VIBRATION ANALYSIS USING DEEP LEARNING FOR PREDICTIVE MAINTENANCE

GOH QI CHEN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

VIBRATION ANALYSIS USING DEEP LEARNING FOR PREDICTIVE MAINTENANCE

GOH QI CHEN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : VIBRATION ANALYSIS USING DEEP LEARNING FOR PREDICTIVE MAINTENANCE

Sesi Pengajian : 2023/2024

Saya GOH QI CHEN mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

□ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☑ **TIDAK TERHAD**

Disahkan oleh:

fakrulradzi

(TANDATANGAN PENULIS) (COP DAN TANDATANGAN PENYELIA)

DR. FAKRULRADZI BIN IDRIS
Pensyarah Kanan
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya
76100 Durian Tunggal, Melaka

Tarikh : 09 JANUARI 2024 Tarikh : 9 Januari 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “Vibration Analysis using Deep Learning for Predictive Maintenance” is the result of my own work except for quotes as cited in the references.

Signature : ………………………………….

Author : …GOH QI CHEN……………..

Date : …9 January 2024………………

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : fakrulradzi ……………………………………

Supervisor Name : …Dr. Fakrulradzi Bin Idris……

Date : …9 January 2024………………

DEDICATION

This study is sincerely dedicated to our dear parents, who have been our constant inspiration and pillars of strength during moments when I felt like giving up. Their unwavering moral, spiritual, emotional, and financial support have been invaluable throughout this journey.

To our siblings, relatives, mentors, friends, and classmates, who generously shared their words of advice and encouragement, contributing significantly to the completion of this study.

To our teachers who believed in my ability to complete the research on time, offering valuable assistance to improve our work. Their encouragement and inspirational stories from their own student days have been a great source of motivation for me.

ABSTRACT

In the era of Industry 4.0, predictive maintenance through vibration analysis has gradually improved across various industrial sectors, especially those incorporating rotating components like bearings and shafts. Conventional maintenance methods, whether reactive, preventive, or proactive, often entail high risks and costs. Reactive maintenance, in particular, results in significant drawbacks, causing downtime, resource wastage, and substantial monthly repair costs. The project proposes employing a Long Short-Term Memory (LSTM) autoencoder deep learning model for predictive maintenance, utilizing acceleration data collected from accelerometers. This data undergoes preprocessing before being fed into the LSTM autoencoder model, implemented using Python with TensorFlow and Keras frameworks in Jupyter Notebook. The project concludes with the LSTM autoencoder model demonstrating low losses (0.0017), highlighting its effectiveness in flagging anomaly conditions and its potential to enhance predictive maintenance in industries with rotating machinery components.

ABSTRAK

Dalam era Industri 4.0, penyelenggaraan meramal melalui analisis getaran secara beransur-ansur meningkat di pelbagai sektor industri, terutamanya dalam sektor yang mengandungi komponen putaran seperti bantalan dan shaft. Kaedah penyelenggaraan konvensional, sama ada reaktif, pencegahan, atau proaktif, sering melibatkan risiko dan kos yang tinggi. Penyelenggaraan reaktif, khususnya, membawa kelemahan yang signifikan, menyebabkan masa berhenti, pembaziran sumber, dan kos penyelenggaraan bulanan yang besar. Projek ini mencadangkan penggunaan model pembelajaran mendalam long short-term memory (LSTM) autoencoder untuk penyelenggaraan meramal, dengan menggunakan data pecutan yang dikumpulkan daripada pengukur pecutan. Data ini mengalami pra pemprosesan sebelum dimasukkan ke dalam model autoencoder LSTM, dilaksanakan menggunakan bahasa Python dengan rangka kerja TensorFlow dan Keras dalam Jupyter Notebook. Projek ini diakhiri dengan model autoencoder LSTM menunjukkan kerugian rendah (0.0017), menyoroti keberkesanannya dalam mengenal pasti keadaan yang tidak normal dan potensinya untuk meningkatkan penyelenggaraan meramal dalam industri yang melibatkan komponen mesin putaran.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Dr. Fakrulradzi Bin Idris, for their unwavering support, guidance, and invaluable insights throughout the research process. Their expertise and encouragement played a pivotal role in shaping the direction of this thesis.

I am also thankful for the assistance and collaboration of my colleagues and peers who provided valuable feedback and constructive criticism, contributing to the refinement of this work.

Special appreciation goes to my family for their constant encouragement, understanding, and patience during the challenging phases of this journey. Their love and support have been my pillar of strength.

Lastly, I want to acknowledge the sources, authors, and institutions whose works and contributions have been instrumental in shaping the theoretical framework of this thesis.

This accomplishment would not have been possible without the support and encouragement of these individuals and organizations. I am truly grateful for their contributions to the successful completion of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

LSTM	:	Long Short-Term Memory
AE	:	Autoencoder
MQTT	:	Message Queuing Telemetry Transport
MEMS	:	Micro-electromechanical systems
MSE	:	Mean Square Error
RMSE	:	Root Mean Square Error
MySQL	:	My Structured Query Language
FFT	:	Fast Fourier Transform

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In this chapter, a simple introduction to the project will be provided. Explanations about the development of vibration monitoring technology will be included to familiarize the reader with the system. This chapter comprises background information, the problem statement, objectives, scopes, and the project outline for the entire project.

1.1 Background

Nowadays, vibration monitoring technology plays an essential role in industrial factories for diagnosing potential machine failures. This technology is crucial because a breakdown in the production chain can lead to significant economic losses [1]. The mechanical vibration signals generated by the machine are analyzed to assess the overall performance of the mechanical equipment or specific mechanical parts during operation [2].

In the early years, vibration monitoring technology could only be implemented with a cable connection. However, wired vibration monitoring technology has its limitations, making it unsuitable for use in all industrial environments. For instance, attaching a wired sensor reliably to the surface of rotating machinery is unsafe. Additionally, the presence of cables may pose risks of human and material accidents. Moreover, the traditional wired technology incurs high installation and maintenance costs, especially in remote monitoring scenarios. On top of that, users need to manually replace old cables with new ones, adding inconvenience due to the low mobility imposed by this type of technology [3,4].

Recently, a new wireless vibration technology, capable of monitoring machines without any cables, has been developed. This technology is more convenient than wired sensors, offering extensive coverage in remote monitoring areas. Furthermore, it is low-cost and easy to install since there is no need for wiring and trenching.

Vibration-based condition monitoring is one of the most widely used approaches for predictive maintenance, given its ease of measurement and data analysis. Vibration analysis is a measurement technique employed to assess a machine's operating condition, identifying potential issues before they lead to failure and extensive damage. Consequently, the implementation of this technology in the industry can lead to simultaneous reductions in maintenance costs and downtime [5].

Condition monitoring methodologies for predictive maintenance can be simplified into four steps. First, vibration data, such as the acceleration signal, is collected from the sensor. Second, the signal undergoes processing to remove noise. Third, the signal is analyzed to detect the presence of any potential failure. Lastly, the system identifies the type of failure and determines the machine's condition by comparing the actual measurement results with manufacturer-set standards. Actions are then decided based on statistical information and years of experience.

Based on the statistics, a factory implementing condition-based predictive maintenance will experience a 25-30 percent decrease in maintenance activities and a 35-45 percent decrease in machine breakdowns. Additionally, it is stated that there is a 20-25 percent increase in production rates and return on investment when the monitoring system is employed in industrial factories [6]. Therefore, the wireless vibration monitoring system has fewer limitations compared to the traditional wired monitoring system.

1.2 Problem Statement

With Industry 4.0, proactive maintenance is being replaced by predictive maintenance. In the industrial field, many machines consist of rotational parts such as bearings or shafts. However, maintenance methods like reactive, preventive, or proactive maintenance pose high risks and costs. Reactive maintenance, in particular, leads to machine malfunctions causing downtime and wastage of raw resources, incurring substantial monthly repair costs [7]. Production halts result in decreased line productivity, especially when replacement times are lengthy. When bearings become faulty, it affects various geometrical parameters in the frequency response [8,9]. Unknown errors in bearing faults can lead to extra maintenance fees and extended downtime, which is not ideal for businesses. Beyond fault identification, the Remaining Useful Life (RUL) of the machine provides crucial information for industries [10]. Without knowledge of the machine lifespan, companies struggle to predict when machines might fail, leading to extended waiting times for new machines or spare parts and significantly increasing maintenance costs. Therefore, this project aims to develop an LSTM deep learning predictive maintenance model using time-series vibration data to analyze the RUL of machines.

1.3 Objectives

The objectives of this project are:

- To design wireless vibration monitoring system
- To develop a deep learning-based LSTM model
- To analyze and compare the performance of the LSTM model with parameter of model loss.

1.4 Project Scope

This project aims to develop a deep learning model for vibration predictive maintenance. The model will utilize data collected from parameters such as acceleration, temperature, and sound. After collecting the data, it will be labeled and fed into the deep learning model as both training and testing data. The primary focus of this project is on the LSTM model. Additionally, other machine learning models will be explored and built to compare the performance of the LSTM predictive maintenance model. The project encompasses both hardware and software implementation.

1.4.1 Hardware

For the hardware component of the system, it is designed to collect raw vibration data from the machine. The setup includes the Hibiscus ESP32, Raspberry Pi 3, resistors, LEDs, TP4056 charging module, 3.7V Lithium rechargeable batteries, and DPDT switches. In this configuration, the Raspberry Pi 3 functions as a broker, facilitating communication between the Hibiscus ESP32 and a laptop simultaneously. The Hibiscus ESP32 module incorporates the MPU6050 chip, which houses an accelerometer sensor and a built-in temperature sensor. The accelerometer within the MPU6050 chip measures vibrations in three axes (X, Y, Z). Green LEDs are connected to both modules to indicate MQTT connection status. A lit green LED signals successful MQTT connection, while an unlit LED indicates a lack of connection to MQTT.

1.4.2 Software

Several software applications are used in developing this project. For designing the hardware system, Arduino IDE, Node-RED, mySQL are used for configuring the hardware in term of communication and data collection. First, the Arduino IDE is used to write the program code for publishing acceleration and temperature data to Raspberry Pi 3 through the MQTT protocol. Then, the Node-RED is used as a programming tool to create the system's functionality by wiring the data flow between nodes using a browser. Next, mySQL is used as a cloud to store the data obtained from raspberry pi. It able to provides data file in several extension such a .db and .csv.

For analyzing the data, Python programming will be used to model the predictive maintenance model. In this project, two models will be designed: a fault diagnosis model and an RUL prediction model. These models can be designed and run in a Jupyter notebook application programmed with Python code.

1.4.3 Deep Learning Algorithm

Deep learning algorithms are a group of machine learning algorithms that mimic the structure of the human brain, particularly neural networks. These algorithms learn and extract meaningful representations or features from large amounts of data without requiring in-depth knowledge of mathematical relationships or specific field knowledge. In this project, the main model designed for anomaly detection will be the LSTM model. Additionally, other models, such as the Convolutional Current Network (CNN), will be developed for performance comparison.

1.5 Chapter Outline

The wireless vibration monitoring system for predictive maintenance was described, focusing on the improvement of the wired vibration monitoring system through the utilization of the Internet of Things (IoT) concept. All the details about this project are defined in each chapter of this report, as shown below.

Chapter 1: In this chapter, it will show a simple introduction of the project. Some explanations about the development of vibration monitoring technology will be considered to acknowledge the system. It consists of background, problem statement, objectives, scopes, and the project outline for the whole project in this chapter.

Chapter 2: This chapter will discuss sources, website, patents and journals that related to the project. It also consists of the existing products that can be found on the market. To briefly understand the project, there are many sources and researches done before.

Chapter 3: This chapter will mention about steps and methods involve completing the project. There are several steps to be applied in designing wireless vibration monitoring system for predictive maintenance. This part consists of project flowchart, methodology that being used and the explanation about tools and components used for this project.

Chapter 4: In this chapter, it shows the result obtained that have been achieved throughout this project. Besides, it will also discuss about the result of the project based on testing of the finished project.

Chapter 5: This chapter will describe about conclusion and recommendation for the wireless vibration monitoring system for predictive maintenance. This section includes project summary, project finding and further recommendation to improve the project.

CHAPTER 2

BACKGROUND STUDY

This chapter will discuss sources, website, patents, and journals that related to the project. It also consists of the existing products that can be found on the market. To briefly understand the project, there are many sources and research done before.

2.1 General Background

2.1.1 Vibration

Vibration is defined as a physical process that occurs in kinetic structures and wheeling machines. Vibration can be generated by different types of sources, such as rotating electric field and shafts, components with bearing, flowing in liquid and ignition phenomenon. There are specific sensors invented depending on various vibrations such as PZT sensors, proximity probes, MEMS and etc. Different sensors with various measuring techniques are used to evaluate displacement, velocity and acceleration. Vibrations are expressed in both periodicity and intensity by frequency and amplitude respectively. Furthermore, vibration can be detected in any condition of machines.

2.1.2 Vibration Analysis

Vibration analysis is an action that detects the level of vibration and also inspects the pattern of vibration signals. Vibration analysis is frequently conducted on 2 characteristics which are time waveform and frequency spectrum. Frequency spectrum is acquired by substituting formula of Fourier Transform on time waveform. The analysis for time domain shows the normality and abnormality of vibration pattern, and these are studied in several parameters such as RMS, crest factor, standard deviation, peak amplitude etc. Moreover, FFT technique is established to ease the process of frequency analysis as it provides quick responses and has high efficiency. Vibration analysis is applied on machines, components, and structures such as bearings, wheels, gearboxes, motors and etc. Vibration analysis is able to detect issues including imbalances, failures on bearing and gearbox, misalignment and etc.

2.1.3 Vibration Monitoring System

A vibration monitoring system is a series of combined apparatus aimed at detecting the health changes of machinery and instruments by evaluating one or more variables. It is essential to many industries for safety purposes and keeping the operating system running. This is due to vibration monitoring helping a lot at forecasting machine failures. The failures include but are not limited to misalignment, bearing faults and etc. Additionally, having a vibration monitoring system could prevent financial losses due to scheduled predictive maintenance. The latest vibration monitoring system applies wireless transmission and long-lasting batteries which have gradually replaced the wired accelerometers that have been used in industry. Vibration monitoring systems are executed by applying vibration trend analysis. It is a process that tracks for irregularities in the vibration system of an operating machine. When the amplitude of vibration increases or decreases significantly with steady state conditions, it is a symptom that changes occurred in machines.

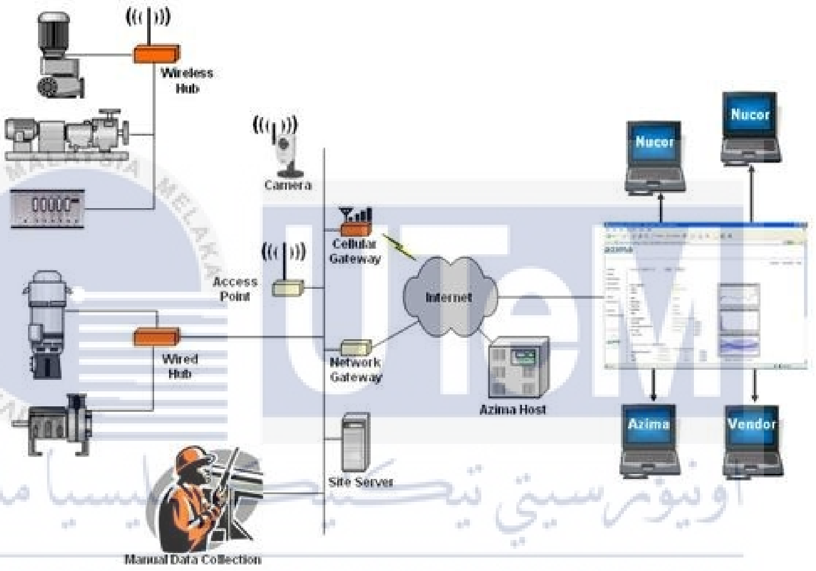


Figure 2.1 Vibration monitoring system

2.1.4 Condition Monitoring System

Condition monitoring is the action of inspecting a specific condition in machines to make certain of the alteration which indicates the occurring faults. The condition that provides alteration could be vibration, temperature and others which are manipulated. It is an important part for predictive maintenance to be scheduled after abnormal pattern of signal is observed to prevent further breakdown and unexpected downtime. Condition monitoring could be used on a variety of equipment, including rotating machines, and additional systems. Conventional type of condition monitoring applies vibration analysis while on the other hand, modern yet innovative technologies such as sensors are used to detect various parameters and notify users when a fault is detected. To fasten the process of being notified, the Internet of Things (IoT) is executed and combined therefore smart machines could act as a platform for engineers in different positions to make decision before system breakdown. Types of condition monitoring include electrical monitoring, motor circuit analysis, oil analysis, thermography, vibration analysis and radiography.

2.1.5 Rotating System

The rotating electric machines are used to convert mechanical energy to electrical energy or vice versa. The three basic types of rotating electric machines are DC electric machines, synchronous machines and induction motors or asynchronous machines. The examples of DC electric machines are DC motors and DC generators while the example of synchronous machines are alternators and synchronous machines. All existing rotating electric machines have two fundamental parts where the first one is rotor, and second part is stator which is stationary. These two parts are fabricated from highly permeable magnetic material, for example silicon steel.

2.1.6 Displacement, Velocity and Acceleration Sensors

Vibration sensors are measuring devices which sense the vibration of movement for the equipment or system which the sensors are attached to. Amplitude and frequency of vibration collected will be studied and investigated. General types of vibration sensors are displacement sensor, velocity sensor, and accelerometer. Accelerometer is the best option among these three sensors. It is mostly used in industrial rotating machines for its simplicity to apply and quite sensitive to the high frequencies during vibration which is typically an indicator of failed machines. Examples of industrial accelerometers are piezoelectric accelerometers and micro electromechanical systems (MEMS) accelerometers.

2.2 Working Principles of System

In this project, the working principles involved are the general knowledge for this project. It may include theory of a data process algorithm, communication ways and the software used to carry out the task.

2.2.1 MQTT communication protocol

MQTT had been widely used in the IoT project. IoT, Internet of Thing ia the interconnection devices such as appliances, devices, sensors, or components [11]. MQTT, Message Queuing Telemetry Transport, is a lightweight messaging protocol for communication between two devices. It consists of a publish-subscribe messaging pattern, where devices can publish the data to assign topic, vice versa, the devices also able to subscribe to the topic that interested to collect.

As shown in Figure 2.2, the sensors and valve or cloud server are type of MQTT clients, they can subscribe and publish the data to the MQTT broker. With the technology of MQTT nowadays, the MQTT can be installed in Raspberry Pi and the microprocessor with Wi-Fi module able to subscribe and publish the data to the MQTT broker.

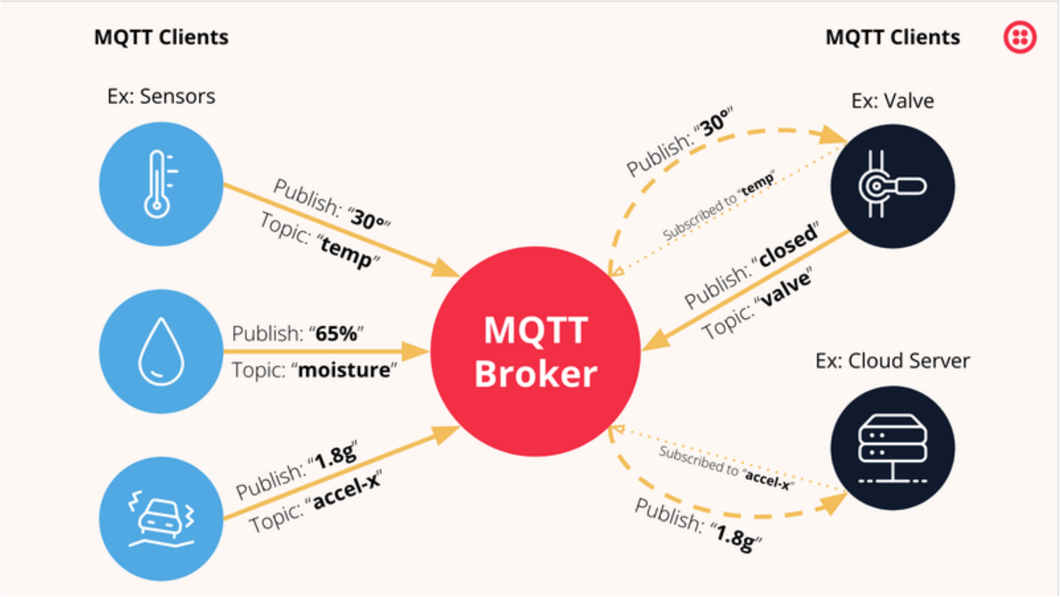


Figure 2.2 MQTT protocol

2.2.2 MySQL

MySQL is an open-source database management system that is used for managing structured data. It is commonly used in web applications and acts as an online database (cloud). It helps to communicate between hardware with the broker. It can operate in several operating systems such as Linux, Solaris, macOS, Windows and FreeBSD. Since MySQL is open source, the application and use of MySQL is free of license and can be modified with the user creativity.

2.2.3 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and explanatory text. It provides an interactive computational environment that supports various programming languages, including Python, R, Julia, and others. Jupyter Notebook is widely used for data analysis, prototyping, machine learning, and scientific computing tasks. In Jupyter notebooks, it is interactive and self-contained, where it consists of series of cells that can contain code, text, equations, or visualizations. Under the notebooks, users can execute the code within the notebook. All the Code cells can contain snippets of code written in the supported programming languages. When a code cell is executed, the code is run in a separate kernel, and the output or result is displayed below the cell. Furthermore, Jupyter Notebook integrates well with libraries for data visualization and plotting, such as Matplotlib, Seaborn, and Plotly. Visualizations can be displayed directly in the notebook, allowing users to create interactive charts, graphs, and other visual representations of their data.

2.2.4 Deep Learning

It is an important branch of machine learning that achieved significant advances in theory [12] and applications to many fields, for example, image processing [13], automatic driving [14], natural language processing[15,16], etc. Deep learning is a subfield of machine learning that focuses on training artificial neural networks to learn and make predictions or decisions without explicit programming. It is inspired by the structure and function of the human brain, specifically the interconnected network of neurons. The key component of deep learning is the artificial neural network, which consists of interconnected nodes or artificial neurons. These neurons receive inputs, apply a weighted sum, pass the result through an activation function, and produce an output. Deep neural networks typically have multiple layers, including an input layer, one or more hidden layers, and an output layer. The learning process in deep learning involves two main steps: training and inference. During training, the network learns from a labeled dataset by adjusting the weights and biases of the neurons to minimize the difference between its predictions and the true labels. This process is often performed using optimization algorithms like stochastic gradient descent. Inference is the phase where the trained model is used to make predictions on new, unseen data.

2.2.5 Fault Diagnosis

In fault diagnosis of the vibration object, acceleration and velocity data are the main parameter that can retrieve the characteristics of the rotatory part. An object vibrates when it moves repeatedly back and forth from its stationary position. This can be demonstrated with a spring where the spring is stretched and forced to move its weight to the minimum limit when force is put on. When force is taken out, it will result in the weight moving upwards, passing through the stationary point to reach the maximum limit. This is due to the stored energy in the spring. In common knowledge, velocity is the first derivative of displacement with respect to time, and known as the rate of change in displacement. Acceleration is the second derivative of displacement, and the rate of change of velocity. In vibration analysis, all rotating machines generate vibrations because of the dynamics of the machine. By determining the amplitude of vibration at specified frequencies, informative values are obtained such as the accuracy of alignment, the condition of used bearings etc.

2.2.6 Anomaly Detection

The identification of anomalies in vibration applications holds paramount significance in the realms of predictive maintenance and condition monitoring across diverse industries, such as manufacturing, aerospace, and energy. Vibration signals serve as key indicators of the condition and operational efficacy of rotating machinery, with anomalies in these signals serving as potential precursors to faults or imminent failures. The primary aim of anomaly detection lies in the discernment of deviations from established norms in vibration patterns, thereby signalling irregularities in the machinery.

An array of methodologies is deployed for anomaly detection in vibration applications, encompassing statistical methods, machine learning algorithms, and signal processing techniques. Statistical approaches often entail the formulation of baseline models representing typical vibration behaviour, with deviations being identified through rigorous statistical analysis. Machine learning strategies leverage algorithms trained on historical data to discern patterns associated with normal operation, facilitating the identification of anomalous behaviour. Signal processing techniques are focused on extracting pertinent features from vibration signals and subsequently comparing them to predefined thresholds. The amalgamation of these methodologies facilitates early fault detection, consequently mitigating downtime, curtailing maintenance expenditures, and augmenting overall operational efficiency within industrial contexts.

2.2.7 Fast Fourier Transform (FFT) Analysis

FFT which known as Fast Fourier Transform is applied when analysing signals for various application domains. FFT performs signals transformation from time domain to frequency domain. In time domain, it is expressed in one waveform that carries the summation of all characteristics. On the other hand, the signal characteristics are expressed in individual frequency components during frequency domain.

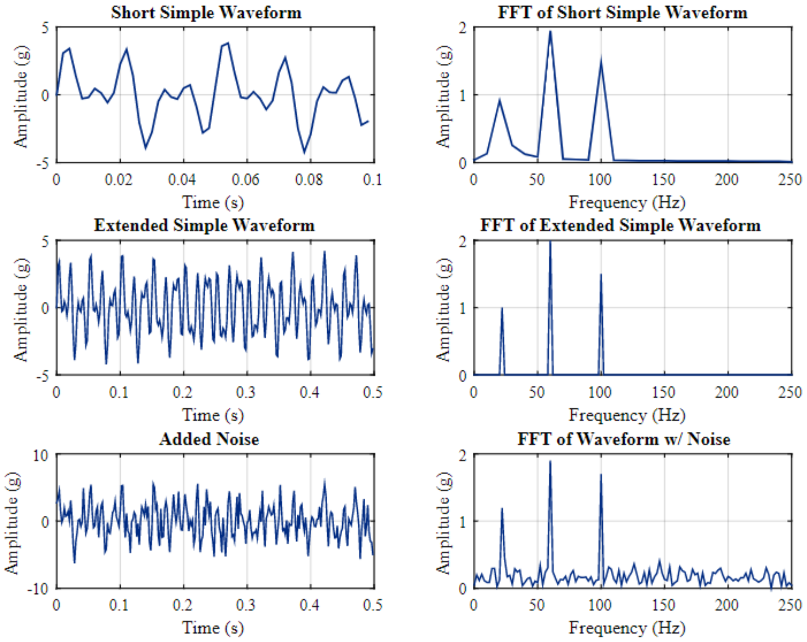


Figure 2.3 Vibration signal in time domain and frequency domain

In frequency domain, examination of measured data is usually the first stage of tracking and analysing signals. FFT analysis provides features where engineers can observe the process of instruments respond at different independent frequencies when performing vibration testing at instruments and components. This has proven that with the aid of frequency spectra, engineers can perform better when dealing with design optimization and defining deflection limitations. With obtained frequency values, FFT spectra can be used to recognize acceptable tolerance graphs. Moreover, notification will be received by engineers when level of critical vibration is beyond the maximum frequencies level. Transformation of FFT spectra is derived using algorithm from DFT which known as Discrete Fourier Transform that can be defined as:

$$A(f_k) = \frac{1}{N}\sum_{n=0}^{N-1} a(t_n)e^{-i\frac{2\pi kn}{N}} \tag{1}$$

Digitized signals from all different kinds of sensors can be analysed with the aid of FFT analysers. Based on its application, the most suitable and related sensors are operated. For instance, accelerometers are utilized when performing machine diagnostics and structural measurements.

2.3 Literature Review

2.3.1 Related Paper

Before design and development of project, several research to similar topic has been done. The first reference is "Vibration analysis of electrical rotating machines using FFT: A method of predictive maintenance" [17]. It is a research paper written by Patil and Gaikwad (2013) with regards to vibration analysis using FFT. This paper mentions that detection of faults should be done to increase the reliability of rotating equipment as the demand from the market has been increasing in a critical way. Therefore, vibration analysis for predictive maintenance is proposed to overcome these issues as it can analyze the fundamental cause within any instruments or plants. To inspect the vibration signals and identify the health conditions of machines, Fast Fourier Transform (FFT) is suggested. By running rotating electrical machines with different operating conditions, the author concludes that vibration analysis could be proceeded only if all machines are in healthy modes and thus producing vibration pattern that is quite stable. While comparing to the stable patterns, the abnormal patterns spotted indicate that the machine tested needs repairing and predictive maintenance could be scheduled. Moreover, this paper also focuses on the restriction of vibration analysis in the traditional way.

Nowadays, industry 5.0 is an evolution of manufacturing technology that utilizes advanced data analytics and machine learning techniques to optimize production process. Therefore, predictive maintenance with machine learning becomes more crucial as technology evolves. With predictive maintenance, the performance of the equipment can be monitored and predict the potential fault condition. However, with machine learning, the data processing can be automated and perform better decision and prediction [18].

The target object to be tested will be a rolling element bearings and gear. It is because both of them are the key components in rotating machinery, which plays an important role in power and motion transmission and is widely used in many major fields of industrial production [19]. For analyze the data, FFT is usually the first analysis to be conducted for the vibration signal from time domain to frequency domain [20].

From the journal "Gearbox fault diagnosis based on multi-scale deep residual learning and stacked LSTM model" written by K.N. Ravikumar, he introduces a fault diagnosis model that includes a multi-scale deep residual learning with a stacked long short-term memory (MDRL-SLSTM) to address sequence data in a gearbox health prediction task. The model achieved better diagnostic performance with vibration data of gearbox. The classification accuracy of 94.08% and 94.33% are attained on bearing datasets and 2nd driving gear of gearbox respectively [21].

Furthermore, in the journal entitled "LSTM-Autoencoder for vibration anomaly detection in vertical carousel storage and retrieval system (VCSRS)" written by Jae Seok Do, the vibration data was collected and analyzed to identify potential issues with the system's operation. An LSTM-autoencoder (long short-term memory) model was used for training and testing further to enhance the accuracy of the anomaly detection process. The combination of the correlation coefficient model and the LSTM-autoencoder resulted in an accuracy rate of 97.70% for detecting anomalies in the vertical carousel system [22].

Next, the journal entitled “A new dynamic predictive maintenance framework using deep learning for failure prognostics” written by Khanh T.P. Nguyen in 2019, he proposed a new dynamic predictive maintenance framework with sensor measurements. In this framework, the prognostics step, based on the Long Short-Term Memory network, is oriented towards the requirements of operation planners. It provides the probability that the system can fail in different time horizons to decide the moment for preparing and performing maintenance activities. For the limitation, the model is only considering for perfect maintenance, the different levels of imperfect maintenances can be investigated [23].

In journal “A Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders” written by Xanthi Bampula in 2021, the author with her team study investigates an approach to enable a transition from preventive maintenance activities, that are scheduled at predetermined time intervals, into predictive ones. To enable such approaches in a cyber-physical production system, a deep learning algorithm is used, allowing for maintenance activities to be planned according to the actual operational status of the machine and not in advance. An autoencoder-based methodology is employed for classifying real-world machine and sensor data, into a set of condition-related labels. However, additional experiments are required to further test its performance and accuracy with a larger dataset of proper data quality in a greater period [24].

Moreover, in a journal, “A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance” written by Nagdev Amruthnath in 2018, the author had chosen a simple vibration data collected from an exhaust fan, and have fit different unsupervised learning algorithms such as PCA T 2 statistic, Hierarchical clustering, K-Means, Fuzzy C-Means clustering and model-based clustering to test its accuracy, performance, and robustness. T2 statistic provided more accurate results compared to GMM method, and no hypothesis was required to identify the relationship between cluster and state. In short, although most algorithms provided nearly similar results, each algorithm provided deeper insight into the data. Hence, if the application is just to detect the faults, T2 statistic would be an excellent tool [25].

Besides, the journal entitled “A Wind Turbine Vibration Monitoring System for Predictive Maintenance Based on Machine Learning Methods Developed under Safely Controlled Laboratory Conditions” written by David Perez and his team in 2023 had presented a method for simple wind turbine vibration monitoring in the laboratory by means of an accelerometer placed on a weathervane under different scenarios, with recording of different amplitudes of vibrations caused at a constant speed of 10 km/h. The variables, trends, and data captured during vibration monitoring were then used to implement a prediction system of synthetic failure using machine learning methods such as: Medium Trees, Cubic SVN, Logistic Regression Kernel, Optimized Neural Network, and Bagged Trees, with the last demonstrating an accuracy of up to 87% [26].

Additionally, the journal entitled “The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system" written by Mustafa Cakir in 2021 had include the data such as sound level, current, rotational speed, and temperature to increase the success of the classification. The data collected from the experimental setup was modelled for classification with popular ML algorithms such as support vector machine (SVM), linear discrimination analysis (LDA), random forest (RF), decision tree (DT), and k-nearest neighbors (KNN). The models were evaluated with accuracy, precision, TPR, TNR, FPR, FNR, F1 score and, Kappa metrics. During the evaluation of all models, it was observed that with the increase in the number of features in the data set, the accuracy, sensitivity, TPR, TNR, F1 score, and Kappa metrics increased above 99% at 95% confidence interval, and FPR and FNR metrics fell below 1%. Although ML models gave successful results, LDA and DT models gave results much faster than others did. On the other hand, the classification success of the LDA model is relatively low. However, DT model is the optimum choice for CMS due to its convenience in determining threshold values, and its ability to give fast and acceptable classification rates [27].

2.3.2 Existing Product

Similar products appeared in market is observed and referenced. The first model is "Phantom EPH-V11 | Triaxial Vibration Sensor" which is embedded with wireless condition-monitoring system and is able to transfer 3 simultaneous FFT and waveform recordings in terms of time at once. Moreover, it provides features of analyzing and visualizing data from sensors with DigivibeMX. [28]

Phantom EPH-V11 triaxial vibration sensor

The second model "Fluke 3561 FC Vibration Sensor" could make predictive maintenance process in short time. It can sense the faults, imbalances and misalignments. Furthermore, it is a wireless yet compact sensor as a portable monitoring device. Its battery usage can last for 3 years which is a useful feature. [29]

Fluke 3561 FC vibration sensor.

The following instrument "Fluke 805 FC" can measure frequencies from 10Hz up to 1kHz in terms of acceleration, speed, and misalignment. Besides that, with the support of accelerometer, it is useful in high-altitude or cornered areas that could not be reached easily. Furthermore, it is adaptable with the Fluke Connect App which includes functions such as mobile data management, remote access to maintenance team etc. [30]

Fluke 805 FC handheld vibration sensor.

From ifm company, they had introduced a vibration monitoring system that provides vibration monitoring, condition monitoring, machine protection process monitoring with easy setup. In this system, it consists of several type of components such as vibration sensors, diagnostic electronics and gateway for wireless vibration sensors and IO-links. This company do provide wide selection for the vibration sensors, including acceleration sensor that measure the raw data of the vibration. Other than the system, this company also provides a software for analysing the vibration data, which named VSE and VNB. In VSE, the data can be configured and visualized.

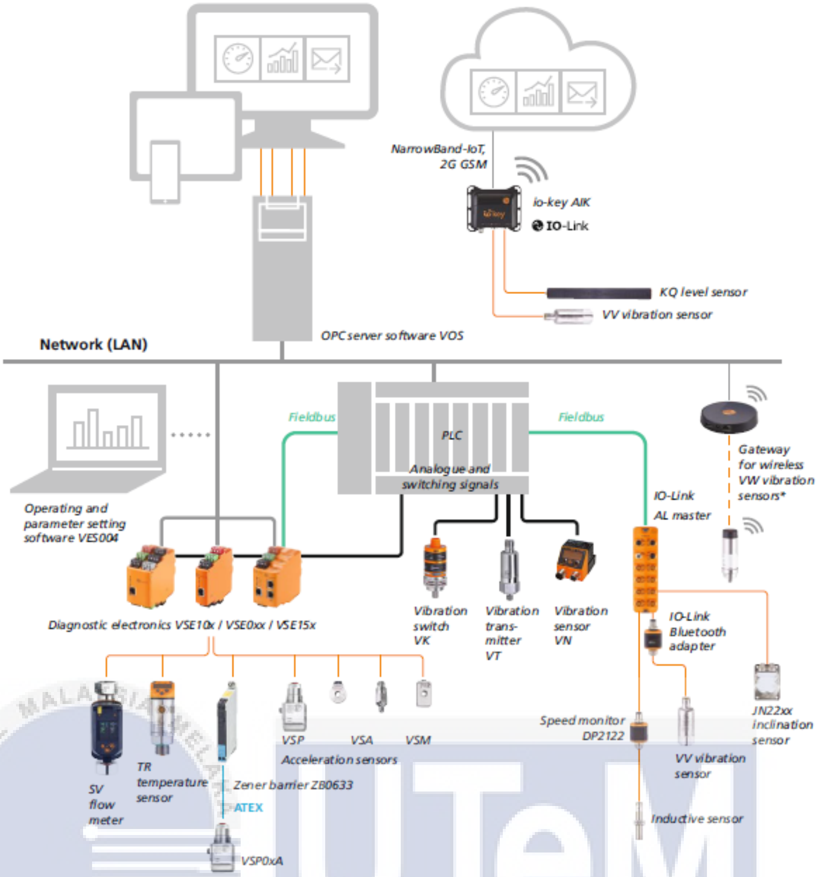


Figure 2.4 Vibration monitoring system of ifm company

CHAPTER 3

METHODOLOGY

This chapter will mention steps and methods involved in completing the project. There are several steps to be applied in designing wireless vibration monitoring system for predictive maintenance. This part consists of a project flowchart, methodology that is being used and the explanation about tools and components used for this project.

3.1 Flow Chart of Project Flow

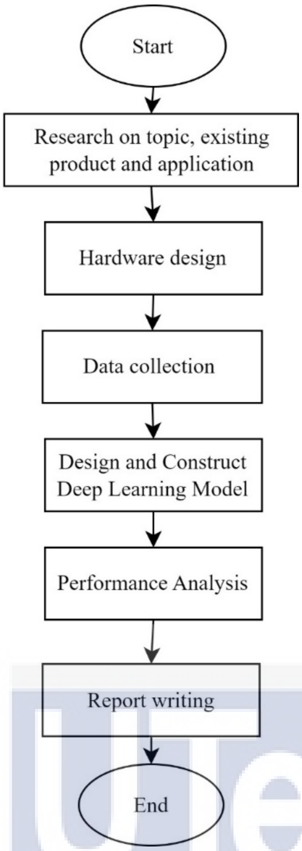

Start
Research on topic, existing product and application
Hardware design
Data collection
Design and Construct Deep Learning Model
Performance Analysis
Report writing
End

Figure 3.1 Overall flow of the project

In the course of this project, the initial phase entails a comprehensive investigation of prior literature and the examination of existing products. The primary objective of this endeavor is to address the void identified in the antecedent scholarly work. Subsequently, a wireless vibration monitoring system is meticulously fashioned, encompassing hardware design implementation on a dedicated testbed to simulate both balanced and unbalanced vibration scenarios. Following this, the construction of an LSTM autoencoder model is undertaken with precision and diligence. Post model creation, an in-depth performance analysis is executed to discern the optimal model amidst various design architectures. Finally, the conclusive stage involves the meticulous composition of the thesis report.

3.2 Flow chart of Overall Process

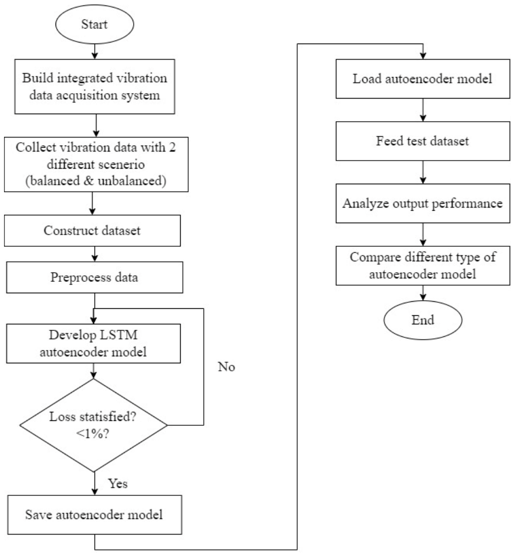

Start
Build integrated vibration data acquisition system
Collect vibration data with 2 different scenerio (balanced & unbalanced)
Construct dataset
Preprocess data
Develop LSTM autoencoder model
No
Loss statisfied? <1%?
Yes
Save autoencoder model
Load autoencoder model
Feed test dataset
Analyze output performance
Compare different type of autoencoder model
End

Figure 3.2 Overall process of the project

Foremost, the inception of the wireless vibration data acquisition system, recognized as the wireless vibration monitoring system, constituted a pivotal undertaking in this initiative. This system was meticulously designed and assembled to facilitate the acquisition of vibration data under two distinct scenarios: balanced and unbalanced cases. After data collection, a rigorous process of construction and preprocessing ensued to prepare the data for integration into the training and testing phases of the deep learning model.

The subsequent phase involved the development of the LSTM autoencoder model, with a stringent criterion ensuring that the loss value remained below the threshold of 1%. Following successful model development, the trained model was preserved and subsequently loaded for the purpose of comprehensive testing. The testing phase involved the introduction of the test dataset into the model, with subsequent performance analysis and comparison executed among diverse design configurations of LSTM and CNN models.

3.3 Flowchart of Data Collection

This subsection delineates the methodology employed for the acquisition of vibration data. The data collection process is orchestrated through the utilization of the ESP32 microcontroller. This microcontroller establishes a seamless connection with the Raspberry Pi 3, functioning as the pivotal server in the communication network bridging the ESP32 with the cloud. Within this architecture, the Raspberry Pi 3 assumes the role of an intermediary, facilitating the bidirectional exchange of data between the ESP32 and the cloud infrastructure.

The collected data is systematically channelled into a MySQL database, configured to accommodate the storage of expansive datasets. This database, designed with the capability to export data efficiently, serves as a repository for the amassed vibration data. The structured data export functionality ensures accessibility and convenience in retrieving datasets for subsequent analysis and utilization.

3.3.1 Flowchart of Program Code of ESP32

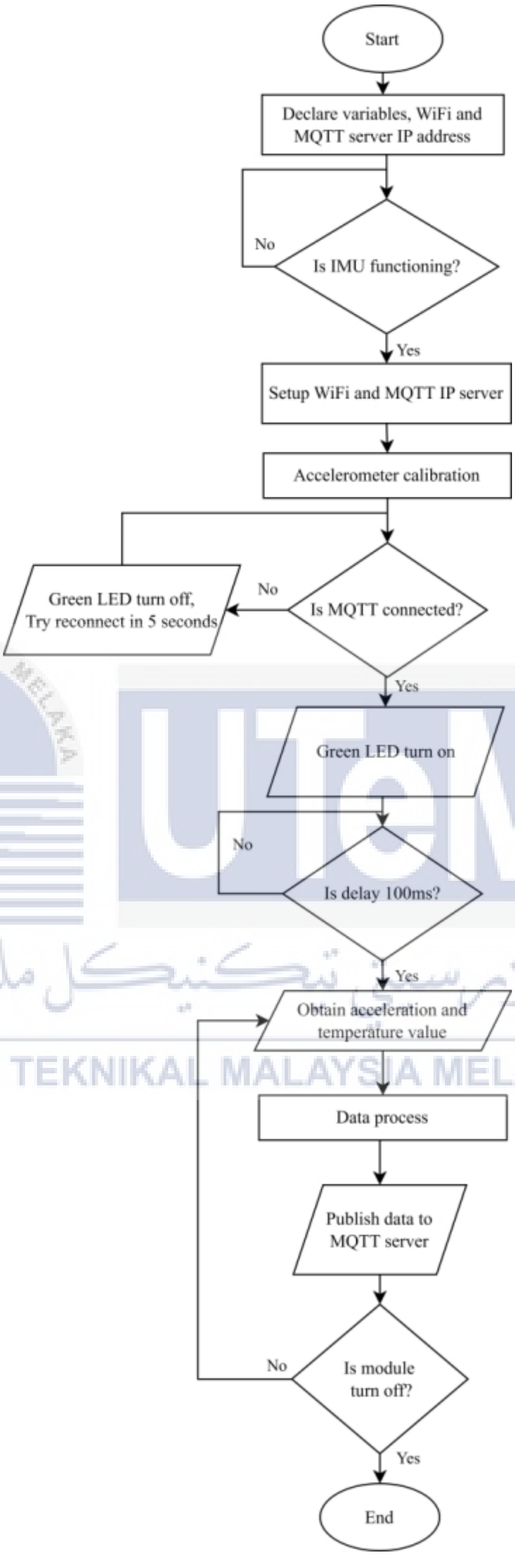

Start
Declare variables, WiFi and MQTT server IP address
Is IMU functioning?
No
Yes
Setup WiFi and MQTT IP server
Accelerometer calibration
Is MQTT connected?
No
Green LED turn off, Try reconnect in 5 seconds
Yes
Green LED turn on
Is delay 100ms?
No
Yes
Obtain acceleration and temperature value
Data process
Publish data to MQTT server
Is module turn off?
No
Yes
End

Figure 3.3 Program code of ESP32 microcontroller

First, all the variables, WiFi and MQTT server IP address, were declared. After that, the IMU was tested first. If IMU is not functioning, it will loop back until the IMU is successfully detected or functions. After that, WiFi and MQTT client servers were set up to build the WiFi and MQTT connection between Raspberry Pi and the module itself.

For the following step, the accelerometer was calibrated by using the average method. At the calibration phase, it will return a mean value that will be subtracted from the instantaneous reading afterwards. Next, the MQTT connection was connected. If the MQTT fails to connect, the green LED will turn off, and the system will try to reconnect after 5 seconds. This process will keep looping until the MQTT is connected.

When MQTT is successfully connected, the green LED will turn on as an indicator for the user. The system will then run for a 100ms delay. The millis() function was used in this system instead of delay(). After a 100ms delay, the module will obtain the acceleration and temperature value and go through a data process such as subtracting calibration value and changing from float type variable into string type variable. The string type variable was a better variable type to be sent with the MQTT protocol. The data will keep sending until the module is powered off. If the module keeps powering, the sensor will continue to obtain the acceleration and temperature data.

3.3.2 Flowchart of Node-RED Flow

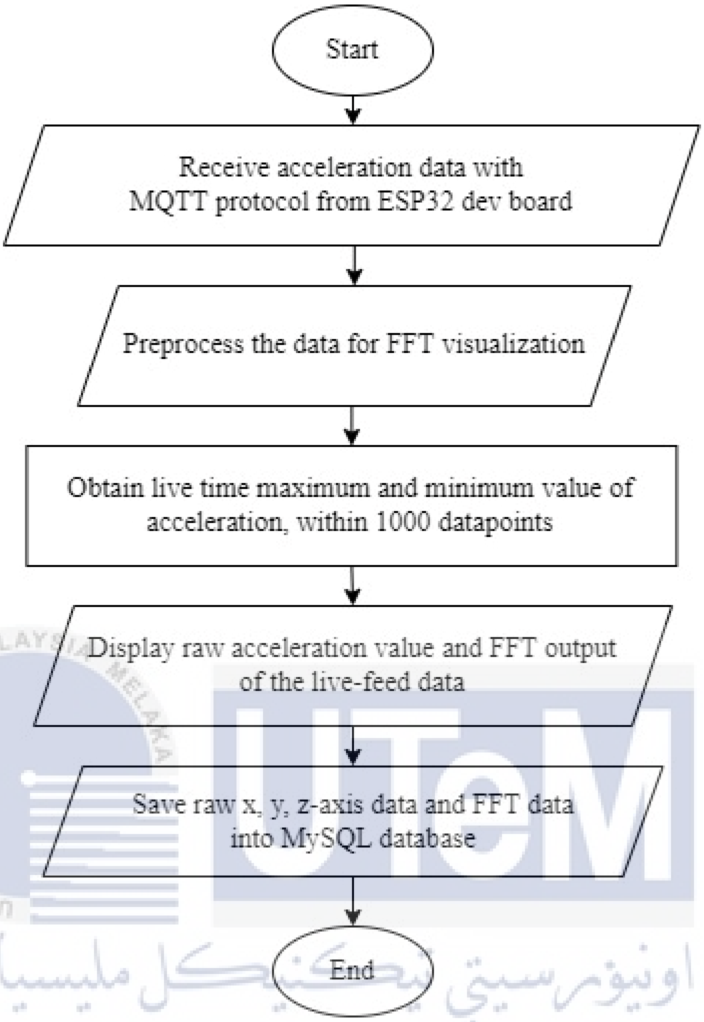

Start
Receive acceleration data with MQTT protocol from ESP32 dev board
Preprocess the data for FFT visualization
Obtain live time maximum and minimum value of acceleration, within 1000 datapoints
Display raw acceleration value and FFT output of the live-feed data
Save raw x, y, z-axis data and FFT data into MySQL database
End

Figure 3.4 Node-RED flow design

The successful development of Node-RED is evident in its seamless execution of expected functions, as depicted in Figure 4.3. Within this Node-RED flow, data acquisition is facilitated through the MQTT node, serving as the initial ingress point for the raw vibration data in terms of acceleration. Subsequently, this data is dynamically plotted onto the user interface in real time, providing a live visualization of the ongoing vibrations.

Following the real-time plotting, a filtering mechanism is implemented on the acquired data, refining it to 1000 datapoints. This filtering process serves to eliminate extremities, enhancing the clarity and interpretability of the real-time data, thereby affording more nuanced insights.

Simultaneously, the raw vibration data undergoes real-time computation for Fast Fourier Transform (FFT). The resultant frequency domain information is extracted and promptly displayed on the user interface. This dual visualization of time and frequency domains provides a comprehensive understanding of the vibration characteristics.

Concurrently, a data storage procedure is executed, with the collected data seamlessly integrated into a MySQL database. The stored dataset encompasses x, y, and z-axis data, along with FFT data from all three axes. This integrated approach not only facilitates real-time analysis but also ensures a comprehensive archival of relevant vibration data for subsequent in-depth examinations.

3.4 Build Deep Learning Model

Within this subsection, a comprehensive elucidation of the deep learning model construction process is provided. The focal deep learning model employed in this undertaking is the LSTM autoencoder, specifically designed for anomaly detection. The entire spectrum of data preprocessing and model development transpired utilizing the Python programming language within the Jupyter notebook environment.

The orchestration of data preprocessing encompasses a series of intricate steps aimed at refining and organizing the input data for optimal utilization within the

LSTM autoencoder architecture. Subsequently, the Python programming language, along with the Jupyter notebook interface, was instrumental in executing the intricate process of model creation. The LSTM autoencoder model, a key component of this project, was meticulously fashioned to discern anomalies within the acquired vibration data, underscoring its pivotal role in the overarching objective of anomaly detection.

3.4.1 Data Preprocessing

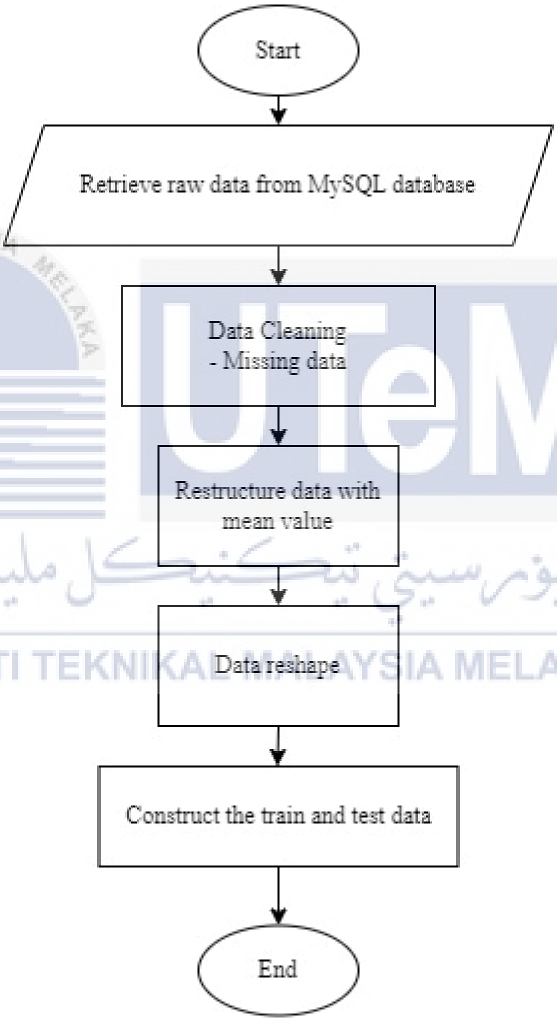

Start
Retrieve raw data from MySQL database
Data Cleaning
- Missing data
Restructure data with mean value
Data reshape
Construct the train and test data
End

Figure 3.5 Flow of data preprocessing

In the realm of data preprocessing, the initial step involves retrieving data from the wireless vibration monitoring system, where the data is stored in a MySQL database. Following this retrieval, a meticulous selection process is undertaken, culminating in the exportation of the data in the form of a .csv file. After this exportation, a critical phase of data cleaning transpires, wherein instances of NaN (indicating unknown values) are systematically expunged.

The data is then subjected to a restructuring process involving the segmentation of data into windows of 100 datapoints, with a singular mean value encapsulating each window. This deliberate reduction in datapoints to a scale of 100 serves a dual purpose: enhancing the efficiency of the subsequent training process by expediting data processing and ensuring the model's resilience to noise. The dataset undergoes further transformation through the application of the. reshape function, resulting in a structured format of (datapoints, timestamps, features).

Following this preparatory phase, the dataset is demarcated into distinct training and testing subsets. This demarcation is a pivotal precursor to the input phase for the LSTM autoencoder, facilitating the robust training and subsequent evaluation of the model's performance.

3.4.2 LSTM Autoencoder Model for Anomaly Detection

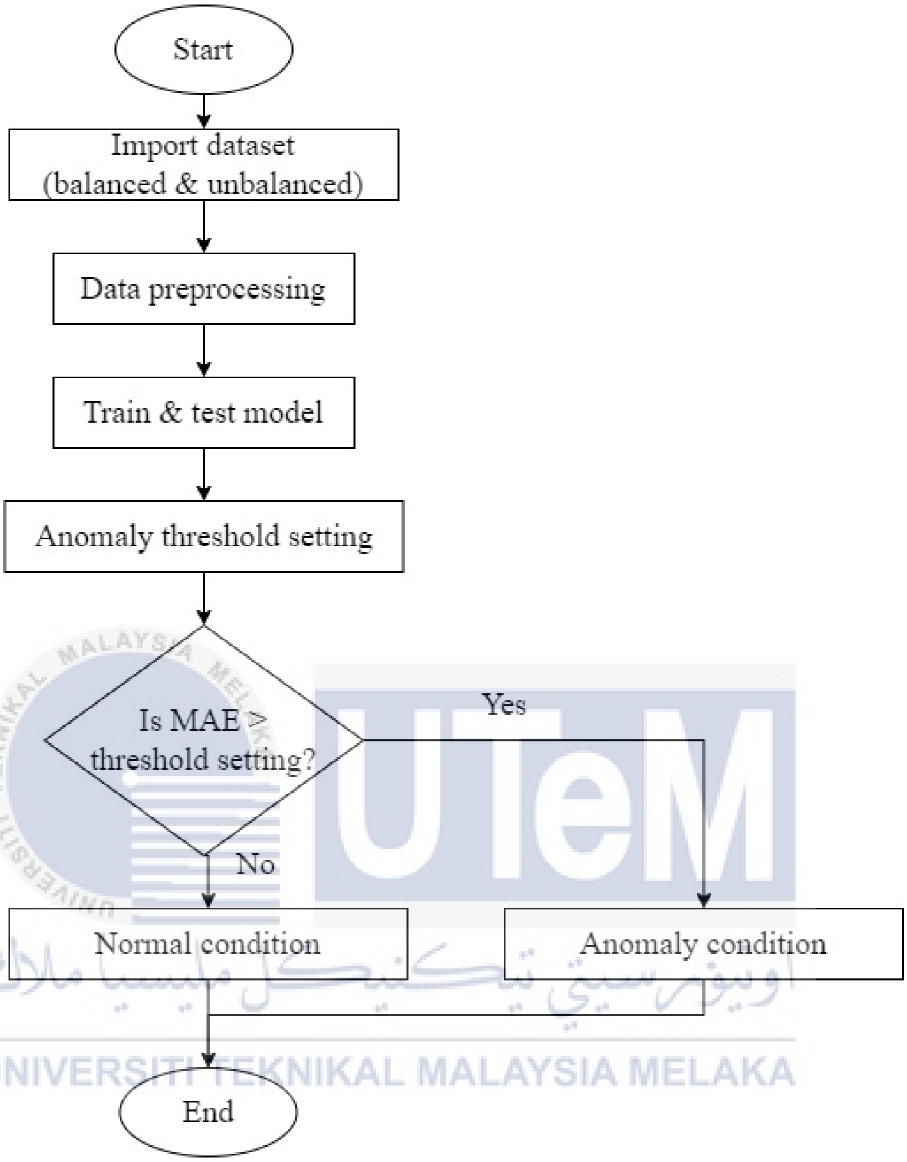

Start
Import dataset (balanced & unbalanced)
Data preprocessing
Train & test model
Anomaly threshold setting
Is MAE > threshold setting?
Yes
No
Normal condition
Anomaly condition
End

Figure 3.6 Flow of LSTM autoencoder model for anomaly detection

Within the framework of this project, the acceleration data underwent training using the LSTM autoencoder model. Commencing with the importation of data, a meticulous data preprocessing phase ensued, optimizing the dataset for subsequent utilization. Following this preparatory step, the model underwent training and testing, leveraging dedicated train and test datasets. The training process culminated in the establishment of an anomaly threshold, a critical parameter for discerning abnormal situations within the vibration data.

Armed with the acquired threshold value, the model adeptly identifies anomaly conditions during operational use. Specifically, when the Mean Absolute Error (MAE) surpasses the predefined threshold, it signifies an anomaly situation in the vibration data. Conversely, if the MAE falls below the established threshold, the model categorizes the condition as normal. This systematic classification mechanism underscores the model's proficiency in distinguishing between normal and anomalous states based on the threshold criterion.

3.5 Components

This subsection provides a comprehensive clarification of the components employed in the design of the wireless vibration monitoring system. The constituent elements of this system encompass the Hibiscus ESP32 microcontroller, Raspberry Pi 3, resistors, and Light Emitting Diodes (LEDs). Each component plays a pivotal role in the overall architecture and functionality of the monitoring system.

3.5.1 Hibiscus ESP32

The Hibiscus ESP32 is an Internet of Things (IoT) development board powered by the dual-core ESP32 microcontroller, embedded with three sensors (APDS9960, BME280 & MPU6050) and two actuators. This module also comes with WiFi and Bluetooth connectivity. The MPU6050 sensor is a 6 Degree of Freedom (DoF) IMU (Inertial Measurement Unit) which consists of an accelerometer and gyroscope. The temperature sensor is also embedded on MPU6050. It can sense acceleration in 3-axis, which is X, Y and Z. Arduino IDE software can be used to program this module. Similary, the green LED is connected to pin 4 of this module.

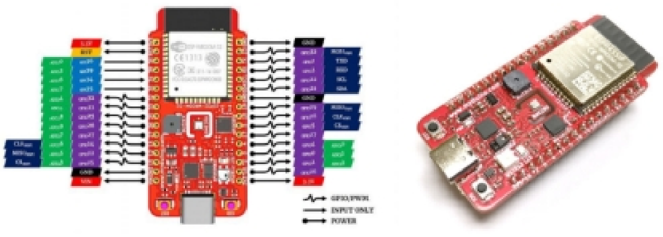


Figure 3.7 Pinout of Hibiscus ESP32

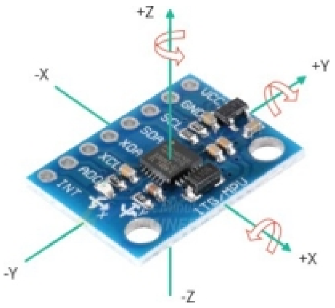


Figure 3.8 Sensors MPU6050 embedded on ESP32 module

3.5.2 Raspberry Pi 3

The Raspberry Pi 3 is equipped with a quad-core 64-bit Broadcom BCM2837 ARM Cortex-A53 SoC processor running at 1.2 GHz. It is one type of small single-board computer (SBCs) that comes with WiFi and Bluetooth connectivity. Therefore, it can be used for the Internet of Thing (IoT) application. In this project, the Raspberry Pi 3 acts as a broker to communicate with Arduino Nano RP2040, Hibiscus ESP 32 and laptop simultaneously.

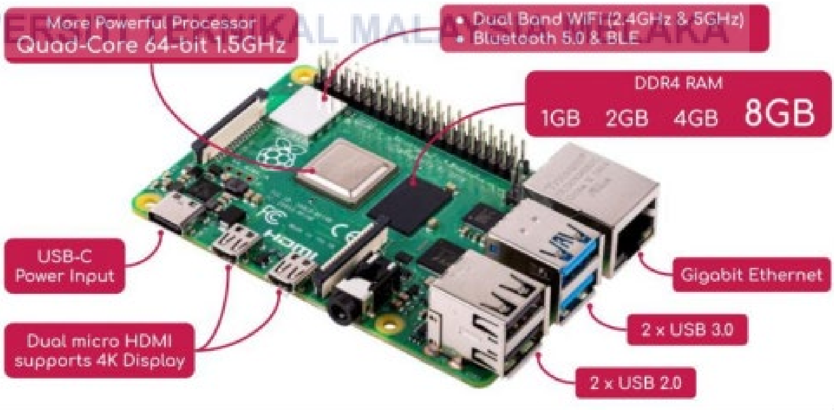


Figure 3.9 Raspberry Pi 3

3.5.3 Resistor

Resistor is a passive two-terminal electrical component that reduces the electric current. The ability of resistor to reduce the current is known as resistance and is shown in units of ohms (symbol: Ω). In this system, a 220Ω resistor is connected in before the green LED to prevent excess current flows which can burn out the LED.

3.5.4 Light Emitting Diode (LED)

The "Light Emitting Diode", known as LED is a specialised type of diode as they have similar electrical characteristics to a PN junction diode. It will pass current in forward bias direction but block the flow of current in the reverse bias direction. LED function as converting electrical energy into light. In this project, one green LEDs are used for each sensor module to indicate the MQTT connection.

3.6 Autoencoder Model

This subsection delves into the theoretical underpinnings and pertinent knowledge associated with the LSTM autoencoder. The initial section provides an elucidation of the autoencoder architecture incorporating Long Short-Term Memory (LSTM) components. Subsequently, a detailed exploration ensues regarding the design and equations governing the LSTM cell, shedding light on the intricate mechanisms that contribute to its efficacy in capturing temporal dependencies.

The autoencoder, when employing LSTM, is structured to encode input data into a latent space representation, and subsequently decode it back to the original form. This iterative process facilitates the extraction and retention of essential temporal features inherent in sequential data, making it particularly adept for applications such as anomaly detection.

Within the LSTM cell design, a comprehensive exposition is provided, encompassing the inherent gates and memory cells. The equations governing the LSTM cell elucidate the intricate interactions within the network, contributing to its capacity for capturing long-term dependencies and mitigating issues like vanishing gradients encountered in conventional recurrent neural networks.

In the final segment, attention is directed towards the choice of the Mean Absolute Error (MAE) as the loss function for model performance analysis. This metric, known for its robustness and interpretability, quantifies the disparity between predicted and actual values, serving as a pivotal gauge for the model's fidelity in capturing the nuances of the input data.

3.6.1 Framework Overview

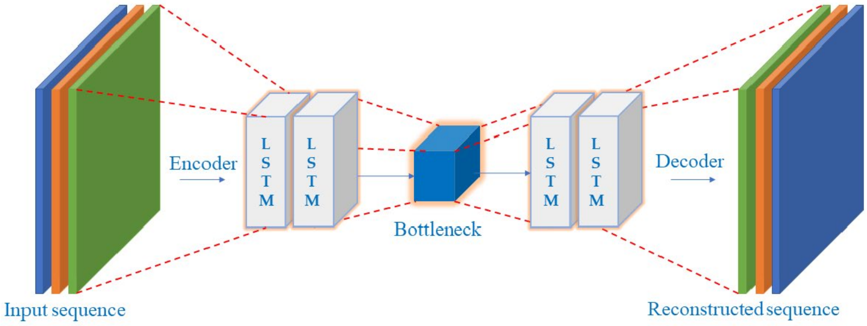


Figure 3.10 LSTM based Autoencoder model

The fault diagnosis methodology employed in this project is grounded in the synergistic application of LSTM and AE architectures. As depicted in Fig. 3.10, the AE comprises an encoder and a decoder, enveloping the input sequence and its corresponding reconstructed sequence. Positioned centrally is a bottleneck, strategically situated between the encoder and decoder components. Introducing LSTM layers between the encoder and decoder fortifies the model's capability to distill salient features from the input data.

The incorporation of LSTM layers facilitates the extraction of pivotal features by dynamically adjusting weight allocations. This adaptive learning process significantly enhances the efficacy of fault diagnosis. The distinctive advantage of this training paradigm lies in its ability to discern and extract key features essential for the accurate reconstruction of the raw vibration signal. Consequently, this feature extraction mechanism proves instrumental in distinguishing between normal and abnormal conditions in rotary parts, thereby optimizing the fault diagnosis efficacy within the proposed framework.

3.6.2 LSTM Cell

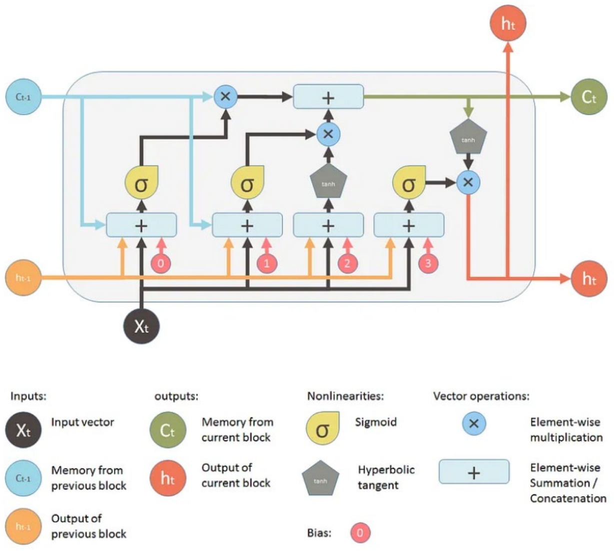


Figure 3.11 LSTM cell

LSTM is a great architecture for learning the characteristics of time-series data by comparing to RNN. LSTM model was first proposed by Ma et al. [31]. In the conventional neural network framework, there existed interconnections among nodes spanning adjacent layers, with no connections established within the same layer. Nevertheless, when dealing with time-series data, the correlation among the data points tends to be robust. Consequently, the conventional neural network architecture proves less adept at efficiently capturing the data's features in the context of time-series information, primarily due to the absence of connections among nodes within each layer.

LSTM addresses the issue inherent in the RNN model, wherein there is a propensity to forget past information. During the feature extraction process using nonlinear transformations, the input data in RNN tends to undergo substantial changes. Fig. 3.11 shows the structure of the LSTM cell.

A crucial aspect of the LSTM model design involves the incorporation of the cell state, forget gate, input gate, and output gate structures. The operational principles and specific calculation processes for these gate structures are elaborated upon in the following detailed description.

3.6.2.1 Forget Gate

The LSTM network merges the hidden layer h_{t-1} and the input x_t to form $[h_{t-1}, x_t]$. The vector f_t is then computed to determine which information should be "forgotten" from the cell state C_{t-1}at time $t-1$. The forget gate's control function is responsible for this calculation as shown in equation (2).

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

Where,

$$\sigma\ = sigmoid\,function$$

$$W_f = weight\,vector\,of\,forget\,gate$$

$$b_f\ = offset\,value\,of\,forget\,gate$$

3.6.2.2 Input Gate

The input gate computes the cell state $\widetilde{C_t}$ to be input, considering the input information $[h_{t-1}, x_t]$. Simultaneously, it calculates the vector it to govern which information will be incorporated into the cell state $\widetilde{C_t}$. The control function of the input gate is responsible for managing this process is shown in equation (3) and (4).

$$\widetilde{C_t} = \tanh(W_c\ \cdot\ [h_{t-1}, x_t] +\ b_c) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

Where,

$\widetilde{C_t} = cell\ state\ value\ to\ be\ input\ from\ new\ input$

$W_c = the\ updated\ weight\ of\ the\ cell\ state$

$b_c = the\ cell\ state\ bias\ value$

$\tanh = hyperbolic\ tangent\ activation\ function$

$i_t = output\ vector$

$W_i = weight\ matrix$

$b_i = bias\ value\ of\ the\ input\ gate$

3.6.2.3 Cell Status

The update of the cell state is determined at time $t-1$. Following the computation of the forget gate ft and the determination of information C_{t-1} to be forgotten, the cell state $\widetilde{C_t}$ is generated based on the data at time t. Utilizing the calculation results of the input gate it, a decision is made to ascertain which information can be input. The update function of the cell state C_t at time t is then carried out as in equation (5).

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t} \tag{5}$$

3.6.2.4 Output Gate

The output result ot of the LSTM model is computed through the "output gate." Subsequently, the cell state Ct at time t determines the information within the output model that is ultimately exported, leading to the final output model result, ht. The control function of the output gate is shown in equation (6) and (7).

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

$$h_t = o_t \times \tanh(C_t) \tag{7}$$

Where,

$o_t = output\ vectors$

$W_o = weight\ vectors$

$b_o = offset\ values$

$h_t = output\ value\ of\ the\ LSTM\ model\ at\ time\ t$

3.6.3 Loss function (MSE)

In this model, the loss function chosen for model training is Mean Absolute Error (MAE). The MAE is a metric used to measure the average absolute differences between predicted values and actual values in a regression problem. In the context of neural network models, which are often employed for regression tasks, the MAE calculates the average absolute discrepancies between the predicted and true values for each data point in the dataset. Mathematically, it is computed as the mean of the absolute differences between the predicted and actual values. Unlike other regression loss functions such as Mean Squared Error (MSE), MAE is less sensitive to outliers, making it a robust choice when the dataset may contain noisy or extreme values. In essence, the MAE provides a straightforward and interpretable measure of the average prediction error, making it a suitable choice for regression problems where the emphasis is on minimizing the absolute differences between predicted and actual values. The equation of MAE is shown in equation (8).

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{8}$$

$MAE = mean\ absolute\ error$

$y_i = prediction$

$x_i = true\ value$

$n = total\ number\ of\ data\ points$

3.7 Gear Test Experiments

Within this subsection, the collection of acceleration data is meticulously detailed, alongside insights into the design and testing apparatus. The testbed, tailored to simulate both normal and abnormal load conditions, encompasses essential components essential for a comprehensive analysis. The constituent elements of the testbed include a water pump motor, pillow blocks, shaft, coupling, plungers, and the load itself. To ensure stability and precision during simulations, all components are securely affixed to a flat wooden tile.

3.7.1 Components of Testbed

In designing the testbed, several essential components are required, including:

- Water pump motor (APM37)
- Pillow block bearing UCP201
- Aluminium shaft 30cm
- Aluminium coupler
- Flange motor shaft coupling 12mm
- Disc plate

3.7.2 Testbed Design

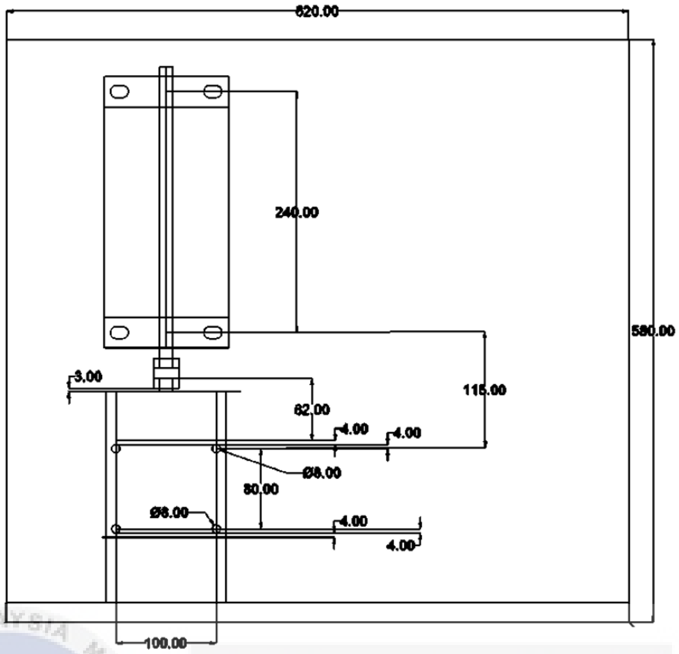


Figure 3.12 AutoCAD drawing for testbed design

Prior to the construction of the testbed, meticulous measurements were conducted to ascertain the precise distances between individual components, ensuring their accurate placement within the designated configuration. This preliminary step, critical for the structural integrity and functionality of the testbed, serves to prevent any deviations from the intended design.

In the creation of the AutoCAD design, a meticulous process unfolded, wherein the dimensions of each component were measured with precision using rulers and measuring tape. This comprehensive measurement regime facilitated the accurate representation of the testbed in the AutoCAD software. The top view of the AutoCAD design as shown in Fig. 3.12, incorporating the measured dimensions, provided a clear and detailed illustration of the component lengths and the spatial arrangement within the designated area.

By incorporating these meticulous measurements and AutoCAD design protocols, the construction process was not only streamlined but also guaranteed the adherence to the intended specifications. This approach ensures that the components remain securely in place, fulfilling their roles effectively within the overall framework of the testbed design.

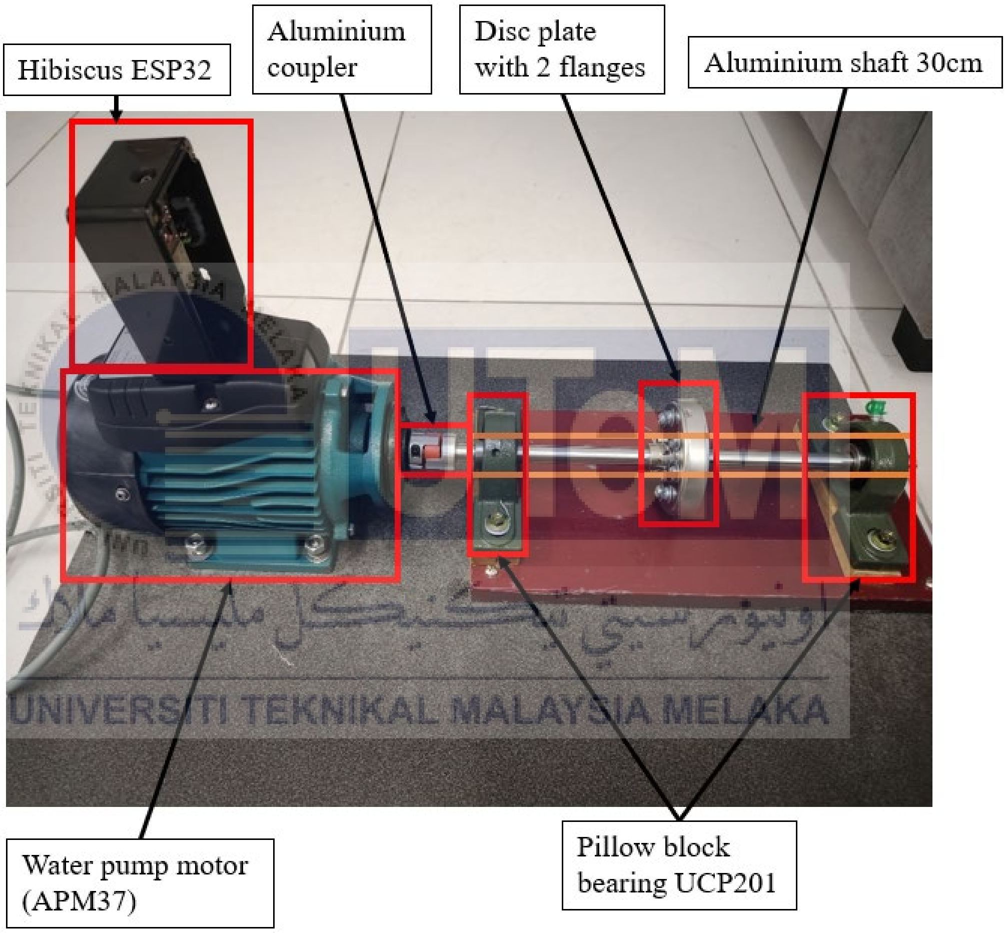


Figure 3.13 Testbed design

Fig.3.13 shows the testbed that had been constructed. The water pump motor (APM37) serves as the primary source of mechanical energy, initiating the rotational motion within the system. Pillow blocks and shafts contribute to the structural integrity and alignment of the system, while the coupling facilitates the transfer of energy between components. Flanges serve a pivotal role in securing the disc plate in a stable position within the assembly. Illustrated in Fig. 3.15, the disc plate incorporates a design comprising 8 bolts and nuts strategically positioned on the middle ring. These bolts and nuts collectively act as the counterweight for the disc plate, influencing its dynamic behavior within the system. Central to the disc plate is a 12mm diameter aperture, designed to accommodate the installation of a 12mm diameter aluminum shaft.

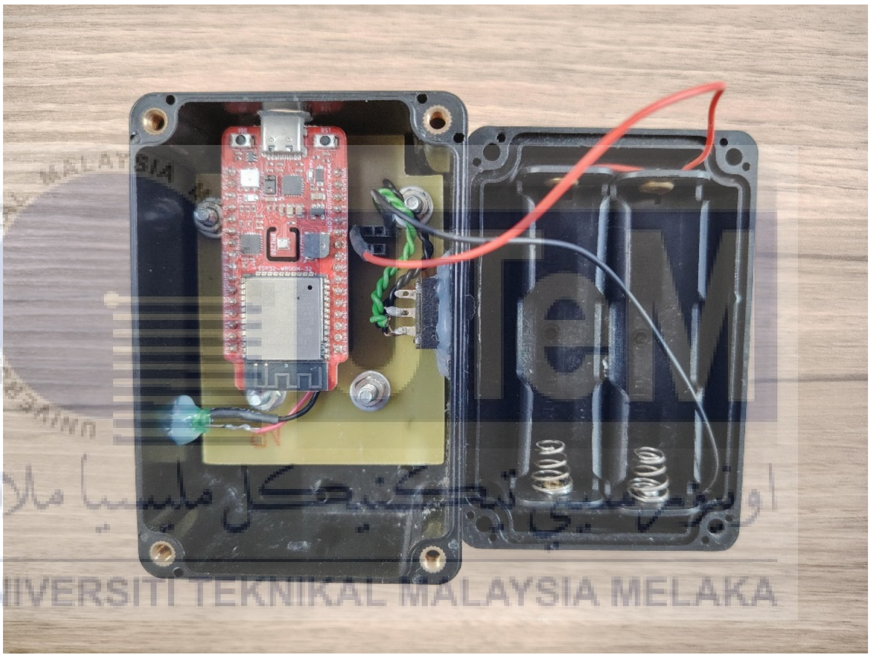


Figure 3.14 Hibiscus ESP32 accelerometer

Facilitating further structural integrity, four holes with a diameter of 3mm each are strategically positioned around the central aperture. These holes play a crucial role by providing anchor points for bolts and nuts, facilitating the secure attachment of the disc plate to the flanges. This meticulous arrangement ensures the robust connection between the disc plate and flanges, emphasizing the importance of precision engineering to uphold the stability and functionality of the overall system.

The meticulous fixation of all components onto a flat wooden tile serves a dual purpose: providing substantial weight for stability during simulations and ensuring a consistent platform for accurate data collection. This systematically designed testbed stands as a reliable foundation for the simulation of normal and abnormal load conditions, facilitating the acquisition of pertinent acceleration data essential for the project's objectives.

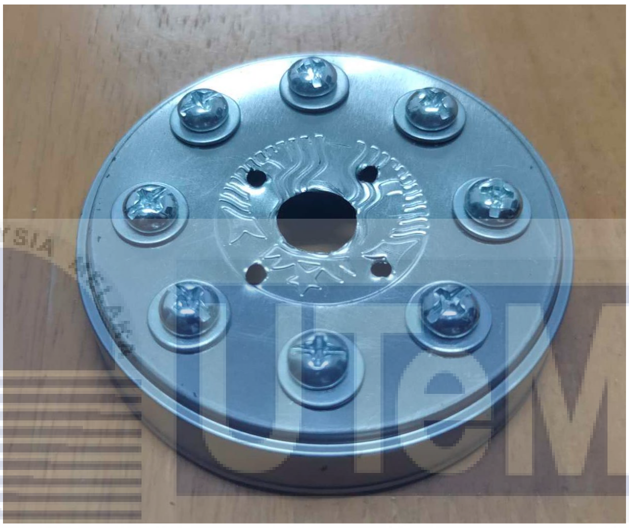


Figure 3.15 Disc plate used in testbed

3.7.3 Experiment Setup

The data collection was done with the experiment setup. This experiment setup able to obtain different type of dataset for analysis. The experiment had been list down in Table 3.1.

Table 3.1: Experiment setup with different conditions

Experiment	Conditions	Speed(rpm)	Data collected
1	No shaft and no load	2900	Acceleration (m/s^2)
2	Shaft only	2900	Acceleration (m/s^2)
3	Balanced load	2900	Acceleration (m/s^2)
4	Unbalanced load	2900	Acceleration (m/s^2)

The first experiment collected the data from the motor purely without any attachment to it. It helps to understand the conditions and operating frequency of the motor. Next, the second experiment acts as the control for the testbed design. The second experiment was carried out with shaft attached on the motor shaft, but without the load. For the third and fourth experiment, the load was attached to the shaft. The difference between both experiments was the weight distribution of the disc plate. For the balanced load, the 8 holes of the disc plate were screwed with 8 bolts and nuts as shown in Figure 3.15. Meanwhile the unbalanced load condition, the 8 holes of the disc plate were partially screwed up, remain 4 holes empty as shown in Figure 3.16.

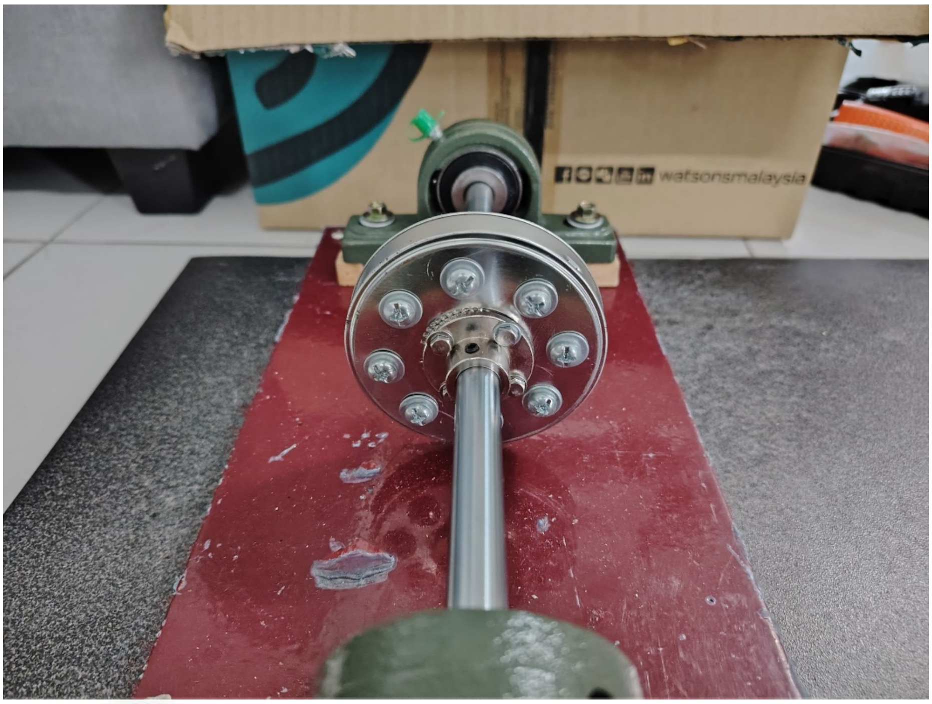


Figure 3.16 Balanced load condition of the testbed

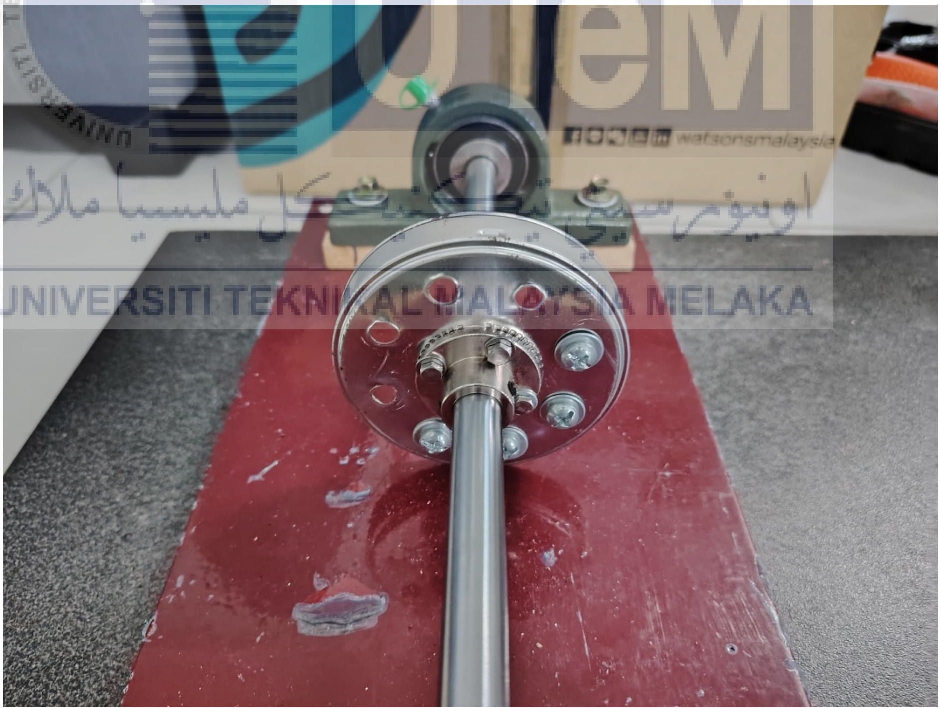


Figure 3.17 Unbalanced load condition of the testbed

For the collected data, the pure motor vibrations are used as validation purpose. Same goes to the data on second experiment. For the third and forth data, the data was collected and preprocess to be model training. Where the balanced load as the train dataset and the unbalanced load act as the test dataset.

In Figure 3.18, the acceleration data for the balanced condition is depicted, showcasing a rhythmic oscillation within the range of 7 to -7 m/s². In contrast, Figure 3.19 illustrates the acceleration data for the unbalanced condition, where the oscillation amplitude expands, fluctuating between 15 and -15 m/s². The unbalanced condition exhibits more vigorous vibrations, and the magnitude of the vibration is notably less stable.

Within the unbalanced condition, distinct spikes are observable at datapoints 312000 and 329000. These spikes exhibit significantly larger magnitudes compared to the relatively stable oscillations between datapoints 314000 and 328000. Following the occurrence of the second spike, the vibration becomes increasingly unstable, characterized by larger magnitudes and erratic behavior.

These visualizations provide a clear representation of the differing vibrational characteristics between the balanced and unbalanced conditions, aiding in the identification of anomalies and further emphasizing the significance of effective anomaly detection in this dynamic system.

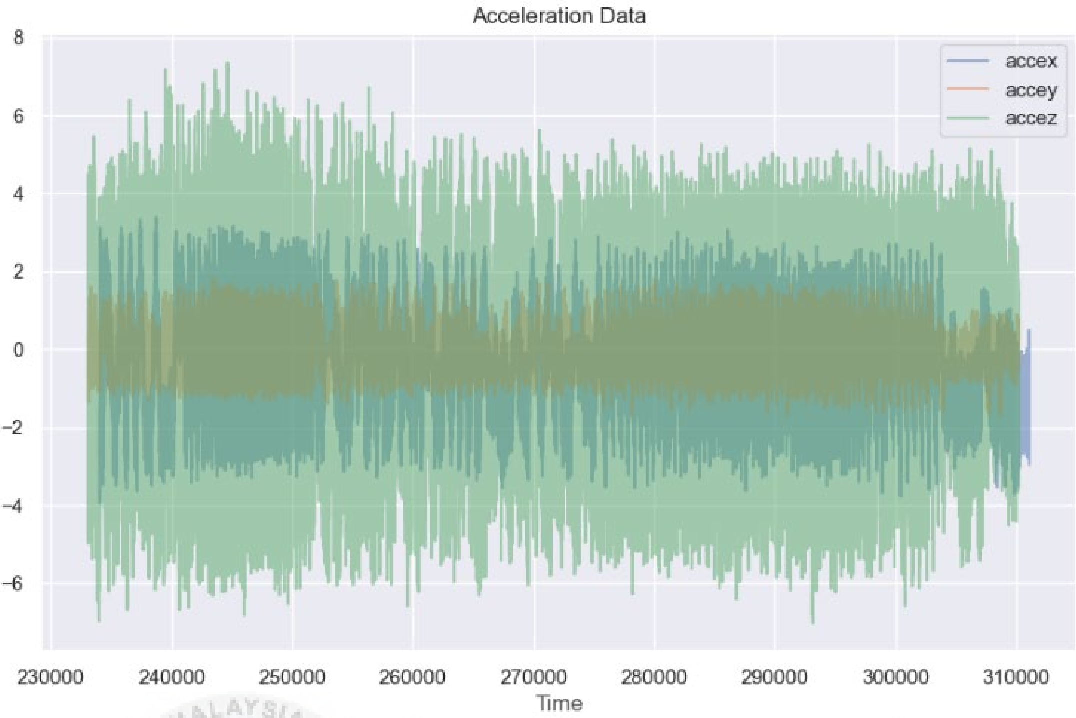


Figure 3.18 Acceleration Data of Balanced Condition

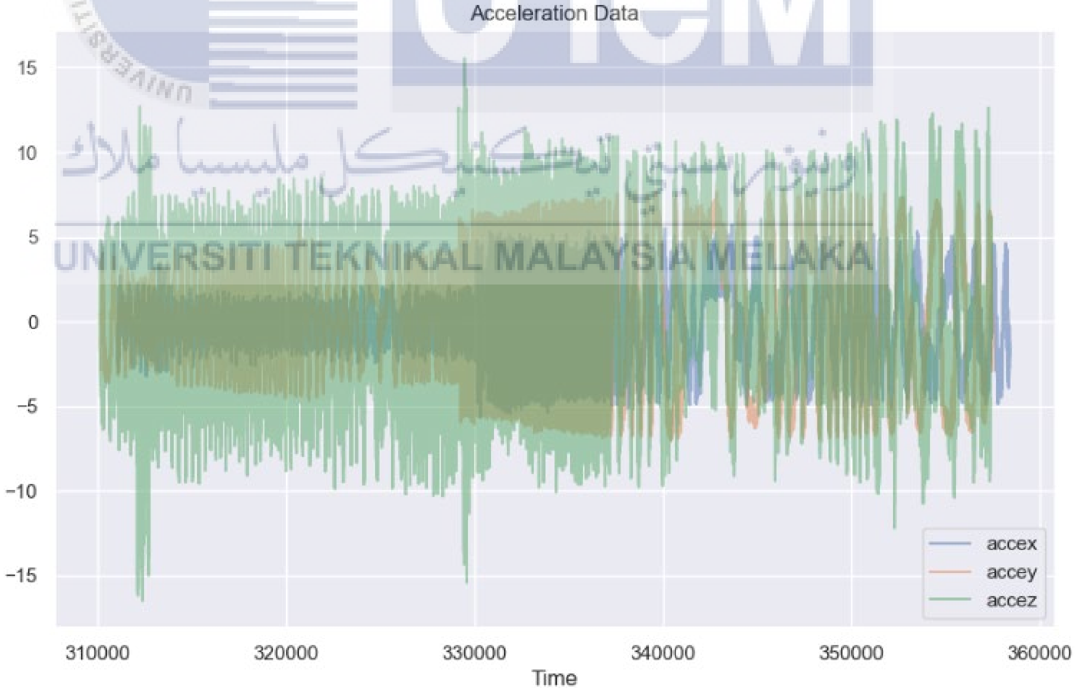


Figure 3.19 Acceleration Data of Unbalanced Condition

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, it will shows the result obtained that have been achieved throughout this project. Besides, it will also discuss about the result of the project based on testing of the finished project.

4.1 Calibration

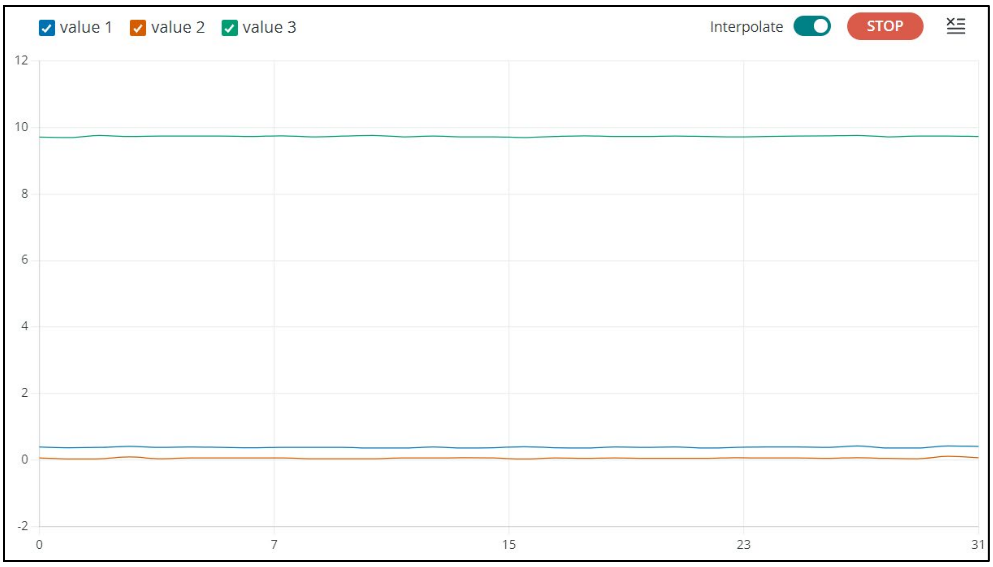


Figure 4.1 Acceleration data before calibration (value1 = x, value2 = y, value3 = z)

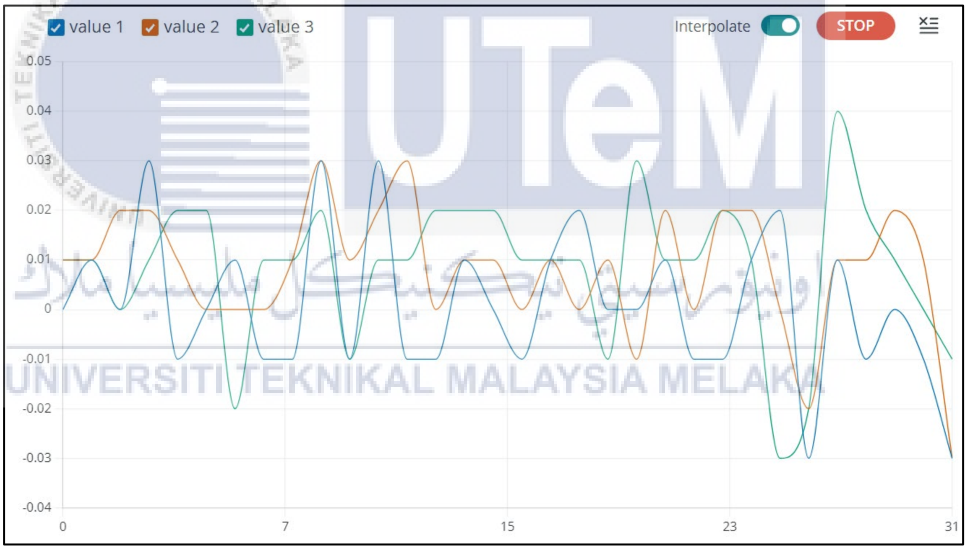


Figure 4.2 Acceleration data after calibration (value1 = x, value2 = y, value3 = z)

Every sensor has its offset and internal noise in the electronic world. With the offset presented on data, the measured data will fluctuate by the offset and cause the data to become unreliable. When the accelerometer MPU6050 embedded in ESP32 is turned on, the raw output of the acceleration is shown in Figure 4.1. The data showed the z-axis with nearly 0.8 offsets, the y-axis with close to 0 offsets and the z-axis with 0.4 offsets. If these offsets are not removed by calibration, the acceleration obtained has been included with these offsets. It makes the value becomes higher and more inaccurate. Hence, it is crucial to carry out the sensor calibration for the accelerometer to obtain accurate acceleration data.

The calibration is done by using the average method. When the module is set up, it will read 2000 samples and cumulatively add up all the measured values. After that, the cumulative value will be divided by 2000 to obtain the mean of the offset value. In this stage, the sensor will run calibration for 2 seconds. At the same time, the sensor also needs to be in a steady state, without any vibration, to obtain the accurate average value. The mean value is also called as calibrate value. Next, this value is subtracted from the acceleration data to get an accurate acceleration measurement. The calibrated data for the three axes had shown in Figure 4.2. It can be observed that the value of the three axes in the steady state had dropped to the range of 0.03 to -0.03, which is significantly low compared to the original offset.

4.2 Node-RED

Node-RED is an open-source flow-based development platform designed for visual programming of Internet of Things (IoT) and automation tasks. Developed by IBM Emerging Technology and later contributed to the open-source community, Node-RED simplifies the creation of applications by allowing users to wire together devices, APIs, and online services in a graphical interface. The platform utilizes a browser-based flow editor that enables users to drag and drop nodes, representing various functions or devices, and connect them to define the workflow. With a vast library of pre-built nodes and support for numerous protocols, Node-RED facilitates rapid development and deployment of IoT applications, making it an accessible tool for both beginners and experienced developers seeking a seamless and intuitive approach to

IoT and automation projects. In the subsection below will shows the code developed in this project and the user interface of the Node-RED.

4.2.1 Node-RED User Interface design

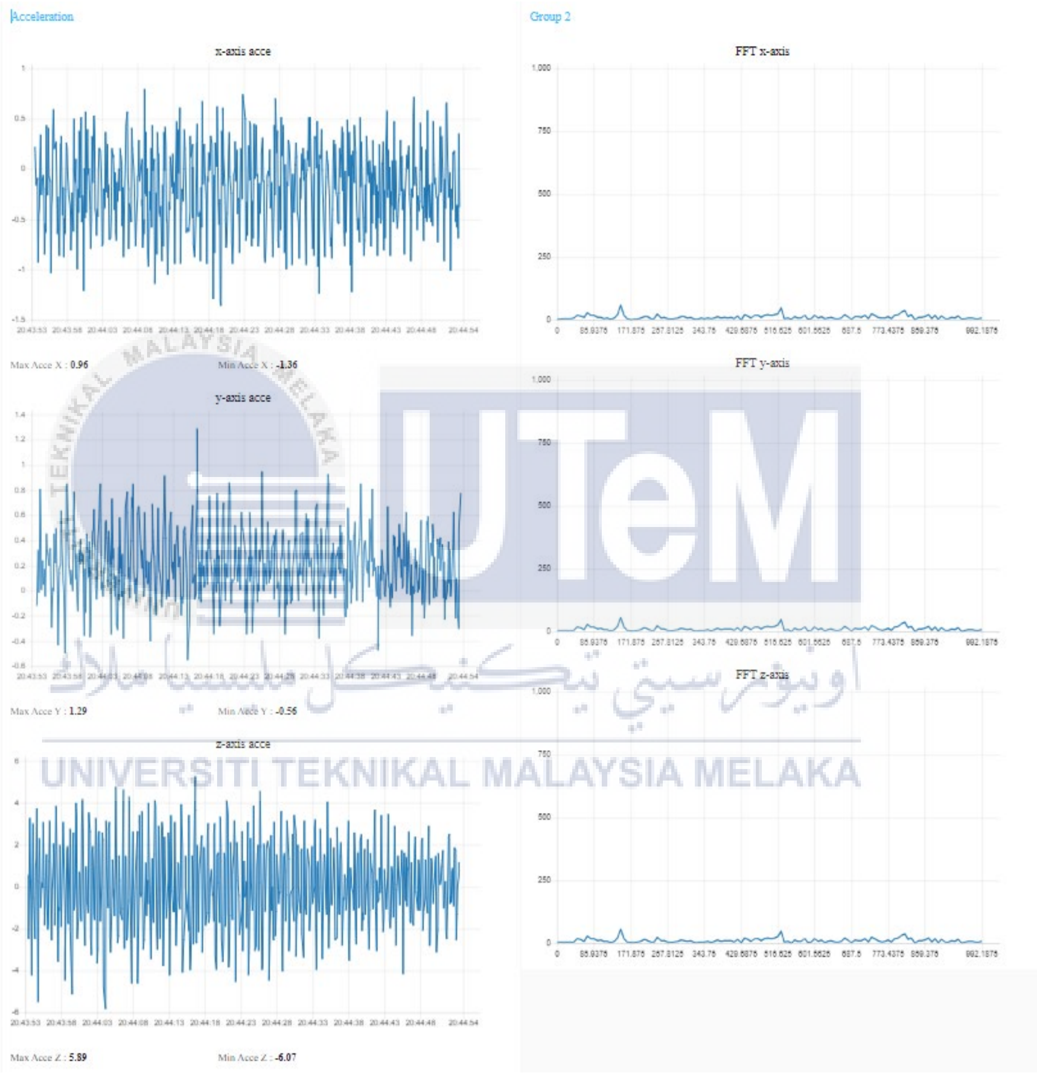


Figure 4.3 User Interface of Node-RED

As the data seamlessly traverses the designated flow, a user-friendly dashboard is generated and conveniently accessed by navigating to the designated website [http://Your_RPi_IP_address:1880/ui]. The structured dashboard, exemplified in

Figure 4.3, serves as an intuitive interface for users to monitor and analyze the real-time vibration data.

The left-hand section of the dashboard provides a comprehensive display of the real-time vibration data across three axes (x, y, z). This graphical representation facilitates a quick and dynamic assessment of the ongoing vibrational patterns, allowing for immediate insights into the system's behavior.

On the right-hand side of the dashboard, the frequency domain of the vibration signal is showcased. This visual representation contributes to a more nuanced understanding of the vibrational characteristics, offering insights into the various frequencies present in the acquired data.

The accessibility and clarity of this dashboard enhances the user experience, providing a centralized platform for real-time monitoring and analysis of the vibration data.

4.3 MySQL Database

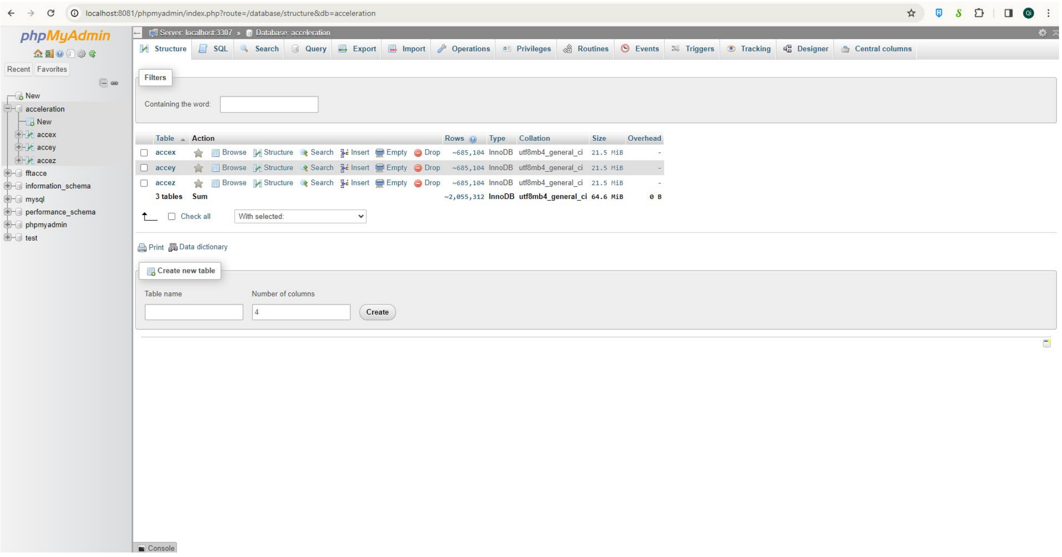


Table	Action	Rows	Type	Collation	Size	Overhead
accex	Browse Structure Search Insert Empty Drop	~685,104	InnoDB	utf8mb4_general_ci	21.5 MiB	-
accey	Browse Structure Search Insert Empty Drop	~685,104	InnoDB	utf8mb4_general_ci	21.5 MiB	-
accez	Browse Structure Search Insert Empty Drop	~685,104	InnoDB	utf8mb4_general_ci	21.5 MiB	-
3 tables	Sum	~2,055,312	InnoDB	utf8mb4_general_ci	64.6 MiB	0 B

Figure 4.4 MySQL database

After the data collection from the Node-RED, the data will stored in the MySQL database. The database was saved locally and can be accessed using phpMyAdmin, which supports HTML (website). In the MySQL database user interface as shown in Figure 4.4, the vibration data was saved in database [acceleration] with three different table named [accex, accey, accez].

4.4 Data Preprocessing

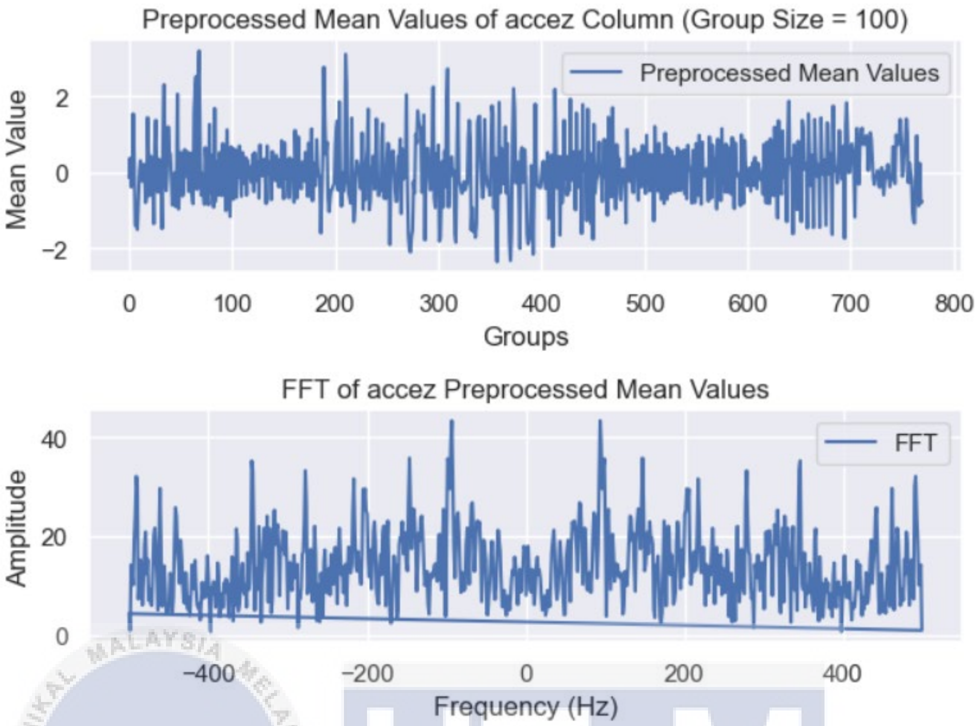


Figure 4.5 Data after preprocessing (balanced data)

Before initiating the model training process, a meticulous data preprocessing stage was executed, as illustrated in Figure 4.5. The top segment of the figure provides a visual representation of the preprocessed mean values of the z-axis acceleration data, whereas the bottom segment elucidates the frequency domain of the preprocessed data.

In the context of model training, the z-axis data was specifically chosen for its heightened reliability in capturing vibrations. This choice is attributed to the favorable orientation of the sensor in relation to the system dynamics. In the depicted figure, the balanced data exhibits a consistent vibration pattern, with values falling between 3 and -2. Notably, the major frequency component of the motor vibration is observed at 100Hz.

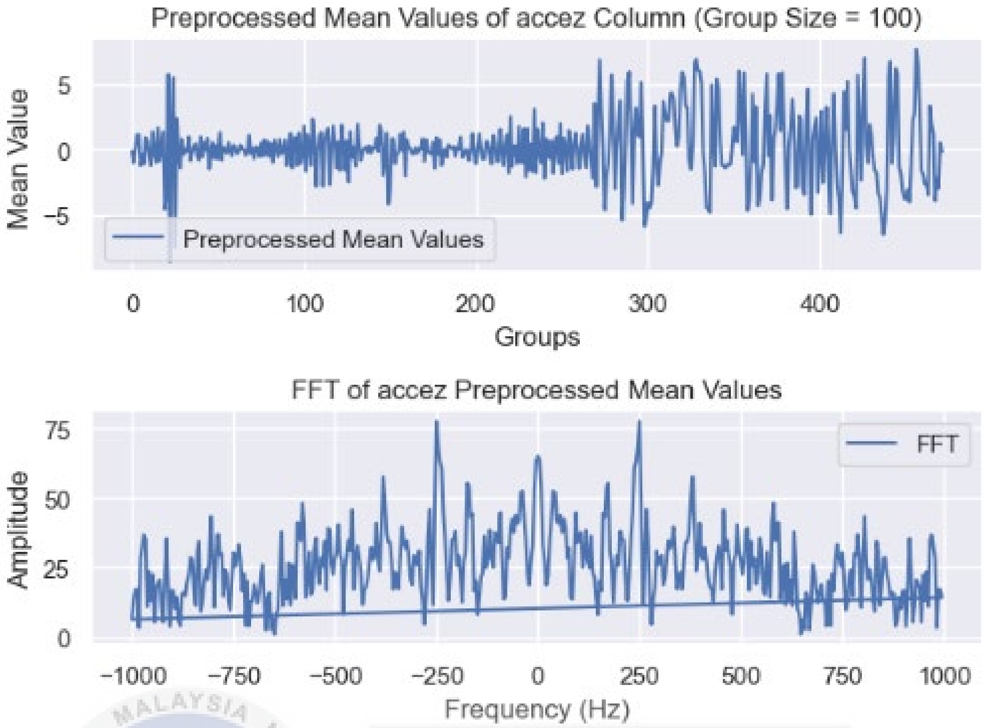


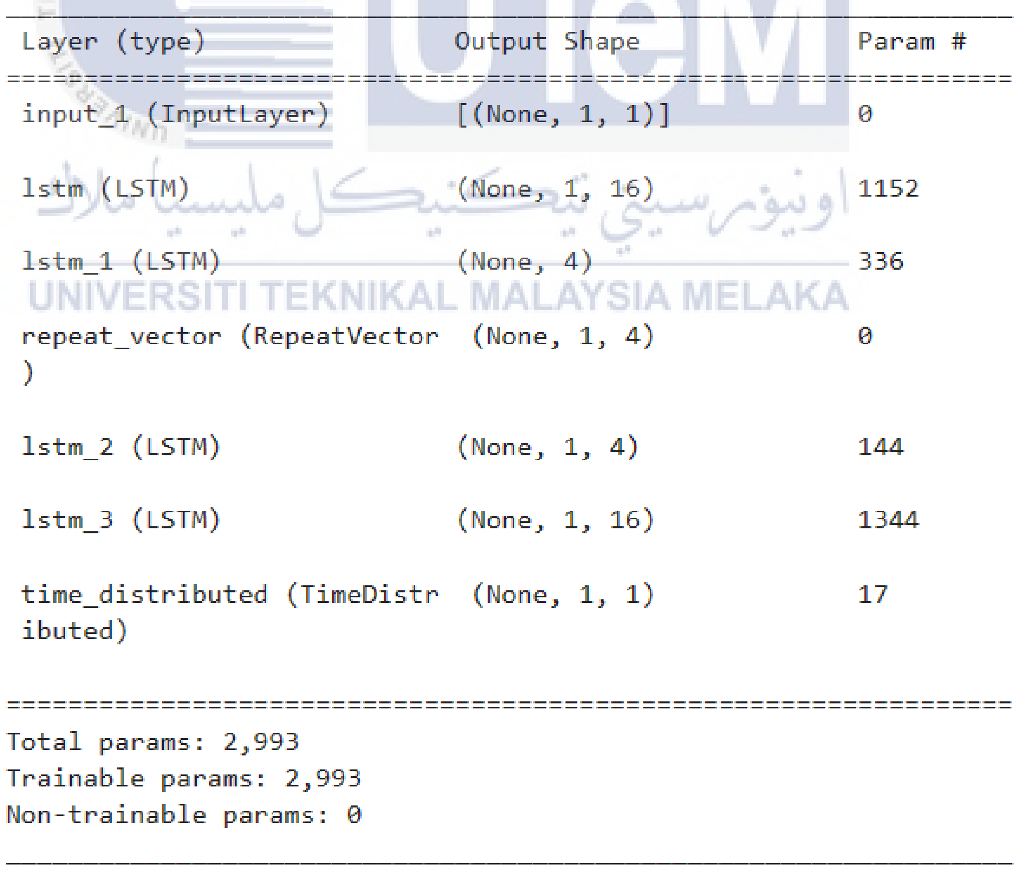
Figure 4.6 Data after preprocessing (unbalanced data)

The aftermath of preprocessing the unbalanced data is vividly depicted in Figure 4.6. The graph exhibits a notable escalation in the magnitude of acceleration, ranging from approximately 7 to -6, indicative of pronounced and unstable vibrations. Notably, at datapoints 20 and 270, discernible shifts in the vibration signal are evident, underscoring the dynamic nature of the system under unbalanced conditions.

Furthermore, an observable shift in the frequency of the system is discerned, with the dominant frequency gravitating towards 250Hz. This alteration in the frequency profile aligns with the expected behavior of the system when subjected to unbalanced conditions. The comprehensive insights derived from this representation facilitate a nuanced understanding of the vibrational characteristics under unbalanced scenarios, laying the groundwork for effective anomaly detection in subsequent stages of the

4.5 Autoencoder Model

The AE can learn the features of the data. With the combination of the LSTM layer, it can extract the important information carried by time series data. A LSTM AE model was developed with 2 layering of LSTM on encoder, with one bottleneck and 2 layering of LSTM on decoder. The model is shown in Figure 4.7. The input layer consists of shape of (none, 1, 1) where the none is the data input, with 1 timestamp and 1 feature to be learned. For the first layer, it consists of 16 neurons of LSTM cell followed by 4 neurons on the second layer. The repeat vector acts as the bottleneck and passes the encoded information to decoder. The decoder will reconstruct the signal back and make predictions with the input data received. The decoder also consists of two layer which consists of 4 neurons and 16 neurons.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 1, 1)]            0

 lstm (LSTM)                 (None, 1, 16)             1152

 lstm_1 (LSTM)               (None, 4)                 336

 repeat_vector (RepeatVector  (None, 1, 4)             0
 )

 lstm_2 (LSTM)               (None, 1, 4)              144

 lstm_3 (LSTM)               (None, 1, 16)             1344

 time_distributed (TimeDistr  (None, 1, 1)             17
 ibuted)

=================================================================
Total params: 2,993
Trainable params: 2,993
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.7 AE model

After the model train, the training had set to 100 epochs which take around 6ms per epoch. The model loses performance during training as shown in Figure 4.8. At the epoch 0 to 10, the model loss gradually decreased and around 20 epochs onward, the model loss floated around 0 and reach steady nearly to 0 at 100 epochs.

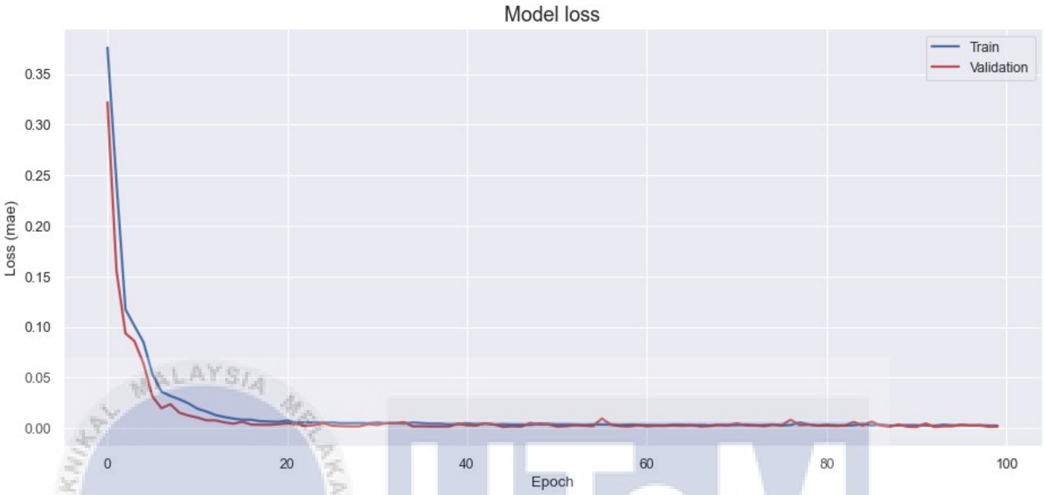


Figure 4.8 AE model loss respect to epoch times

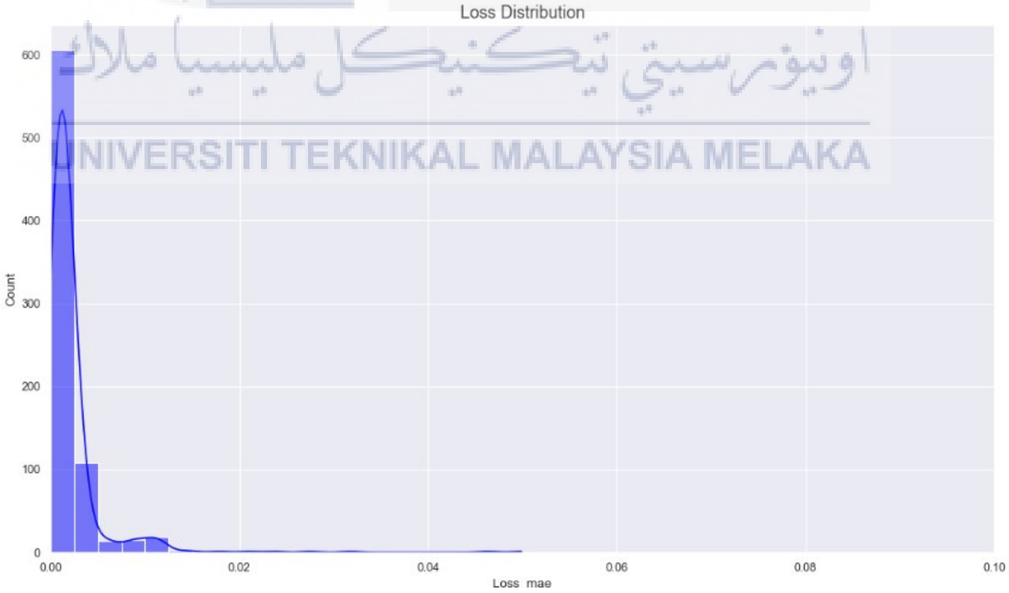


Figure 4.9 Loss distribution during training

After the model train, the loss distribution can be histogram can be plotted to understand more insight of the model. The loss distribution histogram is shown Figure

4.9. From Figure 4.9, the maximum loss distribution of the model train using the balanced data falls around 0.05. Most of the loss falls in 0.003. From this loss distribution, the threshold value for the AE model for identifying anomaly can be obtained and assigned as 0.06.

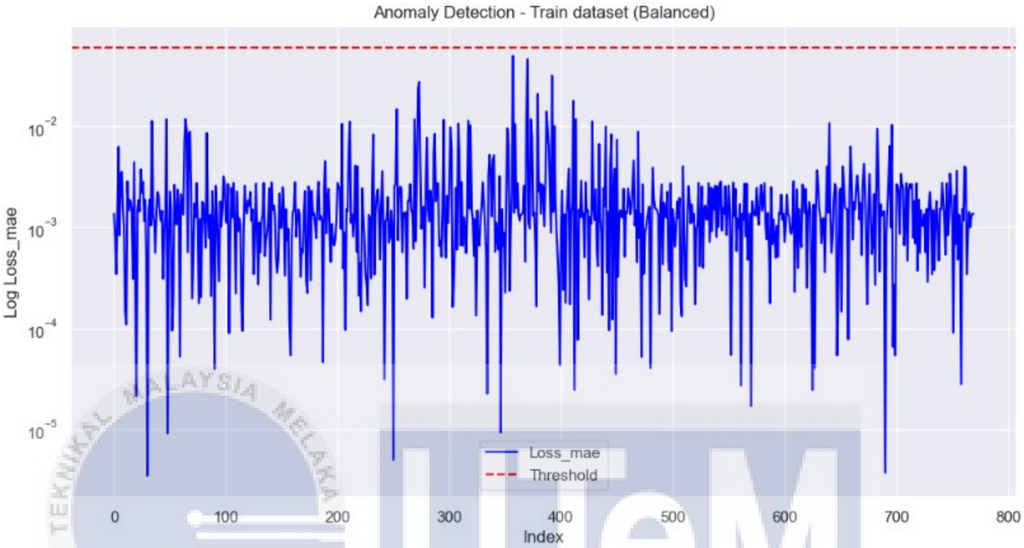


Figure 4.10 Loss MAE of training dataset

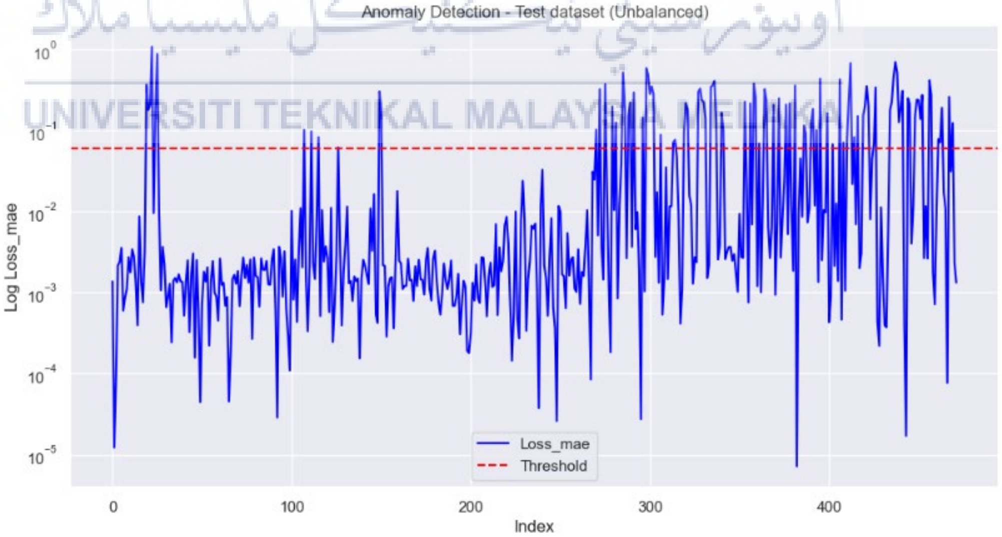


Figure 4.11 Loss MAE of test dataset

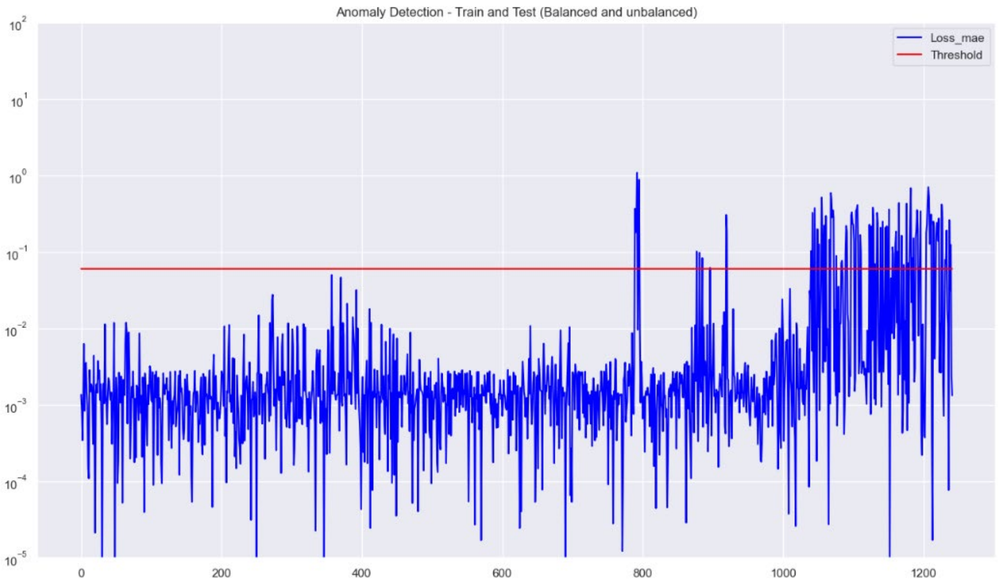
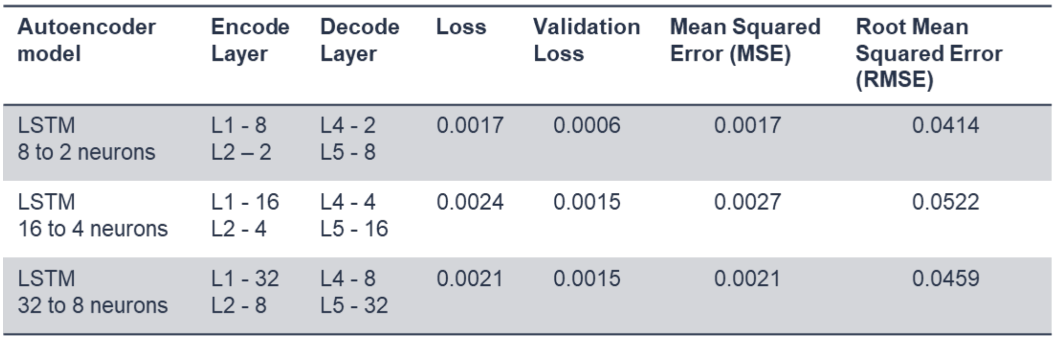


Figure 4.12 Loss MAE of train test dataset

After obtaining the threshold value, the threshold line can be plotted into the model loss refer to the train and test data. In Figure 4.10, it shows loss MAE of the train dataset falls under the threshold line, which the condition is flagged as normal condition. When the test data is fed to the LSTM AE model, the model loss had exceeded the threshold line as shown in Figure 4.11. With the test data, the unbalanced dataset, the anomaly can be flagged out using the threshold line. By concatenating the train and test, the result can be shown more clearly that the balanced dataset is falls under the threshold line while the unbalanced dataset had exceeded the threshold line as shown in Figure 4.12.

Table 4.1: Comparison between different parameters of AE model



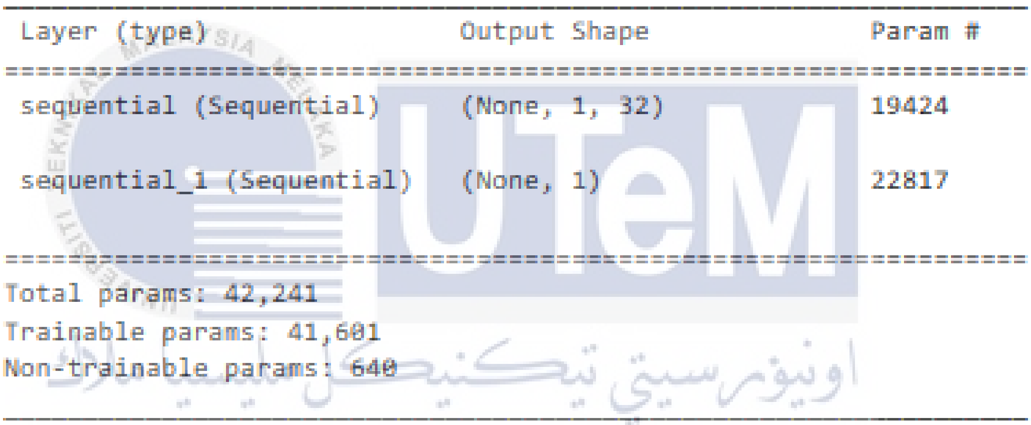
Autoencoder model	Encode Layer	Decode Layer	Loss	Validation Loss	Mean Squared Error (MSE)	Root Mean Squared Error (RMSE)
LSTM 8 to 2 neurons	L1 - 8 L2 – 2	L4 - 2 L5 - 8	0.0017	0.0006	0.0017	0.0414
LSTM 16 to 4 neurons	L1 - 16 L2 - 4	L4 - 4 L5 - 16	0.0024	0.0015	0.0027	0.0522
LSTM 32 to 8 neurons	L1 - 32 L2 - 8	L4 - 8 L5 - 32	0.0021	0.0015	0.0021	0.0459

For the next step, the evaluation of the model's performance is carried out. The LSTM AE had been designed using different parameters. As shown in Table 4.1, the neurons for the encoder and decoder were manipulated with the scale of 4. All models had been trained with similar dataset. The best model among the three models are the LSTM AE with the number of LSTM neurons of 8 on layer 1 and layer 5, and number of LSTM neurons of 2 on layer 2 and layer 4. The lowest loss obtained was 0.0017 with validation loss of 0.0006. The MSE falls on 0.0017 and the RMSE of this model is 0.0414.

Consequently, a detailed comparison between the CNN AE model and the LSTM AE model was conducted. The train and test datasets were obtained using the same dataset and underwent similar steps in the data preprocessing phase. The architectural components of the CNN AE model, namely the encoder and decoder, are visually represented in Figure 4.13.

Upon training the CNN model, the progression of the training loss was plotted, revealing insightful patterns, as depicted in Figure 4.14. The initial training loss commences at 0.19 and undergoes a gradual decrease to 0.11. However, as the epochs progress, the training loss exhibits fluctuations, ultimately settling at 0.11 towards the conclusion of the training process.

Notably, the validation loss of the CNN AE model displays a more dynamic pattern compared to the LSTM AE model. The validation loss experiences pronounced fluctuations, ranging from 0.35 to 0.03 throughout the entire training period. This dynamic behavior in the validation loss underscores the intricacies and challenges associated with training the CNN AE model, revealing nuances in its performance compared to the LSTM AE model.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 1, 32)             19424

 sequential_1 (Sequential)   (None, 1)                 22817

=================================================================
Total params: 42,241
Trainable params: 41,601
Non-trainable params: 640
_________________________________________________________________
```

Figure 4.13 CNN model

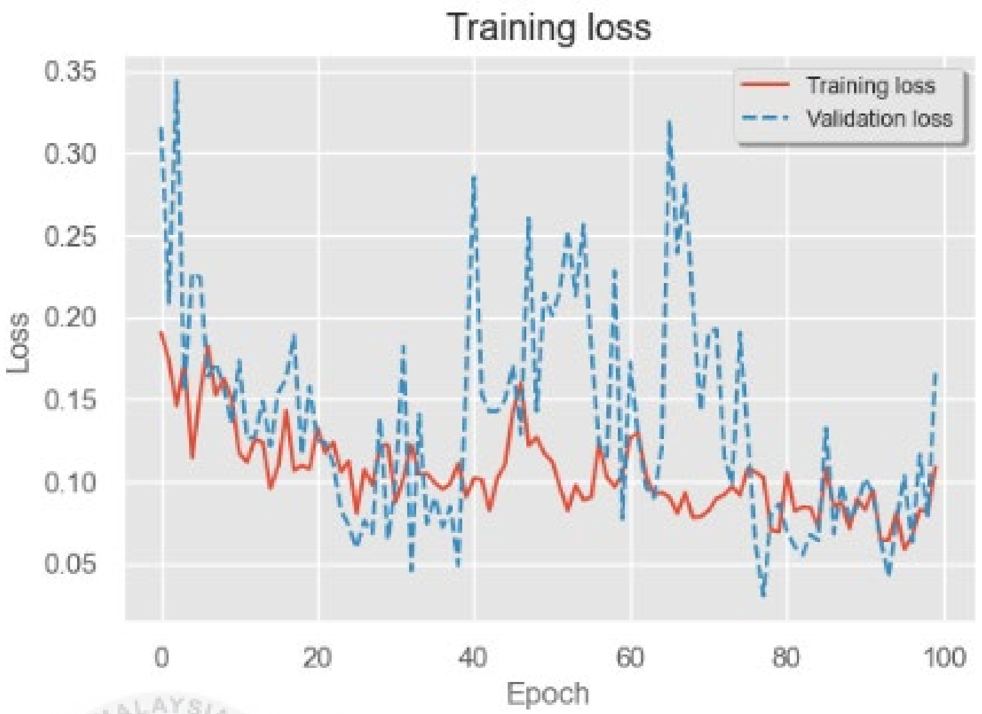


Figure 4.14 CNN model loss respect to epoch times

Moreover, the loss distribution graph has been meticulously plotted, incorporating the train, test, and anomaly datasets, as depicted in Figure 4.15. The graph illustrates the distribution of losses, with the train data showcasing a predominant concentration between 0.05 and gradually tapering off to 0 at the 0.4 loss mark. However, a notable observation is the reflection of losses towards the 0.8 mark, indicative of certain intricacies in the model's learning process.

Notably, when the anomaly dataset is introduced to the model, the loss distribution surpasses the threshold line, effectively flagging anomalies. This capability underscores the model's efficacy in identifying and isolating anomalous patterns within the data.

In comparison to the LSTM AE encoder, the train data does not exceed the threshold value, ensuring that balanced data remains unflagged as anomalies.

Additionally, the training loss for the LSTM AE model is notably lower at 0.0017 in contrast to the CNN AE model, underscoring differences in the performance and training characteristics of the two architectures.

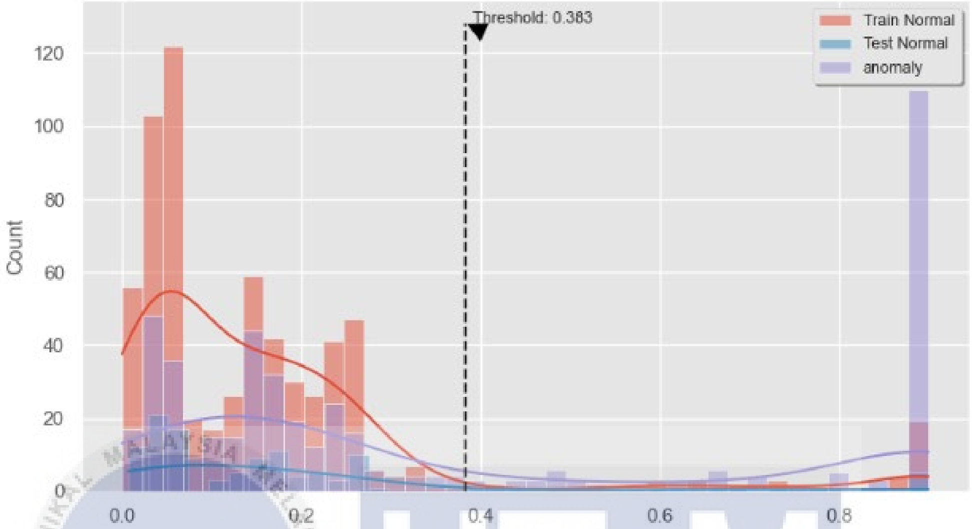


Figure 4.15 Loss distribution of training

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter will describe the conclusion and recommendation for the wireless vibration monitoring system for predictive maintenance. This section includes project summary, project finding and further recommendation to improve the project.

5.1 Conclusion

As a conclusion, a LSTM AE deep learning model had been successfully developed to analysis the vibration signal for the predictive maintenance. The proposed method can identify the potential anomaly of the motor, bearing or any vibration materials. With the help of the LSTM model, the fault can be immediately addressed and scheduled for potential fault before the machine or components run to failure.

This project holds significance across various industries with machinery containing rotating components. By facilitating the timely identification of anomalies in machinery, it enables efficient maintenance scheduling, thereby reducing costs associated with repairs or the replacement of faulty units.

Utilizing vibration analysis through deep learning, our predictive maintenance approach mitigates significant losses resulting from machine failures and associated downtime. This not only aligns with Sustainable Development Goal 8 (SDG8) by promoting sustained economic growth but also contributes to SDG9 by enhancing the reliability of system components through the identification of vibration anomalies.

The primary goal of the project was to design wireless vibration monitoring system. The wireless monitoring system had been successfully developed using Hibiscus ESP32, Raspberry Pi 3 and the platform Node-RED with the database platform, MySQL. Apart from that, a deep learning-based LSTM model had successfully developed to analysis the vibration data. The proposed method LSTM AE model can differentiate the normal and anomaly condition of the testbed motor. Furthermore, analyze and compare the performance of the LSTM model with parameter of model loss also carried out to identify the best LSTM AE model design and compare the LSTM AE model with the CNN AE model. The LSTM AE model with the 8 to 2 neurons design architecture surpass other design with lowest losses of 0.0017 and the loss also significantly lower than the CNN AE model.

5.2 Future Work

Although the LSTM AE model can identify the normal and anomaly conditions of certain applications such as motor, bearing, or any other rotary parts, it is unable to classify the type of fault present in the rotary parts. Moreover, this model unable to predict the RUL of the machine, where RUL also plays a main role in understanding the rotary parts behavior. This can be achieved by collecting more dataset by modifying the testbed to realize other fault conditions such as misalignment, bearing outer wear out, bearing inner wear out etc. For the RUL, the datasets need to collect start from a machine that operates from healthy to fail conditions. This may time consuming to obtained.

REFERENCES

[1] A. Jaafar, N. Soin, and S. W. M. Hatta, “An educational FPGA design process flow using Xilinx ISE 13.3 project navigator for students,” *2017 IEEE 13th Int. Colloq. Signal Process. its Appl.*, no. March, pp. 7–12, 2017.

[2] B. Zhang, L. Zhang, W. Deng, L. Jin, F. Chun, H. Pan, B. Gu, H. Zhang, Z. Lv, W. Yang, and Z. L. Wang, “Self-powered acceleration sensor based on Liquid Metal Triboelectric Nanogenerator for vibration monitoring,” ACS Nano, vol. 11, no. 7, pp. 7440–7446, 2017.

[3] B. Bengherbia, M. Ould Zmirli, A. Toubal, and A. Guessoum, “FPGA-based wireless sensor nodes for vibration monitoring system and fault diagnosis,” Measurement, vol. 101, pp. 81–92, 2017.

[4] Q. Huang, B. Tang, and L. Deng, “Development of high synchronous acquisition accuracy wireless sensor network for machine vibration monitoring,” Measurement, vol. 66, pp. 35–44, 2015.

[5] M Senthilkumar, M Vikram and B Pradeep, "Vibration Monitoring for Defect Diagnosis on a Machine Tool: A Comprehensive Case Study", International

Journal of Acoustics and Vibration., vol. 20, no. 1, 2015.

[6] N. D. R. Ajansı, “Predictive maintenance,” Miller Magazine. [Online]. Available: https://millermagazine.com/blog/predictive-maintenance-2763. [Accessed: 08-Jan-2023].

[7] Peter Poór, David Ženíšek, and Josef Basl, “Historical overview of Maintenance Management Strategies: Development ...,” 2019. [Online]. Available: http://ieomsociety.org/pilsen2019/papers/135.pdf. [Accessed: 12-Apr-2023].

[8] D. S. Chandra and Y. S. Rao, “Fault diagnosis of a double-row spherical roller bearing for induction motor using vibration monitoring technique,” Journal of Failure Analysis and Prevention, vol. 19, no. 4, pp. 1144–1152, 2019.

[9] Y. Jiang, W. Huang, J. Luo, and W. Wang, “An improved dynamic model of defective bearings considering the three-dimensional geometric relationship between the rolling element and defect area,” Mechanical Systems and Signal Processing, vol. 129, pp. 694–716, 2019.

[10] M. Cakir, M. A. Guvenc, and S. Mistikoglu, “The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIOT based Condition Monitoring System,” Computers &amp; Industrial Engineering, vol. 151, p. 106948, 2021.

[11] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, “Mqtt vulnerabilities, attack vectors and solutions in the internet of things (IOT),” IETE Journal of Research, vol. 69, no. 6, pp. 3368–3397, 2021.

doi:10.1080/03772063.2021.1912651

[12] M. R. Minar and J. Naher, “Recent advances in Deep learning: An overview,” arXiv.org, https://arxiv.org/abs/1807.08169 (accessed Jun. 8, 2023).

[13] Deep Learning in Medical Image Analysis - annual reviews, https://www.annualreviews.org/doi/abs/10.1146/annurev-bioeng-071516-044442 (accessed Jun. 7, 2023).

[14] M. Munoz-Organero, R. Ruiz-Blaquez, and L. Sánchez-Fernández, “Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving,” Computers, Environment and Urban Systems, vol. 68, pp. 1–8, 2018. doi:10.1016/j.compenvurbsys.2017.09.005

[15] J. Wehrmann, W. Becker, H. E. Cagnini, and R. C. Barros, “A character-based convolutional neural network for language-agnostic twitter sentiment analysis,” 2017 International Joint Conference on Neural Networks (IJCNN), 2017. doi:10.1109/ijcnn.2017.7966145

[16] T. Hori, J. Cho, and S. Watanabe, “End-to-end speech recognition with word-based RNN language models,” 2018 IEEE Spoken Language Technology Workshop (SLT), 2018. doi:10.1109/slt.2018.8639693

[17] S. S. Patil and J. A. Gaikwad, "Vibration analysis of electrical rotating machines using FFT: A method of predictive maintenance," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-6, doi: 10.1109/ICCCNT.2013.6726711.

[18] "Predictive maintenance using machine learning - javatpoint," www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/predictive-maintenance-using-machine-learning. [Accessed: 13-Apr-2023].

[19] L. Wang and Y. Shao, "Fault feature extraction of rotating machinery using a reweighted complete ensemble empirical mode decomposition with adaptive noise and demodulation analysis," Mechanical Systems and Signal Processing, vol. 138, p. 106545, 2020.

[20] X. E. Novelo and H.-Y. Chu, "Application of vibration analysis using time-frequency analysis to detect and predict mechanical failure during the nut manufacturing process," Advances in Mechanical Engineering, vol. 14, no. 2, p. 168781322210827, 2022.

[21] K. N. Ravikumar, A. Yadav, H. Kumar, K. V. Gangadharan, and A. V. Narasimhadhan, "Gearbox fault diagnosis based on multi-scale deep residual learning and stacked LSTM model," Measurement, vol. 186, p. 110099, 2021.

[22] J. S. Do, A. B. Kareem, and J.-W. Hur, "LSTM-Autoencoder for vibration anomaly detection in vertical carousel storage and Retrieval System (VCSRS)," Sensors, vol. 23, no. 2, p. 1009, 2023.

[23] K. T. P. Nguyen and K. Medjaher, "A new dynamic predictive maintenance framework using Deep Learning for failure prognostics," Reliability Engineering &amp; System Safety, vol. 188, pp. 251–262, 2019.

[24] X. Bampoula, G. Siaterlis, N. Nikolakis, and K. Alexopoulos, "A deep learning

model for predictive maintenance in cyber-physical production systems using LSTM autoencoders," Sensors, vol. 21, no. 3, p. 972, 2021.

[25] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 2018.

[26] D. P. Granados, M. A. Ruiz, J. M. Acosta, S. A. Lara, R. A. Domínguez, and P. J. Kañetas, "A wind turbine vibration monitoring system for predictive maintenance based on machine learning methods developed under safely controlled laboratory conditions," Energies, vol. 16, no. 5, p. 2290, 2023.

[27] M. Cakir, M. A. Guvenc, and S. Mistikoglu, "The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIOT based Condition Monitoring System," Computers &amp; Industrial Engineering, vol. 151, p. 106948, 2021.

[28] Erbessd Instruments, "PHANTOM Vibration Sensor," PE-DS2022.1013.11 datasheet, 2020.

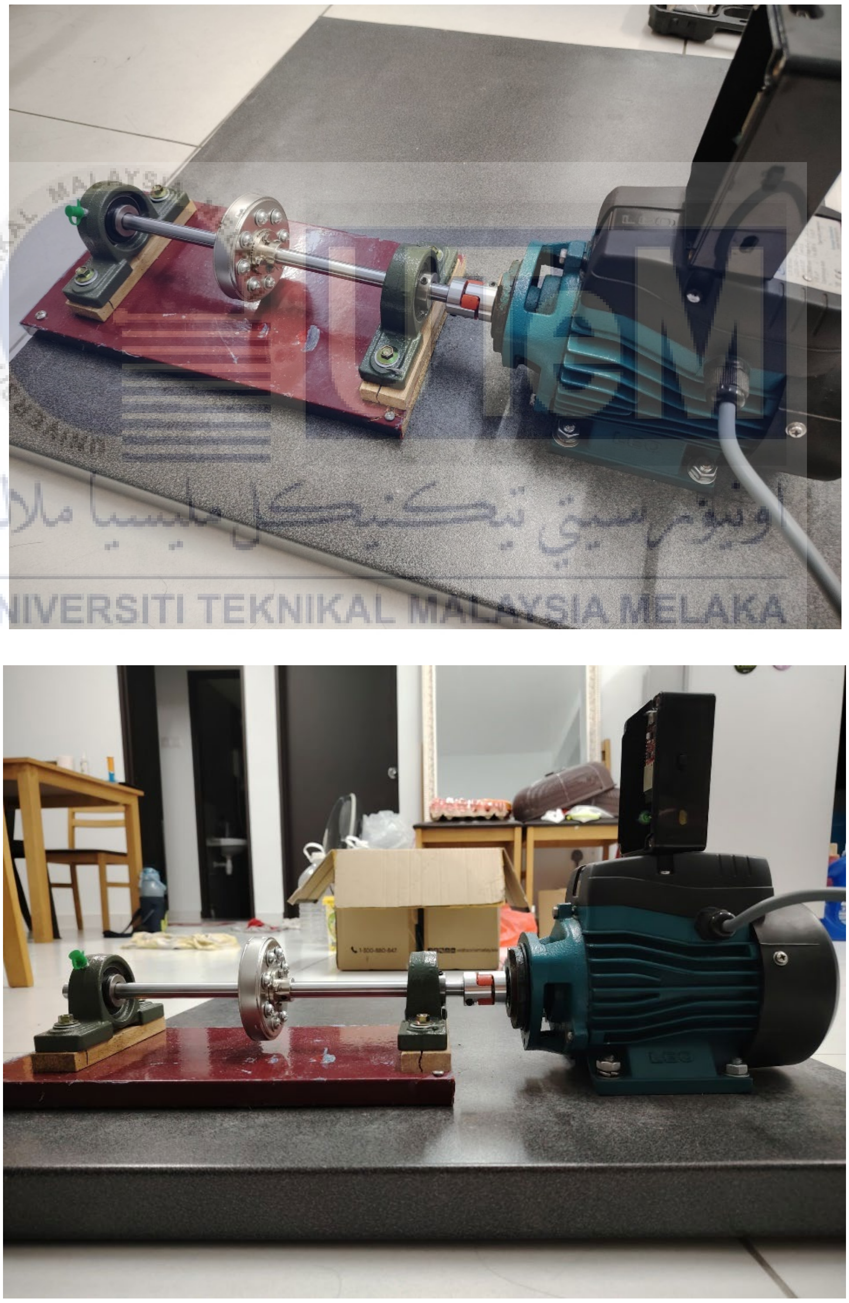
[29] Fluke, "Fluke 3561 FC Vibration Sensor and Fluke 3502 FC Gateway," 6011156b-en datasheet, 2019.

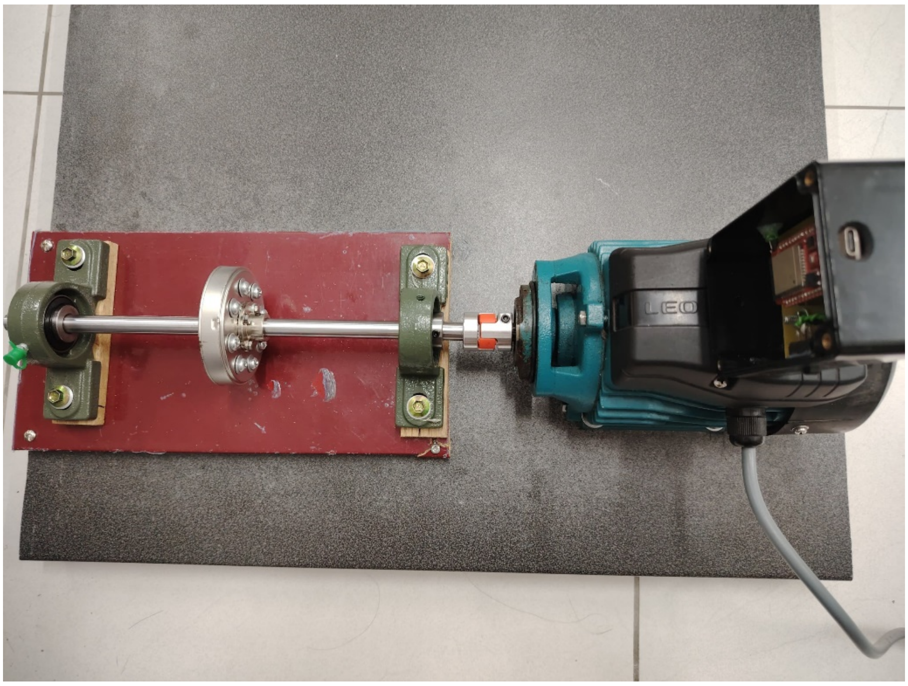
[30] Fluke, "Fluke 805 FC Vibration Meter," 13287-eng Rev 02 datasheet, 2015.

[31] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transp. Res. C, Emerg. Technol., vol. 54, pp. 187–197, May 2015.

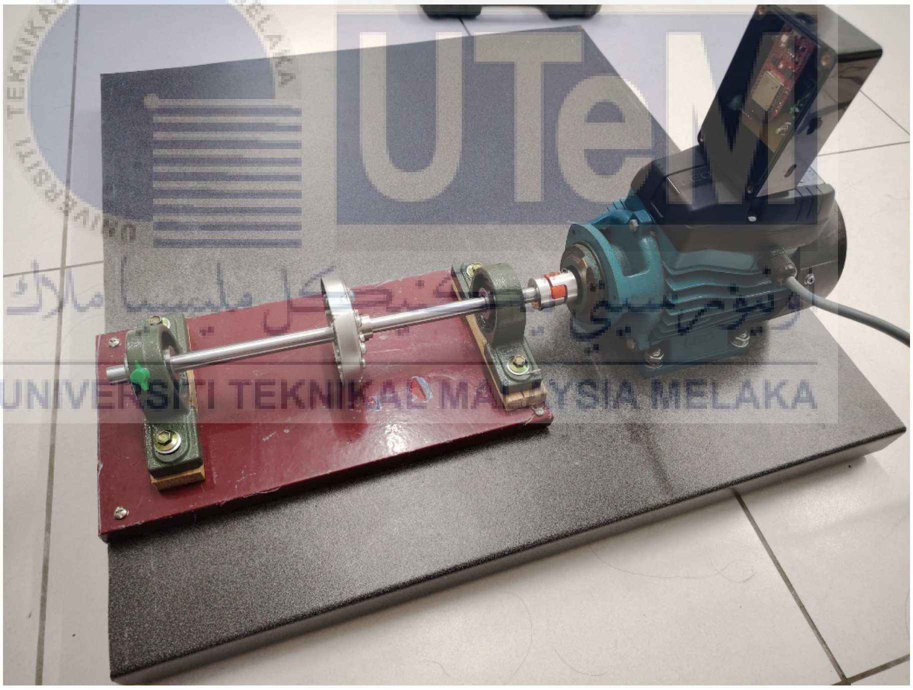
APPENDICES

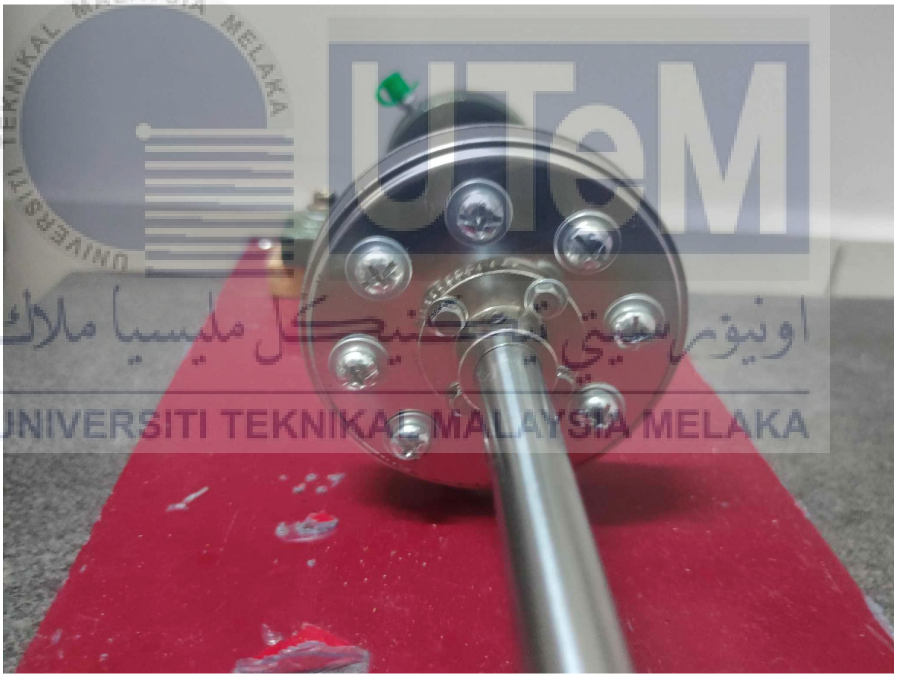
Appendix A: Different shoot angle of testbed



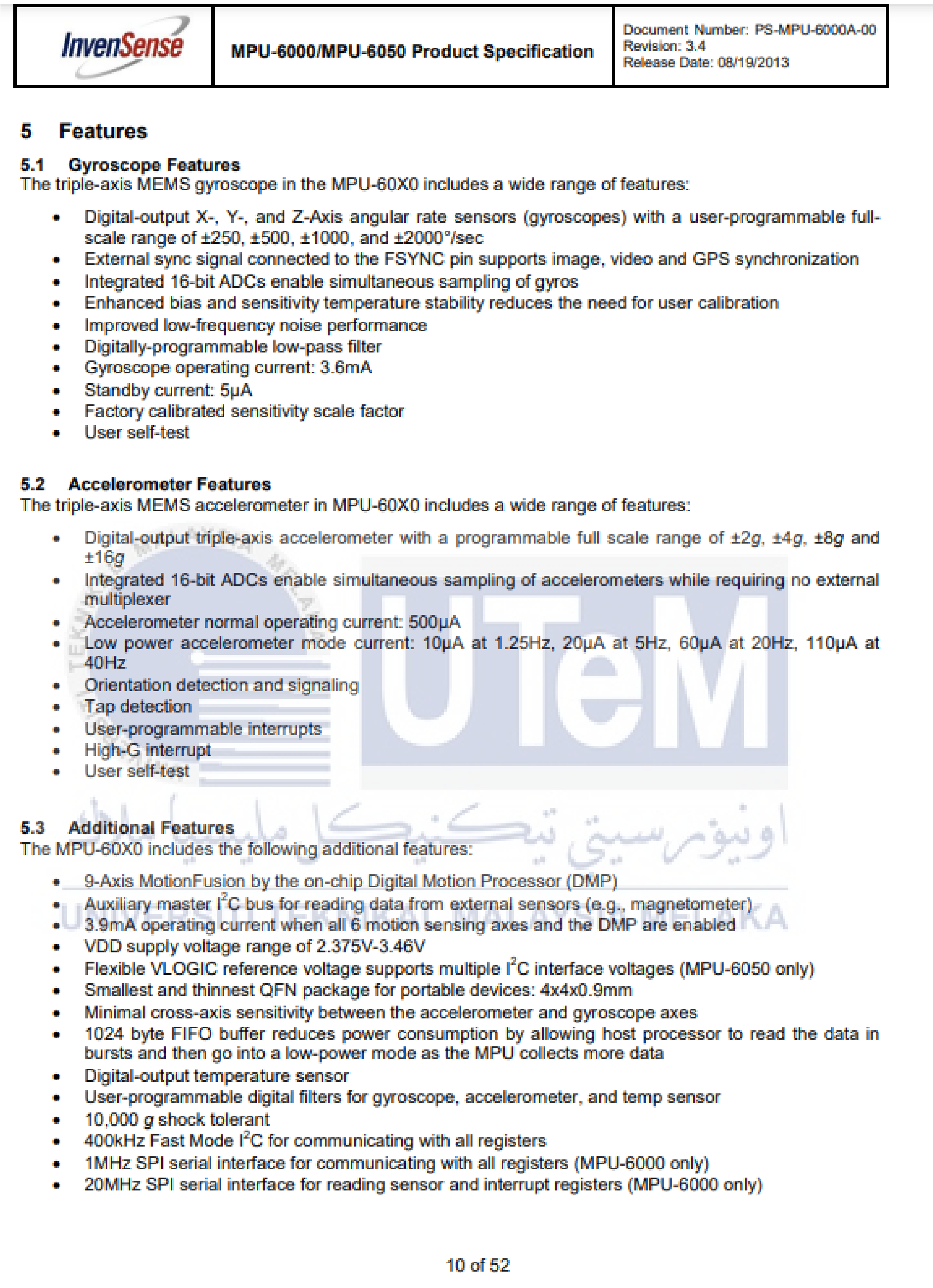


LEO





Appendix B: Datasheet of MPU6050



InvenSense	MPU-6000/MPU-6050 Product Specification	Document Number: PS-MPU-6000A-00 Revision: 3.4 Release Date: 08/19/2013

5 Features

5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ±250, ±500, ±1000, and ±2000°/sec
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5μA
- Factory calibrated sensitivity scale factor
- User self-test

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of ±2*g*, ±4*g*, ±8*g* and ±16*g*
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500μA
- Low power accelerometer mode current: 10μA at 1.25Hz, 20μA at 5Hz, 60μA at 20Hz, 110μA at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I^{2}C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I^{2}C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 *g* shock tolerant
- 400kHz Fast Mode I^{2}C for communicating with all registers
- 1MHz SPI serial interface for communicating with all registers (MPU-6000 only)
- 20MHz SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)