ON THE EDGE EMBEDDED FAULT-DETECTION IN ELECTRO-MECHANICAL SYSTEM

Final Year Project Report

by

Muhammad Abdullah

Syed Muhammad Kamran Shafqat

Uzman Munib

In Partial Fulfillment

Of the Requirements for the degree

Bachelor of Electrical Engineering (BEE)

School of Electrical Engineering and Computer Science

National University of Sciences and Technology

Islamabad, Pakistan (2022)

DECLARATION

We hereby declare that this project report entitled "On the Edge Embedded Fault Detection in Electro-Mechanical Systems" submitted to the School of Electrical Engineering and Computer Sciences (SEECS), is a record of an original work done by us under the guidance of Dr. Usman Zabit and, Dr. Wajid Mumtaz and that no part has been plagiarized without citations. Also, this project work is submitted in partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering.

Team Members	**Signature**
Muhammad Abdullah	
Syed Muhammad Kamran Shafqat	
Uzman Munib	Uzman

Supervisor	**Signature**
Dr. Usman Zabit	

Date:

____

Place:

____

Dedication

We would like to dedicate our work to our beloved and cherished parents whose affection and prayers are a source of strength in our lives and our respected supervisors, Dr. Usman Zabit and, Dr. Wajid Mumtaz for their guidance and pearls of wisdom.

Acknowledgment

All hymns to Almighty Allah who is the supreme authority. Countless gratitude to He, Lord of Lords, who leads us in the dark and aids us in our difficulties. All regards and respects to the Holy Prophet (PBUH) for enlightening our conscience and paving for us the right path with the essence of faith in God.

The acknowledgment is more than a formal obligation; it is backed by all emotional ties we have with those who assisted us in making this report presentable.

It is a matter of great honor for us to have a chance to work under the noble and kind supervision of our advisor Dr. Usman Zabit and co-advisor, Dr. Wajid Mumtaz. Our work would be obscure and incomplete without their sincere collaboration and guidance. We would remain grateful to them forever.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

We propose here an automatic fault diagnosis system as it is an essential part of today's electro-mechanical systems as it characterizes high efficiency and quality production system. Many industrial applications, such as wind turbines and electric automobiles, rely heavily on electromechanical systems.

So, in this project, we are proposing a methodology for fault diagnosis in the electro-mechanical system using machine learning and deep learning techniques. This work proposes an intelligent system to identify incipient faults in electro-mechanical systems.

Advanced signal processing techniques, machine learning techniques, and deep learning techniques are required for feature classification from sensory data. The different features that we have classified include vibration based data, temperature, voltage, and current.

A comparative analysis is also performed on the classification accuracy of different algorithms using the open-source Case Western Reserve University (CWRU) bearing dataset and the time series forecasting SKoltech Anomaly Benchmark (SKAB) dataset to obtain a more intuitive insight.

To validate our approach, we have also used a data set that we have generated from a test rigging system composed of a 12-volt dc gear motor whose shaft is connected to a pulley system.

For validation of real-time implementation efficiency, we have used two types of hardware platforms including a CPU and a Nano Jetson (which is a GPU enabled hardware) connected to our test rigging system through a data acquisition system.

This report will also explore future research directions for improving the performance of our fault prediction model.

Index Terms - Deep learning, Fault Diagnosis, Machine Learning, Real time operating system, Digital Signal Processing, Embedded System.

Chapter 1

INTRODUCTION

Electric Machines are an essential part of many applications in today's industry and electric transportation systems. Studying and developing automatic fault detection systems have received a lot of focus during the last three decades.

Traditionally, fault diagnosis has been handled by human professionals with extensive knowledge and expertise. Mechanical systems, together with their control systems, are becoming more complicated and larger in scale these days. As a result, new methods for fault diagnosis must be developed, not only qualitatively but quantitatively, to meet the demands for the autonomous diagnosis of complex control systems.

Condition monitoring and timely maintenance hold crucial value as it determines the reliability, performance, and safety of a system.

Condition monitoring is defined as the measurement of specific equipment parameters such as machine vibrations, high ambient temperature, current values, voltage values, bearing fault, gear fault, poor lubrication, misalignments, and recognizing any significant changes that could imply an impending failure.

Unsurprisingly, a fault detection system can provide several advantages, including lower maintenance costs, less downtime, longer asset life, cost savings on prematurely altered resources, and the ability to avoid catastrophes.

So, by completing this project we will be able to provide a low-cost and low-powered solution to different industries that are utilizing electromechanical systems.

Chapter 2

LITERATURE REVIEW

2.1 PRIOR WORK

By reviewing the literature, we were able to identify two main categories of fault diagnostic techniques that have been used in previous work.

1. Model-based approach.
2. Data-based approach.

The model-based approach depends upon a fault-free analytical model which will be used to predict faults and develop a robust fault diagnosis algorithm. Mainly this model is obtained from physical rules governing the system. [1, 2]

Moreover, this model is hard to implement, and due to the large number of parameters it requires, it is not feasible. Thus, it limits the algorithm to using one specific diagnostic system. Moreover, it requires complete knowledge of different interdisciplinary engineering fields.

The data-based approach depends upon training data from the system and thus does not require a predefined model for fault diagnostics but considers it as a pattern recognition problem for which a feature extraction and classification approach is used. [2]

In addition, data-driven methods can be adopted for various applications since they do not have to presume a rigid assumption about the system. Due to these facts, data-driven approaches have become popular for fault diagnosis of complex industrial systems. [3]

Data-driven methods have two main stages, namely, feature extraction and feature classification. In the first stage, some distinguishing values, which are known as the features, are constructed from sensory data which is taken from a data acquisition system. In the next stage, features are classified as faulty or fault-free.

The main difference among data-driven methods is basically due to their feature extraction algorithms. Feature extraction is mainly carried out for dimensionality reduction and variability reduction.

When faced with a dataset with a large number of data columns, it's a good idea to consider how many of these data features are useful to the model. There are numerous approaches for reducing data dimensionality to determine how informative each column is and whether they are required.

The most used data-dimensionality reduction techniques are identified as follows,

- Principal component analysis

The principal component analysis has been used in many diverse scientific fields that aims to reduce redundancies in the sensory data and at the same time minimize the information loss. [4]. Authors in [5] have used wavelet transform for feature extraction for fault diagnostics. They were able to get high accuracy but require a large number of computational power and memory.

These methods were obtained from decades of research and a large amount of investment, but these methods have one downfall they required a deep knowledge of each discipline, and designing procedures was complex.

In data-driven methods, prior knowledge is obtained by performing many experiments on the system so that is why nowadays we are developing methods that require less in-depth knowledge. The conventional signal processing methods have all been used by different researchers to obtain satisfactory results from the machine learning and deep learning models.

The methods like fast Fourier transform [6], wavelet transformation (WT) [7], empirical mode decomposition (EMD) [8], ensemble empirical mode decomposition (EEMD) [9], empirical wavelet transform (EWT) [10], wavelet packet transform (WPT) [11], variational mode decomposition (VMD), stochastic resonance, sparse decomposition, etc, have been suggested and implemented.

Signal processing techniques provide suitable accuracies but still, they have some disadvantages such as wavelet transformation comes up with a problem of difficulty in selection of the mother wavelet, the choice of the frequency band and window size which contains the essential information for the fault.

Machine learning is a subfield of artificial intelligence (AI) and computer science that focuses on using data and algorithms to mimic the way humans learn, intending to steadily improve accuracy. For fault diagnosis, many machine learning algorithms have been applied. In recent years, machine learning methods such as support vector machines (SVMs), K nearest neighbors (KNNs), and artificial neural networks (ANNs) have shown to be effective in fault diagnosis of electro-mechanical systems.

Deep learning algorithms are one of the most recent advances in the extraction of a hierarchical layer-wise abstraction model from raw sensory input [12].

In [12], convolutional neural networks have been used for fault diagnosis in rotary machines. In [13] deep neural networks using wavelet transform has been used to implement fault diagnosis on electromechanical system mainly gear and bearing faults in real-time and is then implemented on Nvidia GPU-enabled embedded processor (Jetson TX1) and two other computational hardware and they have obtained an accuracy of 99 and 96 percent for bearings and gearbox fault dataset.

This project aimed to develop a robust fault diagnostics system that classifies faults as well as predicts them before their occurrence for this we used different classical machine learning techniques and state-of-the-art deep learning techniques on bearing and anomaly detection datasets available publicly. Then by analyzing these algorithms we implemented these models on our test rigging system for real-time validation. This validation was done on two different hardware platforms, a CPU and Nano Jetson which is a low-powered architecture design specially for deep learning optimization.

Chapter 3

PROBLEM DEFINITION

It is important to find dependable solutions for diagnostic systems in the scientific study and application of electromechanical systems, which are expected to meet increasingly rigorous standards. At the same time, we need our solutions to move away from the expensive and specialized robust system to a more friendly low-cost budgeted system that is based on cheap hardware and widely available software environments. It is generally easy to enhance capabilities in such flexible diagnostic systems by adding other software modules that use alternative signal processing algorithms.

The traditional method that is being used is mainly based on high-cost architecture or they are based on physical rules governing the system which required expert knowledge of different interdisciplinary engineering fields.

To date, diagnostics have been based on the analysis of easily quantifiable signals utilizing the expertise and experience of human professionals who interpret ongoing data acquired through measurement and analysis systems.

Unfortunately, there is a flaw in this approach: the human expert, who can make mistakes and whose experience is difficult to automate. This is why, in addition to developing diagnostic signal analysis methods, it is vital to identify ways to objectify the defect detection and fault level assessment processes. This can be accomplished by the application of artificial intelligence technologies and techniques, which are becoming increasingly prominent in diagnostics.

According to the available literature, automated diagnostic systems for electromechanical systems (which comprise bearings, gears, and electrical parts as a whole) using artificial intelligence, particularly deep learning approaches, are rarely used and deployed. As a result, the work's goal is to propose a low-cost monitoring and diagnostic system that can operate in both off-line and online modes, based on sensor data obtained from the test rigging system as well as publicly available datasets.

Moreover, all the work done previously is deployed on a high-powered architecture, so another aim of this project was to deploy this on low-cost architecture.

In Pakistan, almost all industries that are using the electromechanical system are using a traditional method that is based on physical rules governing the system and analysis method.

So, providing our solution to this approach would help industries in Pakistan to opt for a low-cost solution for fault diagnostics of electromechanical systems.

Chapter 4

METHODOLOGY

4.1 PROPOSED WORK

In our proposed method we will be using different classical machine learning techniques as well as deep learning techniques for fault classification and fault prediction. Signal processing techniques like wavelet transform were also utilized in preprocessing the data for better localization of features. After training these models we will deploy them on two different types of hardware for the validation of these models in real-time scenarios. Given the resource-constrained nature of IoT-based devices, using machine learning at the edge can provide improved prediction capabilities and effective resource management.

We will also be focusing on reducing the dimensionality and variability of input sensory data in fault diagnosis applications and predicting faults.

In particular, we will explore the benefits of deep learning-based methods over traditional machine learning methods in terms of fault feature classifier performance, as well as additional functionality provided by deep learning techniques that were previously unavailable.

For training and evaluation of different ML and DNN-based algorithms, the following datasets are analyzed:

- CWRU [14], A public domain dataset for bearing faults that are obtained from the test rigging system is shown in figure 4.1.
- SKAB, a time series data for evaluating anomaly detection algorithms.
- Dataset using a customized test rigging system for gathering data with different modalities.

In the next section, a detailed overview of these data sets will be given and how they are useful for fault diagnostics of electromechanical machines.

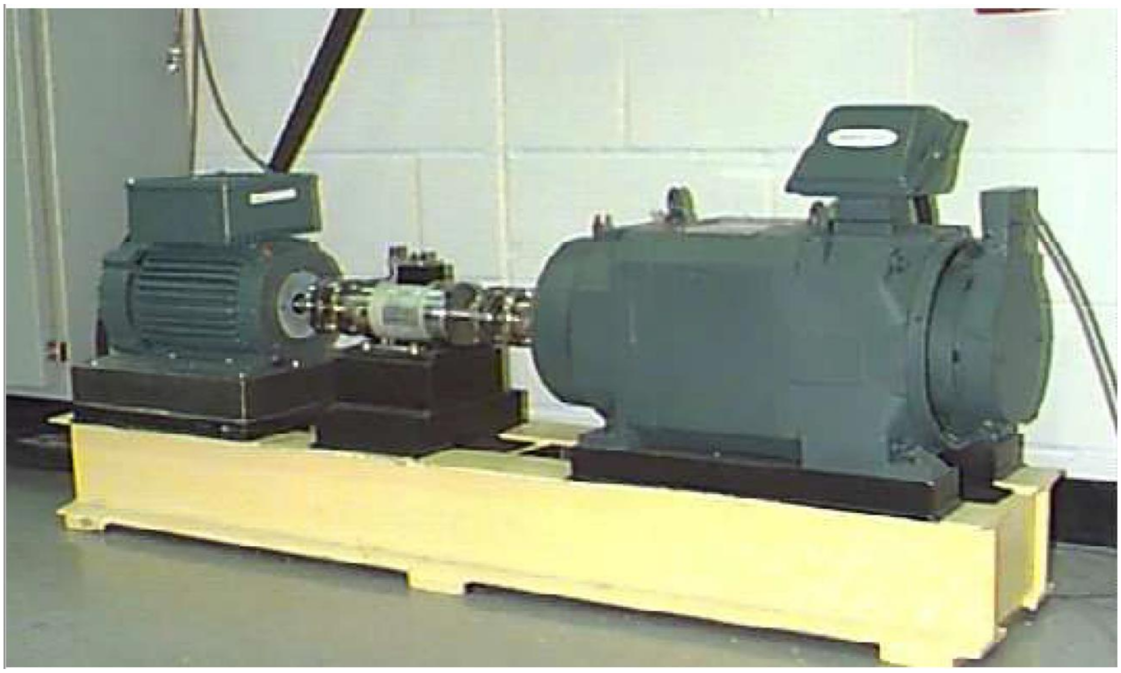

Figure 1 CWRW Test Rigging System (Picture from CWRU data sets website)

Then to validate the practicality of real-time implementation of our learning-based fault diagnosis method, the entire fault diagnosis system is implemented on two types of hardware platforms.

1. Nvidia Nano Jetson
2. CPU i5 intel processor

For our test rigging system, we have designed a data acquisition system on Arduino UNO which is collecting data from vibrational, current, voltage, and temperature sensors. This data acquisition system wirelessly transmits data to the master node using an esp-32 microcontroller. The master node in our case is the Nano Jetson and CPU where we will be deploying our deep learning models.

4.2 DATASETS

A dataset in machine learning is, quite simply, a collection of data pieces that can be used by a computer for analytic and prediction purposes. Deep learning and machine learning need a vast amount of data to give the most excellent results. As the size of data increases the performance of deep learning increases as compared to the classical machine learning techniques. Furthermore, a good dataset should meet specific quality and quantity requirements. Make sure your dataset is relevant and well-balanced for a smooth and quick training experience. As a result, in ML and DL applications, data amount and frequency are critical.

For a robust approach, we are using three different types of datasets. These datasets are openly accessible and have been used for robust fault detection. They are as follows:

1. CWRU Dataset

2. SKoltech Anomaly Benchmark (SKAB)

3. Test Rigging System

4.2.1 CWRU Dataset

Damage to rolling bearings is the most common type of damage in spinning equipment. The percentage share of rolling bearing defects in the total number of machine faults varies depending on the kind and size of the machine, normally in the range of 40%-90%. [15] [16]

The CWRU dataset is a well-known, open-source, and simple-to-use dataset. The dataset in question has been collected and is available on the CWRU website. It contains information on both normal and damaged bearings. Figure 1 depicts the test rigging system that was used to collect the data. A Reliance electric motor with 2 HP power, a torque transducer, dynamometer, and control electronics make up the test bench for evaluating motor performance.

Defects are introduced at a single point of EDM (Electric Discharge Machine) machining.

For this dataset, we get a time series for each defect located in 1 of 3 parts of the bearing: ball, inner race, an outer race. For a clear representation of the structure of rolling bearing with four different cases of misalignment that have been caused by the EDM and led to electromechanical fault refer to figure 2.

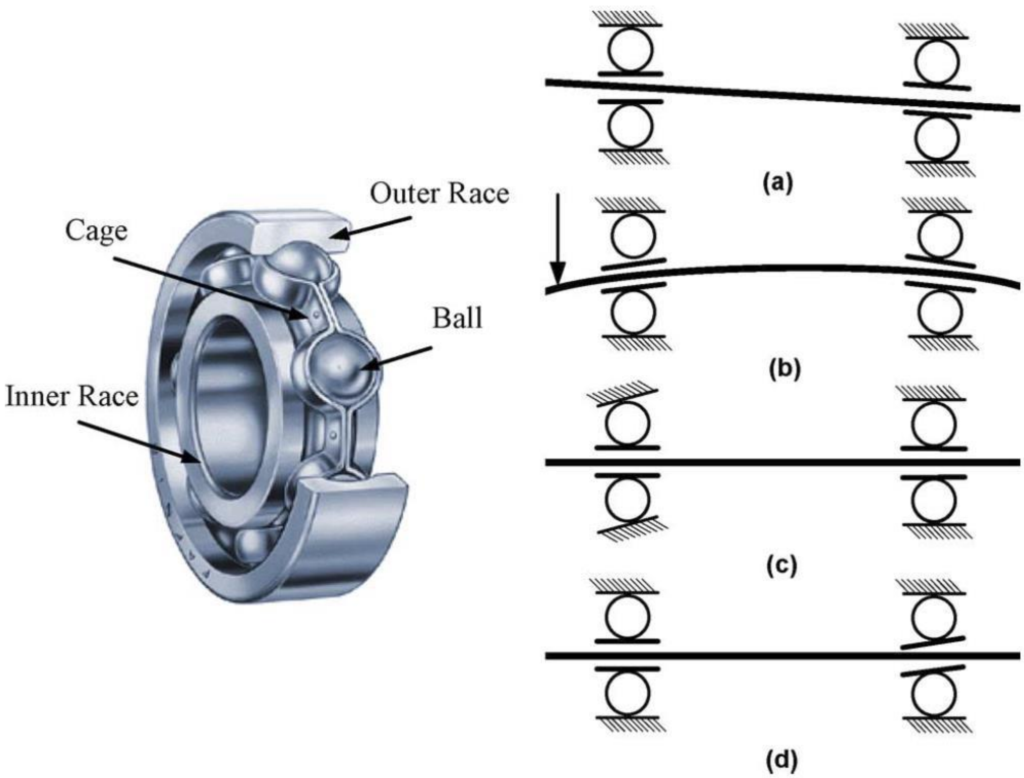

Figure 2 Structure of a rolling-element bearing with four types of common scenarios of misalignment that are likely to cause bearing failures: (a) misalignment, (b) shaft deflection, (c) crooked or tilted outer race, and (d) crooked or tilted inner race [17]

Data on acceleration is collected near and distant from the motor bearings. The data is acquired from several sensors located in various locations. The DE and FE of the motor bearings were fitted with accelerometers, which were used to collect vibrational data.

The sampling frequency at which data was collected was set at 12 kHz and 48 kHz. The data that was obtained from the data acquisition system was then processed in the MATLAB environment and then stored as mat files.

Data from the fan-end was captured at a rate of 12k samples per second. The data collection rate for the normal baseline was 48k samples per second.

Features that were calculated to predict the fault type were as follows: maximum, minimum, mean, standard deviation, RMS (Root mean squared), skewness, kurtosis, crest factor, and form factor. So, total features amount up to 8 and each of these features is computed for time segments of 2048 points which is 0.04 seconds at the 48 kHz accelerometer sampling. Figure 4.2 shows how faults are distributed between two types of features, and it also shows the total number of fault classes in our dataset.

The CWRU-bearing dataset is extensive, varied, and complex. The CWRU bearing dataset is widely used and is the gold standard for verifying various machine learning and deep learning techniques. The masking sources are not included in the dataset, which makes it easier to use [18]. In the following sections, we'll go overusing machine learning and deep learning techniques on the CWRU data sets.

4.2.2 SKAB Dataset

The SKoltech Anomaly Benchmark (SKAB) is a tool for testing anomaly detection algorithms. Outlier detection (anomalies assessed and marked up as single-point anomalies) and changepoint detection are two of the primary problems that SKAB can help with (there are two markups for anomalies) (anomalies considered and marked up as collective anomalies). A control system for water circulation and a monitoring system to monitor the state of water. The details of the different components can be seen in Fig. 3.

The SKAB v0.9 [19] corpus consists of 35 CSV data files, each of which represents a single experiment and contains a single anomaly. The dataset represents a multivariate time series acquired from the test bed's data collecting system.

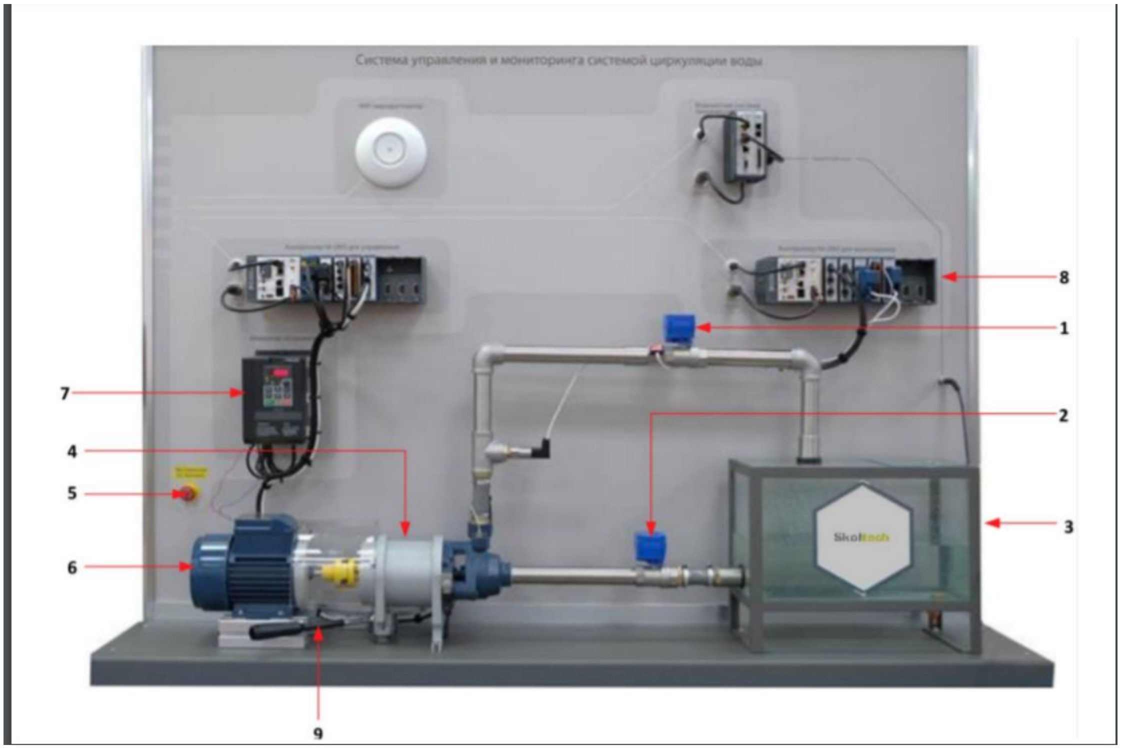

Figure 3 SKAB Test Rigging System (Img taken from Kaggle)

Columns in each data file can be represented as follows: Date Time - Represents dates and times of the moment when the value is written to the database (YYYY-MM-DD hh: mm: ss)

We have used the SKAB dataset and have deployed multivariate LSTM on it to predict if there would be a fault occurring or not. Details of the LSTM model that we have applied for fault forecasting will be covered in the later sections.

4.2.3 Test Rigging System

For real-time validation, prediction, and classification of faults diagnostics of the electromechanical system a customized test rigging was designed. It consists of a data acquisition system that comprises different sensors connected to a motor whose shaft is connected to a system of a pulley.

Sensors include a vibrational sensor (an accelerometer), a voltage sensor, a current sensor, and a temperature sensor connected to Arduino Uno.

We have collected a data set from this system and have trained our models on this dataset. Both classification and prediction have been implemented on this data set in real time.

The sensors are connected to Arduino Uno which then wirelessly transmits data through a microcontroller ESP-32 to another embedded system mainly a CPU or Nano Jetson which is also connected to the receiver ESP-32 module.

Hence Real-time fault prediction and forecasting are carried out on this test rigging system to validate all the models that we trained for CWRU and SKAB dataset and to develop a robust fault diagnostic method for the electromechanical system.

The purpose of our project is to develop a system that when trained with the right data can diagnose faults and with our test rigging system, we have validated our deep learning models for fault diagnostics.

Figure 4 shows the overview of the test rigging system that we have designed for real-time validation of our fault diagnostics models.

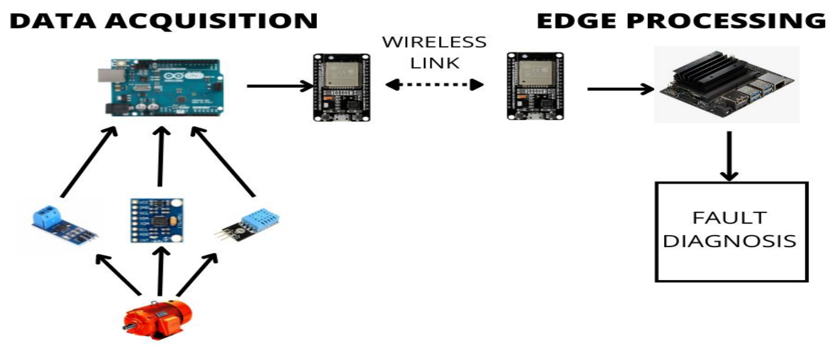

Figure 4 The flow diagram of the test rigging system that we have designed for real-time validation of our fault diagnostics models.

Chapter 5 will cover an overview of all the components that were selected in making the Test Rigging System along with the real-time implementation of our model for fault diagnostics.

In the next section, we will be discussing the different classical machine learning techniques and deep learning techniques that we have utilized for fault prediction and classification on the above-mentioned data set. By validating different classical machine learning techniques and deep learning techniques on the publicly available datasets we will be utilizing the best and optimum approach to our fault diagnostics system.

4.3 CLASSICAL MACHINE LEARNING MODELS

The output of model training can be used for inference, which is used to make predictions based on new data supplied into the model. A machine learning system's learning is reflected by a model. Mathematical functions are analogous to machine learning models. They accept an input data request, generate a prediction based on that data, and then serve a response.

Before the boom of deep learning approaches, a variety of classical machine learning algorithms were very popular.

These algorithms required a lot of complex feature engineering and domain expertise. Moreover, dimensionality reduction techniques were required to reduce the dimensionality of data. This was carried out using techniques like principal component analysis (PCA) etc. Then in the end most important and representative features are fed into the machine learning algorithms for training.

All the classical machine learning models are trained and testing using the data of 10 classes and data for each class are taken at a load of 1hp. The classes are:

- fault1: Ball defect (0.007 inches)
- fault2: Ball defect (0.014 inches)
- fault3: Ball defect (0.021 inches)
- fault4: Inner race fault (0.007 inches)
- fault5: Inner race fault (0.014 inches)
- fault6: Inner race fault (0.021 inches)
- fault7: Normal
- fault8: Outer race fault (0.007-inch, data collected from 6 O'clock position)
- fault9: Outer race fault (0.014 inches, 6 O'clock)
- fault10: Outer race fault (0.021 inches, 6 O'clock)

Some of the classical machine learning techniques that we have applied to CWRU datasets are as follows:

4.3.1 Support Vector Machine (SVM)

SVM is a kind of machine learning technique that can be used for classification and regression applications. They improve on fundamental machine learning algorithms by adding features that make them more efficient at certain tasks.

One work that is very popular in our field is SVM used for identifying bearing faults. [20]. In our work, we have used the SVM algorithm and deployed the SVM algorithm on the CWRU dataset and have analyzed how well it classifies faults concerning other classical machine learning and deep learning algorithms based on a set of defined metrics.

The data was preprocessed using a standard scalar and then split into test and train sets for training and validating the model.

We have used 10-fold cross validation to find the optimum value of the gamma and C value and then have trained train support vector classifier. The decision function that was used was One v/s Rest (OVR) as it was a multivariate classification of 10 different faults. Polynomial Kernel was used as the kernel function to represent the similarities of training samples in feature space over the polynomial of the original data available to the polynomial function.

The results we gained from SVM were very optimal and the accuracy achieved was equal to 96.1 percent. The confusion matrix shown in figure 5 is obtained from inputting testing data of the CWRU model to a trained support vector classifier gives the distribution of how many classified values are truly positive, true negative, false positive, and false negative.

The confusion matrix gives a very clear view of how well the model has been classified. The confusion matrix for testing data that was separated from the training data at the start is shown in figure 5.

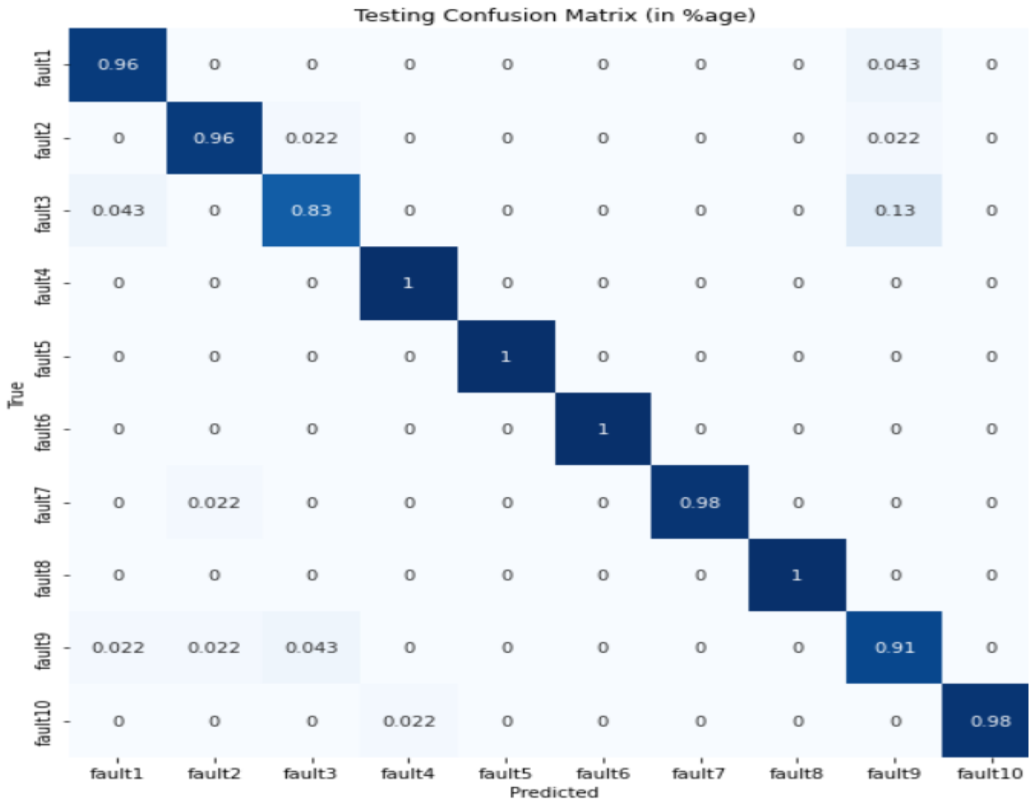

Figure 5 Confusion matrix obtained from inputting testing data into the Support Vector Classifier trained from the CWRU training set.

4.3.2 K Nearest Neighbor

k-Nearest Neighbor is one of the simplest classification algorithms that doesn't require any training and data can be fed directly into the classifier.

Minmax Scalar was used to preprocess the data of CWRU that was present in form of the feature matrix. In the feature matrix, each fault is divided into 230 segments. Each segment represents a row in the matrix. The total number of segments mounts to 2300 segments as there is a total of 10 faults. To determine the class of a test data point, we calculate its distance to k nearest training points. As labels for the training set are known, the class of the test point is decided by a majority vote among the k training labels.

The only hyper-parameter of this algorithm is the value of k. It decides the number of neighbors to choose to do classification. The appropriate value of k can be chosen by trying out different values.

In [21], k-NN is used as the main method for a data mining-based ceramic bearing fault classifier based on acoustic signals, which is an early application of the k-NN classifier on bearing fault diagnostics.

The testing accuracy achieved from K-NN was equal to 93.9 percent which is very high concerning the complexity of the model used. K-NN is a very simple classifier so achieving high accuracy is good but still, the classification results from the confusion matrix can be seen as not so good as diagonal values are not so high.

Boundaries are very distinguishable between 10 faults for their feature values. One of the plots between the distribution of faults according to their crest and mean values calculated is shown in figure 6.

The confusion matrix for deploying the model on the testing data that was separated from the training data at the start is shown in figure 7. Compared to the SVM model K-NN model has not performed so well but still, we have to keep in mind computational power required for K-NN is less than that of the SVM model.

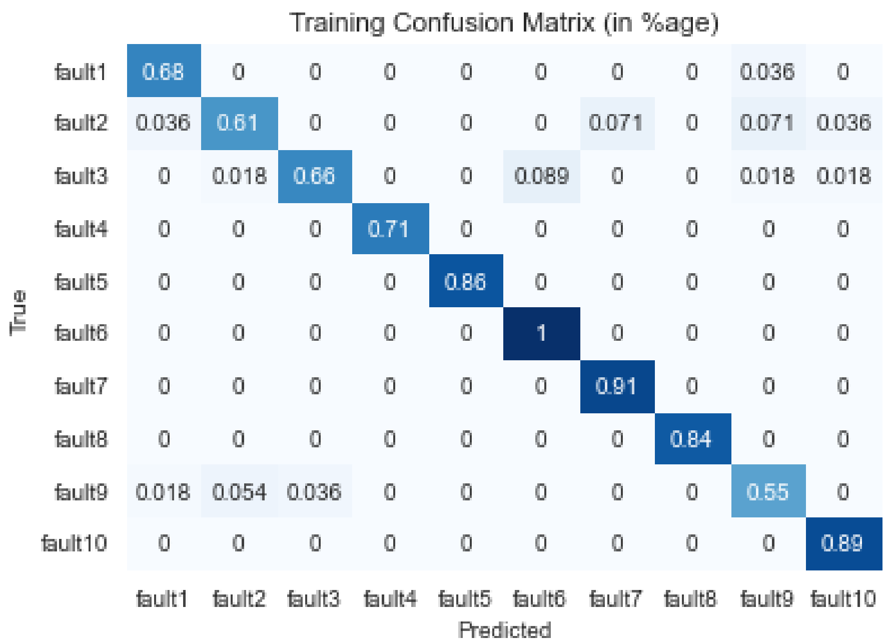

Figure 6 Confusion matrix obtained from inputting testing data into K-NN classifier trained from CWRU training set.

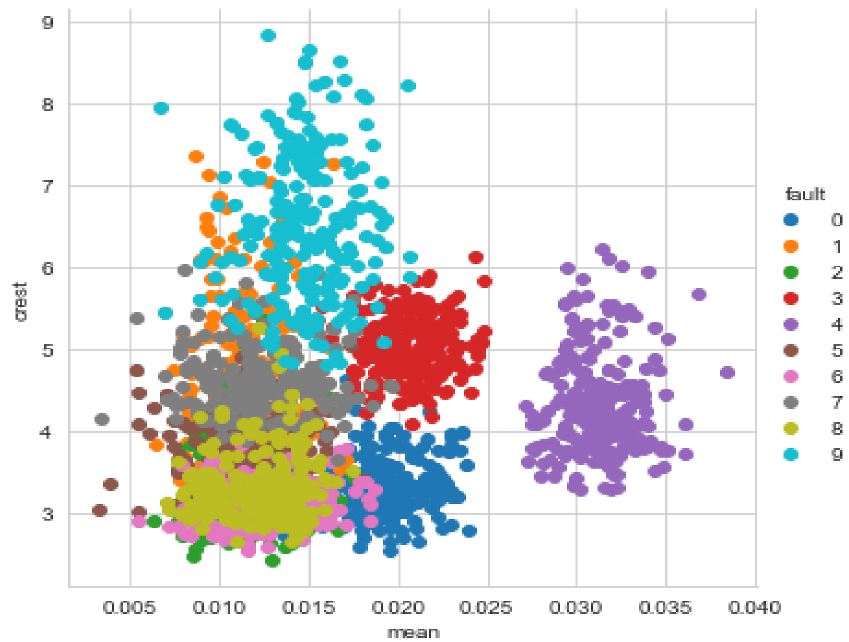

Figure 7 The figure shows the distribution of 10 faults for their crest and mean values calculated in the feature matrix.

After analyzing different classical machine learning techniques, we will now move towards the deep learning techniques that are more popular nowadays.

4.4 DEEP LEARNING ALGORITHMS

Deep learning is a part of machine learning and has helped achieves a great amount of flexibility. Deep learning methods are popular nowadays because they can learn the distinctive features directly from the data thus removing the need for feature engineering that is required in most classical machine learning techniques.

The DL-based approach does not require human expertise or prior knowledge of the problem and is thus used in a wide variety of fields. Thus, deep learning architecture can be easily implemented on new problems relatively easily as compared to other techniques.

Deep learning models required a large amount of data for maximized performance, so it is optimum to use a data set that is as large as possible for maximum accuracy.

For validating deep learning models for predicting and forecasting tasks two types of models have been utilized.

For time series forecasting a multivariate LSTM was used on the SKAB dataset and for classification a Convolutional Neural Network was used on the CWRU dataset.

4.4.1 Convolutional Neural Network (CNN)

Convolutional neural networks are distinguished from other neural networks as they show superior results as compared to the traditional neural networks.

They have three fundamental types of layers mainly known as:

- Convolutional layer
- Pooling layer
- Fully connected layer

The convolutional layer is the first layer of the convolutional neural network and can be followed by another convolutional layer or pooling layer. In the end, the final layer of the convolutional neural network is the fully connected layer. The complexity of CNN models keeps on increasing after each passing layer and parameters keep on increasing. CNN focused on finding the distinct features from the data and as layers keep on increasing more distinctive elements are recognized and were built upon the lower-level features.

CNN has been used in previous work for fault classification of the electromechanical system. The main focus of this work was fault classification of bearings and gears.

In the next section, we will focus on the architecture and results of the CNN model we deployed on the CWRU dataset. We also are applying a continuous wavelet transform to the time series dataset to get better localization of features and then these wavelets will be fed into the CNN model for training.

4.4.1.1 CNN Using Raw Time Domain Data

Temporal raw data of CWRU is obtained from a different accelerometer placed in different places in the CWRU test rigging system.

Raw temporal data is then preprocessed using a standard scaler to normalize the data values.

Raw temporal data is divided into three sets mainly a training set, and a testing set and for cross-validation, a third validation set is also separated from the dataset. Data was thoroughly randomized before splitting them into these sets.

The training set will be used to train the CNN model. A 2-dimensional CNN will be deployed on this training set which is validated using the validation set and then in the end will be tested by a testing set that was separated at the start of preprocessing.

Figure 8 shows the basic architecture of the CNN-based bearing fault classifier. The training set is then sent to a convolutional layer for feature extraction and downsampling, followed by a pooling layer.

To further deepen the network, this convolution-pooling process is repeated several times. The CNN architecture additionally adds a dropout layer. Finally, the output from the hidden layers will be passed through one or more fully connected layers, with the result being passed via a top classifier based on SoftMax or Sigmoid functions to identify if a bearing fault exists.

The accuracy achieved by deploying the trained model on the test dataset was 98.5 percent on the intel processor and about 96.3 percent of accuracy was achieved on Nano jetson.

The confusion matrix obtained from deploying the model on the test dataset is shown in figure 9. The results are comparatively better than both SVM and KNN combined. CNN's ability to automatically learn features can help in finding faults at their initial stages of fault that have no explicit characteristics frequency.

The H-5 file for the model was approximately equal to a size of 400 KB which is that it can be deployed very quickly and efficiently on different hardware platforms in our case we have used Nano Jetson and Intel Core I5 Processor.

The time to classify a batch of test data fed into the CNN classifier is about 0.183 seconds.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 24, 24, 32)        2624
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 4, 4, 32)          82976
_________________________________________________________________
dropout_1 (Dropout)          (None, 4, 4, 32)          0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_4 (Dense)              (None, 96)                6240
_________________________________________________________________
dense_5 (Dense)              (None, 10)                970
=================================================================
Total params: 101,066
Trainable params: 101,066
Non-trainable params: 0
_________________________________________________________________
```

Figure 8 Architecture of 2-Dimensional CNN based classifier

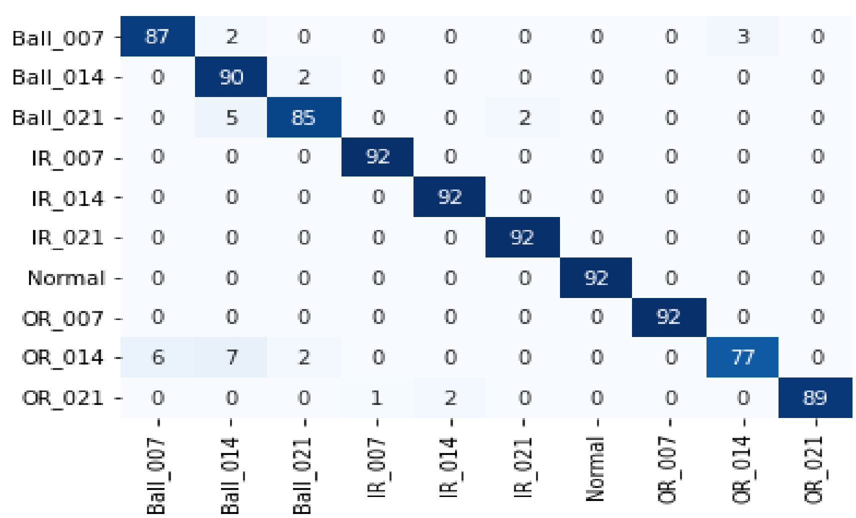

Figure 9 Confusion matrix for 2-d CNN CWRU fault classifier

4.4.1.2 Continuous Wavelet Transform (CWT)

Continuous wavelet transformation has been a very important part of the analysis of time-frequency information. It provides better localization of features as compared to short-time Fourier transform by providing variable time-frequency resolution.

It is useful for studying changes in time domain signals by localizing the frequency contents of a signal in time.

Equation of CWT is given below:

$$\boldsymbol{T(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\psi^{*}\frac{(t-b)}{a}dt}$$

a = Scale factor

b = location of the wavelet

ψ = Wavelet function

x = time domain signal

The wavelet function that we have used in our case is Morlet also known as Gabor wavelet which is a multiple of a complex exponential multiplied with a gaussian window. The formula for the Morlet wavelet is given below:

$$\boldsymbol{\psi(t) = \exp(-i\omega_0 t)\exp\left(-\frac{t^2}{2}\right)}$$

ω_0 = frequency

From each category of fault temporal data are collected in segments of length 1024. Continuous wavelets transform (CWT) is then applied to this segment of data at 64 different scales. So, the output of the wavelet transform is of size (64×1024). This is similar to the image that we see after the wavelet transform. As the input size to the 2d-CNN model that we have created is (32×32), we resize the (64×1024) image into (32×32) using TensorFlow.

There is no overlapping between different segments. For each category of fault, 460 such segments are taken. The total size of the data thus becomes (4600, 32, 32)

Figure 10 and figure 11 show how wavelet transformation of the raw temporal data provides a better localization of frequency contents in time.

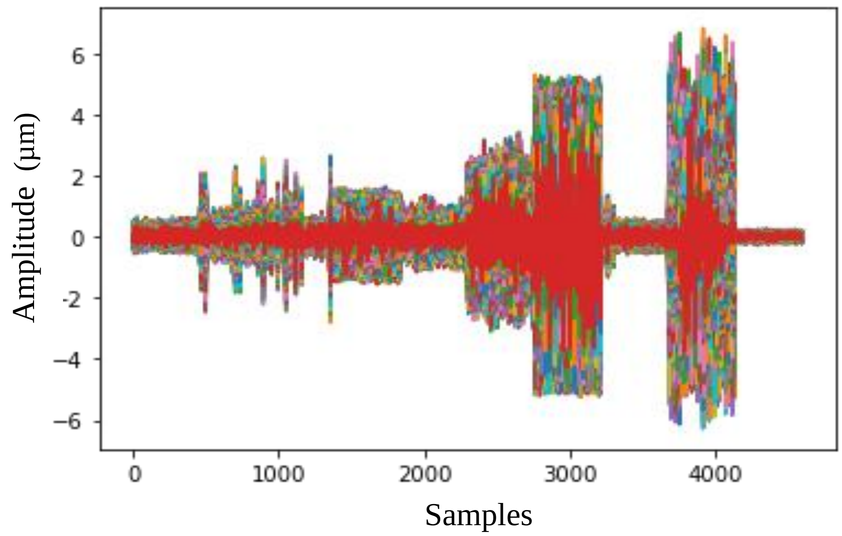

Figure 10 Plot of CWRU temporal raw data

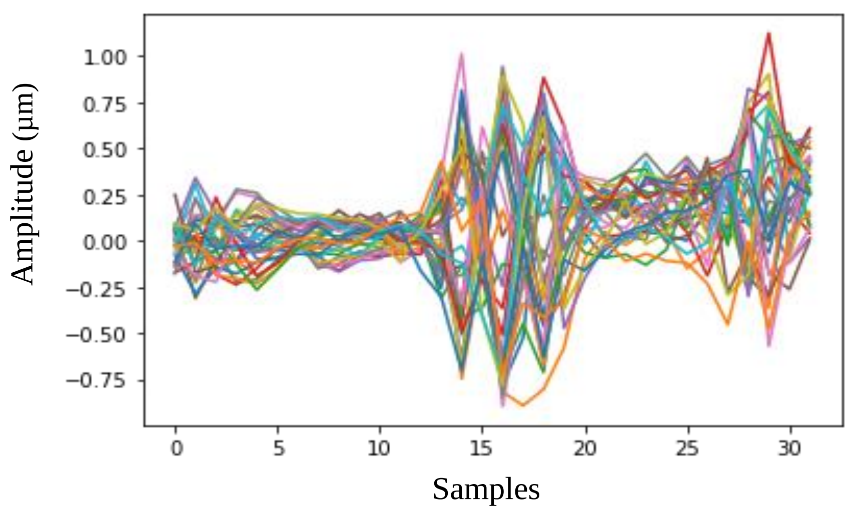

Figure 11 Plot of CWRU after taking CWT of the temporal raw data

4.4.1.3 CNN Using Wavelet Transform

Computational time increases considerably while taking the CWT of the raw temporal CWRU data. 2-D CNN classifier is trained based on the similar architecture that we have trained earlier with some slight changes in the filters of dense layers.

Accuracy is considerably increased and is about 99.5 percent which is the highest we have obtained while validating all types of deep learning and machine learning models.

The architecture of the 2-d CNN used to classify data after wavelet transform is shown in figure 13. The confusion matrix for the CWRU CNN classifier has performed best among all the other models. For viewing the confusion matrix refer to figure 12.

Deep learning model weights are initialized randomly. Due to the inherent non-deterministic nature of the processing, we will get different answers if we run the same model twice.

We can observe that by simply changing the data into the frequency domain the accuracy has been increased considerably.

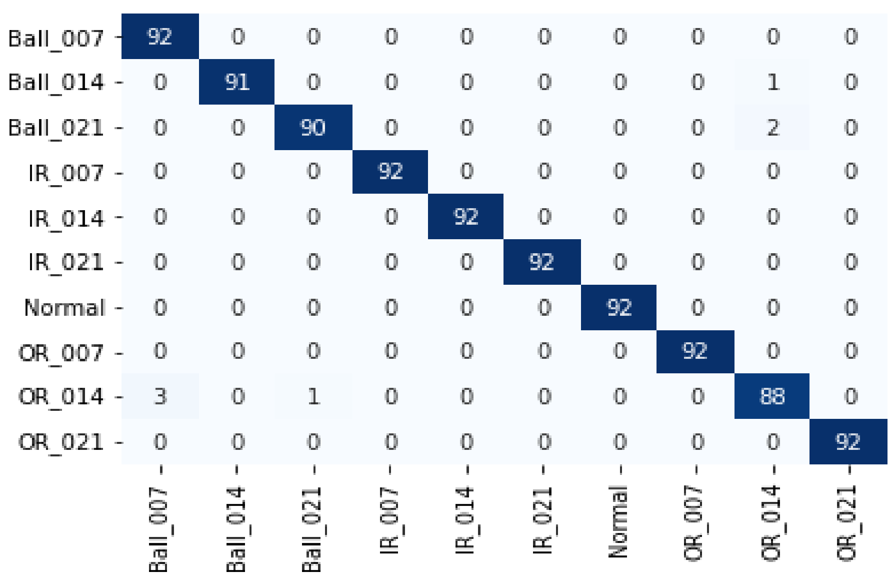

Figure 12 Confusion matrix for 2-d CNN CWRU fault classifier

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 30, 30, 32)        320
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 32)        36896
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 10, 32)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 32)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 800)               0
_________________________________________________________________
dense_3 (Dense)              (None, 120)               96120
_________________________________________________________________
dense_4 (Dense)              (None, 80)                9680
_________________________________________________________________
dense_5 (Dense)              (None, 10)                810
=================================================================
Total params: 143,826
Trainable params: 143,826
Non-trainable params: 0
_________________________________________________________________
```

Figure 13 Architecture of 2-Dimensional CNN based classifier for CWRU Wavelets

4.4.2 Multivariate LSTM On SKAB Dataset

For time series forecasting of faults, we have deployed multivariate LSTM. Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods is practically their default behavior, not something they struggle to learn.

Preprocessing of the SKAB data set is done through a min-max scalar and thus data is in the range of 0 to 1. Normalization makes it easier to find distinctive patterns in the sequence and thus all values in a range pattern are more visible.

LSTM model architecture that has been made for SKAB fault prediction is shown in Figure 14.

Then data is divided into sequences of input usually three inputs at a time are fed to the LSTM model for time series prediction. The feature used for predictions is the data itself which includes the vibrational values of the x, y, and z axis, current, voltage, and temperature values.

Mean square logarithm error (MLSE) loss is used as the loss function in the LSTM deployment as it considerably reduces the effect of large losses to a small value. This loss is the mean over the seen data of the squared difference between the log-transformed true and predicted values.

The formula for MLSE is given below:

$$\boldsymbol{L(y,\hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$$

$\boldsymbol{y}$ = true value

$\boldsymbol{\hat{y}}$ = predicted value

N = total number of observations in the data set

Figure 15 shows a lot of actual values and the values that were predicted by our LSTM model.

Current and voltages values are targeted in this dataset, and these are stored as time series values in the SKAB dataset. We were able to use LSTM to predict whether the fault will occur in near future or not.

The final loss value that was obtained from the LSTM model between the actual and predicted value was equal to 0.414.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_6 (LSTM)                (None, 1, 60)             15600
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 60)             0
_________________________________________________________________
lstm_7 (LSTM)                (None, 1, 64)             32000
_________________________________________________________________
dropout_5 (Dropout)          (None, 1, 64)             0
_________________________________________________________________
lstm_8 (LSTM)                (None, 64)                33024
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 260
=================================================================
Total params: 80,884
Trainable params: 80,884
Non-trainable params: 0
```

Figure 14 LSTM architecture used for SKAB time series forecasting

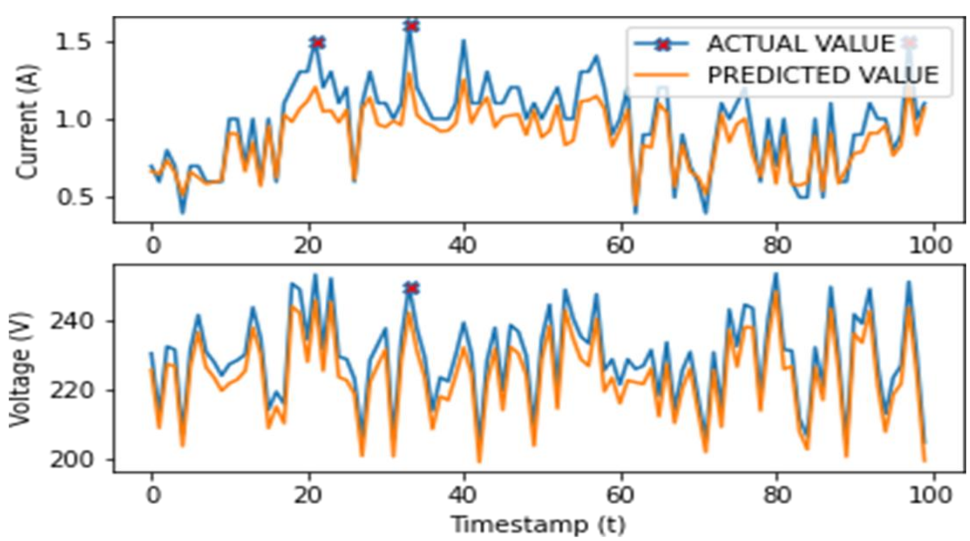

Figure 15 Prediction that we have made SKAB time series forecasting along with the actual values.

By validating all these models, we have concluded for a fault classification convolutional neural network with data prepared using continuous wavelet transform works the best. The model that was made for classification was light and was easily deployed on nano jetson and CPU. For prediction, LSTM works the best and gives the best results for fault forecasting.

For a comparison between different classical machine learning models and deep learning, models refer to figure 16.

Now we will create a CNN model for deployment of our test rigging system fault. On the test rigging data set an LSTM will be used to predict is there going to be a fault in near future or not and if the fault occurs our CNN will classify what type of fault it is. Deployment, architecture, and results obtained from real-time validation on the test rigging system will be shared in the next chapter.

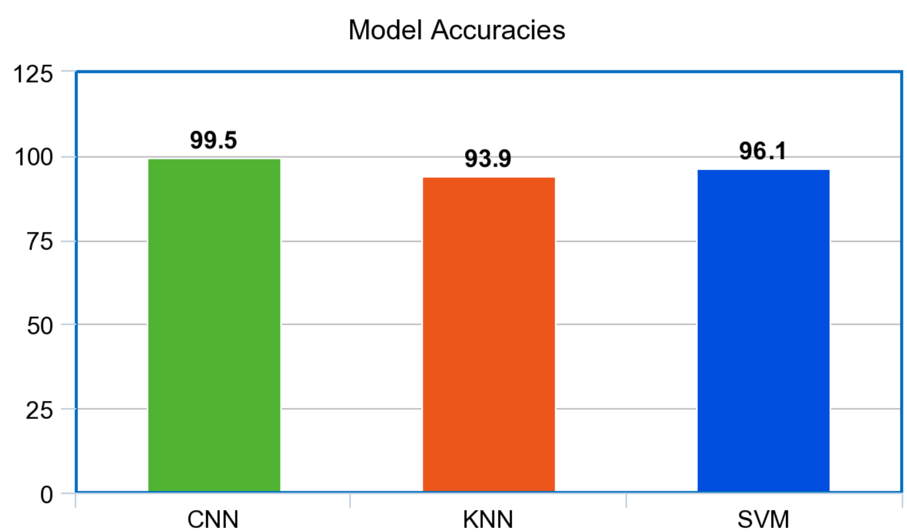

Figure 16 Accuracy comparison between different ML and DL models

Chapter 5

Real Time Implementation

5.1 CNN AND LSTM MODEL

To perform fault diagnostics in real time, we have deployed two types of models. LSTMs for fault prediction on raw temporal data acquired from the test rigging system. If LSTMs predict that a fault is going to occur, then the same temporal data is fed into the second model CNN classifier which will classify the fault type.

A flow chart for the whole process is displayed in figure 17.

For fault forecasting, in real time LSTM architecture is similar to LSTMs made for the SKAB dataset. A clean data set was acquired from the test rigging system. This data was normalized using a standard scalar, so all values are in the range of -1 to 1. LSTMs model is then trained using this clean dataset. Data is fed into the LSTMs model as sequences of integer values. Mean square logarithmic error was used as the loss function for the LSTM. A graphical representation of faulty data is shown in figure 18. Figure 18 shows that if the pattern of the predicted stream matches the actual stream, then no fault is predicted and if this is not the case then a fault is predicted.

In real time the data samples from data acquisition in sent to the hardware platform and if a fault is forecasted then the data is fed to the CNN classifier for fault classification.

1D CNN classifier has been deployed on the temporal data available from the data acquisition system for fault classification. Parameters in the model have been set to the minimum possible value thus reducing the size of the model considerably so that it can be effectively and efficiently deployed in Real time on the edge.

The basic architecture of the 1D-CNN-based electromechanical fault classifier is illustrated in figure 19. The training set is then passed over to a convolutional layer for feature extraction, followed by a pooling layer for downsampling.

The accuracy achieved by deploying the trained model for real-time data incoming from the data acquisition system is equal to 93.1 percent.

Figure 17 Flow diagram for real-time implementation

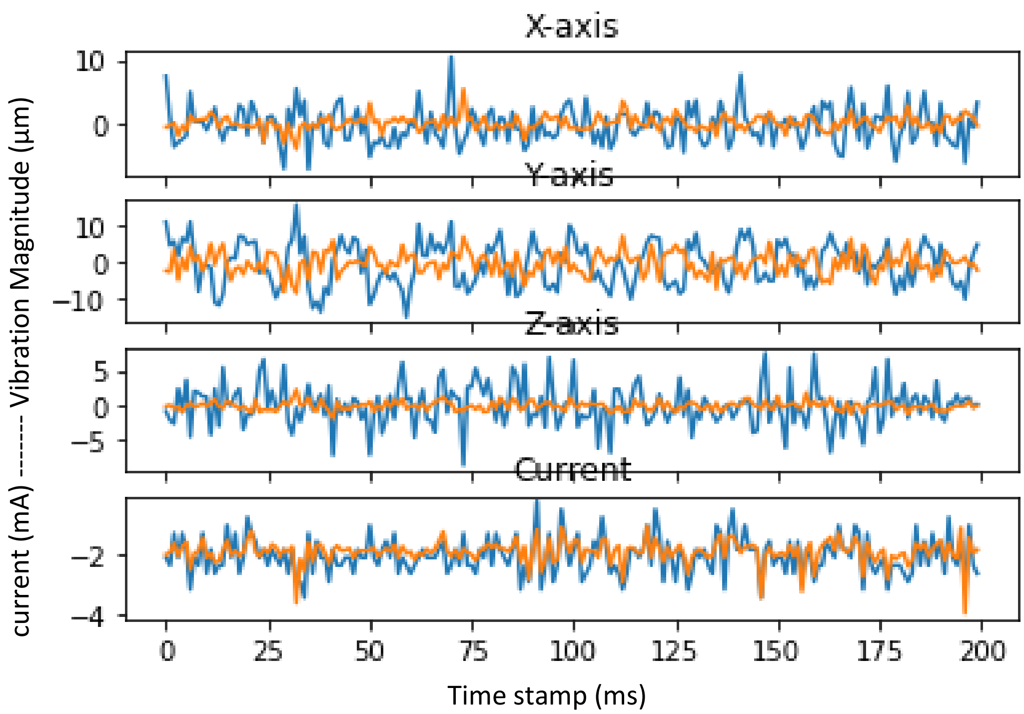

Figure 18 A graphical representation of the predicted output from the LSTMs model and the actual model

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_29 (Conv1D)           (None, 5, 98)             196
_________________________________________________________________
max_pooling1d_29 (MaxPooling (None, 5, 98)             0
_________________________________________________________________
conv1d_30 (Conv1D)           (None, 5, 128)            12672
_________________________________________________________________
max_pooling1d_30 (MaxPooling (None, 5, 128)            0
_________________________________________________________________
flatten_14 (Flatten)         (None, 640)               0
_________________________________________________________________
dense_42 (Dense)             (None, 120)               76920
_________________________________________________________________
dense_43 (Dense)             (None, 80)                9680
_________________________________________________________________
dense_44 (Dense)             (None, 3)                 243
=================================================================
Total params: 99,711
Trainable params: 99,711
Non-trainable params: 0
```

Figure 19 Architecture for real-time CNN fault classifier

5.2 DEPLOYMENT

For the deployment of our system, we built a custom electromechanical system that can be modified to induce faults on our will. This test rigging system consists of a pulley system connected with a belt. One of the pullies is connected to a gear motor, which when powered, rotates the belt. Figure 20 shows our test rigging system.

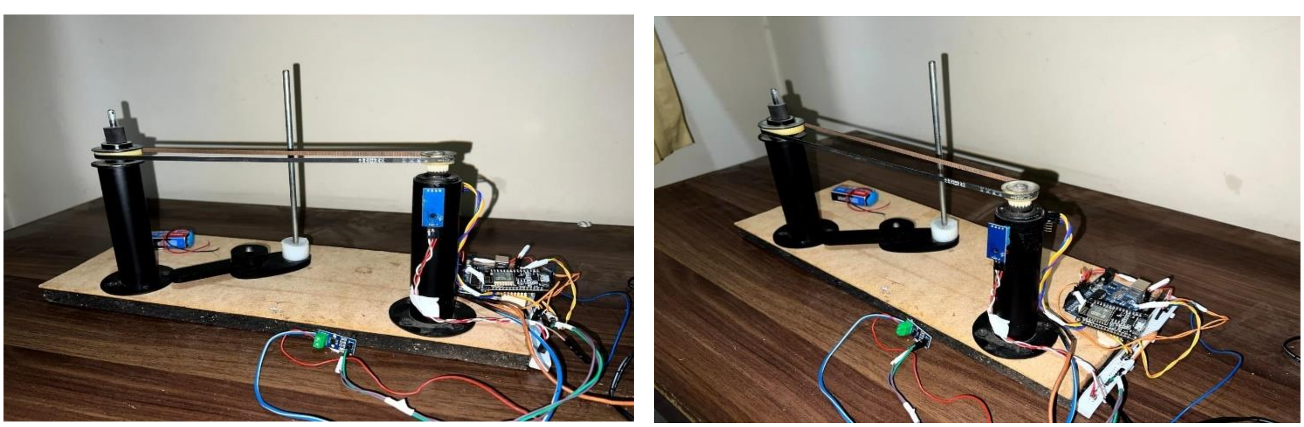

Figure 20 Test rigging system

5.2.1 Faults

To deploy the system, faults must be induced in the EMS. We deployed three major types of faults in the system which are the cause of more than 80% of the damage to machinery in industries.

- Hindrance / Friction
- Vibrational jerk
- Rising temperature

5.2.1.1 Hindrance / Friction

Friction or hindrance in EMS can be caused due to multiple reasons like overloading, and poor maintenance. This fault was added to our test rigging system. An increase in current is visible in figure 20 when this fault is induced.

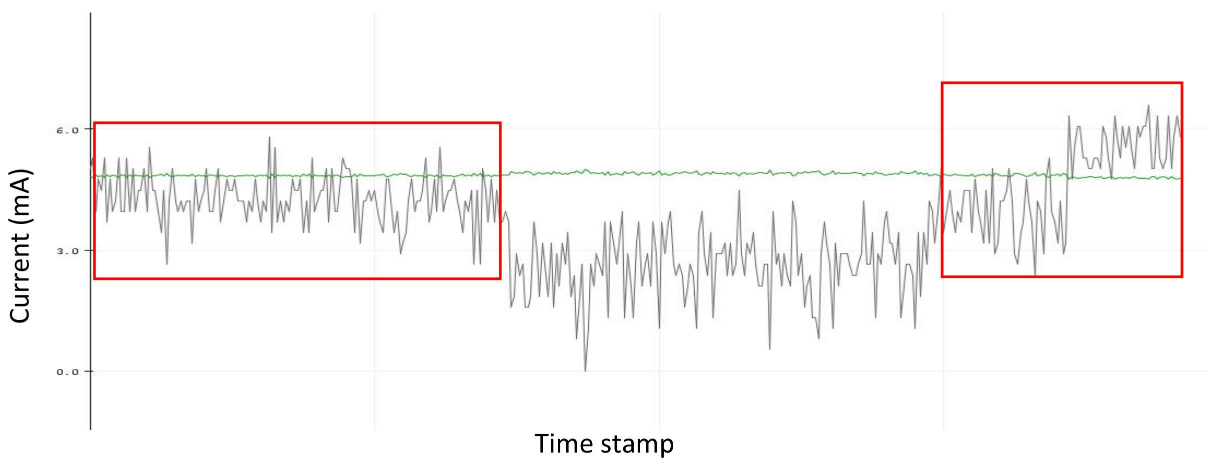

Figure 21 Current pattern for hindrance/friction in EMS

The value of current during the fault can be seen in the red box. We took 30,000 readings with this fault induced. The average current in this condition and normal condition can be seen below:

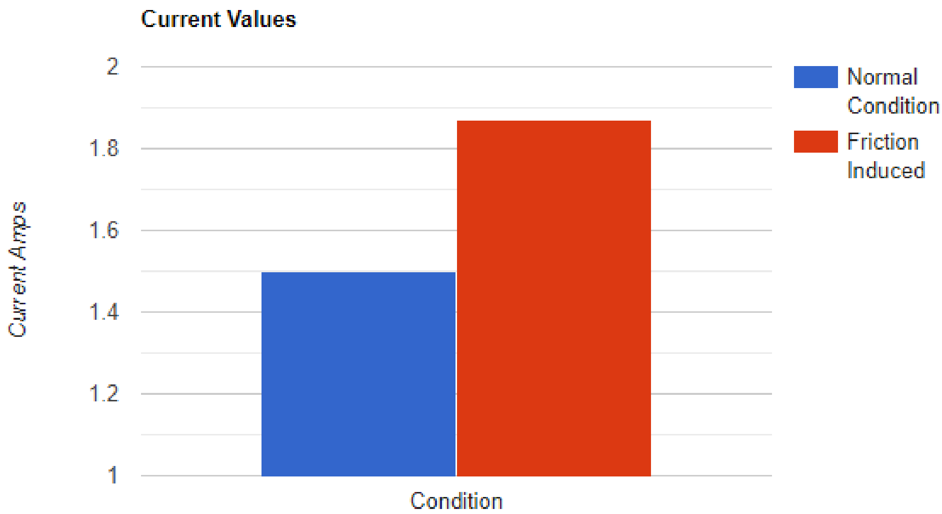

Figure 22 Average current value in EMS

5.2.1.2 Vibration Jerk

The vibration pattern is the most vital form of information that tells about the health of a machine. A faulty machine changes its vibrational pattern. We induced a vibrational fault and capture the data. The difference between normal and faulty system patterns can be seen.

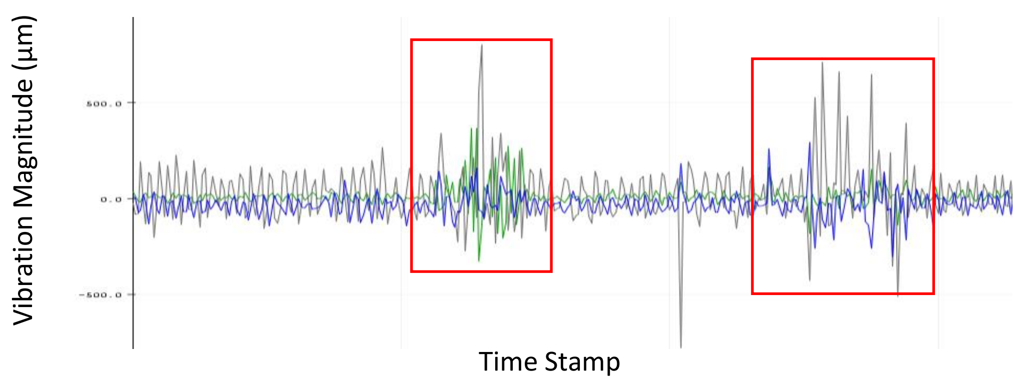

Figure 23 Vibration pattern of EMS

The faulty pattern can be seen in red boxes in figure 23.

5.2.1.3 Temperature Fault

The temperature of an EMS can cause a lot of damage if it is out of the optimal range. The sudden increase of our system was also increased as a fault to our system to capture the data.

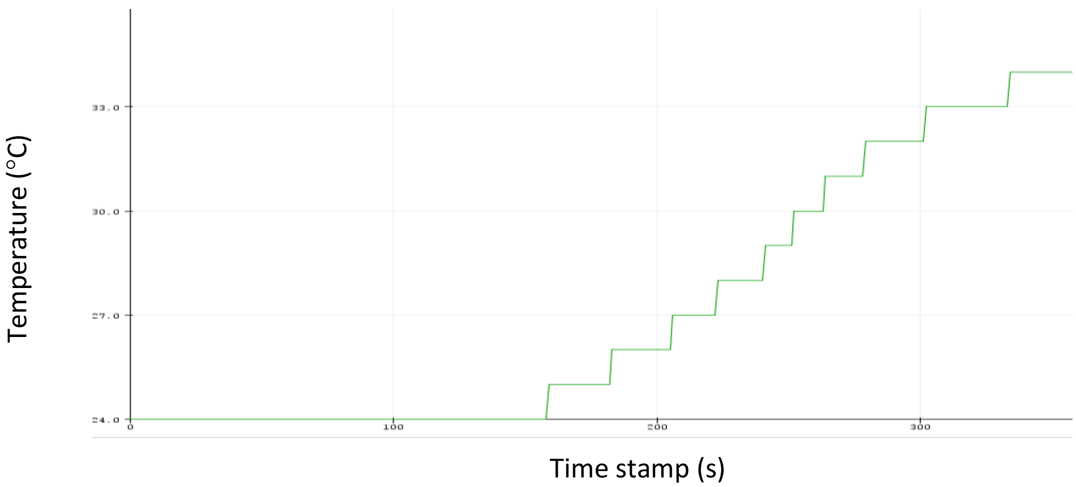

Figure 24 Temperature pattern of faulty EMS

5.3 REAL-TIME IMPLEMENTATION

For the real-time implementation of our system, we need to develop a data acquisition system. This data acquisition system will be responsible for collecting the data at a specified rate and feeding the data to our neural network which is deployed on the edge device that is Nvidia Nano Jetson in our case.

The overall architecture of the implemented system is as follows

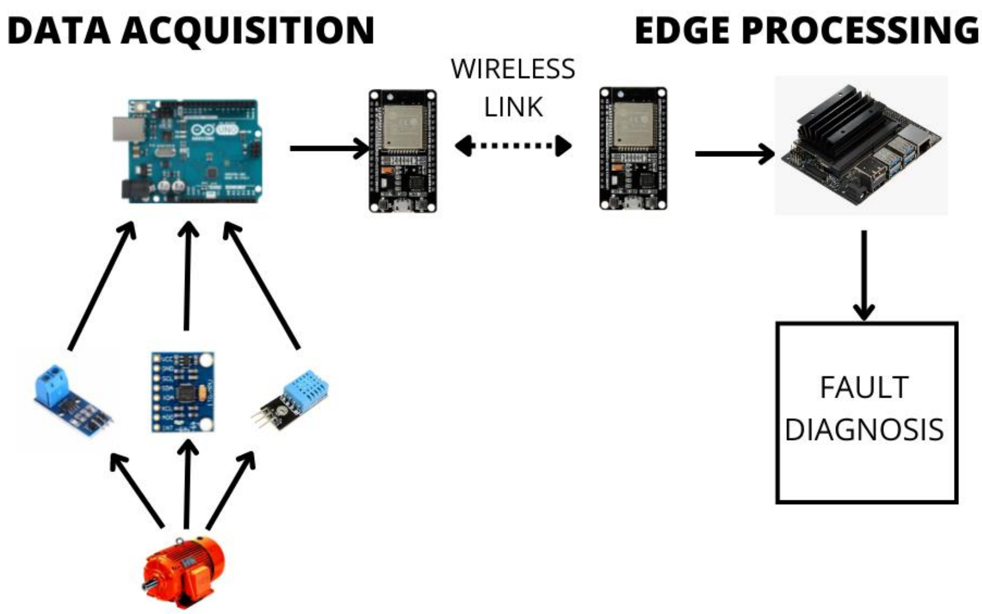

Figure 25 Implemented architecture of the system

The breakdown of the architecture is explained below:

5.3.1 Data Acquisition System

The data acquisition system is composed of three sensors and a mother processor dedicated to collecting the data from EMS. The sensors used are:

- Current/voltage sensor
- Temperature sensor
- Vibration sensor

5.3.1.1 Current Sensor

The sensor used for the current and voltage measurement for our EMS is **ACS 712**. When the load or friction in the parts of EMS is increased, a change in the pattern of current and voltage is observed which is beneficial in training the models. This sensor itself comes in three variants of 5 amps, 20 amps, and 30 amps. For our application, the need was fulfilled by the 5 amps variant.

- The measurement range of ACS 712 is from -5 amps to 5 amps.
- The current resolution of the sensor is 113 mA.
- The sensor can work in the presence of 2100 Vrms.

5.3.1.2 Temperature Sensor

Electro-mechanical systems when exposed to overloading and poor working conditions, poor maintenance can lead to overheating. This increase in temperature results in a faulty system. Thus, to keep this factor in the training of the neural network, a **DHT11** sensor is used to collect the data. This sensor is used as it suited our application. Some specifications of the sensors are given below:

- The temperature measurement ranges from 0°C to 50°C.
- The sensor has a 1°C resolution.

5.3.1.3 Vibration Sensor

The most vital information about an electromechanical system can be extracted by the vibration pattern of the EMS. Due to any fault in bearings, conveyor belts, or miscellaneous faults, vibration pattern tends to change a big time. To capture the vibration pattern, we have used **MPU6050.** The features of this sensor are mentioned below:

- This is a three-axis accelerometer, which means that it measures the vibrational/acceleration values along three axes i.e. x-axis, y-axis, and z-axis.
- The sensor has a resolution of the 16-bit analog to digital conversion.
- The range of vibration that it can measure is ±16g.
- The sampling rate is up to 8kHz.

Now, all these sensors are connected to Arduino Uno which collects all the data in a list of 6 values:

x-axis value	y-axis value	z-axis value	Current	Voltage	Temperature

The values are sampled at **a 4Hz** frequency. The optimum baud rate is set at 112500 bauds. Data was collected for training the neural network and for real-time processing. 50,000 samples for each fault induced were collected at this rate. For the real-time implication, this collected data is serially transferred to **ESP3212E Node MCU.** This ESP module is used to transmit the data wirelessly to the edge processor, where the data is processed. **UART** protocol is used between the communication of Arduino and the ESP module.

5.3.2 Wireless Communication

There is a need for wireless communication between the data acquisition so we can benefit from the remote access and monitoring of the machinery. For this purpose, **ESP3212E Node MCU** is used. This node MCU acts as a transmitter and sends the data to the receiver. For this purpose, we have used the Server-Client protocol. An access node is made on the receiver end which receives the data from the connected sender. The data is sent via **HTTP** protocol. The data is secured by a unique password, which should be known at the time of the configuration.

5.3.3 Edge Deployment

The data received at the edge device via the ESP32 wireless module is passed on to the neural network. The device processes the data and it gives us the output as the results.

The edge device we used to deploy our network was **Nvidia Nano Jetson.** The nano jeston comprises a Quad-core Arm A57 CPU processor and 128 core NVIDIA maxwell^{TM} GPU. It has 2 GB RAM. This edge device gave us optimized performance and compatibility with our deployed network.

Chapter 6

RESULTS

The results of using multiple models on data sets are tabulated below:

MODEL	ACCURACY
CNN using wavelet transform	99.5 %
CNN on raw data	96.8 %
SVM	96.1 %
KNN	93.9 %
Our deployed model (LSTM leading to CNN)	**93.1 %**
LSTM	0.414 (loss)

Although the individually trained models have high accuracies up to 99.5%, the overall accuracy of the model is low. The major reason for the dropped accuracy is using both LSTM and CNN models simultaneously. The processing of data from LSTM to CNN combined gives us the accuracy of 93.1%

The confusion matrix of the final result is shown in the figure below

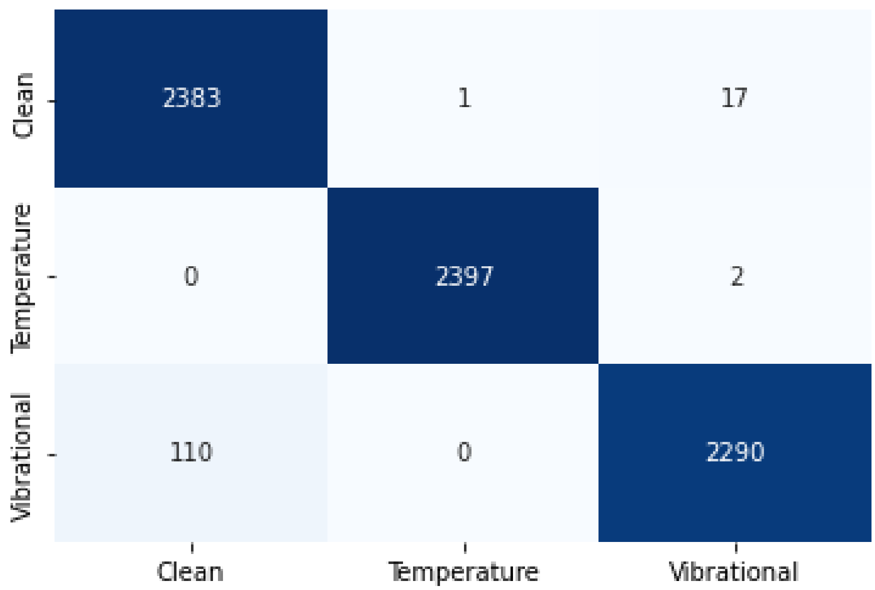

Figure 26 Confusion matrix of the final deployed system

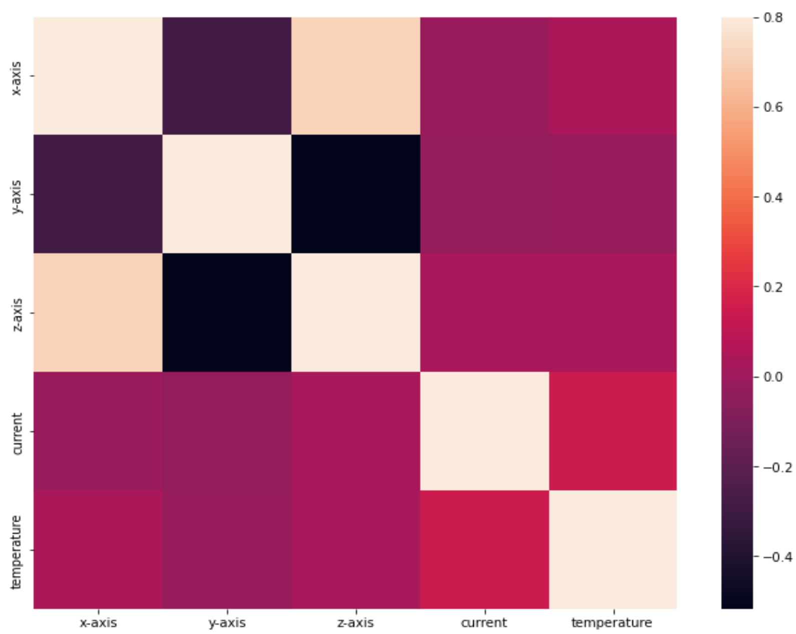

Figure 27 Heatmap of the final deployed system

CONCLUSION

With the development of the industries, the number of machines and electromechanical systems is increasing at a high pace. The more the machines, the greater the chance of faulty systems leading to catastrophe. To counter this situation, and to know about the health of EMS, and the classification of faults, a cheap and sustainable solution was required. With every moving day, the amount of data generated worldwide is increasing. Artificial intelligence and deep learning models become more and more accurate with the increasing data.

Thus, we provide a low-cost and low-power solution to the fault classification system in EMS. With the help of data collected from the EMS, the deep learning model of CNN and LSTM is trained and deployed on the dedicated edge device i.e. Nvidia Nano Jetson. When deployed on the edge, the data is collected and transferred to the edge device over a wireless link for processing.

The classification accuracy on the real-time implementation of the model is 93.1% which is competitively provided the test rigging system and facilities we could use. The developed classification model is robust and has been cross verified by training on CWRU and SKAB datasets which provide 99.5% accuracy and 0.414 loss.

Chapter 7

REFERENCES

[1] S. a. F. C. a. B. S. Simani, "Diagnosis techniques for sensor faults of industrial processes," *Control Systems Technology, IEEE Transactions on,* vol. 8, pp. 848-855, 10 2000.

[2] M. a. Z. M. a. N. M. a. A. B. Heydarzadeh, "A Wavelet-Based Fault Diagnosis Approach for Permanent Magnet Synchronous Motors," *IEEE Transactions on Energy Conversion,* vol. 34, pp. 761-772, 2019.

[3] X. D. a. Z. Gao, "From the model, signal to knowledge: a data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics,* vol. 9, pp. 2226--2238, 2013.

[4] D. W. Scott, "Multivariate density estimation: theory, practice, and visualization," no. John Wiley \& Sons, 2015.

[5] M. V. a. H.-J. Kang, "Wavelet kernel local fisher discriminant," *IEEE Transactions on Instrumentation and Measurement,* vol. 64, pp. 3588-3600, 10 2015.

[6] V. K. R. a. A. R. Mohanty, "Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform," *Mech. Syst. Signal Process.,* vol. 21, pp. 2607-2617, 2007.

[7] Z. a. C. F. a. H. Y. Peng, "Vibration signal analysis and feature extraction based on reassigned wavelet scalogram," *Journal of Sound and Vibration - J SOUND VIB,* vol. 253, pp. 1087-1100, 2002.

[8] C. C. a. B. Z. a. G. M. a. D. W. a. Y. Yuan, "Wasserstein Distance based Deep Adversarial Transfer Learning for Intelligent Fault Diagnosis," *CoRR,* 2019.

[9] X. W. a. C. L. a. F. B. a. X. B. a. K. L. Shao, "Fault diagnosis of diesel engine based on adaptive wavelet packets and EEMD-fractal dimension," *Mechanical Systems and Signal Processing,* vol. 41, pp. 581-597, 2013.

[10] C. a. G. M. a. G.-P. J. a. J. O. a. R. H. Castejón, "Automatic Selection of the WPT Decomposition Level for Condition Monitoring of Rotor Elements Based on the Sensitivity Analysis of the Wavelet Energy," *International Journal of Acoustics and Vibrations,* vol. 20, pp. 95-100, 2015.

[11] C. a. G. M. a. G.-P. J. a. J. O. a. R. H. Castejón, "Automatic Selection of the WPT Decomposition Level for Condition Monitoring of Rotor Elements Based on the Sensitivity Analysis of the Wavelet Energy," *International Journal of Acoustics and Vibrations,* vol. 20, pp. 95-100, 2015.

[12] L. a. Y. D. Deng, "Deep learning: methods and applications," *Foundations and trends in signal processing,* vol. 7, pp. 197-387, 2014.

[13] M. a. K. S. a. N. M. a. H. H. a. A. G. a. C. G.-A. Heydarzadeh, "FEATURE LEARNING USING DEEP NEURAL NETWORKS FOR FAULT DIAGNOSIS IN ELECTROMECHANICAL SYSTEMS," 2019.

[14] D. a. S. J. Neupane, "Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review," *IEEE Access*, vol. 8, pp. 93155-93178, 2020.

[15] J. Faiz, A. Takbash and Mazaheri-Tehrani, "E. A Review of Application of Signal Processing Techniques for Fault," *AUT J. Electr. Eng,* vol. 49, p. 109–122, 2017.

[16] F. Immovilli, A. Bellini, R. Rubini, and C. Tassoni, " Diagnosis of Bearing Faults in Induction Machines by Vibration or Current Signals: A Critical Comparison," *IEEE Trans. Ind. Appl.,* vol. 46, pp. 1350-1359, 2010.

[17] T. Harris, "Rolling Bearing Analysis," *Hoboken, NJ, USA:* vol. 3, 1991.

[18] F. a. Z. X. a. W. W. a. W. Y. Ding, "A Fault Feature Extraction Method of Motor Bearing Using Improved LCD," *IEEE Access,* vol. 8, pp. 220973-220979, 2020.

[19] I. D. a. K. V. O. Katser, "Skoltech Anomaly Benchmark (SKAB)," 2020.

[20] A. R. a. A. K. Nandi, "Detection and classification of rolling element bearing faults using support vector machines," *IEEE Workshop on Mach. Learning Signal Process,* pp. 153-158, 2005.

[21] D. a. R. L. a. Z. J. a. Z. M. He, "Data Mining Based Full Ceramic Bearing Fault Diagnostic System Using AE Sensors," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council,* vol. 22, pp. 2022-2031, 2011.

[22] X. W. a. C. L. a. F. B. a. X. B. a. K. L. Shao, "Fault diagnosis of diesel engine based on adaptive wavelet packets and EEMD-fractal dimension," *Mechanical Systems and Signal Processing,* vol. 41, pp. 581-597, 2013.