DEVELOPMENT OF ANOMALY DETECTOR FOR MOTOR BEARING CONDITION MONITORING USING FAST FOURIER TRANSFORM AND LONG SHORT TERM MEMORY (LSTM)-AUTOENCODER

BY

JIMOH, Oladebo Suliat
MTech/SICT/2018/8818

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF TECHNOLOGY (M.TECH) IN COMPUTER SCIENCE

NOVEMBER, 2021

ABSTRACT

Motor bearings have over the years been one of the components which aids efficiency in industries as it helps smooth running of rotary machines in industries. However, it is important to note that the rate of degradation of motor bearings vary from one machine to another. This phenomenon is unavoidable but vary as a result of operational and environmental factors such as the time of operation of machines where installed, ambient temperature, load factor and maintenance ethics like constant oiling. For sustainability of machines, safe points of these factors have to be considered to avoid anomaly of bearings that could lead to high maintenance cost such as bearing faults, fatigue and accelerated aging or even complete breakdown which accounts for 30% to 40% failures of machines. With this effect, production in industries could halt as a result of prolonged downtime due to anomaly. Furthermore, it is important to note that complete breakdown could be catastrophic especially in automobiles and heavy duty production machines which could as a result of sudden failure during operation, jet parts from the machine leading to accidents and sometimes death. This therefore suggests that consistent monitoring of the health status of bearings is important so as to ensure efficiency and avert complete breakdown. Aside that, it is also important to detect this anomaly much earlier via sensitive methods so as to be fore warned. However, to avoid downtime, it is a good practice to constantly monitor the component so as to aid early detection of anomaly before a break down. Over the years, a lot of researches has been done in the detection of anomaly in motor bearing using time sequence data which has evolved via the use of Artificial intelligence (AI) these days. However, since bearings can run for months or years without anomaly, there is a need to use an unsupervised AI model which could be trained with data characterized with normal bearing operation and can flag an anomaly when an outlier is detected. Despite the use of AI techniques, often times, anomaly is predicted at a high threshold signifying low sensitivity. This is because of the sequential data generated is often laced with noise from sensor and therefore characterized by low signal to noise ratio (SNR). Furthermore, for better accuracy, the noise within the data generated as a result of the sensors used has to be taken into consideration and worked upon using a digital signal processing technique such as Fast Fourier Transform (FFT) so as to aid fast computation and anomaly detection at low signal to noise ratio. To improve detection, this research work presents the Development of an Anomaly Detector for Motor Bearing Condition Monitoring using Fast Fourier Transform (FFT) and Long Short Term Memory (LSTM)-Autoencoder (AE). This was achieved via the use of Fast Fourier Transform (FFT) and Long Short Term Memory (LSTM)-Autoencoder which helped to detect the anomaly at a threshold less than 0.2 and also attain an accuracy of 91.3%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PLATES

GLOSSARY

Abbreviation	Meaning
AI	Artificial Intelligence
ANFI	Adaptive Neuro Fuzzy Inference
ANN	Artificial Neural Network
CBM	Conditional Based Maintenance
CNN	Convolutional Neural Network
CSC	Cyclic Spectral Correlation
CSCoh	Cyclic Spectral Coherence
DAD	Deep Learning Anomaly Detection
DWT	Discrete Wavelet Transform
FFT	Fast Fourier transform
FS	Full scratch
FT	Fourier Transform
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
KNN	K-nearest neighbors
LSTM	Long Short Term Memory
NASA	National Aeronautics and Space Administration

OP-SWPT	Optimized Stationary Wavelet Packet Transform
PCA	Principal Component Analysis
QCD	Quick change detection
RF	Random Forest
RNN	Recurrent Neural Network
RUL	Remaining useful life
SDA	Stacked Denoising Autoencoder
SNR	Signal to noise ratio
SS	Single scratch
SSAE	Sparse Stack Autoencoder
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
VSI	Virtual Spectrum Imaging
WT	Wavelet Transform
WPT	Wavelet Packet Transform

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Motor bearings are regarded as one of the most important parts of rotating machines such as turbines, industrial machines and automobiles (Choudhary *et al.*, 2020; Glowacz, 2019; Jin *et al.*, 2019; Novaes *et al*., 2019). The importance of this component cannot be over emphasized since they are responsible for the smooth running of the rotary parts of machines. Therefore, it is important to detect any anomaly behavior in such component. Before now, the traditional way of detecting anomaly is after complete breakdown. This however, is costly for industries. In the quest to reduce this cost, researchers have device a means of continuous monitoring of the machine via the use of sensors to convert physical occurrence into electronic signals (Jiang *et al.*, 2019). In the analysis of the signals which is characterized with repeated patterns, there are instances when a particular pattern is off the regular pattern as shown in Figure 1.1. This is called anomaly or outlier (Chalapathy and Chawla, 2019). Therefore, anomaly detection is the process used to highlight signals that have patterns different from the regular pattern that are repeating. According to Hawkins in 1980, he defined anomaly as an observation that deviates with huge significance from other observations that have similar patterns.(Chalapathy and Chawla, 2019).

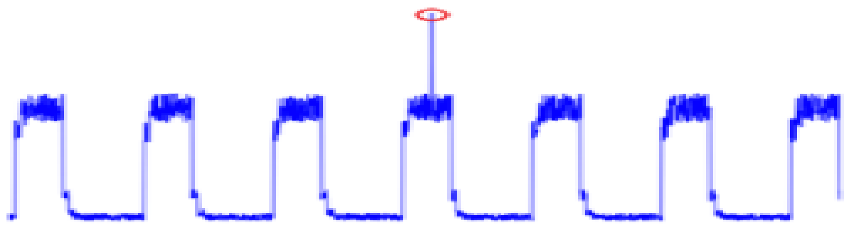

Figure 1.1: Illustration of anomaly (Chalapathy and Chawla, 2019).

These days, outliers have been the intelligent way for early detection of anomaly carried out without the machine breaking down. Failure to detect anomaly early may lead to the total failure of such machine which is costly in an industrial outfit. (Lee *et al.*, 2018; Jiang *et al*., 2019; Mohendra *et al*., 2018). This (early detection of anomaly) can only be ascertained via the continuous monitoring of the machine which is called in line monitoring (Jiang *et al*., 2019). Early anomaly detection also known as prognostic Health Management of machines, according to Lee *et al.* (2018) or condition monitoring according to Jiang *et al.* (2019) involves conventional anomaly detection and prediction model which has the ability to notice aging in machine parts. These make use of signals which are preprocessed to enhance feature extractions of the signal data (Jiang *et al*., 2019).

In recent times, Artificial Intelligence has been employed in the detection of anomaly in machines. Some techniques used includes the Support Vector Machine (SVM), Bayesian Classifier, Neural Networks and deep learning methods (Jiang *et al*., 2019). However, the imbalanced nature of the dataset which is generated from sensors poses challenges in accurate anomaly detection especially in terms of classification since the data isn't labelled. However, it is important to note that these data in time series format is needed to predict time for anomaly and time for complete breakdown.

According to Chalapathy and Chawla (2019), the reason why the use of Artificial Intelligence which includes the use of machine learning and deep learning is most preferred compared to the traditional methods of anomaly detection is that unlike the traditional method which data may not be available to suspect an anomaly, the use of Artificial intelligence (AI) makes use of data generated so as to detect variant. This however, aids detection before breakdown. Among these AI approaches, deep learning is observed to outperform the traditional machine learning because of its flexibility in learning and representation of data in hierarchical form in a neural network thereby increasing good performance in detection (Chalapathy and Chawla, 2019). As a result, its applications have been boundless as it was used by Adewumi and Akinyelu (2017) to detect financial fraud, used to detect intrusion in cyber space by Kwon *et al*. (2019), used for Big data anomaly detection in the internet of things (IoT) (Mohammadi *et al*., 2018) and medical image analysis (Litjens *et al*., 2017). According to Lee *et al.* (2018) the use of deep learning for anomaly detection has been on the increase because of its ability to receive signal data and automatically preprocessing it before extracting features needed (Lee *et al.*, 2018). This therefore validates the use of deep learning approach in this research. Generally Deep Learning Anomaly Detection is categorized based on the following:

Type of Input Data: The type of DAD used in detection of anomaly could be dependent on the type of data used, there are two categories of data. They are sequential data which includes voice, text, time series data, and the non-sequential data. The later includes images and other kinds of data. According to Chalapathy and Chawla (2019), Deep Learning models known for detection of anomaly using sequential data includes Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM). While the techniques often used for non-sequential data remains CNN, autoencoders and its variants. These data can further be categorized as high dimensional or low dimensional data depending on the number of attributes that is within it.

Based on Label Availability: The target colon of Data can be labeled or unlabeled. This helps to categorize DAD into Supervised, Semi supervised and Unsupervised.

Supervised DAD involves the use of data with labeled data to train a binary or multi-class classifier so as to detect anomaly. The binary classification usually will have two classes which can be Normal and Abnormal. While the multi-class DAD data will have targets having more than two classes. However, it is important to note that this method of anomaly detection is not so popular because data are not often labelled in real life application.

Semi Supervised DAD involves the use of data which are partially labelled. In this type of DAD, the data for the normal sequence is easily labeled. With this, outlier is detected. An example of such is the use of Autoencoders. In other words, the model is trained with normal data so that anomaly can be detected when abnormal data is detected.

Unsupervised DAD uses the intrinsic properties of data to detect anomaly. In other words, this aids the automatic labelling of unlabeled data. This method has been observed to outperform the traditional Principal Component Analysis (PCA), SVM, Vapnik and Isolation Forest techniques which often is applied in health and cyber space. (Chalapathy and Chawla, 2019).

With all these DAD approaches towards the detection of anomaly, the use of unsupervised learning especially autoencoders is chosen in this research because of its application in real life scenarios. Furthermore, the need for LSTM cannot be overemphasized since it is often used for large sequence data.

In general, to use any deep learning technique for the detection of anomaly, huge data is needed. It is important to note that since the data generated is done from signals which comes from sensors, the actual signal or rather the raw signal may be characterized with electrical noise and low signal to noise ratio (SNR). According to Vo *et. al.* (2017), these raw signals are often times not suitable for outlier detection. This is because even though deep learning has overwhelmed other methods such as the use of Support Vector Machine (SVM), Artificial Neural Network (ANN), and Adaptive Neuro Fuzzy Inference (ANFI), during the process, the variable dependencies can be used without indicating the outlier detection. Detection may be difficult as noise levels can give us false alarms or no alarm. In other words, the generation of weak signals by some of these sensors can hinder detection as the threshold which is calculated based on error margin of normalcy may have been set too high based on stronger signals. Therefore, the higher this threshold, the less effective the approach will be in detecting weak signals. Furthermore, the use of time stamped data directly may be cumbersome and slow in the process of anomaly detection. To achieve detection faster with better sensitivity and accuracy, Jabczyńskę and Szczęśniak (1995) proposed Fast Fourier transform (FFT) as a robust technique for data characterized by low SNR. This helps to convert sequence data from time domain to frequency domain. With the use of this technique, feature extraction can be done easily (Jabczyńskę and Szczęśniak, 1995; Vo *et al.*, 2017) even though salient especially in low SNR characterized data. Other benefits of using the FFT technique is that, it aid anomaly detection faster as it reduces computational time by reducing complex multiplications and acts as a filter (Tan *et al.*, 2017) to aid the cleaning up of the data. Therefore, this research presents the Development of Anomaly detector for motor bearing condition monitoring using Fast Fourier Transform and LSTM-Autoencoder.

1.2 Statement of the Research Problem

Anomaly especially in bearings is one reason for the down time of machines. Recently, a lot of researchers have done extensive work towards anomaly detection. In their research which involves the use of data generated from sensors, large thresholds are observed as a result of the use of sensor data which is characterized by noise and therefore characterized with low SNR irrespective of the technique used. This low SNR often times results into false alerts of anomaly and the inability for the deep learning models to sense anomaly of weak signals (low sensitivity). To improve detection of anomaly, it is important to clean up the data using digital signal processing techniques like FFT so as to be able to detect anomaly at lower thresholds which also indicates improved sensitivity.

Thus, this research work aim to detect anomaly in the presence of noise and low SNR through development of anomaly detector for motor bearing condition monitoring using Fast Fourier Transform (FFT) and Long Short Term Memory (LSTM)-Autoencoder.

1.3 Aim and Objectives

The aim of this research work is to develop an Anomaly detector for motor bearing condition monitoring using Fast Fourier Transform and Long Short Term Memory (LSTM) - Autoencoder. This will be achieved via the following objectives.

1. To clean the noise contained in the data and transform the data using Fast Fourier Transform (FFT).
2. To develop anomaly detection model based on Long Short Term Memory (LSTM) – Autoencoder (AE).
3. To hybridized the approach in (1) and (2) into the anomaly detection model.
4. To evaluate the performance of the approach in (3) using threshold and accuracy performance metrics.

1.4 Significance of the Study

From literature, it is obvious that as a result of how important bearings are in machines, several approaches has been used to detect anomaly. This approaches ranges from the use of supervised learning to semi supervised learning to unsupervised learning.

However, as a result of imbalance dataset and difficulty in getting complete labeled datasets, unsupervised learning has been used more of late. In line with this, lot has been done using autoencoders. However, using autoencoders alone may be advantageous since it has the ability to capture datasets from low signal, but, owing to the fact that the features are temporal, one will need an autoencoder that will also be able to remember the immediate past features of the data. This therefore, justifies the reason why we use LSTM Autoencoder in this research. This however suggests that LSTM Autoencoders are preferred for these kinds of challenge. Despite the fact that many researches has been done on this, literatures reviewed did not consider improvement on sensitivity. They only focus on accuracy. However, as earlier mentioned, the use of this technique (LSTM Autoencoder) may not be fast in the detection of anomaly especially with time sequence data. For that reason, this research uses FFT with LSTM Autoencoder since it has been observed from reviews that the use of digital signal processing especially FFT could aid signal cleanup and faster computation.

1.5 Scope of the Study

The scope of this research is limited to the detection of anomaly in motor bearings at a threshold lower than 0.2 which was achieved by (Ahmad *et al.*, 2021).

CHAPTER TWO

2.0 LITERATURE REVIEW

Over the years, as a result of the quest to run efficient industrial process involving the use of rotary machines, anomaly detection in motor bearing have been looked into by a lot of researchers. A lot of researchers have come up with techniques to detect impending defects in motor bearing before it becomes permanent damage. These efforts are reviewed in this chapter.

2.1 Data Generation for Anomaly Detection

Data generation has over the years been done from signals of different kinds. These signals includes acoustic signals, infrared thermograph method and, vibration signal (Choudhary *et al.*, 2020; Glowacz, 2019; Jin *et al.*, 2019). According to Glowacz (2019), the most efficient is infrared thermograph method. This is because, apart from it being a contactless method of generating data, the data generated is not affected by the load vibrations and speed fluctuations that affect the vibration method of data generation. Also, it is not affected by background noise which affects the acoustic method of data generation. The drawback of this method of data generation is it being too expensive to deploy. Also, the method may generate insignificant information and noise in the image generated which may affect accuracy of predictions.

Another way to generate data is via acoustic signals (Glowacz, 2019). This method which makes use of a pickup microphone is contactless. However, its limitation is that, it picks up background noise which may not be useful. These noises could lead to false trigger or no trigger if not properly handled.

The use of vibration remains the most widely used (Jin *et al.*, 2018). Though, it is a cheaper medium and can analyze electrical fault and mechanical fault (including bearing anomaly) in machines, it is limited as it is affected by random noise, speed fluctuations, load vibrations and background noise (Glowacz, 2019; Hruntovich *et al.*, 2019). However, because of its cheap nature this kind of data generation is considered in this research.

2.2 Fault Detection in Motor Bearing via Vibration Analysis

For better industry, by providing smart ways of operation, many scientist and researchers have come up with ways to evaluate the most crucial part of the industrial machine. One of such machines is the motor bearing which aids the smooth running of the machine to provide efficient services. The reason for the analysis of the part remains to avert down time which may result from the brake down of this component.

In literatures, a lot of method have been employed to aid the analysis of this part. One of the methods includes the use of vibration analysis (Khadersab and Shivakumar, 2018). This analysis involves the collection of vibrational data which is analyzed using different techniques which includes the use of Fourier Transform, (FT), Fast Fourier Transform (FFT), Inverse Fast Fourier Transform (IFFT), Short Time Fourier Transform (STFT) and Wavelet Transform (WT).

Khadersab and Shivakumar, (2018) presented how FFT and IFFT was used to analyze failure in bearings. In their submissions, the healthy vibration data and the data characterized with faults were compared using these techniques to accurately access the failure in the bearing. In their experiment a piezo-electric accelerometer sensor was used to take vibration data. The data acquired is interfaced with an FFT algorithm called EL-

Calc. Afterwards, the FFT signal was used to generate IFFT and spectrogram. At the end of the experiment, defects were effectively identified.

According to Sulka *et al*. (2019) causes of unwanted vibration and the extent of fault can be estimated. In their presentation, FFT and STFT was used to determine defect in bearings. The difference between these two techniques is that unlike FFT that makes no use of a time window, STFT uses a time window to achieve analysis of a particular vibration data. After using both techniques, it was observed that FFT shows larger amplitude in the vibration signal analyzed. Also, the amplitude depends on the depth of damage.

In treating vibrational signal which are non-periodic, Abouelanouar *et al*. (2018) worked on fault detection via the use of wavelet transform. In their review, WT was used to detect faults in gears and bearings. In his conclusion, WT is a powerful tool for detection of faults (Abouelanouar *et al*., 2018).

Other applications where digital signal processing has been used include detection of tampering in images used in multimedia. This was presented by Kanwal (2019). In their presentation, the method used to aid the detection of tampering was FFT with local texture descriptors together with SVM classifier. At the end of the research, detection accuracy was increased.

In a review done by Barot and Kulkarni (2021), detailed presentation was done on the different techniques needed to aid the detection of anomaly. Generally, it was emphasized that there is a need to use digital signal processing which involves the use of FFT, WT, Discrete Wavelet Transform (DWT) and Wavelet Packet Transform (WPT). Furthermore, it was stated that the need for denoising cannot be over emphasized. Also, weak signals have to be extracted since the signals are always masked with noise. One way to do this includes the use of wavelet-based signal denoising which aids the increase of the signal to noise ratio. Aside the signal processing, the author presented a review of artificial intelligent methods used in diagnosing faults in bearings. According to the author, diagnosis could be to determine whether there is fault in a machine or not, to find the exact position of the fault and lastly to predict the trending of fault development. For fault recognition, the author presented convex optimization, mathematical optimization as well as classification, statistical learning and probability-based methods. Types of classifiers that can be used includes the k-Nearest Neighbor, Bayesian classifiers, Support Vector Machine and Artificial Neural Network which are all part of artificial intelligence.

All these are diagnostic measures to machine health as described by Lee *et al*. (2018). This however, will not aid the vision of smart industry which involves data gathering via internet of things (IoT), cloud computing and big Data analysis to aid prognostic measures to avoid break down. This industry is also known as industry 4.0 (Lee *et al*., 2018). In their presentation, a lot was reviewed on the different machine health maintenance and the different technique used, some of which includes the use of analytical models, experimental model, fuzzy logic, Time domain analysis, Fourier series, Numerical model, Artificial Neural Network, WT and much more. To achieve this, different type of data gathering techniques were reviewed. This includes vibrational signals, Acoustic signals via the use of acoustic emission sensor, force signal, temperature, and electric signals. Furthermore, techniques used to detect anomaly was reviewed. These techniques includes the use of Big Data, Hybrid algorithms, machine learning algorithm such as support vector machine (SVM), statistical model, neural networks and Empirical mode decomposition.

2.3 Machine Learning used in Anomaly Detection

Machine learning is an aspect of data science that aids machine intelligence. This paradigm describes the ability for machines to take decisions based on past experiences which is represented as data. The implementation of machine learning is achieved in three models as shown in Figure 2.1. This includes supervised learning, semi supervised learning and unsupervised learning (Pittino *et al.*, 2020; Chalapathy and Chawla, 2019; Shen *et al.*, 2019).

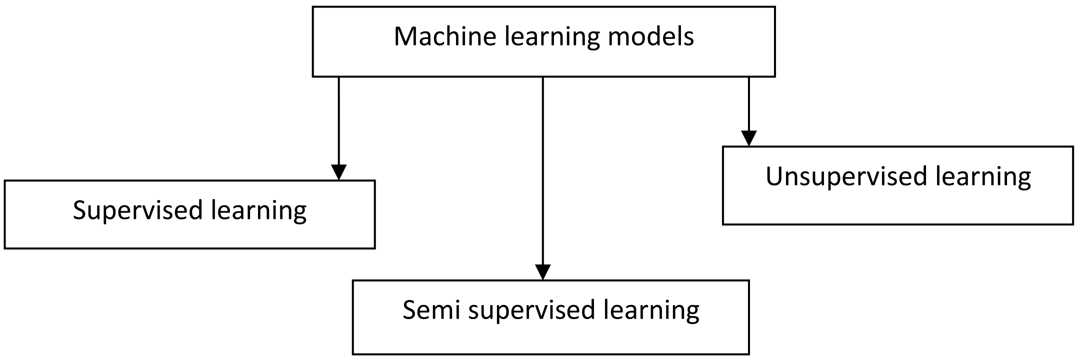

Figure 2.1: Type of machine learning models.

Supervised learning involves the use of data that is labeled. Usually, this method of machine learning is used for classification problems. As described in many literatures this method seems to be the easiest way to detect bearing fault anomalies (Shen *et al.*, 2019). However, in real life it may be challenging to get such data (Shen *et al.*, 2019; Chalapathy and Chawla, 2019; Shen *et al.*, 2020). Furthermore, according to Shen *et al.* (2019), during anomaly detection, it is difficult to state when the fault started and how long it lasted. In other words, it may be easy to get time series data which indicates healthy condition of the bearing but one may not be able to tell when the defect started since it takes hours, days or months before motor bearings are characterized by defects which may not be noticed (Meire and Karsmakers, 2019; Shen *et al.*, 2020). Furthermore, the data collected can be imbalanced (Meire and Karsmakers, 2019) since much of data will not be collected at fault state. Also, for systems that could be catastrophic in nature, it is not advisable to run such for a long time so as to avoid accidents (Shen *et al.,* 2020). This however, makes supervised learning less attractive for automatic anomaly detection in motor bearing.

Semi supervised learning describes machine intelligence based on data which are not all labeled. This is typical of time series data from motor bearing. This is because the healthy state can be labeled, the point where the fault or anomaly is detected can be labeled and the point of machine failure can be labeled. However, the point between the point where anomaly was detected and the point of failure may be difficult to label (Shen *et al.,* 2019). As a result, this data can be used for semi supervised learning (Liu *et al.,* 2019) since models like autoencoders can take advantage of labeled data to highlight outliers (Ren, 2019). The challenge of using the semi supervised learning is that it involves complex procedures as discussed by Shen *et al.* (2019). This however makes semi supervised learning less attractive for this research.

Unsupervised learning is machine learning model that deals with unlabeled data which is the characteristic of data generated from motor bearing. This model detects outliers as anomaly based on intrinsic properties of the data (Ren, 2019). As a result, it is considered more flexible and easier to use to the automatic anomaly detection in motor bearing. This method does not need time wasting and cumbersome process of labeling nor do some cumbersome data wrangling. Instead, regression models are built to aid prediction of anomaly. However, since the imbalance nature of the data can’t be avoided, it is better to train a regression model with the normal data and look for outliers (Chalapathy and Chawla, 2019). One way to achieve this is to compress the data to a small representation and regenerate the output from the compressed data. The output data is expected to look like the input data. However, the difference in these will give an alert of an anomaly. The process in which the data is compressed is called encoding which is a function of autoencoders. Furthermore, the process of regenerating the input data from compressed data is called decoding. Generally, autoencoders are neural networks that deduce features of low-level signals by comparing the differences between the input data and the output data. To achieve this, the data is first compressed (or under goes encoding) and then from the compressed data the output is generated or decoded. According to Meire and Karsmakers (2019), autoencoders are efficient tools for low level signals which is the characteristics of bearing signals. As shown in Figure 2.2, X_T is inputted and compressed to be Z_T. Afterwards, it is decoded or the output is generated as ϕ. If the output deviates from the input then, an outlier is detected hence an anomaly.

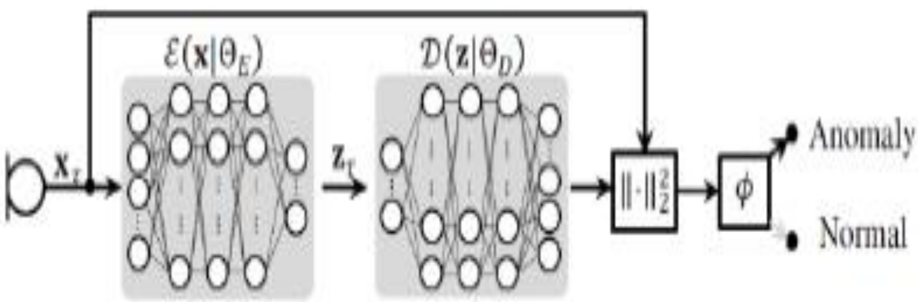

Figure 2.2: Diagram of an autoencoder network (Meire and Karsmakers, 2019)

It is important to note that the data from the motor bearing is characterized with time sequence. Therefore, it can be regarded as time sequence data. To handle such data, Recurrent Neural Networks (RNN) are usually used (Ren, 2019). However, it does not have the ability retain information of former time stamp. To overcome this, Long Short Time Memory (LSTM) algorithm was built. This has the ability to retain information of former time stamp.

Having studied the nature of the data generated by motor bearing, characterized with time stamp which can be encoded or compressed and reconstructed to solve anomaly issues;

this research therefore focuses on automatic anomaly detection in motor bearing using LSTM-Autoencoder.

2.4 Related Works on Artificial Intelligence

In reducing the downtime of machines in industries, authors in (Jin *et al*., 2018) and (Jin *et al*., 2019) proposed a data driven approach for bearing prognostic based on Kolmogorov-Smirnov test, self-organizing map and unscented Kalman filter as shown in Figure 2.3. In their approach, the first step taken was to detect bearing degradation process also known as anomaly detection by learning the historical data generated via vibration sensors. The second process involves the prediction of the remaining useful life (RUL) of the bearing with the aid of degradation model and unscented Kalman filter.

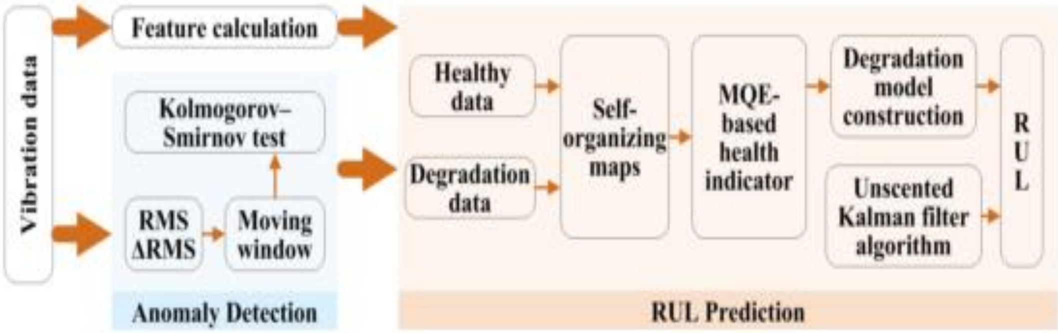

Figure 2.3: Data driven approach for bearing prognostic based on Kolmogorov-Smirnov test, self-organizing map and unscented Kalman filter (Jin *et al*., 2019)

Usually, in many research, many anomaly detection is achieved via the use of large data generated by setup similar to Figure 2.4. However in real life these data may not be available in the quantity required for detection. For that reason, authors in (Shen *et al*., 2020) proposed a few shot learning approach based on model-agnostic meta-learning. This approach aids classification using limited data. Compared to Siamese-network based benchmark study approach, their approach improved in accuracy by 25%.

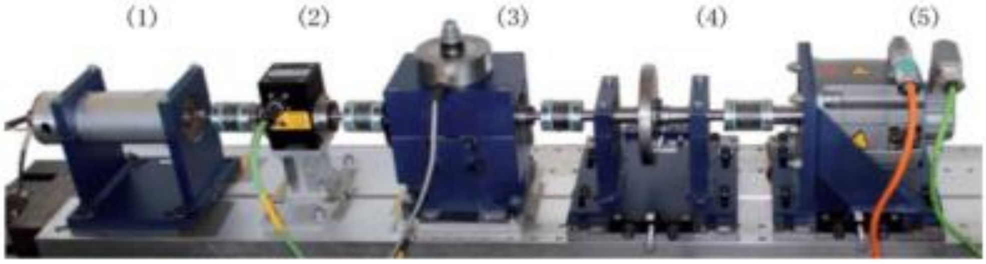

Figure 2.4: Experimental set up of machine for data collection. (1) electric motor (2) torque measurement shaft (3) rolling bearing test module (4) fly wheel (5) load motor (Shen *et al*., 2020).

Shen *et al.* (2019), in their presentation emphasized the use of internet of things (IoT) for data gathering in preparation to detect anomaly in bearing. Their work which aims at ensuring adequate maintenance before a complete breakdown of machines, gave an expose of different techniques used to track the degradation of the system. Such technique involves the use of signal processing for better performance in degradation tracking. Cyclic Spectral Correlation (CSC) and Cyclic Spectral Coherence (CSCoh) has been proven to be powerful tool for signal processing. Furthermore, as a result of the strain or difficulty in getting labeled data via experimental setup in Figure 2.5, fault detection was done via the used of semi supervised learning and Support data Description.

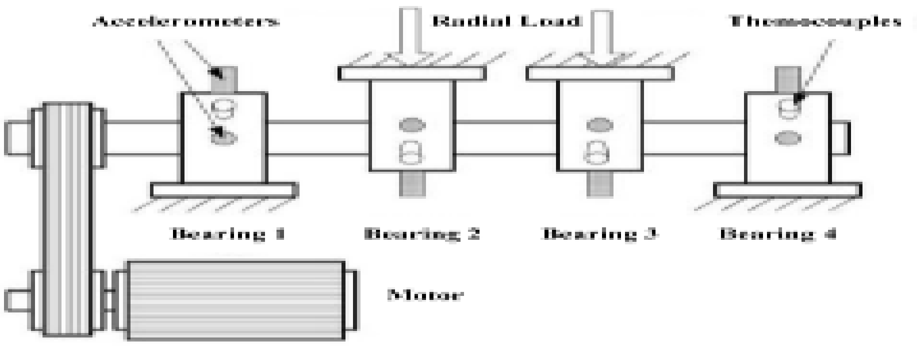

Figure 2.5: Experimental set up for data gathering used for semi supervised learning (Liu *et al*., 2019).

After the run to fail experiment, result indicated that bearing anomaly detection can be done accurately via the use of a semi supervised vector detecting device and CSCoh.

Syan *et al.* (2020) presented a case study which utilizes Conditional Based Maintenance (CBM) for centrifugal pumps as part of safety for critical water system used to mitigate fire hazard (Syan *et al.*, 2020). This was done so as to monitor the operational condition of centrifugal pumps. To achieve this, vibration data was gathered so as to investigate if the conditional based maintenance will identify different faults in the centrifugal pump. The work flow of the research as shown in Figure 2.6 involves the determination of best practice or approach of CBM in the lab compared to the CBM approach in the industry. Furthermore, they determine a research gap and design an experimental study to fill such gap. Afterwards, data is being collected, results analyzed and limitation highlighted.

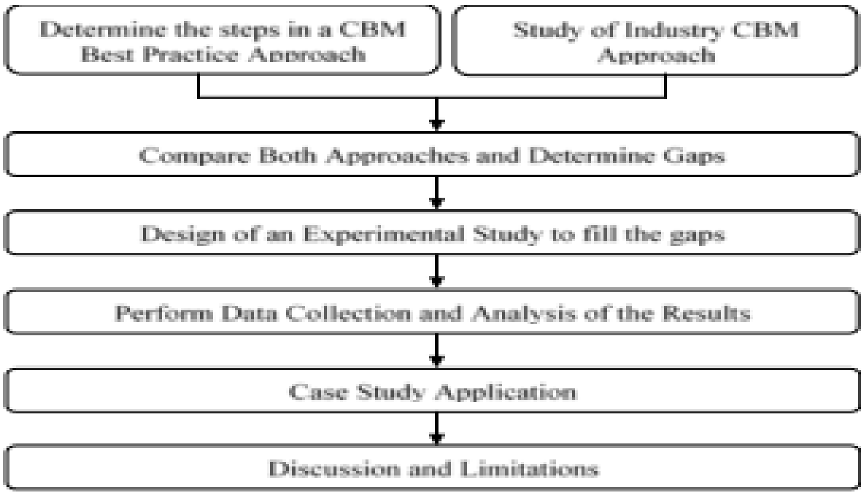

Figure 2.6: Work flow of the research (Syan *et al.*, 2020).

Results show that, in a place where single fault was studied, the accuracy of detection was 100%. For multiple faults, the accuracy was 67%. However, the overall classification accuracy is 76.5%. To improve classification accuracy, it was suggested that artificial intelligence should be used in fault detection.

Detecting anomaly using current approach has been described by authors in (Boniol *et al.*, 2020) as cumbersome and expensive especially when having recurrent anomaly. Also, the prior knowledge of the domain is important so as to aid effective detection. To solve this difficult and cumbersome trait, they used NorM: a technique used for domain-agnostic anomaly detection. The technique detects recurring anomaly when compared to a model that represents normal behavior. The result of this technique shows high performance (Boniol *et al.,* 2020).

In industry 4.0, the advent of smart meters and IoT has improved the gathering of data which is instrumental to the detection of anomaly in industrial machines (Pittino *et al*., 2020). According to authors in (Pittino *et al.*, 2020), even though machine learning has been used over the years to extract information from the dataset which seems impossible for humans to do, the important aspect of anomaly detection remains the derivation of models which aids the detection of faults in the bearing. This aids detection of malfunctioning bearing before complete breakdown of machines.

The authors in Ren (2019) emphasized the need of deep learning techniques in detecting anomaly in unmanned vehicles. This is done so as to ensure safety of the unmanned vehicle. Among several techniques, deep learning was chosen because of the high dimension of enormous data which is used for auld determination especially in bearings. Furthermore, the method adopted for this research is the use of X-ray images for classification which is rather expensive.

Saeki (2019) used visualization techniques specifically Convolutional Neural Network (CNN) for the detection of motor bearing anomaly. In his work, vibration dataset was captured and predictions were made to state the health status of the bearings. These predictions were compared with analysis done by experts. Results show that the technique is helpful in providing useful information as regards to the health status of motor bearings.

Aside motor bearing, anomaly detection has been applied to gears as stated in (Schmidt and Heyns, 2019). In their presentation, they argued that many technique used for anomaly detection involves the use of historic fault data which may not be available. According to them, in situations like this, techniques such as discrepancy analysis are used. This technique assumes that machine condition is the same throughout the signal in the model optimization process. In other words, no localized damage is present. To determine or identify localized fault, continues wavelet transform and principal component analysis is used to determine the divergence of gear under consideration. Afterwards, Bayesian data analysis technique was used to infer the presence of localized anomaly.

Mahalanobis-Taguchi, a method used to ascertain the extent of damage in a logical system as elaborated by Asakura *et al.* (2020) was used to detect anomaly in logical systems. In their work, the technique was applied on a large scale vertical transfer system. Calculations to achieve proper values for the technique was developed based on simple excitation using shaker (Asakura *et al*., 2020). Other researchers such as authors of Cooper *et al.* (2020) worked on anomaly detection in milling tools. The method used for the work is generative adversarial networks. The data used was acoustic based. Authors in Wang *et al.* (2020) used K-nearest neighbor to detect anomaly for machineries.

The author in Nath (2020) disclosed one major problem in anomaly detection. In his presentation, the abrupt change in sequential data is a problem in most anomaly detection. To solve this, low latency anomaly detection based on Quick Change Detection (QCD) is needed for effective detection. This minimizes the delay in detection of anomaly observed in sequential data. This is useful since in most models, Post Change Distribution model may not be available. This however, has been used for bearing fault detection in turbines.

Authors in Sohaib and Kim (2019) proposed the use of bi-spectrum analysis and Convolutional Neural Network (CNN) to be able to detect faults in bearings. In the work, the bi-spectra of the vibrational signal was first extracted. Afterwards, CNN based on stochastic optimization function (Sohaib and Kim, 2019) was proposed to extract the interclass of the bi-spectra. At the end of the research, detection was done more accurately than previous work.

In the quest to deal with sequence data, the author in (Pandarakone, 2018) used an online detection based on deep learning approach. In his work, Fast Fourier Transform was used in spectral analysis on data generated from load current of the stator coil so as to aid feature extraction. Afterwards, Convolutional Neural Network (CNN) was trained with the extracted features to aid classification. This method which has extended applications was also used to detect multiple faults in bearings such as Single Scratch (SS), and Full Scratch (FS). The average accuracy of the system was 88.17%.

Authors in Abid *et.al.* (2019), in their study, presented a technique to aid the detection of faults in bearings and the extent at which the damage was done. In other words, the method used in the study detects the fault and the severity of the fault. The technique used in their study is called the Optimized Stationary Wavelet Packet Transform (OP-SWPT); an advanced form of a digital signal processing technique called Wavelet Transform. Even though Wavelet Transform was used for signal processing, the authors attested to the need to process the signals because of the noise which could emanate from the sensors (Lee *et. al.*, 2020). This can be done via the use of Fast Fourier Transform (FFT) which is used to analyze the signal in the spectral domain. Another signal processing identified by the author is the Short Time Fourier Transform which often times is used when the location to the feature extracted is needed. Lastly, the Wavelet Transform (WT) is used for none stationary signal. Results show that the technique is efficient in detecting different types of bearing faults. Also, it was reported that the technique detected faults faster.

Authors in (Pourpanah *et. al.*, 2018) applied anomaly detection on Unmanned Aerial Vehicle (UAV) motors and propeller. The current signal was used with fuzzy adaptive resonance neural network which is an unsupervised learning scheme to aid early detection of faults in UAV motors. A vibration signal was used to detect anomaly in propellers using Q-learning based Fuzzy ARTMAP neural network. For the selection of subset of features, the Genetic algorithm was used.

Authors in (Egaji *et. al.*, 2020) emphasized on the need to use available data (usually vibration data) to aid avoidance of downtime in industries. In their presentation, as a result of noise in data which results from sensor inefficiency, it was suggested that digital signal processing methods such as FFT could be used to aid better output. In their approach, features were extracted from the data. These features were used to train the neural network. For ease of detection, Principal Component Analysis (PCA) was used to reduce the dimension of the data from 24 to 1 dimensional space. The output from PCA is then used as an input to a regression model which reconstructs the input. The error between the input and the reconstructed output reveals anomaly detection. The regression models used in this presentation includes Support Vector Machine (SVM), Random Forest (RF) and K-Nearest Neighbors (KNN).

To aid proper functioning of industrial machines, industrial fans have been used to aid cooling. However, authors in (Gong *et al*., 2018) presented an online solution to detect anomaly in industrial fans. To achieve this, acoustic signals was used with an intelligent prediction integrated system with the internet. Furthermore, Acoustic Signal

Enhancement Filter and Adaptive Kalman Filter was used for feature extraction and detection.

In the quest to improve the concept of industry 4.0, Neupane and Seok (2020) set up an experimental platform shown in Figure 2.7. In their presentation, vibration data was collected and a machine learning algorithm was used to detect anomaly. The process involved is shown in Figure 2.8. However, it is important to note that signal processing had to be done with the machine learning so as to get efficient result.

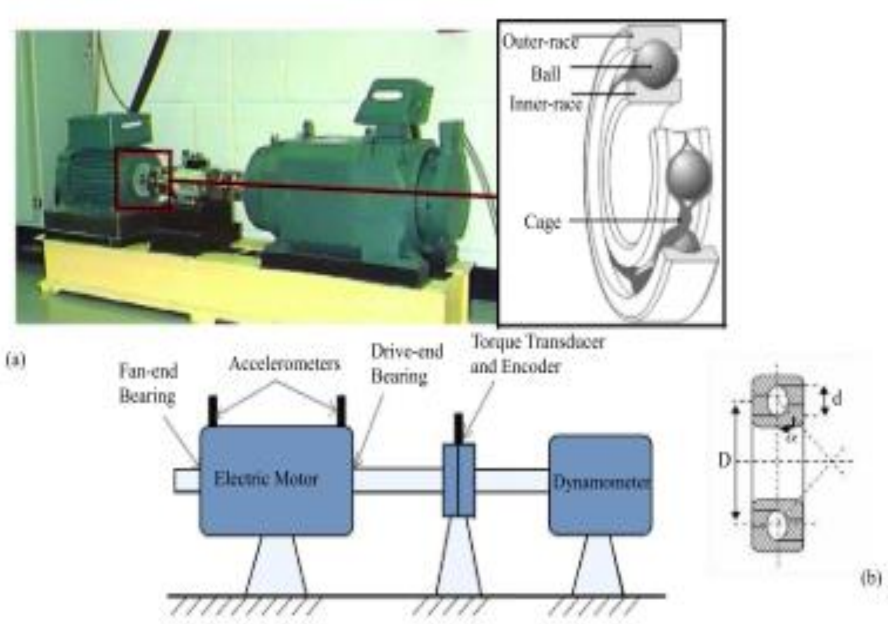

Figure 2.7: (a) Experimental set up of Case Western Reserve University to generate data set. (b) the cross sectional view of the bearing.

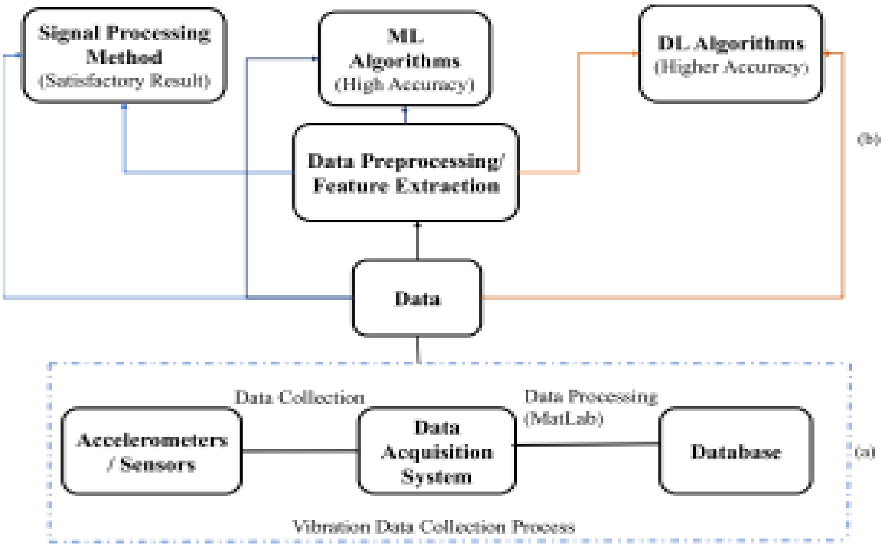

Figure 2.8: Process of detection used by the CWRU

In their presentation, the vibration data gathered was worked on using FFT among other digital signal processing techniques. The reason for choosing FFT is because it can serve for both stationary and none stationary signals.

Sohaib and Kim (2018) presented the use of Complex Envelope Spectra and Stacked Sparse Autoencoder based Deep Neural Network for the detection of anomaly in bearings of machines. To aid the detection, a fault diagnostic scheme was developed so as to overcome the fluctuations of the shaft speed. The detection was made easy via the use of the Complex Envelope Spectra.

Authors in (Malla and Panigrahi, 2019) presented how vibration signals was used to determine bearing failure which may result from lack of lubrication, contamination, inaccurate mounting and dismounting, misalignment, false brinelling, corrosion, electric damage and much more. The method used include time domain, frequency domain and time-frequency domain analysis. From the results, the use of Wavelet Transform with ANN and fuzzy logic gave favorable result.

All these contributions of different researchers are enumerated in Table 2.1, none, to the best of our knowledge did work on the improved sensitivity of LSTM-Autoencoder used for anomaly detection in motor bearing. This therefore justifies the research work.

2.5 Summary Table for the Related Works

Table 2.1: Summary table for the related works

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
1.	Vibration Analysis Techniques for Rotating Machinery and its effect on Bearing.	Khadersab and Shivakumar (2018)	Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT).	FFT was seen to aid better detection of faults.
2.	Vibration analysis and comparison of the damaged and undamaged of rolling ball bearing.	Sulka *et. al.* (2019)	FFT and STFT.	FFT was observed to have more amplitude difference making it easier to detect anomaly.
3.	Application of wavelet analysis and its interpretation in rotating machines monitoring and fault diagnosis.	Abouelanouar *et. al.* (2018)	WT.	The conclusion is that wavelet transform is a powerful tool for anomaly detection.
4.	Detection of Digital Image Forgery using Fast Fourier Transform and Local Features.	Kanwal (2019)	FFT with local texture descriptor together with SVM classifier.	Detection accuracy of anomaly was increased.

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
5.	Technological Evolution in the Fault Diagnosis of Rotating Machinery: A Review.	Barot and Kulkarni (2021)	Review on FFT, WT, DWT, Wavelet Packet Transform (WPT). Furthermore, Deep learning techniques such as KNN, SVM, and ANN was reviewed.	
6.	Machine health management in smart factory: A review.	Lee *et al*. (2018)	Review on analytical models, experimental model, Fuzzy logic, Time domain analysis, Fourier series, Numerical model, Artificial Neural Network and WT to detect anomaly. Furthermore, Big Data, Hybrid algorithms, Machine learning algorithm such as Support Vector Machine (SVM), Statistical model, Neural networks and Empirical mode decomposition was reviewed for the purpose of diagnosis of anomaly.	

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
7.	A Data-Driven Approach for Bearing Fault Prognostics.	Jin *et. al.* (2018)	Kolmogorov-Smirnov test, self-organizing map and unscented Kalman filter	The degradation of bearing and the remaining useful life of the bearing was predicted.
8	A Data-Driven Approach for Bearing Fault Prognostics.	Jin *et. al.* (2019)	Kolmogorov-Smirnov test, self-organizing map and unscented Kalman filter.	The degradation of bearing and the remaining useful life of the bearing was predicted.
9.	Few-Shot Bearing Anomaly Detection.	Shen *et al*. (2020)	A few shot learning approach based on model-agnostic meta-learning.	Detection was made possible with little available data which may not be easy with other approaches with an improved accuracy of 25%.

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
10.	Semi-Supervised Learning of Bearing Anomaly Detection via Deep Variational Autoencoders.	Shen *et al*. (2019)	Cyclic Spectral Correlation (CSC) and Cyclic Spectral Coherence (CSCoh). Also auto encoders were used for prognostics.	Semi-supervised vector detecting device and CSCoh was emphasized as a tool for anomaly detection in bearing.
11.	A Case Study for Improving Maintenance Planning of Centrifugal Pumps Using Condition-Based Maintenance.	Syan *et al*. (2020)	Conditional Based Maintenance (CBM) for centrifugal pumps.	Although, very good classification accuracy was observed, but it was suggested that artificial intelligence should be used for better performance.
12.	Automatic anomaly detection in large sequence.	Boniol *et. al.* (2020)	NorM: a technique used for domain-agnostic anomaly detection	The technique shows high accuracy in detecting recurring anomaly.
13.	Deep Learning Methods Applied to Anomaly Detection in Vehicle Manufacturing and Operation.	Ren (2019)	Deep Neural Network	Though expensive but shows impressive results which overcomes the problem of the use of few data for anomaly detection.

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
14.	Low Latency Anomaly Detection with Imperfect Models.	Nath (2020)	Low latency anomaly detection algorithm, which is based on the framework of Quickest Change Detection (QCD).	The author achieved a reduction in the delay of detection.
15.	Visual explanation of neural network based rotation machinery anomaly detection system.	Saeki (2019)	Convolutional Neural Network (CNN).	The technique was proven to be useful in detection of anomaly.
16.	Localised gear anomaly detection without historical data for reference density estimation.	Schmidt and Heyns (2019)	Continuous wavelet transform, principal component analysis, was used to determine localized faults while Bayesian data analysis technique was used to infer the presence of localized anomaly.	Localization of anomaly was achieved.
17.	Anomaly Detection in a Logistic Operating System Using the Mahalanobis– Taguchi Method.	Asakura *et al.* (2020)	Mahalanobis-Taguchi.	Anomaly detection was achieved in machines tools.

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
18.	Anomaly detection in milling milling tools tools using using acoustic signals and generative generative adversarial networks.	Cooper *et al.* (2020)	Generative Adversarial Networks as Anomaly Detector.	Increase in classification accuracy with about 24.49%.
19.	Anomaly detection for machinery by using Big Data Real- Time processing and clustering technique.	Wang *et al.* (2020)	KNN.	The results shows that the technique detect anomaly in machinery.
20.	Fault Diagnosis of Rotary Machine Bearings Under Inconsistent Working Conditions.	Sohaib and Kim (2019)	Bi-spectrum analysis and Convolutional Neural Network (CNN).	A better detection of anomaly was achieved.
21.	Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches.	Neupane and Seok (2020)	FFT.	FFT proves to be efficient in anomaly detection.

Table 2.1: Summary table for the related works continue …

S/N	Title of paper	Author (s)/Year	Method	Result/Remark
22.	Reliable Fault Diagnosis of Rotary Machine Bearings Using a Stacked Sparse Autoencoder-Based Deep Neural Network.	Sohaib and Kim (2018)	Complex Envelope Spectra and Stacked Sparse Autoencoder based Deep Neural Network.	Complex Envelope Spectra helped to achieve detection of anomaly easily.
23.	Review of Condition Monitoring of Rolling Element Bearing Using Vibration Analysis and Other Techniques.	Malla and Panigrahi (2019)	Wavelet Transform with ANN and Fuzzy logic.	Better detection compared to existing techniques.

From the review, it is observed that researchers either used only digital signal processing or only machine learning. However, from the search no one has improved detection using FFT and LSTM autoencoders as we used in this research.

CHAPTER THREE

3.0 RESEARCH METHODOLOGY

3.1 Overview

This chapter presents the method used to achieve the aim an objective of this research. This includes the use of National Aeronautics and Space Administration (NASA) data on motor bearings, the use of FFT, LSTM-Autoencoder technique to achieve anomaly detection. To achieve this, Python was used as a tool to write the code. Also, the work flow in Figure 3.1 was followed. In the work, the data was collected, preprocessed with pandas. Furthermore, it was then processed with FFT. To aid detection, the model was then developed. The structured data characterized with normalcy was then fed into the model and was trained. Afterwards, the model was tested with abnormal data and the performance of the model was then evaluated using two metrics; threshold also known as anomaly score and accuracy.

`

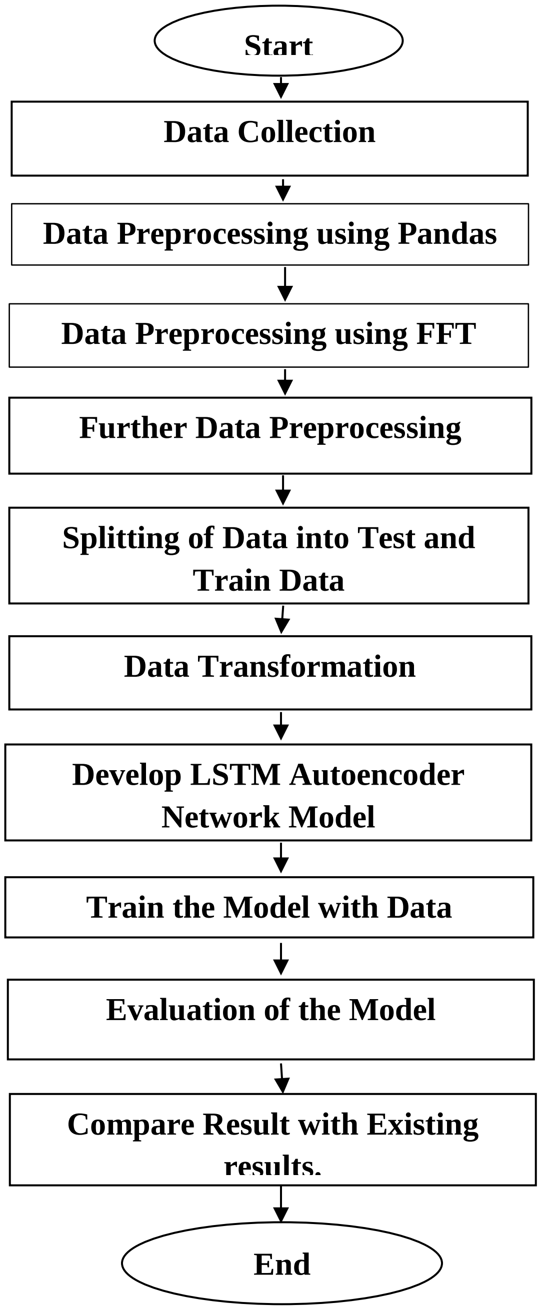

Figure 3.1: Model Diagram.

3.2 Data Collection

The data used in this research is sensor reading obtained from vibration sensor fastened to the motor bearing by NASA Acoustics. These data were generated from four bearings that were run till failure with constant load. The data which was taken at 10-minutes interval contains 20800 data points per bearing read at sampling rate of 20 kHz.

Plate I shows part of the data collected in files named with dates of collection. Opening the first file, the content within similar to others is shown in Plate II.

- 2004.02.12.10.52.39
- 2004.02.12.11.02.39
- 2004.02.12.11.12.39
- 2004.02.12.11.22.39
- 2004.02.12.11.32.39
- 2004.02.12.11.42.39

Plate I. Raw data collected in files named with date time stamp.

0.112	-0.010	-0.005	0.022
0.049	-0.012	-0.046	0.020
-0.027	-0.017	-0.012	0.098
-0.110	0.029	0.112	0.056
0.007	-0.024	-0.044	0.054
0.122	0.051	-0.007	-0.012
0.134	0.007	0.034	-0.017
-0.015	-0.027	0.002	0.027
-0.168	-0.037	-0.107	0.042
-0.061	-0.090	0.020	0.046
0.012	-0.059	0.100	0.012
0.027	0.061	-0.068	-0.032
-0.056	0.002	-0.081	0.022
-0.115	-0.049	-0.044	0.056
0.015	0.042	-0.046	0.005
0.100	0.049	-0.125	-0.020
0.088	0.076	0.020	-0.015
-0.012	0.056	-0.049	-0.071
-0.049	0.073	-0.015	-0.027
0.068	0.100	-0.051	0.051
0.103	0.232	0.012	-0.002
-0.007	-0.005	0.142	-0.110
-0.032	0.005	-0.027	-0.110

Plate II. Unorganized data before processing within the file for just one file.

3.3 Data Preprocessing

The data used in this work was from four motor bearing. The vibration data obtained was collected for each of the bearing, tagged in different files named with dates of collection. However, to keep the data together, pandas was used to bring the four data together as shown in Plate III after all the dependencies (Libraries) were imported. Furthermore, the data was quarried to verify if there are missing data. Also, the data type and the shape of the data was checked so as to ensure that the data conforms to the format which is required for the deep learning algorithm.

	Bearing 1	Bearing 2	Bearing 3	Bearing 4
2004-02-12 10:52:39	0.060236	0.074227	0.083926	0.044443
2004-02-12 11:02:39	0.061455	0.073844	0.084457	0.045081
2004-02-12 11:12:39	0.061361	0.075609	0.082837	0.045118
2004-02-12 11:22:39	0.061665	0.073279	0.084879	0.044172
2004-02-12 11:32:39	0.061944	0.074593	0.082626	0.044659

Plate III: Dataset of the motor bearing after preprocessing

3.4 Data Preprocessing using FFT

Fast Fourier Transform is a technique which employs the transformation of time sequence signals to their frequency domain so as to extract latent behavior of the signal source. In other words, the FFT decomposes 'N' points time domain signal to 'N' time domain signal composing of a singular point. Afterwards, the FFT algorithm calculates the 'N' frequency spectral which is corresponding to 'N' time domain. Lastly, the 'N' spectral will then be synthesized in one frequency spectrum. This algorithm is guided by the equation:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{i2\pi kn}{N}} \qquad (1)$$

Where N is is the size of the domain and $X_k = X_0, X_1 \ldots\ldots X_{N-1}$ is converted to another sequence number $x_n = x_0, x_1 \ldots\ldots x_{N-1}$ is the signal. In other words, x_n sinusoid with frequency of $\frac{k}{n}$ is a cross correlation sequence of X_k. Furthermore, it must be noted that the 'N' points must be in the form of 2^n. This means the N time points must be within this range else it won't be able to capture the whole data. During the decomposition, the levels of decomposition is given as $log_2 N$. For example, a 16-point signal also represented as (2^4) is broken down into x stages where $x = log_2 2^4$. This results to a break down into 4 stages. During the break down, the original samples are re-ordered via bit reversal. Table 3.1 shows a sample re-ordering.

Table 3.1: **Data re-ordering table**

Normal sample	Binary representation	Decomposed outcome	Binary representation
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0110
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011
14	1110	7	0111
15	1111	15	1111

Furthermore, the algorithm then finds the frequency spectrum of 1 point time domain which is equal to itself. The last thing the algorithm did was to combine the N frequency spectral in the reverse order of the time domain decomposed into. This combination, results into the frequency spectrum of the time domain signal. All this is represented in the following pseudocode:

Fast Fourier Transform (FFT) Algorithm

//For general case let the input G of any case have a sequenced

// $G = (a_0, a_1 ...., a_{N-1})$

//Note that N is a power of 2. Also, we want to return output values of H given as:

// $H = A(x) = \sum_{j=0}^{N-1} a_j x^j$

//where H is a polynomial similar to equation 1, evaluated at Nth root of unity and
//a is the coefficient of the polynomial.

if $N = 1$ then return (a_0)

if $N > 1$ then //calling the Fourier transforms recursively.

$(s_0, s_1, ..., s_{n/2-1}) = FFT((a_0, a_2, ..., a_{N-2})w^2)$ // this deals with even sequence.

$(s'_0, s'_1, ..., s'_{n/2-1}) = FFT((a_1, a_3, ..., a_{N-1})w^2)$ // this deals with odd sequence.

for $j = 0$ to $\frac{N}{2} - 1$

$r_j = s_j + w_N{}^j s'_j$

$r_{j+\frac{N}{2}} = s_j - w_N{}^j s'_j$ // the negative sign is from the odd.

//Note that w is the primitive Nth root of unity if $w^0, ...w^{N-1}$ are root
//of the unity

return $(r_0, r_1, ..., r_{N-1})$

end for
end if
end if

//How long will it take to do this computation?

$T(N) = 2T(N/2) + O(N)$

//N is the size of the problem and O(N) is the order of N as solved in the equation
//above.

// $= O(N \log_N)$

//which is much better if the system is to run for $O(N^2)$.

The use of this technique aids fast computation and the extraction of salient characteristics in the signal of the data collected.

3.5 Further Data Preprocessing

To ensure that any data point doesn't look so odd from the others, a further preprocessing was done to normalize the data. The normalization process ensures that all data points is converted to numbers within 0 to 1. Furthermore, since LSTM usually uses a three-dimensional tensor form of data, it is important to reshape the data which is two (date time stamp, feature) dimensional to three dimensional (date-sample, time-sample, features).

3.6 Data Splitting

To aid the use of autoencoder network for the purpose of anomaly detection, the data has to be subdivided to train and test data. Usually, using the autoencoder model, there are two ways in which the data can be split into test and train dataset. One way is to get all the data set characterized with both normal and abnormal data and then split it the traditional way using ratios which can be 60% training data 40% test data. Some other researchers use 70% training data and 30% test data. Basically, the ratio of split it based on the researcher's instinct. The challenge with this approach is that it is used for labeled dataset. The second method is the use of only the normal data to detect anomaly. This is the approach used since the data used is unlabeled. This leaves the data with faulty characterization as the test data. In this research, the train dataset was obtained after a plot of the data was observed to get both the normal and faulty signal.

3.7 Model Description

As intended, the technique used in this research after data clean up via FFT remains LSTM Autoencoder. To be able to use this, it is important to understand the concept.

3.7.1 Autoencoder

Autoencoder is a neural network that aid the replication of the input at the output. The model which consists of an encoder with an encoding function E(x) and a decoder with a decoding function of D(x) first compresses the input into a latent space. Afterwards, the input is recreated at the output via the decoder, Figure 3.2 and Figure 3.3 shows the architecture of the model. As a result of its use, the model is used for the detection of anomaly. To do that, the output sequence is compared to the input sequence to see if there is an error in the replication. To detect anomaly, a threshold also known as anomaly score is used as a bench mark for allowable error. Beyond the threshold, anomaly is flagged. For this reason, it is used for speech recognition and face recognition.

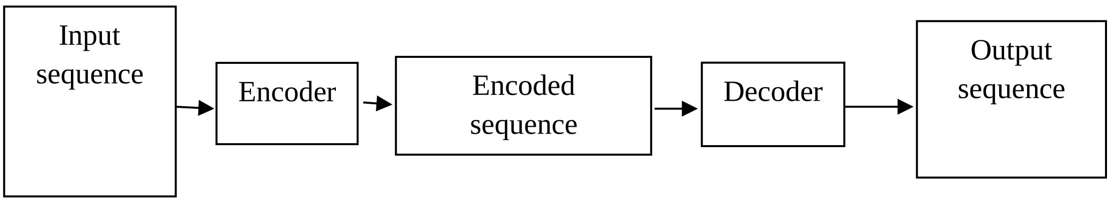

Figure 3.2: Architecture of an autoencoder model.

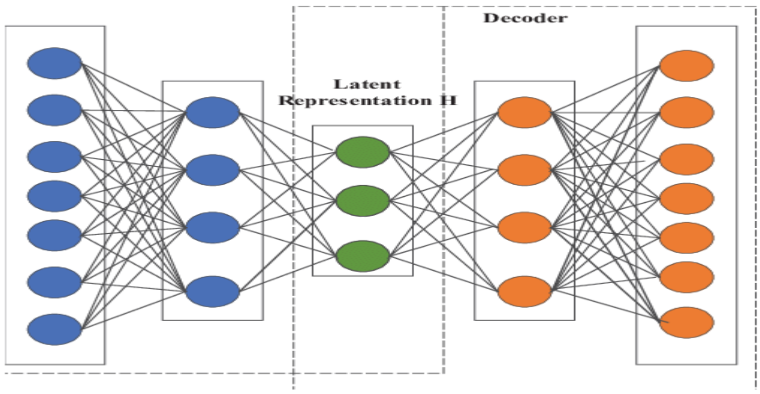

Figure 3.3: Representation of an autoencoder (Mac *et al.*, 2018)

If the encoder and decoder transition is represented as α and β respectively. The transition can be represented as α: $X \rightarrow F$ this shows the transition of series data X transformed to a compressed data F which is reconstructed by β: $F \to X$ Therefore, the encoder decoder transition can be represented as $\alpha, \beta = \arg\min_{\alpha,\beta} \|X - (\alpha \circ \beta)\|^2$. In other words, if $x \in \mathbf{R}^d = X$ at the input is to be reconstructed at the output, it will first be mapped to $\hbar \in \mathbf{R}^p = F$ where $\hbar = \gamma(\mathsf{W}x + \mathsf{b})$. γ is the activation function, W is the weight and the bios is b at the encoder. The output is then reconstructed as $x' = \gamma(\mathsf{W}'\hbar + \mathsf{b}')$ at the decoder. The process is presented in the pseudocode below:

Autoencoder Algorithm

```
//INPUT: Normal dataset X
//V = x^(j)  j = 1,2....N  this is the abnormal data sequence
//Considering a threshold µ
//OUTPUT: reconstruction error ||x = x̂||
//Let the factors for training the autoencoder be ϕ and ŋ
for j =0 to N do
      reconstruct the input at the output as g_ϕ(f_ŋ(x^j))
      reconstruction error (j) = ||x^(j) − g_ϕ(f_ŋ(x^j))||
      if reconstruction error > µ then
            x^(j) is an anomaly
            else
            x^(j) is not an anomaly
      end if
end for
```

3.7.2 Long Short-Term Memory Network (LSTM)

LSTM is a form of Recurrent Neural Network (RNN) that aids the retention of long term dependent variables between data at given time in past records (Nguyen *et al*., 2020). The neural network which is characterized with three control gates such as input gate, output gate and forget gate obtained from sigmoid neural net layer and point wise multiplication, exists, as a form of chain repeated module of neural networks. Note that since RNN is used for time series vector, unlike the traditional neural network that cannot handle sequence data that relate with each other both past and present, the RNN can help handle these sequence data. In other words, traditional neural network which is to produce y when x is inputted, does so and will never use y again. This makes it forgetful. However, this is not so for RNN as it can still use past data in the future giving the deep learning model its own memory. This means that the output of RNN is a function of past data as shown in Figure 3.4:

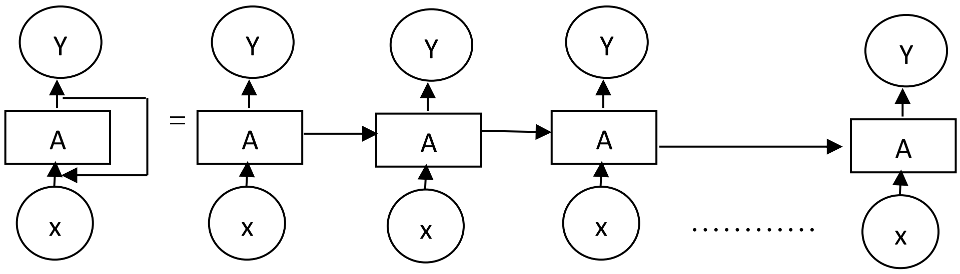

Figure 3.4: RNN network.

However, as a result of the network learning every single detail in the sequence, a problem called varnishing gradient occurs. This means when the network learns all, it will forget because its weight becomes too small for learning to occur. This is what necessitated the development of RNN.

LSTM, a form of RNN as shown in Figure 3.5 has the ability to read data sequentially as vectors $s = \{s_1, s_2, s_3 ..., s_t .....\}$ where $s_t \in k^x$ represents vector readings of x having x dimensions at time t.

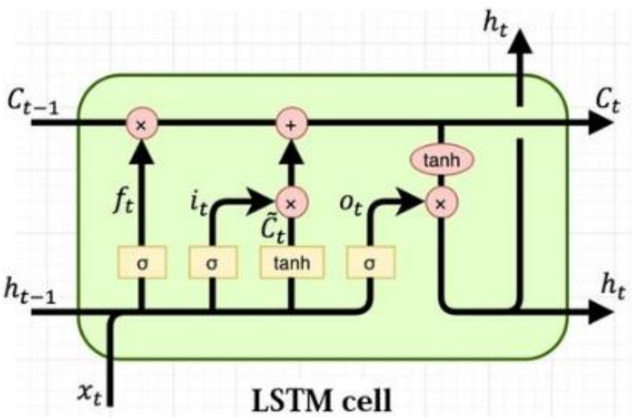

Figure 3.5: Symbolic representation of LSTM (Varsamopoulos *et al*., 2018)

This neural network consists of four gates that helps it to store relevant information and forget irrelevant ones. These gates include the forget gate, remember gate, learn gate and use gate.

The principle of operation of the model is thus: The transport system through which the sequence travel through the neural network is called a cell state (C_t). This helps to move information all through the sequence chain as information gets added or removed along the journey. This is achieved via the sigmoid function (σ) which is at the forget gate. The sigmoid function scales the information from 0 to 1. Whichever information is closed to 0 is to be forgotten. While those close to 1 are to be remembered. Therefore as (X_t) is made available at time t, the network passes through the sigmoid activation function to decide whether the information is important or not. The model decides what old information is to be forgotten by outputting the number between 0 and 1. In other words, when signal (X_t) and the previous hidden state (h_{t-1}) is inputted into the model, it passes through the forget gate which uses the sigmoid function given in equation (2).

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f \qquad (2)$$

Where, f_t is the forget gate function, (h_{t-1}) is the previous hidden state, W_f is the weight of the function, (b_f) is the bias vector parameter the subscript f is the forget gate and t time. This equation checks if this new information outweighs old information. If it is close to 1 as discoursed, it will forget old information and retain the new one. However, if not, it will retain the old information and discard the new information. The retained value is then used to create the input value as expressed in equation (3).

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i \qquad (3)$$

where i is the input. This is then used to form the candidate value (Ĉ) using the tanh function as shown in equation (4).

$$\hat{C}_t = tanh(W_c \cdot [h_{t-1}, X_t] + b_c \quad (4)$$

Where c is the cell state. Afterwards, the new cell state will be updated with the old cell state, forget gate value, input value and candidate value as shown in equation (5).

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (5)$$

Finally, an output has to be decided. This will be based on the cell state. To do this, a sigmoid function runs to decided what part of the cell state needs to be outputted. This is then multiplied by the output of the tanh function of the candidate value. This is expressed in equation (6) and (7).

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

Where o is the output and h_t is the new hidden state. In summary, three parameters are passed to the model. This includes the hidden state (h_t), previous cell state and the input data (X_t). The hidden state and the input data are combined together and are fed to the forget layer. Here relevant data are remembered and irrelevant once are forgotten. As a result of this, a candidate layer is created, this layer contains variable values that are added to the cell state. The combine is then fed into the input layer after which the information in the combine is needed to be added to the new cell state. After the computations of the forget layer, input layer and the combine layer, the cell sate is computed using the vector and the previous cell state. This is now used to compute the output. All these are described in the pseudocode below:

Long Short-Term Memory (LSTM) Algorithm

```
// pseudo code for LSTM
//Input pct = previous cell state, phst= previous hidden state, x=input
Def lstm(pct, phst, x):
               Cm=phst+x // Cm is combine
               Forget = fl(Cm) // fl is the forget layer
               candidate= cl(Cm) // cl is the candidate layer
              x= inputlayer(Cm)
              cell_state=pct * Forget * candidate * x
              out_put= outputlayer(Cm)
              ht=out_put*tanh*  cell_state
              return ht and cell_state.

cell_state=[0, 0, 0]
ht=[0. 0. 0]
for x in x:
            cell_state, ht= lstm(cell_state, ht, x)
end for
```

3.7.3 LSTM Autoencoder

Having known the use of an auto encoder as a model that aid the detection of anomaly via the compression of large data into a smaller vector space, it is important to note that the use of this model alone may be challenging as time series data which is used in the model will with time get large and the computation will be cumbersome even though not all information in the time series data is needed. To solve this problem, it is important to use the LSTM with the autoencoder to give LSTM-Autoencoder. The LSTM will aid the learning and keeping of important part of the data and forget the irrelevant part of the data. This will reduce the amount of data consumed by the model. To create this model, the LSTM is used after the encoding in the autoencoder. The representation of LSTM autoencoder is shown in Figure 3.6. In the illustration, the input data which is a time series data x_u, x_d is fed into the autoencoder for compression to Z . This compressed data is then fed into the LSTM which helps to select the most important features of the data to be remembered. This help to reduce the amount of data needed for reconstruction further reducing computation. During reconstruction, a threshold is selected. If the error is above a certain point, anomaly is detected.

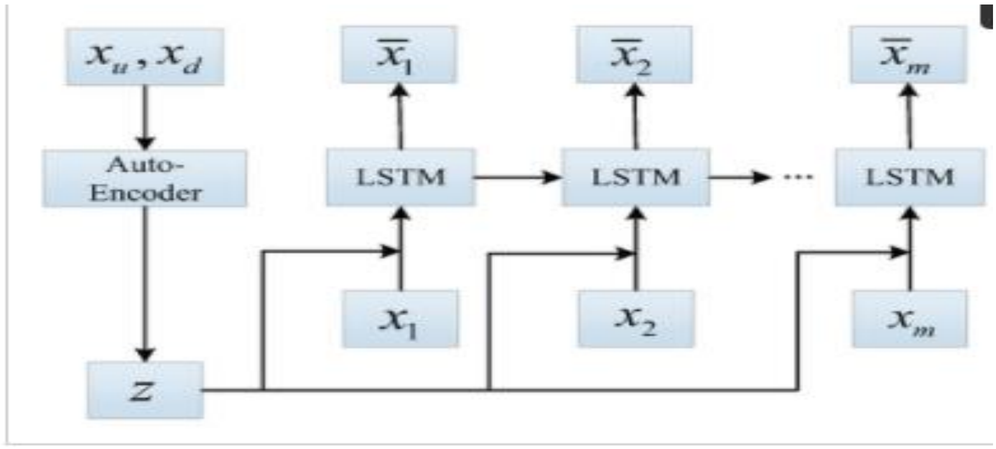

Figure 3.6: LSTM-Autoencoder model diagram (Wei *et al*., 2019).

All the process and equations of the LSTM autoencoder is presented in pseudocode by Wei *et al*. (2019) as shown below:

$$L = \frac{1}{2}\sum_{s} \|s - s^\wedge\| \times \|s - s^\wedge\| \quad (8)$$

LSTM Autoencoder Algorithm

```
//INPUT: the training set Xu, //Xu is the dataset
//OUTPUT: prediction result X.
//Preprocess the data Xu to give Xuu
//Split Xuu to AutoEncoder training set XAE and XAD validation dataset.
//Initialize the weight matrices of AutoEncoder randomly.
//Put XAE into AutoEncoder.
if L(X, Y) < d then
	Calculate the error L(X, Y)
	Use the back propagation training the AutoEncoder.
else
	End the training.
end if
//Generate the characteristics of the input dataset Zt.
for t = 0 to epoch do
	Put Zt into the LSTM, and do for forward propagation.
	Generate output
	Calculate error.
	Use the back propagation to update parameters.
	Use forward propagation to update network status ht,
end for
//Add LSTM after the encoder of AutoEncoder to form AE-LSTM.
//Fine-tuning the whole network, training initialization parameters.
//Input XAD test data in AE-LSTM to generate the predicted value X.
Return X.
```

3.8 Development of LSTM Autoencoder Network with FFT

The model used in this research is the LSTM autoencoder with FFT. The LSTM autoencoder was used so as to aid the detection of anomaly of time series data. The FFT technique also was used to extract latent feature of a signal in a frequency spectrum. This can be seen in the pseudocode below:

Model Algorithm (LSTM Autoencoder with FFT)

```
//INPUT: the training set Xu, // Xu is the dataset
//OUTPUT: prediction result X.
//Preprocess the data Xu to give Xuu
//process the data using FFT to give Xuu’
//Split Xuu’ to AutoEncoder training set XAE and XAD validation dataset.
//Initialize the weight matrices of AutoEncoder randomly.
//Put XAE’ into AutoEncoder.
if L(X, Y) < d then
       Calculate the error L(X, Y)
       Use the back propagation training the AutoEncoder.
else
       End the training.
end if
//Generate the characteristics of the input dataset Zt.
for t = 0 to epoch do
       Put Zt into the LSTM, and do for forward propagation.
       Generate output
       Calculate error.
       Use the back propagation to update parameters.
       Use forward propagation to update network status ht,
end for
//Add LSTM after the encoder of AutoEncoder to form AE-LSTM.
//Fine-tuning the whole network, training initialization parameters.
//Input XAD’ test data in AE-LSTM to generate the predicted value X.
Return X.
```

3.9 Training and Testing

The model was then trained with normal ball bearing data using 100 epochs at batch size of 10. Afterwards, it was tested with data characterized with anomaly. The training loss and loss distribution was computed and plotted so as to serve as a guide to select a threshold. After the threshold was selected, the bearing fail plot was presented.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Overview

This chapter presents the results of the research. This includes the description of the data, the preprocessing of the data, conversion of the data from time domain to frequency domain and the feature extraction from the data. Afterwards, the LSTM autoencoder network was built and the loss computed. From the loss distribution, the threshold was computed and the anomaly observed graphically.

4.2 Data Description

The data used was obtained from NASA repository. The data contains four files that contain 20800 data point each read at a sampling rate of 20 kHz. Plate IV below shows the characteristics of the data.

	Bearing 1	Bearing 2	Bearing 3	Bearing 4
count	982.000000	982.000000	982.000000	982.000000
mean	0.080951	0.078543	0.081351	0.047830
std	0.040200	0.011789	0.011607	0.009549
min	0.001168	0.000767	0.000716	0.001699
25%	0.060773	0.074240	0.076829	0.043951
50%	0.062021	0.075206	0.078187	0.044524
75%	0.083277	0.077458	0.080575	0.048130
max	0.453335	0.161016	0.151299	0.119047

Plate IV: Description of the dataset.

From the description, it is observed that the data has a shape of 982 by 4 with a mean between 0.048 to 0.081. The standard deviation of the dataset is within 0.009 to 0.040, the minimum value captured is 0.0007 and the maximum is 0.453.

4.3 Data Preprocessing

The data gathered may naturally come with some irregularities which makes it unfit for direct use as shown earlier in Plate I and Plate II. For this reason, there is need for data pre-processing. During the process of preprocessing, first, the dataset which comes in numerous files labeled according to date time stamp was merged. This merging, featured in Plate V helps to gather the data in one file so as to be used in the proposed model. In the merged files, it is observed that the data was captured with date time stamp which qualifies it to be a sequence data and hence can be used for LSTM autoencoder.

	Bearing 1	Bearing 2	Bearing 3	Bearing 4
2004-02-12 10:52:39	0.060236	0.074227	0.083926	0.044443
2004-02-12 11:02:39	0.061455	0.073844	0.084457	0.045081
2004-02-12 11:12:39	0.061361	0.075609	0.082837	0.045118
2004-02-12 11:22:39	0.061665	0.073279	0.084879	0.044172
2004-02-12 11:32:39	0.061944	0.074593	0.082626	0.044659

Plate V: Data representation of the four bearings merged into one file.

Furthermore, the data was divided into test data and train data. Note that the train data is the data of the normal working bearing. Also, the split was done so that in the validation dataset, both the normal and the anomaly is captured so as to observe the point when the anomaly occurred. While the training data is mainly the normal data. However, from the raw data, it is not possible to observe the normal data and the point of abnormality. Also, since the data may be characterized with noise, FFT was employed to clean up the data and to observe the point suspected as anomaly.

4.4 Data Transformation using FFT

The need for frequency domain signal cannot be overemphasized as it exposes latent characteristics of the signal. To achieve this, FFT was used.

From Figure 4.1a below, it is difficult to deduce if there are anomaly because of the random nature of the signals as mentioned before. As observed in Figure 4.1b the normal data signal (Training data set) is less random and therefore no suspicion of anomaly.

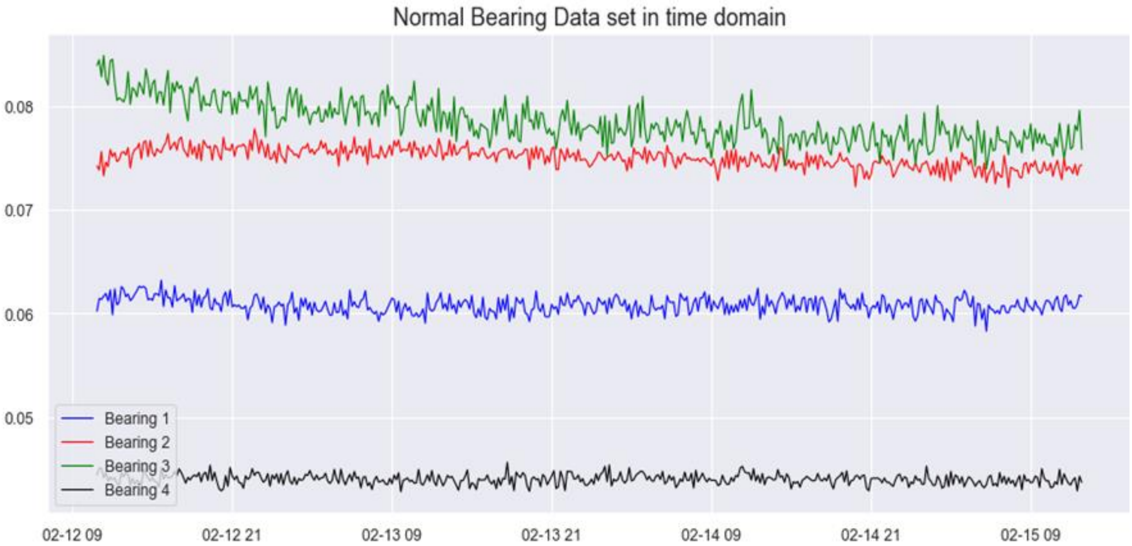

Figure 4.1a: Graphical representation of the normal data (train data set) in time domain.

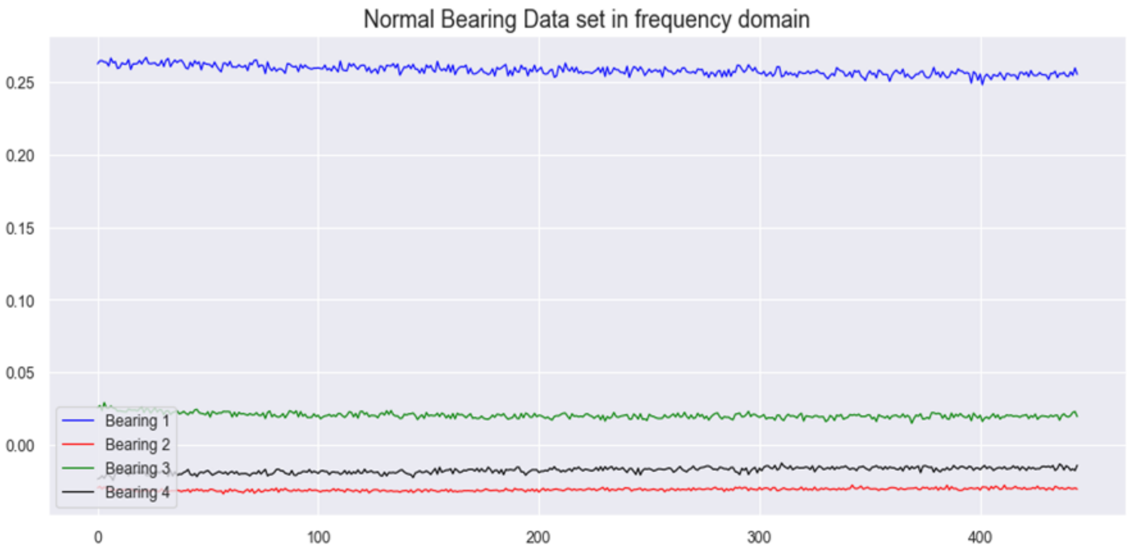

Figure 4.1b: Graphical representation of the train data set in frequency domain.

Also, in Figure 4.2a below, although there are signs of anomaly in Bearing 1 and Bearing 4. However, it is difficult to ascertain where the anomaly of Bearing 2 and Bearing 3 starts from. This is not same with the abnormal data set (Test data set) in frequency domain shown in Figure 4.2b. This therefore validate the need of the frequency domain using FFT to transform the signals.

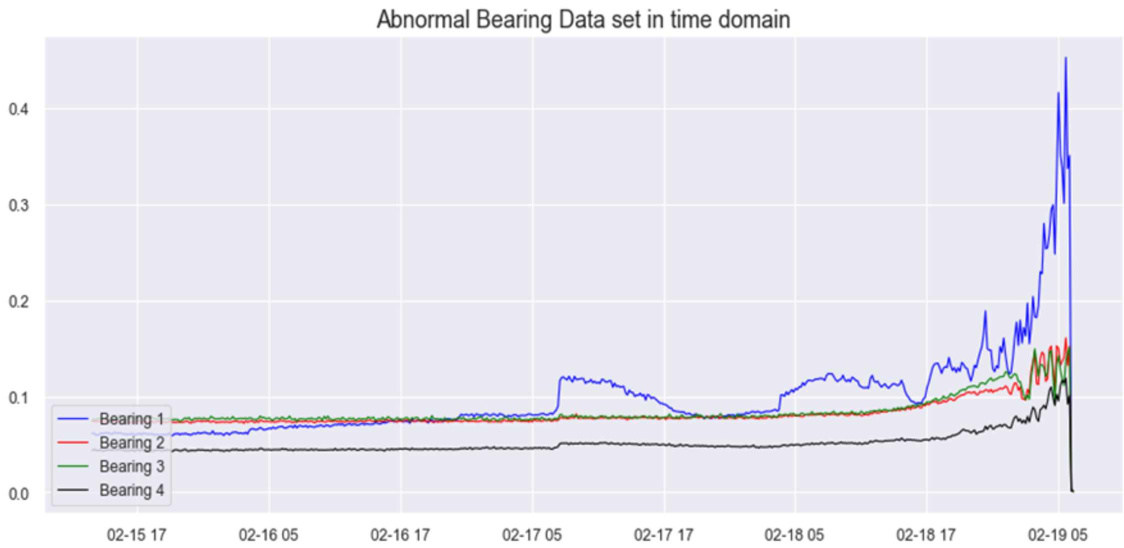

Figure 4.2a: Graphical representation of the abnormal data (test data set) in time domain.

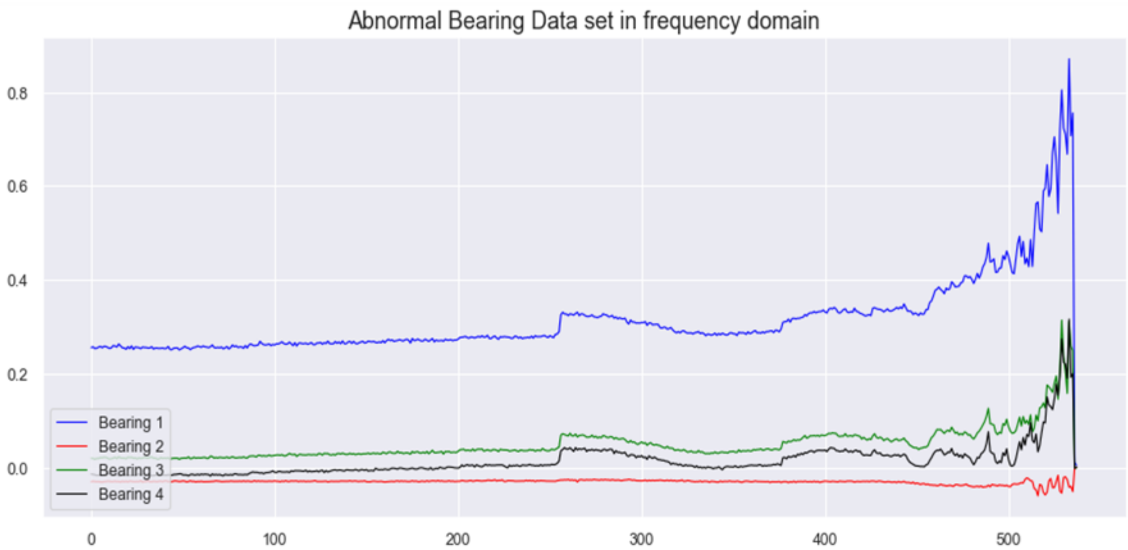

Figure 4.2b: Graphical representation of the test data set in frequency domain.

4.5 Further Preprocessing

Since the data to be fed into the LSTM Autoencoder is a three-dimensional data, there is need to transform the data from two dimension shown in plate VI to a three-dimensional data. This was achieved as shown in plate VII.

```
Dataset shape: (982, 4)
```

Plate VI: Shape of the data originally collected.

```
Training data shape: (445, 1, 4)
Test data shape: (538, 1, 4)
```

Plate VII: Shape of the data generated from the two-dimensional data originally collected.

4.6 LSTM Autoencoder Network Used

After the reshaping of the data as required by the LSTM autoencoder, the model was built and the data fed into it. The structure of the model is shown in Plate VIII below with LSTM layers taking the shape of an autoencoder.

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 1, 4)              0
_________________________________________________________________
lstm_11 (LSTM)               (None, 1, 200)            164000
_________________________________________________________________
lstm_12 (LSTM)               (None, 25)                22600
_________________________________________________________________
repeat_vector_3 (RepeatVecto (None, 1, 25)             0
_________________________________________________________________
lstm_13 (LSTM)               (None, 1, 25)             5100
_________________________________________________________________
lstm_14 (LSTM)               (None, 1, 200)            180800
_________________________________________________________________
time_distributed_3 (TimeDist (None, 1, 4)              804
=================================================================
Total params: 373,304
Trainable params: 373,304
Non-trainable params: 0
```

Plate VIII: The summary of the LSTM Autoencoder network.

From the Plate VIII above, the dimension of the input data to be trained is [1, 4]. At the first layer of the LSTM network, as intended, it is observed that the number of nodes is 200. This then further reduces to 25 nodes at the next layer. This layer is repeated as the encoder layer of the autoencoder. Furthermore, the number of nodes at the decoding layer which aids reconstruction of the data is 200. At the end of the training, the output yields a [1, 4] dimensional data which is a replica of what is at the input layer. Generally, the number of nodes was used so as to achieve lower threshold which is one of the objectives of the research.

4.7 Loss Model

To evaluate the performance of the model, the loss is computed. This was done when the network was trained at 100 epochs. Figure 4.3 shows how much the training dataset is different from the validation dataset.

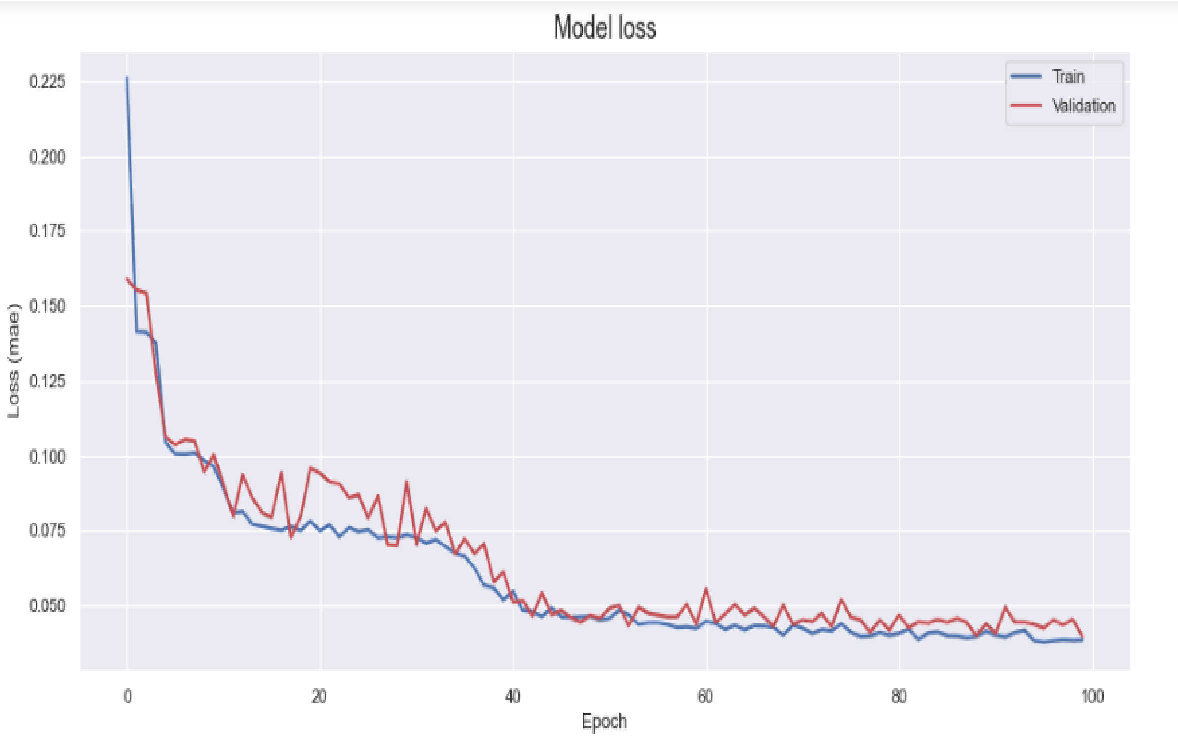

Figure 4.3: Graphical representation of Loss Model.

4.8 Loss Distribution

This parameter is needed so as to be able to calculate the threshold. Figure 4.4 shows the loss distribution plot. From the plot it is observed that the threshold can be below 0.126.

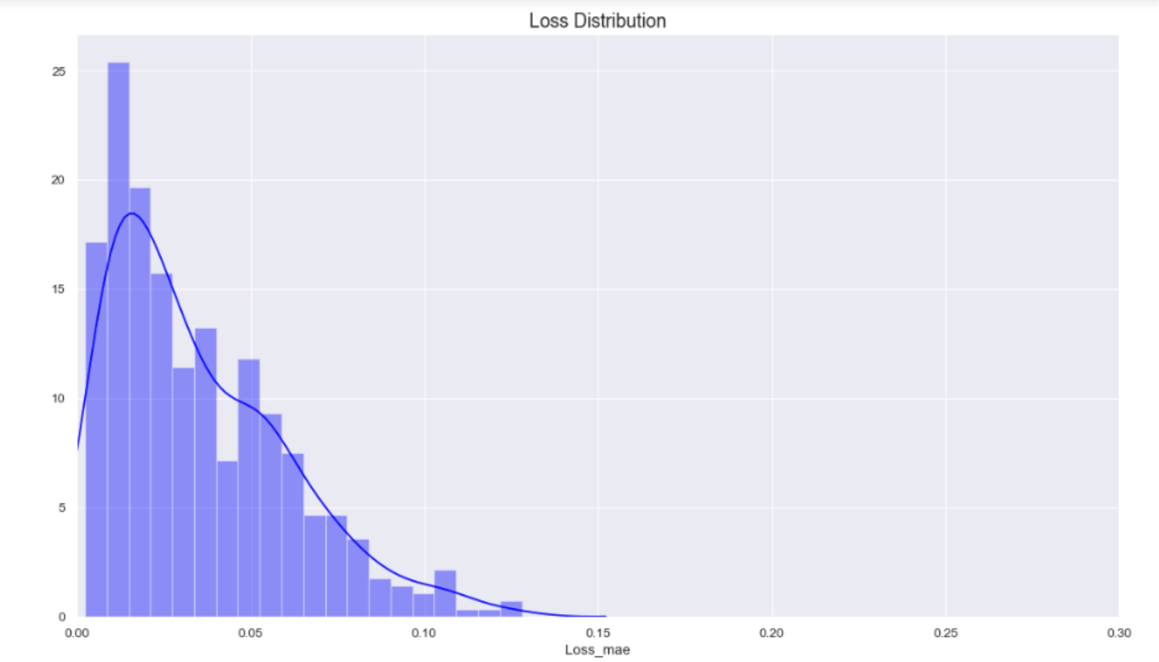

Figure 4.4: Graphical representation of loss distribution.

From the above, it is observed that a lower threshold can be selected based on the plot. Therefore, a threshold of 0.126 was chosen based on the result of the plot.

4.9 Anomaly Detection

The figure 4.5 below shows that, at the selected threshold the model was able to detect anomaly as the red line is the threshold. Every signal above the red line is considered anomaly.

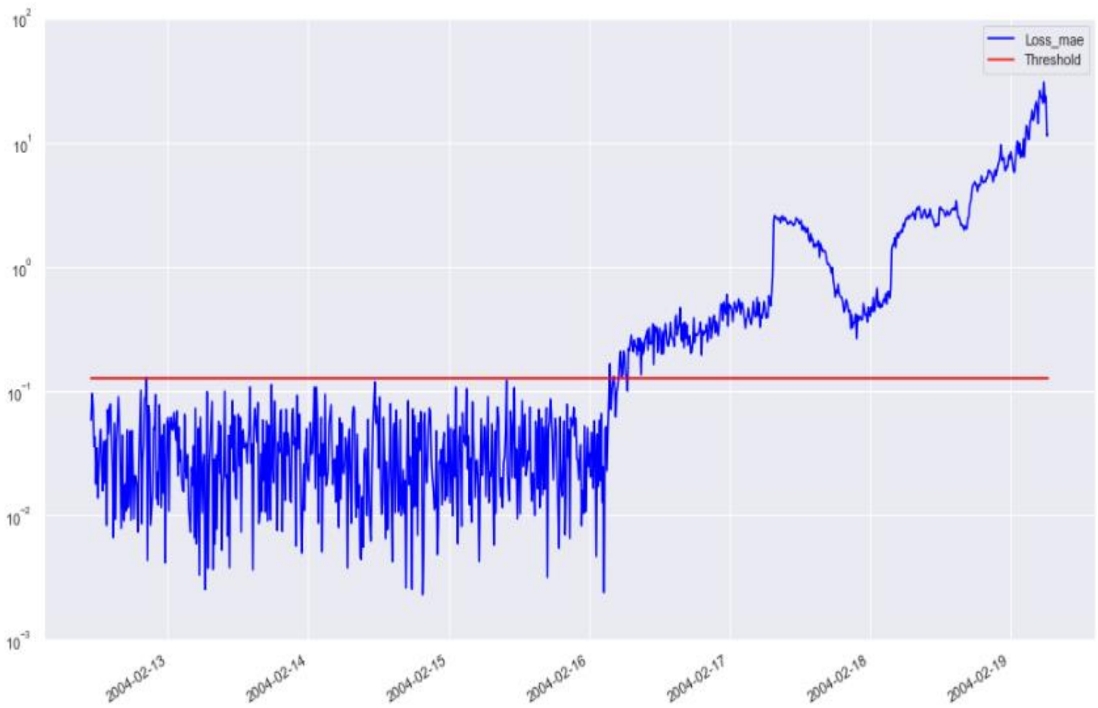

Figure 4.5: Graphical illustration of anomaly detection.

4.10 Comparison of Results

As shown in Figure 4.6 below the threshold observed by authors in (Ahmad *et al*., 2021) using similar dataset with LSTM autoencoder without the use of digital processing techniques is above 0.2. This is also described as the anomaly score. In their work, each dataset (RM2, RM3, RM4, RM5) was treated separately and various anomaly score was observed. However, comparing it with the method adopted in this research, it is observed that the use of FFT reduces the threshold to 0.126. This means the use of FFT with LSTM-Autoencoder is capable to detect weak signals compared to the use of the same model without digital signal processing. Furthermore, compared to existing methods reviewed by authors in (Sohaib and Kim, 2018), the model used in this research tends to perform competitively well in terms of accuracy. Figure 4.7 below shows the comparison of the existing methods with the method in this research. Compared to Virtual Spectrum Imaging (VSI), Artificial Neural Network (ANN) and Stacked Denoising Autoencoder, LSTM autoencoder with FFT performs better in terms of accuracy. However, its performance is low compared to Sparse Stack Autoencoder (SSAE).

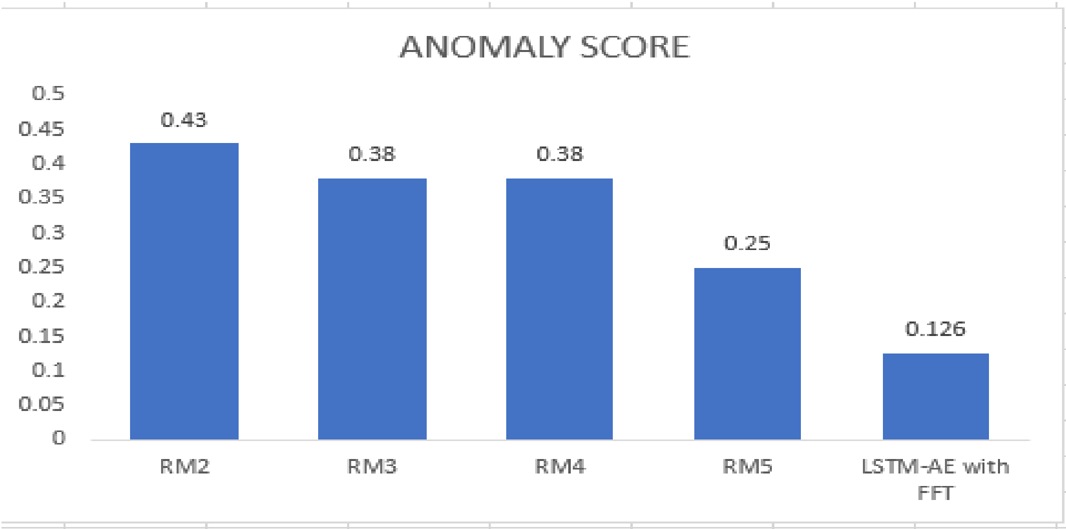

Figure 4.6: Threshold (anomaly score) of bearing data.

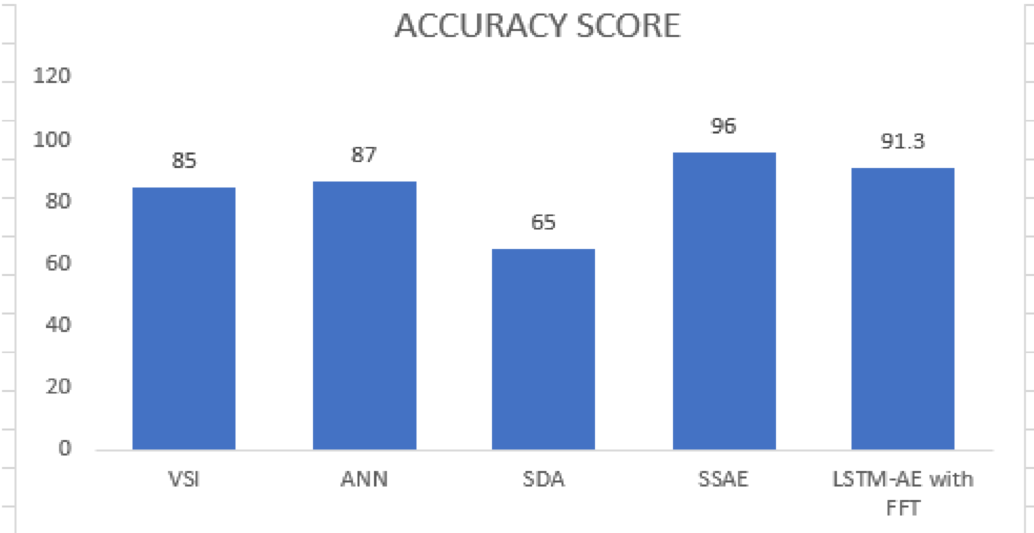

Figure 4.7: Graphical representation of the accuracy score.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 Summary

From this research work, it is observed that lower threshold of 0.126 and even below this can be selected to detect anomaly in motor bearing compared to the various threshold (anomaly score) observed by authors in Ahmad *et al*. 2021 using similar dataset with same LSTM autoencoder but without the use of digital signal processing techniques.

5.2 Conclusion

In this research, the focus has been to detect anomaly in motor bearing. Although, several researches has been done in terms of finding techniques which will aid the detection of anomaly better, however, the aim of the research is to detect such anomaly in weak signals. To achieve this, the LSTM autoencoder with FFT was employed. The FFT was used to clean up and transform the data so as to reduce complex multiplications, aid faster computation and diagnose if anomaly exist while the LSTM autoencoder was used to validate and detect the anomaly before it happens. As a result of the use of **Fast Fourier Transform** (FFT), the technique used **LSTM Autoencoder** was able to detect anomaly at low threshold of 0.126 compared to every other techniques that has been used. Also, the technique in this research work tends to compete favorably with existing techniques as discoursed in previous chapters.

5.3 Contribution to Knowledge

The research work done has been able to contribute to knowledge by hybridizing FFT and LSTM-Autoencoder to improve the sensitivity in detecting anomaly in motor bearing.

5.4 Recommendation

Since Sparse Stack Autoencoder appears to be better in terms of accuracy, it is recommended that FFT could be used with the technique to see if there can be improvement in terms of early detection at a lower threshold and better accuracy.

5.5 Future work

In the future, aside detection, localization of the detected anomaly can be achieved via the use of machine learning and also statistical models. This is to aid fast maintenance process.

REFERENCES

Abid, F. B., Sallem, M., and Braham, A. (2019). Optimized SWPT and Decision Tree for Incipient Bearing Fault Diagnosis. *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 231–236.

Abouelanouar, B., Elamrani, M., Elkihel, B., and Delaunois, F. (2018). Application of wavelet analysis and its interpretation in rotating machines monitoring and fault diagnosis . A review. *7*(4), 3465–3471. https://doi.org/10.14419/ijet.v7i4.15452

Adewumi, A. O., and Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of Systems Assurance Engineering and Management*, *8*, 937–953. https://doi.org/10.1007/s13198-016-0551-y

Ahmad, S., Styp-rekowski, K., Nedelkoski, S., and Kao, O. (2021). *Autoencoder-based Condition Monitoring and Anomaly Detection Method for Rotating Machines*.

Asakura, T., Yashima, W., Suzuki, K., and Shimotou, M. (2020). Anomaly Detection in a Logistic Operating System Using the Mahalanobis – Taguchi Method.

Barot, A., and Kulkarni, P. (2021). Technological Evolution in the Fault Diagnosis of Rotating Machinery : A Review. *18*(2), 9–26. https://doi.org/10.9790/1684-1802010926

Boniol, P., Linardi, M., Roncallo, F., and Palpanas, T. (2020). Automated Anomaly Detection in Large Sequences. 1834–1837. https://doi.org/10.1109/ICDE48307.2020.00182

Chalapathy, R., and Chawla, S. (2019). Deep Learning for Anomaly Detection: A Survey. 1–50. http://arxiv.org/abs/1901.03407

Choudhary, A., Goyal, D., and Letha, S. S. (2020). Infrared Thermography based Fault Diagnosis of Induction Motor Bearings using Machine. *XX*(XX), 1–8. https://doi.org/10.1109/JSEN.2020.3015868

Cooper, C., Zhang, J., Gao, R. X., Wang, P., and Ragai, I. (2020). Anomaly detection detection in in milling milling tools tools using using acoustic signals and generative generative adversarial networks. *Procedia Manufacturing*, *48*(2019), 372–378. https://doi.org/10.1016/j.promfg.2020.05.059

Egaji, O. A., Ekwevugbe, T., and Griffiths, M. (2020). A Data Mining based Approach for Electric Motor Anomaly Detection Applied on Vibration Data. 330–334.

Glowacz, A. (2019). Fault diagnosis of single-phase induction motor based on acoustic signals. *117*, 65–80.

Gong, C. A., Lee, H., Chuang, Y., Li, T., Su, C. S., Huang, L., Hsu, C., Hwang, Y., Lee, J., and Chang, C. (2018). Design and Implementation of Acoustic Sensing System for Online Early Fault Detection in Industrial Fans.

Hruntovich, N. V, Hruntovich, N. V, Kapanski, A. A., Petrov, I. V, and Kostyleva, E. E.

(2019). Vibration diagnostic of electric motor roller bearings. *02008*, 3–6.

Jabczyńskę, J. and A, Szczęśniak (1995). Digital processing of Doppler signals using Fast Fourier Transform. *Optica ÁppUcata, VoL, XXV*(4), 281–289.

Jiang, W., Hong, Y., Zhou, B., He, X., and Cheng, C. (2019). A GAN-Based Anomaly Detection Approach for Imbalanced Industrial Time Series. *IEEE Access*, *7*, 143608–143619. https://doi.org/10.1109/ACCESS.2019.2944689

Jin, X., Member, S., Que, Z., Sun, Y., Guo, Y., Qiao, W., and Member, S. (2019). A Data-Driven Approach for Bearing Fault Prognostics. *9994*(c). https://doi.org/10.1109/TIA.2019.2907666

Jin, X., Que, Z., Guo, Y., and Qio, W. (2018). A Data-Driven Approach for Bearing Fault Prognostics. 1–8.

Kanwal, N. (2019). Detection of Digital Image Forgery using Fast Fourier Transform and Local Features. *International Conference on Automation, Computational and Technology Management ICACTM*, 1–6.

Khadersab, A., and Shivakumar, S. (2018). Vibration Analysis Techniques for Rotating Machinery and its Vibration Analysis Techniques for Rotating Machinery and its effect on Bearing Faults *Manufacturing Engineering Society International Conference 2017*. *Procedia Manufacturing*, *20*, 247–252. https://doi.org/10.1016/j.promfg.2018.02.036

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., and Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing*, *22*, 949–961. https://doi.org/10.1007/s10586-017-1117-8

Lee, G., Kim, M., Quan, Y., Kim, M., Joon, T., Kim, Y., Yoon, H., Min, S., Kim, D., Mun, J., Oh, J. W., Choi, I. G., Kim, C., Chu, W., Yang, J., Bhandari, B., and Lee, C. (2018). Machine health management in smart factory : A review. *32*(3), 987–1009. https://doi.org/10.1007/s12206-018-0201-1

Lee, H., Li, G., Rai, A., and Chattopadhyay, A. (2020). Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft. *Advanced Engineering Informatics*, *44*(February), 101071. https://doi.org/10.1016/j.aei.2020.101071

Lee, K., Kim, J. K., Kim, J., Hur, K., and Kim, H. (2018). CNN and GRU combination scheme for Bearing Anomaly Detection in Rotating Machinery Health Monitoring. *1st IEEE International Conference on Knowledge Innovation and Invention, ICKII 2018*, 102–105. https://doi.org/10.1109/ICKII.2018.8569155

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*(1995), 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, C., Gryllias, K., Liu, C., Gryllias, K., Vector, A. S., and Description, D. (2019). A semi-supervised Support Vector Data Description- based fault detection method for rolling element bearings based on Cyclic Spectral Coherence To cite this version : *HAL Id : hal-02188765 A Semi-supervised Support Vector Data*

Description- based Fault Det.

Mac, H., Truong, D., Nguyen, L., Nguyen, H., Tran, H. A., and Tran, D. (2018). Detecting attacks on web applications using autoencoder. *ACM International Conference Proceeding Series, December*, 416–421. https://doi.org/10.1145/3287921.3287946

Malla, C., and Panigrahi, I. (2019). Review of Condition Monitoring of Rolling Element Bearing Using Vibration Analysis and Other Techniques. *Journal of Vibration Engineering and Technologies*, *7*(4), 407–414. https://doi.org/10.1007/s42417-019-00119-y

Meire, M., and Karsmakers, P. (2019). Comparison Of Deep Autoencoder Architectures For Real-time Acoustic Based Anomaly Detection In Assets.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., and Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys and Tutorials*, *20*(4), 2923–2960. https://doi.org/10.1109/COMST.2018.2844341

Mohendra, R., Sumon, K. B., Bapi, K., and Pradeep, K. G. (2018). A Stacked Autoencoder Neural Network Based Automated Feature Extraction Method For Anomaly Detection in On-line Condition Monitoring. *Advanced Condition M Onitoring System Advanced Condition M Onitoring System*.

Nath, S. (2020). Low Latency Anomaly Detection with Imperfect Models.

Neupane, D., and Seok, J. (2020). Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches : A Review. 93155–93178.

Nguyen, H., Tran, K. P., Thomassey, S., Hamad, M., Nguyen, H., Tran, K. P., Thomassey, S., Forecasting, M. H., and Detection, A. (2020). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in Supply Chain Management To cite this version *: HAL Id : hal-03083642 Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoenco*.

Novaes, G. De, Leite, P., Maurício, A., André, P., Rosas, C., Stosic, T., and Stosic, B. (2019). Entropy measures for early detection of bearing faults. *Physica A*, *514*, 458–472. https://doi.org/10.1016/j.physa.2018.09.052

Pandarakone, S. E. (2018). Deep Neural Network based Bearing Fault Diagnosis of Induction Motor using Fast Fourier Transform Analysis. *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, 3214–3221.

Pittino, F., Puggl, M., Moldaschl, T., and Hirschl, C. (2020). Automatic Anomaly Detection on In-Production Manufacturing Machines Using Statistical Learning Methods. *SENSORS*, 1–17. https://doi.org/10.3390/s20082344

Pourpanah, F., Zhang, B., Ma, R., and Hao, Q. (2018). Anomaly Detection and Condition Monitoring of UAV Motors and Propellers. *2018 IEEE SENSORS*, *61773197*, 1–4.

Ren, R. (2019). Deep Learning Methods Applied to Anomaly Detection in Vehicle

Manufacturing and Operations. September.

Saeki, M. (2019). Visual explanation of neural network based rotation machinery anomaly detection system. *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 1–4.

Schmidt, S., and Heyns, P. S. (2019). *Localised gear anomaly detection without historical data for reference density estimation*.

Shen, Zhang. Fei, Ye. Bingnan, Wang. Thomas, G., H. (2019). Semi-Supervised Learning of Bearing Anomaly Detection via Deep Variational Autoencoders.

Shen, Z., Fei, Y., Bingnan, W., and Thomas, G. H. (2020). Few-Shot Bearing Anomaly Detection. *AtXiv Preprint ArXv.2020.12851*, 1–21.

Sohaib, M., and Kim, J. (2018). Reliable Fault Diagnosis of Rotary Machine Bearings Using a Stacked Sparse Autoencoder-Based Deep Neural Network.

Sohaib, M., and Kim, J. (2019). Fault Diagnosis of Rotary Machine Bearings Under Inconsistent Working Conditions. *IEEE Transactions on Instrumentation and Measurement*, *PP*(c), 1. https://doi.org/10.1109/TIM.2019.2933342

Sulka, P., Sapietova, A., Dekys, V., Stalmach, O., Sulka, P., Sapietova, A., Dekys, V., and Stalmach, O. (2019). Vibration analysis and comparison of the damaged and undamaged rolling ball of bearing rolling ball. *Transportation Research Procedia*, *40*, 511–518. https://doi.org/10.1016/j.trpro.2019.07.074

Syan, C. S., Ramsoobag, G., Mahabir, K., and Rajnauth, V. (2020). A Case Study for Improving Maintenance Planning of Centrifugal Pumps Using Condition-Based Maintenance. *42*(2), 17–24.

Tan, C. S., Mohd-mokhtar, R., and Arshad, M. R. (2017). Underwater Acoustic Distance Measurement using Fast Fourier Transform Overlap. 0–4.

Varsamopoulos, S., Bertels, K., and Almudever, C. G. (2018). Designing neural network based decoders for surface codes Accelerated BWA-MEM View project hartes View project Designing neural network based decoders for surface codes. November, 1–12. https://www.researchgate.net/publication/329362532

Vo, Q., Fang, F., Zhang, X., and Gao, H. (2017). *S*urface recovery algorithm in white light interferometry based on combined white light phase shifting and fast Fourier transform algorithms. October. https://doi.org/10.1364/AO.56.008174

Wang, Z., Zhou, Y., and Li, G. (2020). Anomaly detection for machinery by using Big Data Real- Time processing and clustering technique.

Wei, W., Wu, H., and Ma, H. (2019). An autoencoder and LSTM-based traffic flow prediction method. *Sensors (Switzerland)*, *19*(13), 1–16. https://doi.org/10.3390/s19132946

APPENDIX

```
import os
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler

#from sklearn.externals import joblib
import joblib
import seaborn as sns
sns.set(color_codes=True)
import matplotlib.pyplot as plt
%matplotlib inline

from numpy.random import seed
from tensorflow import set_random_seed
import tensorflow as tf
tf.logging.set_verbosity(tf.logging.ERROR)

from keras.layers import Input, Dropout, Dense, LSTM, TimeDistributed, RepeatVector
from keras.models import Model
from keras import regularizers
##########################################################################
seed(10)
set_random_seed(10)

data_dir='data/bearing_data'
merged_data=pd.DataFrame()

for filename in os.listdir(data_dir):
    dataset = pd.read_csv(os.path.join(data_dir, filename), sep='\t')
    dataset_mean_abs = np.array(dataset.abs().mean())
    dataset_mean_abs = pd.DataFrame(dataset_mean_abs.reshape(1,4))
    dataset_mean_abs.index = [filename]
    merged_data = merged_data.append(dataset_mean_abs)

merged_data.columns = ['Bearing 1', 'Bearing 2', 'Bearing 3', 'Bearing 4']
##########################################################################
merged_data.index = pd.to_datetime(merged_data.index,
format='%Y.%m.%d.%H.%M.%S')
merged_data = merged_data.sort_index()
merged_data.to_csv('Averaged_BearingTest_Dataset.csv')
print("Dataset shape:", merged_data.shape)
merged_data.head()
##########################################################################
merged_data.describe()
##########################################################################
train = merged_data['2004-02-12 10:52:39': '2004-02-15 12:52:39']
test = merged_data['2004-02-15 12:52:39':]
print("Training dataset shape:", train.shape)
```

```
print("Test dataset shape:", test.shape)
###########################################################################
fig, ax = plt.subplots(figsize=(14, 6), dpi=80)
ax.plot(train['Bearing 1'], label='Bearing 1', color='blue', animated =True, linewidth=1)
ax.plot(train['Bearing 2'], label='Bearing 2', color='red', animated = True, linewidth=1)
ax.plot(train['Bearing 3'], label='Bearing 3', color='green', animated =True, linewidth=1)
ax.plot(train['Bearing 4'], label='Bearing 4', color='black', animated =True, linewidth=1)
plt.legend(loc='lower left')
ax.set_title('Normal Bearing Data set in time domain', fontsize=16)
plt.show()
###########################################################################
# transforming data from the time domain to the frequency domain using fast fourier transform
train_fft = np.fft.fft(train)
test_fft = np.fft.fft(test)
###########################################################################
# frequencies of the healthy sensor signal
fig, ax = plt.subplots(figsize=(14, 6), dpi=80)
ax.plot(test['Bearing 1'], label='Bearing 1', color='blue', animated =True, linewidth=1)
ax.plot(test['Bearing 2'], label='Bearing 2', color='red', animated = True, linewidth=1)
ax.plot(test['Bearing 3'], label='Bearing 3', color='green', animated =True, linewidth=1)
ax.plot(test['Bearing 4'], label='Bearing 4', color='black', animated =True, linewidth=1)
plt.legend(loc='lower left')
ax.set_title('Normal Bearing Data set in frequency domain', fontsize=16)
plt.show()
###########################################################################
fig, ax = plt.subplots(figsize=(14, 6), dpi=80)
ax.plot(test['Bearing 1'], label='Bearing 1', color='blue', animated =True, linewidth=1)
ax.plot(test['Bearing 2'], label='Bearing 2', color='red', animated = True, linewidth=1)
ax.plot(test['Bearing 3'], label='Bearing 3', color='green', animated =True, linewidth=1)
ax.plot(test['Bearing 4'], label='Bearing 4', color='black', animated =True, linewidth=1)
plt.legend(loc='lower left')
ax.set_title('Abnormal Bearing Data set in time domain', fontsize=16)
plt.show()
###########################################################################
# frequencies of the degrading sensor signal
fig, ax = plt.subplots(figsize=(14, 6), dpi=80)
ax.plot(test_fft[:,0].real, label='Bearing 1', color='blue', animated =True, linewidth=1)
ax.plot(test_fft[:,1].imag, label='Bearing 2', color='red', animated = True, linewidth=1)
ax.plot(test_fft[:,2].real, label='Bearing 3', color='green', animated =True, linewidth=1)
ax.plot(test_fft[:,3].real, label='Bearing 4', color='black', animated =True, linewidth=1)
plt.legend(loc='lower left')
ax.set_title('Abnormal Bearing Data set in frequency domain', fontsize=16)
plt.show()
###########################################################################
# normalize the data
scaler = MinMaxScaler()
X_train = scaler.fit_transform(train)
X_test = scaler.transform(test)
scaler_filename = "scaler_data"
```

```
joblib.dump(scaler, scaler_filename)
##########################################################################
X_train
##########################################################################
X_test
##########################################################################
# reshape inputs for LSTM [samples, timesteps, features]
X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
print("Training data shape:", X_train.shape)
X_test = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
print("Test data shape:", X_test.shape)
##########################################################################
# define the autoencoder network model
def autoencoder_model(X):
 inputs = Input(shape=(X.shape[1], X.shape[2]))
 L1 =  LSTM(200,  activation=  'relu', return_sequences=True,
kernel_regularizer=regularizers.l2(0.00))(inputs)
 L2 = LSTM(25, activation='relu', return_sequences=False)(L1)
 L3 = RepeatVector(X.shape[1])(L2)
 L4 = LSTM(25, activation='sigmoid', return_sequences=True)(L3)
 L5 = LSTM(200, activation='relu', return_sequences=True)(L4)
 output = TimeDistributed(Dense(X.shape[2]))(L5)
 model = Model(inputs=inputs, outputs=output)
 return model

# create the autoencoder model
model = autoencoder_model(X_train)
model.compile(optimizer='adam', loss='mae', metrics=['accuracy'])
model.summary()
##########################################################################
# fit the model to the data
nb_epochs = 100
batch_size = 10
history = model.fit(X_train, X_train, epochs=nb_epochs, batch_size=batch_size,
 validation_split=0.05).history
##########################################################################
predict=model.predict(X_test)
##########################################################################
# plot the training loss
fig, ax = plt.subplots(figsize=(14, 6), dpi=80)
ax.plot(history['loss'], 'b', label='Train', linewidth=2)
ax.plot(history['val_loss'], 'r', label='Validation', linewidth=2)
ax.set_title('Model loss', fontsize=16)
ax.set_ylabel('Loss (mae)')
ax.set_xlabel('Epoch')
ax.legend(loc='upper right')
plt.show()
##########################################################################
# plot the loss distribution of the training set
X_pred = model.predict(X_train)
```

```
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = pd.DataFrame(X_pred, columns=train.columns)
X_pred.index = train.index
scored = pd.DataFrame(index=train.index)
Xtrain = X_train.reshape(X_train.shape[0], X_train.shape[2])
scored['Loss_mae'] = np.mean(np.abs(X_pred-Xtrain), axis = 1)
plt.figure(figsize=(16,9), dpi=80)
plt.title('Loss Distribution', fontsize=16)
sns.distplot(scored['Loss_mae'], bins = 20, kde= True, color = 'blue');
plt.xlim([0.0,.5])
#########################################################################
# calculate the loss on the test set
X_pred = model.predict(X_test)
X_pred = X_pred.reshape(X_pred.shape[0], X_pred.shape[2])
X_pred = pd.DataFrame(X_pred, columns=test.columns)
X_pred.index = test.index
scored = pd.DataFrame(index=test.index)
Xtest = X_test.reshape(X_test.shape[0], X_test.shape[2])
scored['Loss_mae'] = np.mean(np.abs(X_pred-Xtest), axis = 1)
scored['Threshold'] = 0.16
scored['Anomaly'] = scored['Loss_mae'] > scored['Threshold']
scored.head()
#########################################################################
# plot bearing failure time plot
scored.plot(logy=True, figsize=(16,9), ylim=[1e-3,1e2], color=['blue','red'])
#########################################################################
# calculate the same metrics for the training set
# and merge all data in a single dataframe for plotting
X_pred_train = model.predict(X_train)
X_pred_train = X_pred_train.reshape(X_pred_train.shape[0], X_pred_train.shape[2])
X_pred_train = pd.DataFrame(X_pred_train, columns=train.columns)
X_pred_train.index = train.index
scored_train = pd.DataFrame(index=train.index)
scored_train['Loss_mae'] = np.mean(np.abs(X_pred_train-Xtrain), axis = 1)
scored_train['Threshold'] = 0.16
#scored_train['Anomaly'] = scored_train['Loss_mae'] > scored_train['Threshold']
scored = pd.concat([scored_train, scored])
#########################################################################
# plot bearing failure time plot
scored.plot(logy=True, figsize=(16,9), ylim=[1e-2,1e2], color=['blue','red'])
#########################################################################
# save all model information, including weights, in h5 format
model.save("Cloud_model.h5")
print("Model saved")
```