencedirect.com/science/article/pii/S0888327010002530.

[49] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley amp; Sons, Inc., USA, 1987. ISBN 0471852333.

[50] Cátia M. Salgado, Carlos Azevedo, Hugo Proença, and Susana M. Vieira. *Noise Versus Outliers*, pages 163–183. Springer International Publishing, Cham, 2016. ISBN 978-3-319-43742-2. doi: 10.1007/978-3-319-43742-2_14. URL https://doi.org/10.1007/978-3-319-43742-2_14.

[51] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 582–588, Cambridge, MA, USA, 1999. MIT Press.

[52] George Seif. Understanding the 3 most common loss functions for Machine Learning Regression. 2019. URL https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3.

[53] Letham B. Taylor SJ. Forecasting at scale. 2017. URL https://doi.org/10.7287/peerj.preprints.3190v2.

[54] Sutapat Thiprungsri and M. Vasarhelyi. Cluster analysis for anomaly detection in accounting data: An audit approach 1. *The International Journal of Digital Accounting Research*, 11:69–84, 2011.

[55] Diego A. Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. A mixture of gaussians hidden markov model for failure diagnostic and prognostic. In *2010 IEEE International Conference on Automation Science and Engineering*, pages 338–343, 2010. doi: 10.1109/COASE.2010.5584759.

[56] Diego Alejandro Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. A data-driven failure prognostics method based on mixture of gaussians hidden markov models. *IEEE Transactions on Reliability*, 61(2):491–503, 2012. doi: 10.1109/TR.2012.2194177.

[57] Diego Alejandro Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. A data-driven failure prognostics method based on mixture of gaussians hidden markov models. *IEEE Transactions on Reliability*, 61(2):491–503, 2012. doi: 10.1109/TR.2012.2194177.

[58] A. Ya. Chervonenkis V. N. Vapnik. class of algorithms for pattern recognition learning. *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 937–945, 1964.

[59] Helu M Vogl GW, Weiss BA. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *J Intell Manuf. 2019 Jan;30(1):79-95.*, 2016. doi: 10.1007/s10845-016-1228-8.Epub.

[60] Shiguo Wang. A comprehensive survey of data mining-based accounting-fraud detection research. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 1, pages 50–53, 2010. doi: 10.1109/ICICTA.2010.831.

[61] www.mathworks.com (Online). Rolling element bearing fault diagnosis - mathworks, May 2021. URL `https://www.mathworks.com/help/predmaint/ug/Rolling-Element-Bearing-Fault-Diagnosis.html`.

[62] H. P. Wynn. The advanced theory of statistics : Sir Maurice Kendall, A. Stuart and J.K. Ord, Vol. 3, 4th ed. (Griffin, High Wycombe, 1983) [UK pound]37.50, pp. 780. *International Journal of Forecasting*, 2(4):506–507, 1986. doi: https://doi.org/10.1002/for.3980040310. URL `https://ideas.repec.org/a/eee/intfor/v2y1986i4p506-507.html`.

[63] Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. Fault detection by mining association rules from housekeeping data. In *In Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2001.

[64] Rong Yao, Chongdang Liu, Linxuan Zhang, and Peng Peng. Unsupervised anomaly detection using variational auto-encoder based feature extraction. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7, 2019. doi: 10.1109/ICPHM.2019.8819434.

[65] Dit-Yan Yeung and Yuxin Ding. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition*, 36(1):229–243, 2003. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(02)00026-2. URL `https://www.sciencedirect.com/science/article/pii/S0031320302000262`.

[66] Antonio J. Zapata-Sierra, Alejandro Cama-Pinto, Francisco G. Montoya, Alfredo Alcayde, and Francisco Manzano-Agugliaro. Wind missing data arrangement using wavelet based techniques for getting maximum likelihood. *Energy Conversion and Management*, 185:552–561, 2019. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2019.01.109. URL `https://www.sciencedirect.com/science/article/pii/S0196890419301773`.

[67] Zhang372. Upsampling and downsampling lab, September 2013. URL `https://www.projectrhea.org/rhea/index.php/Upsampling_and_downsampling_lab`.