Predictive Maintenance of Electrical Machines Using Deep Learning

APOORV RANJAN

2022PSP0005

A Dissertation Submitted to

Indian Institute of Technology Jammu

In Partial Fulfillment of the Requirements for the Degree of

Master of Technology in

Communication And Signal Processing

Department of Electrical Engineering

May, 2024

Abstract

This research work addresses the important issue of keeping electrical machines in good working condition by using vibration data to predict when maintenance is needed. Ensuring these machines work reliably and avoid major failures is crucial for industrial operations. Traditionally, checking the health of these machines involves physical inspections and measurements, which are time-consuming and prone to human error. Therefore, there is a need for new techniques that can overcome these limitations.

In recent years, machine learning and deep learning have shown great promise in improving the accuracy and reliability of condition monitoring and predictive maintenance. Instead of using traditional machine learning algorithms, this research uses the Bidirectional Encoder Representations from Transformers (BERT) model, which is typically used for understanding language and natural language processing-related tasks. This study adapts b to classify the health of machines by analyzing vibration data and distinguishing between healthy and unhealthy conditions.

The data for this research work was sourced from Kaggle, an open-source online platform known for its diverse and high-quality datasets. The dataset is named unbalance detection of a rotating shaft using vibration data. These data consist of numeric data coming from vibration sensors in a tabular form and stored in a CSV file format. Before training the model, the data were carefully preprocessed. This involved converting data to a list and then converting it to a string and properly labeling data under healthy and unhealthy conditions.

BERT's ability to understand context, originally designed for text, is used here to interpret the patterns and anomalies in the vibration data. Although the data are not text, their sequential nature allows BERT to capture the necessary information. The model was specifically fine-tuned for the task of binary classification, distinguishing between healthy and unhealthy motor states. This involved extensive training on the preprocessed vibration data, and optimizing the model's parameters to minimize errors and

improve accuracy.

To ensure the model's robustness and reliability, multiple stages of training and validation were conducted. The dataset was divided into training, validation, and testing subsets to thoroughly evaluate the model's performance. Key performance metrics such as accuracy, precision, recall, and F1-score were used to assess its effectiveness. Additionally, cross-validation techniques and hyperparameter tuning helped prevent overfitting, ensuring the model generalizes well to new data.

The results of this study show that the BERT model can be highly effective in predictive maintenance and it is also better than convolution neural networks, fully connected neural networks, and Random Forest (RF) which was used in the existing research paper on this dataset. The model accurately distinguished between healthy and unhealthy states of electrical machines, proving that BERT's ability to understand context can be successfully adapted for vibration data analysis. This success highlights the potential of advanced machine learning and deep learning techniques in industrial maintenance, offering accurate and timely predictions that can be integrated into maintenance schedules. This enables real-time monitoring and proactive interventions, helping to prevent machine failures.

In summary, this thesis demonstrates that using the BERT model for classifying machine health based on vibration data is both feasible and effective. The innovative use of a language model for this task opens up new possibilities for research and application in predictive maintenance, promising significant improvements in the reliability and efficiency of electrical machines. Future research could explore adding other types of data, like temperature, voltage, or sound signals, to further enhance the model's predictive power and validate its practical use in real industrial applications.

Contents

List of Figures

List of Tables

List of Nomenclatures

ANN: Artificial Neural Network
ML: Machine Learning
BERT : Bidirectional Encoder Representations from Transformers
DL: Deep Learning
NN: Neural Network
DGA: Dissolved Gas Analysis
CNN : Convolutional Neural Network
PE: Positional Embedding
LM: Language Model
FC : Full Connected
FP: False Positive
FFT: Fast Fourier Transform
FN: False Negative
MSE: Mean Squared Error
IITJMU: Indian Institute of Technology, Jammu
IOT : Internet Of Thing
FCNN: Fully Connected Neural Network
NLP: Natural Language Processing
MAE: Mean Absolute Error
M.Tech: Master of Technology
TN: True Negative
TP: True Positive
HMM: Hidden Markov Model
RF: Random Forest
SGD: Stochastic Gradient Descent
GRU: Gated Recurrent Unit
LSTM: Long Short-Term Memory
RNN: Recurrent Neural Networks
MHA: Multi-Head Attention
MLM: Masked Language Modelling
NSP: Next Sentence Prediction
BCE: Binary Cross-Entropy
AdamW: Adam with Weight Decay
CSV: Comma-Seperated Values
σ**:** Standard deviation or logistic sigmoid function
$\approx$**:** Approximately equal to
$=$**:** Equality

i, j, k**:** Indices used to represent data points or samples
m**:** Number of column
n**:** Number of row
$\sum$**:** Summation symbol
X**:** Input or feature matrix
Y**:** Output or target variable
p_i**:** Probability of instances
x_i**:** Data point in training dataset
$f_i(x)$**:** Predicted output.
N**:** Number of decision tree.
y**:** Output Variable.
w_i**:** Weight corresponding to input features
e^{-z}**:** Exponential function
$y_{pred}^{(i)}$**:** Predicted value
$y_{true}^{(i)}$**:** Actual value
b**:** Bias term

1. INTRODUCTION

A variety of machinery is relied upon us daily, yet without maintenance, all machines eventually break down. With predictive maintenance, you can predict when a machine breakdown will happen. In this manner, you may maximize equipment lifespan, plan maintenance, improve inventory management, and get rid of unscheduled downtime. With predictive maintenance, you may calculate a machine's failure time. It is easier to determine when to arrange maintenance for your equipment when you are aware of the expected failure time. In addition to foretelling a potential breakdown, predictive maintenance helps you locate malfunctions in your intricate machinery and determine which pieces require repair.

1.1 Motivation

The motivation of the study is to gain knowledge of the different types of industrial maintenance and the difficulties associated with them. In addition, to acquire and utilize machine learning algorithms and sophisticated analytics approaches to forecast industrial machinery failures. To prevent a breakdown in the plant or manufacturing line, this will assist the maintenance crew in making repairs and scheduling maintenance in advance of issues.

1.2 Background

The role of machines in our daily lives is increasing these days. Whether flying, traveling, building houses, roads, or other infrastructure, there is a dependence on machines. Not only do devices save time, but they also boost output. Various machine types are essential to both the automobile and manufacturing industries. Some of the machinery utilized in these industries is quite complicated and requires frequent maintenance to carry out their daily tasks, and very few of them are simple to use and run. Both productivity and maintenance costs were raised by this maintenance. In light of the

COVID-19 epidemic, the majority of industries have undergone a digital transformation. Additionally, the manual maintenance procedure needs to be automated. Ensuring critical asset reliability and supporting on-demand production requirements through the application of advanced analytics techniques. The oil and gas industry was the first to use predictive maintenance. Predictive maintenance is replacing reactive maintenance in the automotive sector as well.

1.3 Industrial Maintenance

Due to early equipment failure, maintenance costs are often higher in many industries than operating and production expenses. Any industry's capacity to turn a profit is typically dependent on its maintenance procedure. In industries, maintenance is typically performed when machinery reaches a specific age or breaks down. While periodic maintenance is a wonderful idea, it doesn't tell you anything about the equipment's long-term health. Depending on the resource, several types of maintenance can be carried out to maximize the reliability of the production lines and the equipment. Figure 1.1 lists the most typical categories of industrial maintenance.

1. Preventive Maintenance
2. Reactive Maintenance
3. Predictive Maintenance

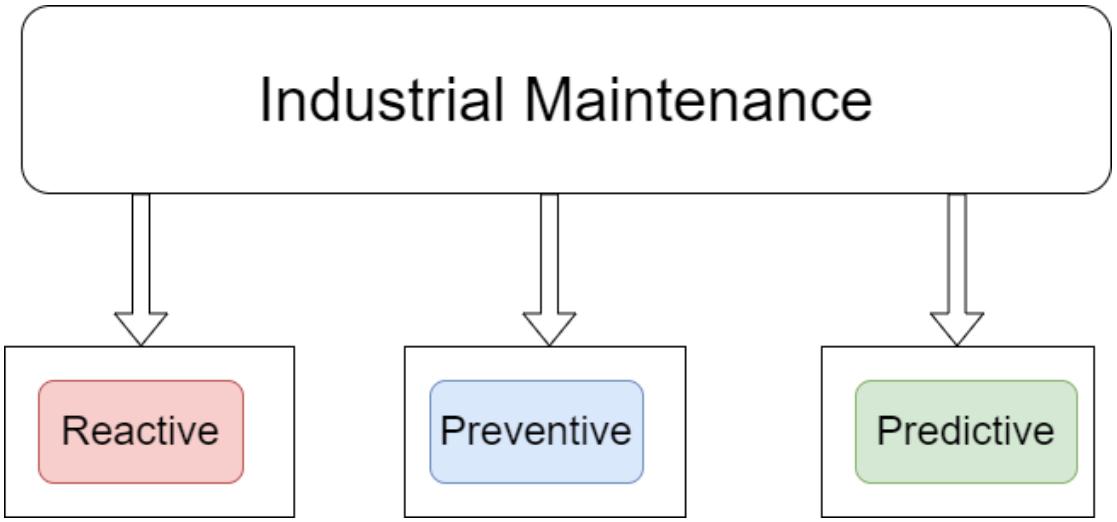

Figure 1.1: Industrial Maintenance Types

1.3.1 Preventive Maintenance

This method replaces the machine or component before it breaks down. Unscheduled Maintenance can be avoided using this. As indicated in Figure 1.2, the maintenance will be carried out at regular intervals. These are its disadvantages

- The equipment or component is not used to its maximum potential.
- Over maintenance is carried out.

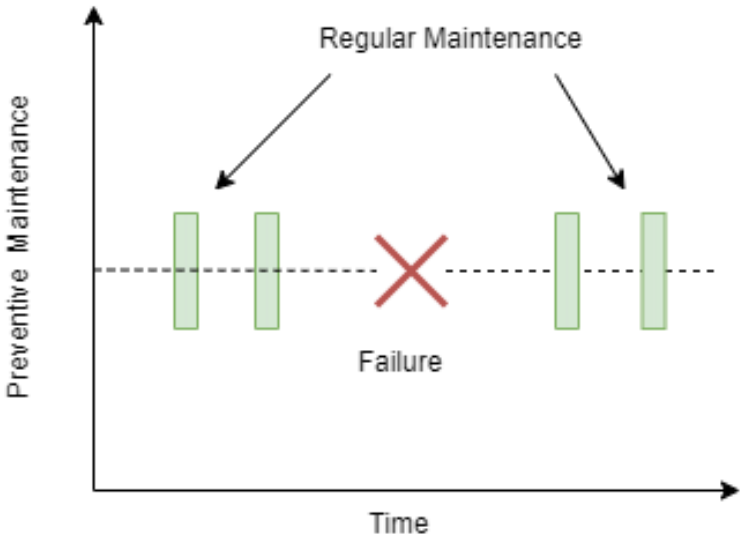

Figure 1.2: An illustration of Preventive Maintenance

Regular maintenance has several drawbacks, including:

- Increased breakdown times.
- Decreased productivity resulting from routine maintenance.
- Overmaintenance of certain machinery or equipment.
- Increased operation costs.
- Shorter machine life
- Requires more skilled labor to maintain the machinery

1.3.2 Reactive Maintenance

Under this method, maintenance can be carried out when machinery or its parts malfunction or stop working. Typically, as seen in Figure 1.3, maintenance is carried out

following an equipment breakdown. Despite the equipment or component being used for its entire lifespan, this strategy has the following drawbacks:

- Unscheduled Maintenance.
- Downtime is increased.

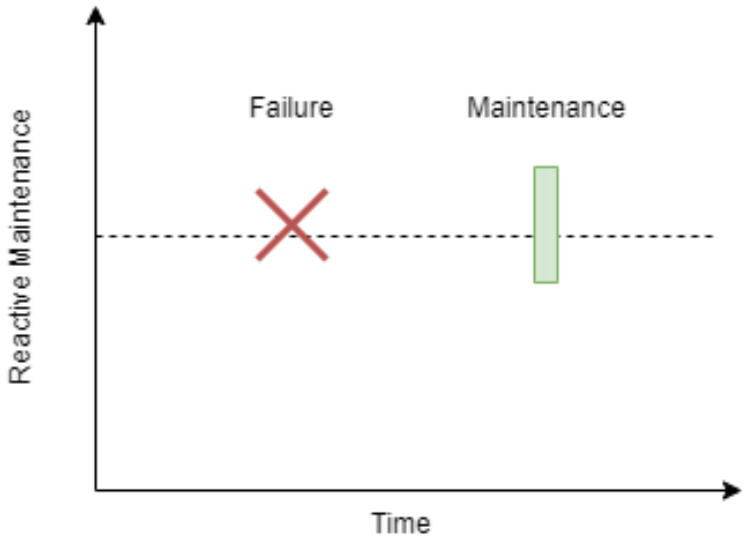

Figure 1.3: An illustration of Reactive Maintenance

1.3.3 Predictive Maintenance

As shown in Figure 1.4, it anticipates faults and performs repairs on machinery or equipment before issues arise. Only the machines or components that are on the verge of failure are replaced. This approach extends the equipment's lifespan. Predictive maintenance has several benefits, including

- Ability to minimize unscheduled downtime.
- Condition monitoring can assist in identifying faults or the health of the equipment, preventing expensive equipment breakdown.
- By decreasing inspection and early repair, it reduced the scheduled downtime.

The disadvantage of this approach is the heavy initial expenses associated with developing a system of that standard.

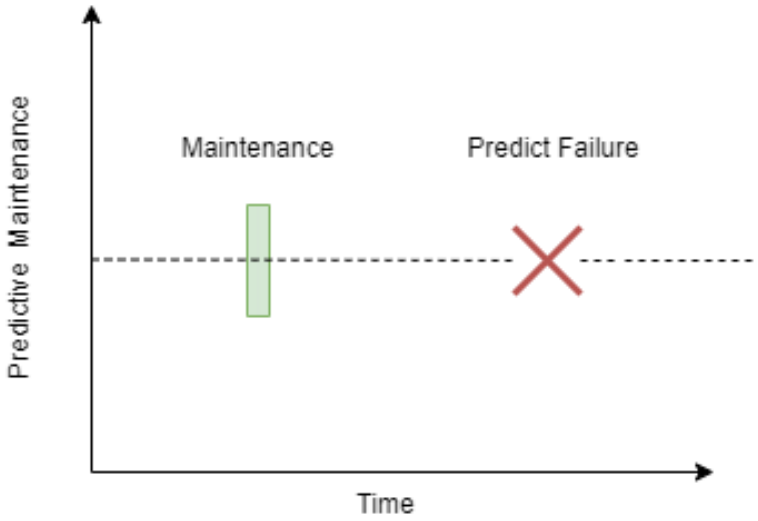

Figure 1.4: Predictive Maintenance overview

1.4 Problem Defination

Rotating machines are essential components in numerous industrial applications, including wind turbines, oil and gas extraction, automotive production, manufacturing, mining, recycling facilities, and hydropower. These machines play a pivotal role in maintaining operational efficiency and productivity. However, rotating machines are prone to errors and imbalances that can result in significant financial losses due to costly repairs, reduced productivity, and potential total machinery or plant failure. Early detection of flaws, particularly imbalances in rotating shafts, is crucial to preventing severe damage and unexpected breakdowns.

Imbalances in rotating shafts can lead to excessive vibrations, which cause premature wear and tear on roller bearings and other critical machinery parts. This not only raises the cost of replacement parts but also shortens the service life of the machinery. Therefore, identifying imbalances at an early stage is vital for reducing maintenance costs, avoiding unnecessary production interruptions, and extending the lifespan of machines. The problem at hand is to develop a reliable method for detecting imbalances in rotating shafts. This can be framed as a binary classification problem, where the objective is to classify machines into two categories: healthy and unhealthy, based on the analysis of vibration data. By achieving accurate early detection, industries can enhance machine reliability, optimize maintenance schedules, and ensure uninterrupted production processes.

1.5 Proposed Solution

The rapid rise of ML and DL in industrial automation is important to note. With affordable access to data from sensors and IoT devices, storing this information in databases has become more feasible. The data coming from the sensors mounted on machines that are under both healthy and faulty states simplifies the process of building and training ML and DL models to predict the current and future states of industrial machinery. This predictive capability helps the technical team avoid unscheduled maintenance.
For solution deep learning techniques are used, specifically the BERT architecture from transformers, to predict whether a machine is in a healthy or unhealthy state. This enables the maintenance team to repair mechanical parts of machines or replace them proactively, preventing damage before they occur.

1.6 Thesis Objective

1.6.1 Classification of Machines

- The goal is to develop a Deep Learning Algorithm that can be used to solve the problem discussed in section 1.4.
- The main focus of the study is the analysis of vibration data from rotating machinery. The dataset used in this research was sourced from Kaggle, an established website that hosts datasets related to data science and machine learning and is freely available.
- The Bert Transformer Model is used for the classification task i.e. to detect the health of the machine. Measurements of accuracy, precision, recall, F1-score, and Average Loss were taken to assess their performance.

1.6.2 Outline of the thesis chapters

The thesis starts by presenting an introduction to the objectives, which is followed by a thorough examination of existing literature related to the topic. The thesis is structured into different chapters. Chapter 1 will discuss about introduction along with the thesis objective. Chapter 2 will be for the Literature Review. Chapter 3 discusses the Dataset used for the task and Measurement setup to collect the Vibration data. Chapter 4 discusses the basics of the Neural Network, Transformers, and the BERT Architecture. Chapter 5 is focused on Data Preprocessing and Dataset Plot. Chapter 6 is the last chapter which is about the Results and Conclusion.

2. Literature Review and Research Gap

2.1 Literature Review

This section includes an overview of the literature on rotating machine shaft unbalance detection and transformer condition monitoring. To identify the gaps in the field's successful approaches and to understand the current research trends, a survey of recent literature is conducted. Author in [11] has done the unbalanced detection of a rotating shaft using vibration data. The author used algorithms like Convolution neural network (CNN), Fully connected Neural network (FCNN), Random Forest (RF), and Hidden Markov Model (HMM). The different accuracy achieved is 94% and 93.6% (with 2 and 4 convolution block) using CNN, 91.6% is the accuracy achieved using FCNN, 93.2% is the accuracy achieved using Random Forest algorithm, and using HMM the author achieved an accuracy of 95%. The limitation of this paper is that with the smaller unbalances, wider variations between the different approaches were found.

The author in [12] has done fault detection of the motor considering the speed of the motor. The paper proposes a deep learning algorithm for motor fault diagnosis that considers motor speed, a factor not included in previous studies. To address the increased data variance due to variable speed, a simplified CNN is used to reduce model complexity and overfitting while achieving high classification accuracy (over 99% with sufficient data). This method offers advantages for industrial applications but is limited to the amount of training data. The authors propose the use of the Internet of Things (IoT) to collect motor signals and create a big data archive for improved performance.

Author in [13] explored using artificial intelligence to diagnose faults in power transformers. Their approach relied on dissolved gas analysis (DGA) content, a technique commonly used to identify transformer problems. The study compared two different

methods for training the AI model, known as optimizers: Adam and Stochastic Gradient Descent (SGD). During training, Adam performed exceptionally well, with accuracy rates of 90% as opposed to SGD's relatively low 68–70%. However, this initial lead wasn't quite as impressive when applied to real-world scenarios. Here's the catch: both optimizers delivered similar accuracy (around 70-75%) when tested on real data. This shows that Adam's greater variance could be a weakness. In machine learning terms, variance refers to how much a model's performance can fluctuate between different training datasets. A high variance may indicate overfitting, a condition in which the model finds it difficult to adjust to new information because it has become too familiar with the details of the training set. The researchers readily acknowledged other limitations to their approach. SGD's lower overall accuracy makes it a less promising optimizer. Additionally, the study identified the need for more advanced techniques within the model's architecture to truly unlock its potential. While Adam's training prowess is undeniable, these limitations underscore the need for further exploration. By refining the model's architecture and potentially investigating other optimization methods, researchers hope to develop a more robust fault detection system for power transformers. This could involve creating a system that achieves both high accuracy during training and testing, while also reducing the risk of overfitting. Ultimately, such a system would lead to a more reliable and efficient diagnosis of faults in power transformers.

Author in [14] proposes a novel deep learning approach for Non-Intrusive Load Monitoring (NILM). It combines feature learning with transformer-enhanced convolution neural networks (CNN), incorporates a temporal scaling module for multi-scale information, and utilizes a decoder with residual gated recurrent unit (GRU) modules and a transformer-based CNN module. This approach achieves high accuracy (over 91%) and F1 scores in experiments, outperforming existing methods. However, the study acknowledges limitations: currently it only analyzes active power data and the model's size leads to high computational costs. Future work should explore incorporating more variables and designing a more lightweight model for broader applicability.

In [15] research proposes a deep learning method for monitoring the health of bearings using vibration data. The method first converts the complex vibration signals into a simpler format using an acceleration envelope detection algorithm. Then, a convolution neural network (CNN) is trained to identify the condition of the bearing (healthy, inner

race fault, outer race fault, etc.) based on these transformed signals. Finally, a separate Long Short-Term Memory (LSTM) network is used to estimate the remaining useful life of the bearing. Experiments showed that the CNN achieved high accuracy (up to 100%) in identifying bearing conditions, while limitations were identified in setting the failure threshold for remaining useful life estimation. This method offers a promising approach for automating bearing health monitoring in various applications.

In [16] researchers investigated a Transformer-based deep learning model for predicting ventilator pressure, aiming to refine ventilator control. The model, trained on a dataset from Google Brain, utilized residual connections to extract features from time-series ventilator data during K-Fold cross-validation. This approach achieved a mean absolute error of 0.13 on an unseen test set, ranking among the top performers in a competition. While limitations like data requirements, computational demands, and real-time implementation need to be addressed, this study highlights the promise of Transformer models for improved ventilator pressure prediction and potentially more precise ventilator control. Author in [17] tackles fault detection in industrial motors using a deep learning approach that addresses the limitations of traditional convolutional models. The proposed method utilizes a multi-scale deep learning architecture with three convolutional blocks employing different kernel sizes to capture features at various resolutions from raw vibration signals. An attention mechanism is then implemented to analyze the importance of these multi-scale features and combine them effectively. This approach offers improved fault detection performance compared to traditional methods, especially in simulations with limited training data. However, potential limitations include the computational complexity of the attention mechanism and the need for substantial training data for optimal model performance. Overall, this research presents a promising approach for fault detection in industrial motor drives using deep learning.

2.2 Research Gap

It has been concluded from our extensive analysis of the literature that a range of datasets and machine-learning techniques have been used in previous studies, including CNN, FCNN, HMM, and RF. The investigations did, however, reveal several flaws, such as underfitting, overfitting, insufficient hyperparameter tuning, feature selection concerns, poor model architectures, the use of inferior optimizers, and incorrectly con-

figured kernel initializers and layer counts.

It describes a novel methodology that utilizes the power of BERT to achieve higher accuracy than previous research. This method, in contrast to earlier ones, performs exceptionally well at contextual understanding in numerical vibration data, which is a great help in handling complicated health monitoring conditions.

The method used precisely tunes the hyperparameters of BERT to maximize its performance. By carefully adjusting these parameters, the aim is to strike a perfect balance between model complexity and its ability to generalize to unseen data. By fine-tuning the BERT model to obtain the most appropriate details from the vibration data, the rotating shaft unbalance is identified with improved classification accuracy.

Additionally, the best optimizer available for the model is used, which guarantees effective convergence and better results. With a better optimizer, the model can be trained more efficiently and better results can be obtained. Furthermore, careful consideration has been given to feature selection and preprocessing methods. The preparation techniques make sure that the input data is normalized, transformed, and model-ready. Additionally, strong feature selection techniques have been taken into practice, finding the most essential and informative features for the model's training. With the help of this stage, noise and unnecessary information are removed, improving accuracy.

3. Electrical Machine Predictive Maintenance Data

3.1 Introduction

- Vibration of the machine is the oscillation of the machine from its initial position.
- The use of different technologies allows not only the identification of whether a machine is healthy or faulty but also the pinpointing of the exact part where the fault is located.
- The fault can be identified at an early stage in the rotatory machine using vibration sensors to protect the machine from future damage and production downtime.

3.2 Measurement Setup

The experimental setup includes a galvanized steel bracket supporting a WEG GmbH UE 511 T electronically commutated DC motor against an aluminum base plate, controlled by a WEG GmbH W2300 motor controller. This motor drives a system that induces imbalances and monitors resulting vibrations. The motor's speed, ranging from 300 to 2300 RPM, is adjusted by varying the voltage to the motor controller. A coupling from Orbit Antriebstechnik GmbH connects a 12 mm diameter, 75 mm long shaft to the motor's drive shaft. An imbalance holder, created with a 3D printer using nylon material, secures the shaft passing through a galvanized steel roller bearing block. Varying weights in the imbalance holder, a 52 mm diameter disc with symmetrical recesses, allow for precise imbalance simulation. The motor mount and bearing block have vibration sensors from PCB Synotech GmbH that are linked to a 4-channel data-collecting system for monitoring. To calculate the motor's rotation speed, the system digitizes the rotor position signal. Figure 3.1 provides a visual representation of the measurement setup, and Figure 3.2 provides a schematic of the configuration.

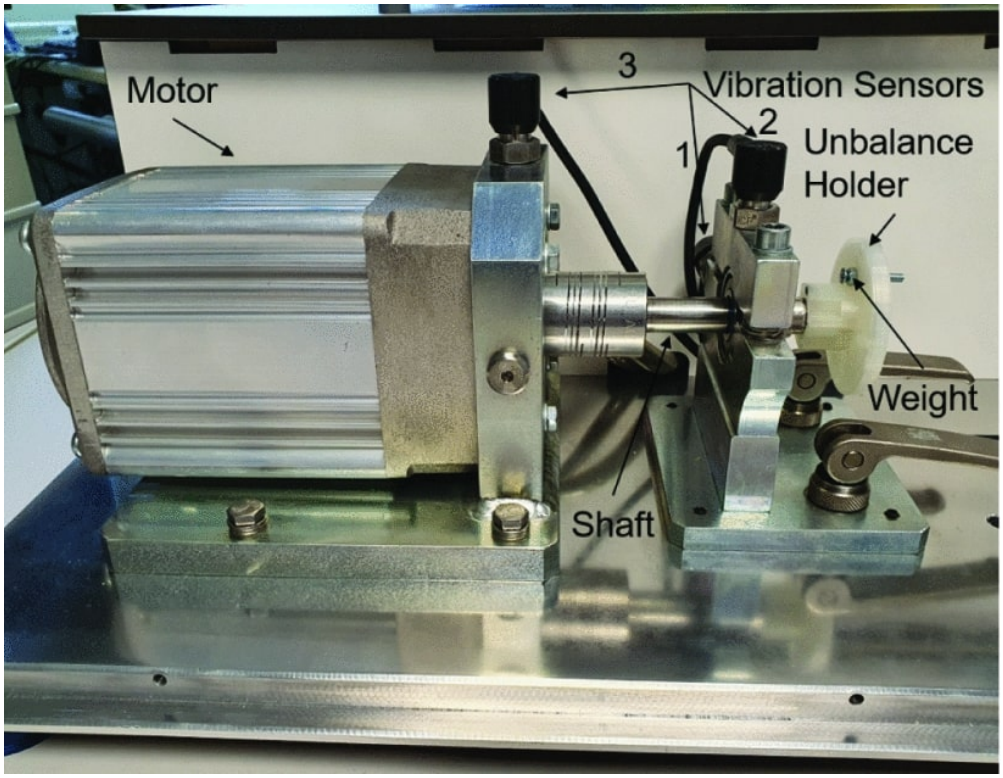

Figure 3.1: Measurement Setup. [11]

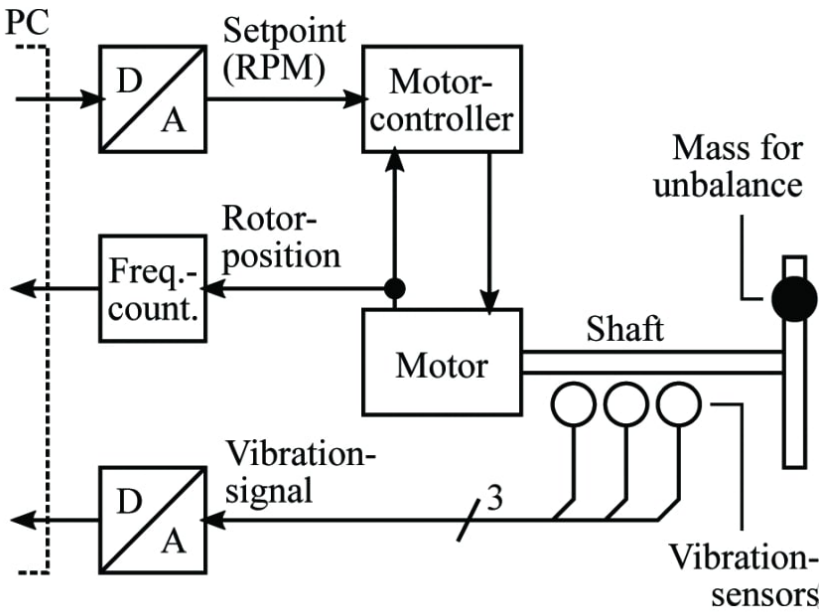

Figure 3.2: Schematic Diagram [11]

3.3 Open Source Dataset

The data set used in this research is open-source data from Kaggle. The vibration data in this dataset has been recorded with different levels of unbalances. Minor unbalances have a minimal impact on the signals detected by vibration sensors. The dataset was carefully designed to be replicable, relevant to industrial settings, and to accurately depict a practical scenario. To cater to diverse industrial needs, vibration data was recorded at different rotational speeds and presented in a tabular format (CSV) for enhanced usability. Four datasets were gathered, each corresponding to a specific level of unbalance, along with an additional dataset representing a balanced condition. These datasets are stored as CSV files, each containing five columns. Parameters for each sample include: Vin: Input voltage in volts (V). Measured RPM: Rotational speed in revolutions per minute. Vibration 1, Vibration 2, and Vibration 3: Signals from three vibration sensors. Each column in the dataset has a sampling frequency of 4096 Hz which means 4096 values per second to ensure precise measurements. The rotation speed values were adjusted to match this sampling rate for data consistency. Measurements were conducted separately for each unbalanced level, resulting in distinct datasets for development and evaluation purposes. The CSV files are named "1D.csv," where the letter indicates the dataset's purpose (either "D" for development/training or "E" for evaluation), and the number signifies the level of unbalance (ranging from "0" for no unbalance to "4" for high unbalance).

4. Proposed Deep Learning Models for Predictive Maintenance.

This section deals with the methodology employed in this thesis. The outline of techniques used is specified. More precisely, the focus is on BERT and Transformers.

4.1 Introduction

Feedforward neural networks, commonly known as Neural Networks (NN), are at the backbone of the subjects discussed within this section. Besides consistently delivering excellent outcomes, the key strength of NNs lies in their versatility. They can effectively analyze both structured and unstructured data for various learning tasks, including supervised (which is for labeled data), unsupervised (which is for unlabeled data), and hybrid learning scenarios. Since the beginning, the popularity and use of neural networks have increased dramatically due to the introduction of the backpropagation algorithm, which was first proposed by Rumelhart et al. in 1986. Other mathematical optimization techniques that have contributed to this trend include the various forms of Stochastic Gradient Descent (SGD) and the ongoing development of computational capabilities. An illustration of the basics of Fully connected Neural Network (FCNN) architecture, comprising an input layer and output layer with a single hidden layer is shown in Figure 4.1. The model's learning parameters encompass the individual connections between all neurons, depicted as weights represented by black arrows connecting the layers, denoted by **W**. Every link between the input layer's (blue), hidden units' (green), and output layer's (yellow) neurons corresponds to a particular $w \in W$. Figure 4.1 shows the fundamental design of a fully connected neural network. It consists of an input layer with three neurons representing distinct features, an output layer with two neurons, and a single hidden layer with four neurons (or hidden units). In this architecture, a bias

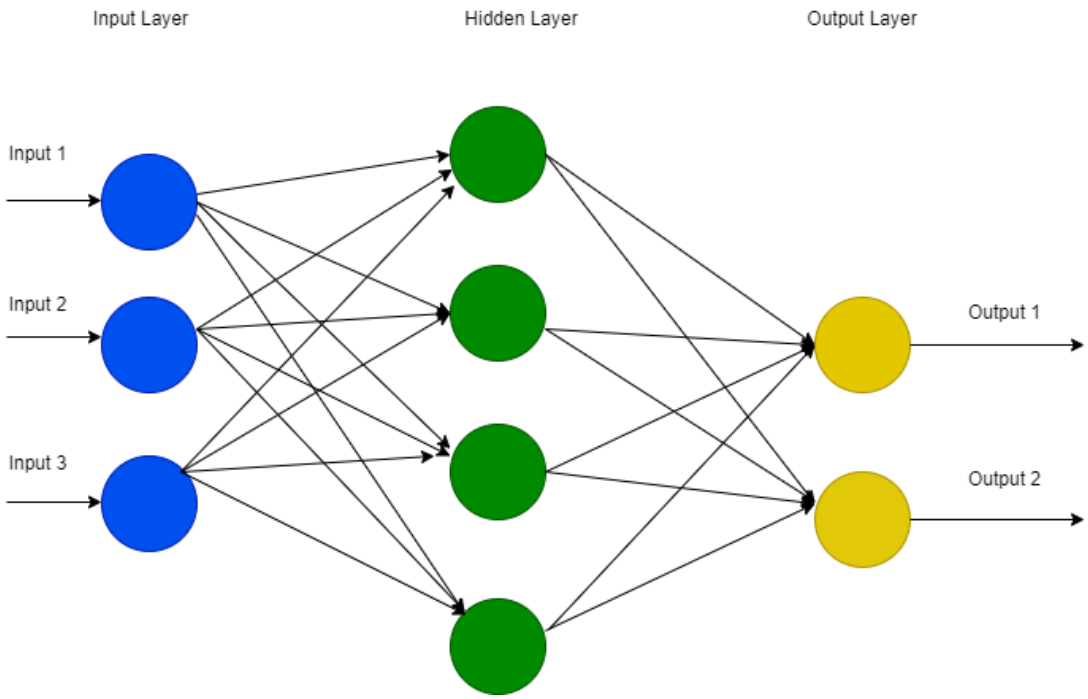

Figure 4.1: A basic fully connected neural network architecture.

term is typically added to each layer, except for the output layer, to enhance the model's learning capabilities. The connections between the layers, represented by the weights, are crucial for the model to learn and make predictions from the input data. Each connection between the neurons of the input layer, the hidden units, and the output layer corresponds to a specific weight value. This simple yet effective architecture serves as a foundation for more complex neural network models used in various applications.

When data is fed into the model, each input (i.e. feature) undergoes multiplication by the weight associated with the connection between two units, followed by the addition of a bias term. The result of this computation is then processed by an *activation function*, which triggers the neuron connected to that specific connection. This process occurs at each layer within the network. Mathematically hidden layer input is expressed as

$$W^{\intercal}x + b \tag{4.1}$$

Where W^T is the transposed weights matrix; x is the input vector or the features; and b is the vector containing the bias terms. The sigmoid function is being used for the classification of the output of the final layer which is defined as follows, in multi-label classification problems:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

A loss function is used to compare the output of the model during training to the true class. The weights are continuously updated by backpropagation[1]. For regression issues, the mean squared error loss is commonly utilized, while for classification problems, the binary cross entropy loss is defined as

$$L(\hat{y}_n, y_n) = -\sum_{c=1}^{C} y_{n,c} * \log(\hat{y}_{n,c}) + (1 - y_{n,c}) * \log(1 - \hat{y}_{n,c}) \tag{4.3}$$

for a multiple-class classification problem, where $\hat{y}_n$ is the predicted probabilities of the model for n = 1 ... N, y_n which, in such case, is the true label and c= 1 ... C which represents the single class in our case. Real labels are typically binarized; if the instance does not belong to that class, it is set to 1; otherwise, it is set to 0, creating a binary vector whose length is equal to the number of classes in the problem. Thus, equation 4.3's sum always has one of its two terms multiplied by 0. Furthermore, to accelerate a training or tuning process, multiple instances are typically entered into the model. A batch is another term for a group of input instances. In this case, each instance in the batch has its loss computed independently. After that, the weights are updated using the overall loss, which is determined by an average of all of these losses by 1/β, where β is the number of instances.

NNs are referred to as feedforward networks since all data is evaluated by flowing from the input to the output layer through the in-between hidden layers without any feedback connection. When modeling data sequences where each instance is connected to the others in some way, this becomes problematic. NNs are extended to the circumstance where it has been enabled these connections generate Recurrent neural networks (RNN), which are earlier forms of the transformers family of models that will be used in this thesis. In section 4.2, the latter's structure is examined in further detail.

[1]It is a supervised learning algorithm used in neural networks to update the weights by minimizing the error between the predicted and actual outputs through gradient descent

4.2 Transformers

In 2017, a novel architecture fundamentally transformed natural language processing (NLP) by resolving problems associated with the use of RNNs and CNNs without their use. Due to their higher parallelization than RNNs or CNNs, transformer-based models, which were developed by Vaswani et al.(2017) [1], have resulted in considerable increases in the models' performances as well as a reduction in training time. The Attention mechanism (discussed in more detail in section 4.2.1) makes this feasible. A transformer's fundamental structure is depicted in Figure 4.2.

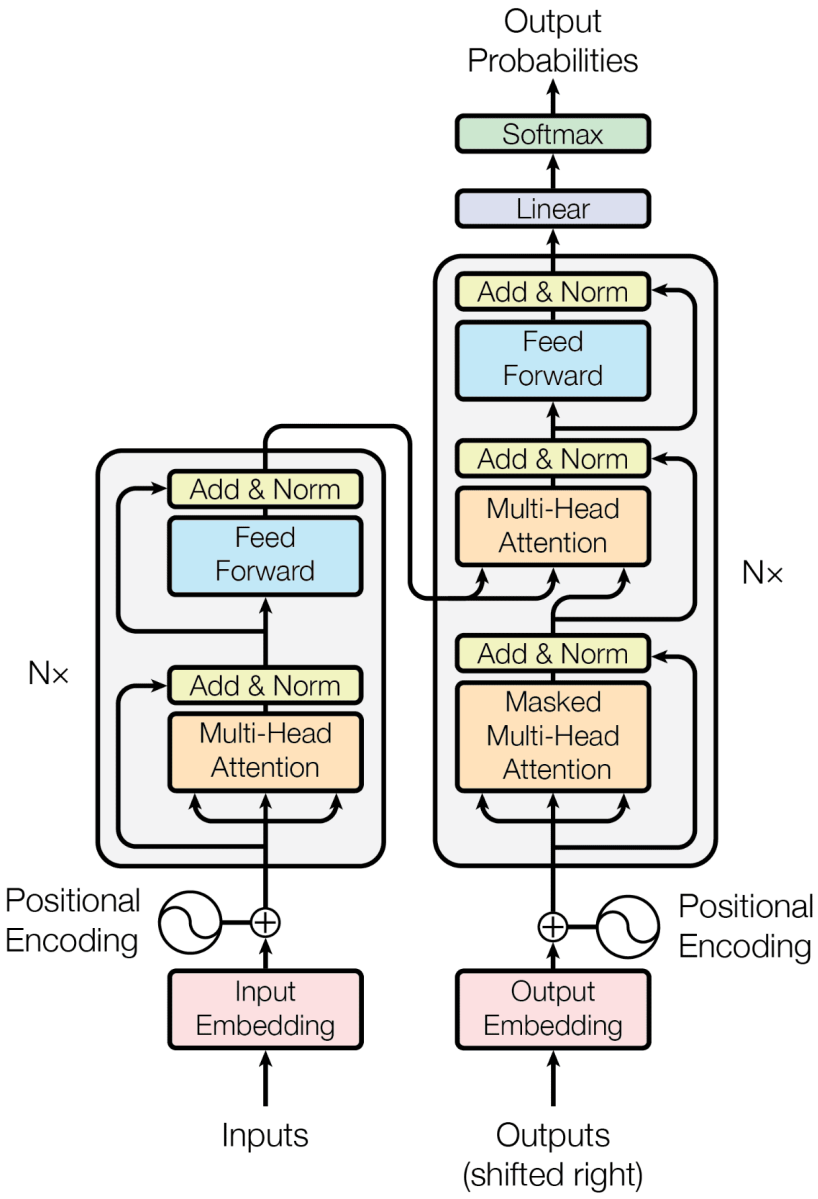

Figure 4.2: Basic Architecture of Transformer. [1]

The image's left encoder and right decoder are the two primary parts that are easily distinguished. The model has to be given additional information regarding the sequence's elemental order because there are no recurrences or convolutions. If this weren't the case, every embedding would always yield the same representation, regardless of the context. This is accomplished by incorporating within the embeddings a positional encoding, or vector of the same dimension. To compute these, the authors suggest using the sine and cosine functions, which are, respectively, defined as

$$\mathrm{PE}_{\mathrm{p},2n} = \sin\left(\frac{\mathrm{p}}{10000^{2n/\mathrm{dmodel}}}\right) \tag{4.4a}$$

$$\mathrm{PE}_{\mathrm{p},2n+1} = \cos\left(\frac{\mathrm{p}}{10000^{2n/\mathrm{dmodel}}}\right) \tag{4.4b}$$

where p is the position and n is the dimension of the input token[2] and d_{model} is the dimension of the output coming by the model. These two functions were chosen together because of their linear characteristics, which facilitate the model's ability to learn which tokens to focus on. However, alternative techniques may theoretically be used to calculate the positional encoding. In the preprocessing stage, the output is transformed by the encoder into an abstract, continuous sequence containing all the learned information for that input. The encoder is composed of a stack of identical layers, although only one layer is shown in the illustration. Each layer consists of two main sub-layers: a fully connected neural network and a Multi-head attention (MHA) layer. Both of these sub-layers are normalized before passing on to the next sub-layer. Additionally, each sub-layer includes a residual connection, which provides the model with extra information. The output from the encoder is then passed to the MHA sub-layer of the decoder. The decoder uses this output to generate text sequences. The decoder is also made up of a stack of identical layers, with N[3] layers used in their experiment. The key difference between the encoder and decoder layers is that the decoder includes an extra Masked MHA sub-layer before its MHA sub-layer. The decoder's final output is fed into a softmax function, which calculates the probability of each word, and a linear layer, which acts as a classifier. After the loss is calculated, the model's weights are

[2]This indicates that the sine function is used to calculate the positional embedding for tokens that are at even positions in the sequence, and the cosine function is used for tokens that are at odd positions.

[3]N=6 for their experiment's encoder and decoder

updated through backpropagation. Because decoders are autoregressive, their outputs, once preprocessed similarly to the encoder outputs, are fed back into the model as new inputs. One of the advantages of Transformers is that the encoder and decoder can be used independently as separate models. Only the encoder will be used in this thesis.

4.2.1 Attention!

The Attention mechanism, or simply attention, is the neural network behind transformers. The three fundamental components of this technique—a query (q), a key (k), and the value (v)—were initially put forth by Bahdanau et al. (2014). [2]. It can be described as a function that maps vectors representing queries, keys, values, and outputs to output when combined with pairs of keys and values. Self-Attention is another name for the attention mechanism found in encoders. A model can correlate each word (token) in an input sequence with its other words due to this type of attention. The objective is to train the model to identify which previous tokens require processing to process the current token. This is accomplished with a scoring function that calculates " significance " ratings for each query-key pair. Let us examine this mechanism's operation step-by-step. The queries, keys, and values vectors in the matrices will be represented by the matrix notations Q, K, and V. The attention mechanism employed by the transformer design is depicted in Figure 4.3. Three different linear fully linked layers receive the embedding in parallel. These produce queries, keys, and values matrices as their output. They represent three distinct abstractions of our embedding, in our opinion. For computing these, the weights (matrices) are W^Q which is for Q, W^K used for K, and W^V is for V, in that order. Since they are the model's typical parameters, they are learned throughout the training process. Next, the Scaled Dot-Product Attention will be computed (Figure 4.3(A).

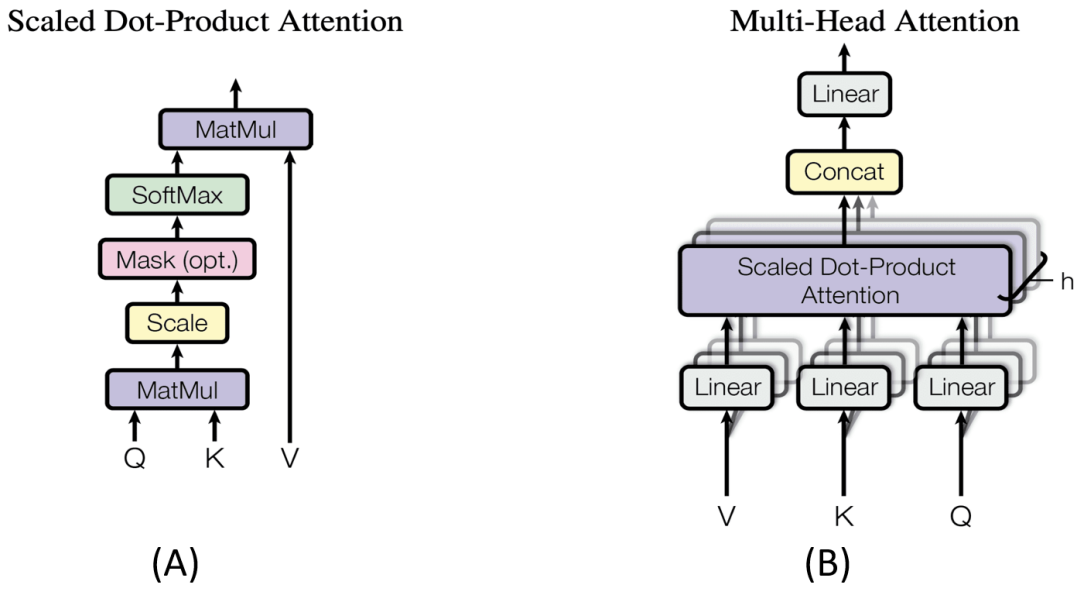

Figure 4.3: Attention mechanism with the scaled dot-product attention (A) and the multi-head attention (B) [1]

In this process, Q and K undergo a dot-product multiplication of the matrix. This process yields a (scoring) matrix, the entries of which, or scores, indicate the relative relevance of the other tokens for a specific token. Therefore, the focus on the other tokens is quantified in this matrix (the higher the score, the higher the focus). Next, this scoring matrix is scaled by $\sqrt{d_k}$, where d_k is the key's dimension. This allows gradients to be more stable because doubling values could lead to explosive consequences and the vanishing gradient problem (Pascanu et al., 2012) [3]. The scaled final score for each query-key pair is then calculated using a softmax, producing probabilities that is

$$\alpha_{ij} = \frac{\exp(q_i^T k_j)}{\sum_{l=1}^{n} \exp(q_i^T k_l)} \tag{4.5}$$

where α_{ij} denotes the attention score for the dot product of i$^{\text{th}}$ query vector and j$^{\text{th}}$ key vector. This suggests that when scores rise, so do scores fall, providing the model greater confidence in determining which tokens to attend. This action multiplies the outcome by V. The softmax output determines the relative importance of each character in the values matrix. The Scaled Dot-Product mathematical formulation of the attention technique is

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dk}}\right) V \tag{4.6}$$

where k^T represents the transpose of the K Matrix. Separate identical layers perform this entire process h times in succession. The system as a whole is known as Multi-

Head Attention, and it is represented in Figure 4.3 (B). A single layer is referred to as a head. The model extrapolates information by examining multiple (independent) representations of the same input at the same time. In essence, h unique fixed-dimension projections of Q, K, and V are discovered. Before being processed further, these are concatenated together. Therefore, Multi-head Attention can be defined as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{4.7}$$

and Single-attention head as

$$\text{head}_h = \text{Attention}(QW_h^Q, KW_h^k, VW_h^V) \tag{4.8}$$

with $W_h^Q \in \mathbb{R}^{\text{d}_{\text{model}} \times d_q}, W_h^K \in \mathbb{R}^{\text{d}_{\text{model}} \times d_k}, W_h^V \in \mathbb{R}^{\text{d}_{\text{model}} \times d_V}$ and $W^O \in \mathbb{R}^{hd_v \times \text{d}_{\text{model}}}$ as the matrix of the parameter, and d_v is the dimension of the values vector W^O or the weights of the outputs, is the weights matrix of the final layer after concatenation. The base variation of BERT, which is employed in this work, has h = 12 and for each head $d_k = d_v = d_{\text{model}}/h$ = 768/12 = 64.

4.3 Bidirectional Encoder Representations from Transformers (BERT)

This section introduces BERT, the main model which is used in this research. By pre-training deep bidirectional representations from unlabeled text, Devlin et al. (2018) [4] provide BERT as a pre-trained model that can be swiftly adjusted on specific downstream tasks. Figure 4.4 shows the general architecture of our used version. The authors propose two versions: a large version and a base version. The version used here is the basic one, consisting of a stack of twelve encoders. To calculate the Multi-Head Attention, each of them independently uses 12 attention heads. The result is embeddings with a dimension of d_{model} = 768, or roughly 110 million parameters. Before BERT, a common LM purpose was to anticipate the next token for an input sequence. Rather, BERT receives training on two primary tasks: Next Sentence Prediction (NSP) and Masked Language Modelling (MLM).

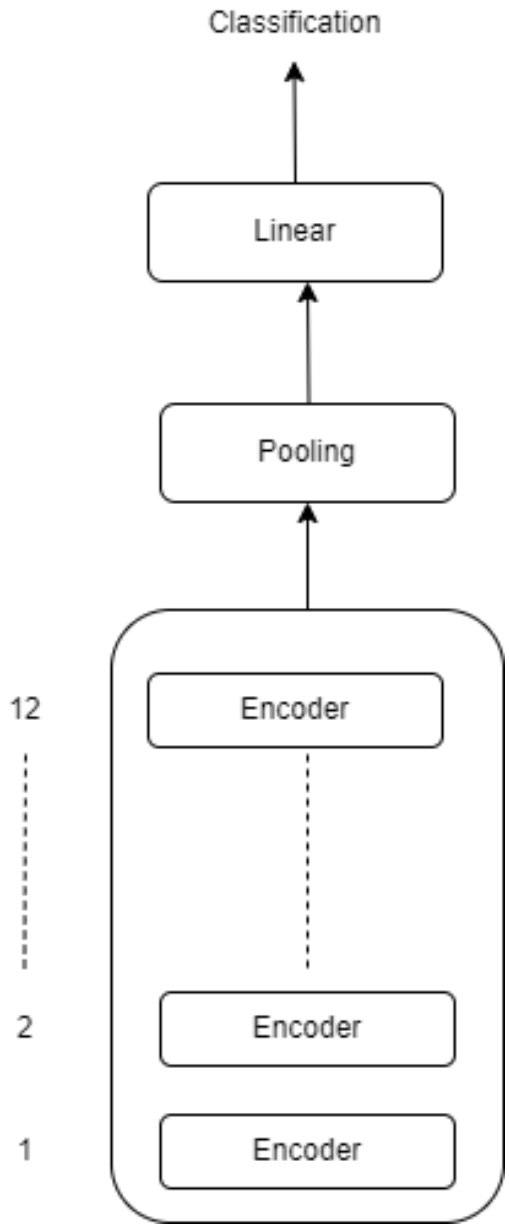

Figure 4.4: Architecture of BERT

4.3.1 Masked Language Modelling

Masked LM is a task in NLP aimed at predicting missing words or tokens within a given sequence of text. Here's how it works with an example: Imagine the sentence: "Sheila bought [MASK] from the store to bake a cake." In a Masked Language Modeling task:

1. **Masking Tokens**: Tokens in the input sequence are randomly selected to have a specific percentage (usually 15%) replaced with a unique [MASK] token. For our example, let's mask the word "flour". So, the sentence becomes: "Sheila bought [MASK] from the store to bake a cake."

2. **Training Objective**: The objective is to train a model that can predict, from

the context provided by the surrounding words, what word or token should take the place of the [MASK] token. In our example, the model should predict that "[MASK]" should be replaced by "flour" because it fits logically with the context of baking a cake.

3. **Contextual Learning**: During training, the model learns from the tokens that surround the masked [MASK] token. As a result, the model is better able to forecast missing words with accuracy since it can comprehend the relationship between words and their contexts.

By training on masked language modeling tasks, models like BERT can effectively capture nuanced linguistic relationships and improve their performance on various downstream NLP tasks such as text classification, sentiment analysis, or in our case unbalance detection.

4.3.2 Next Sentence Prediction

Next Sentence Prediction (NSP) is an important part of how BERT learns to understand the relationship between two sentences in a text. Imagine you have two sentences: "The team finished their project early." and "The client was very impressed."BERT learns from NSP training to determine if the second sentence flows naturally from the first. This facilitates BERT's understanding of sentence structure, which is important for activities like comprehending inquiries and drawing logical conclusions from text.

During NSP training, BERT adjusts how it processes sentences. It adds special markers to show if a token (like a word) belongs to the first sentence or the second. These markers help BERT figure out where one sentence ends and another begins. Additionally, BERT keeps track of each token's position in the sequence, ensuring it understands the correct order of words. For example, if it sees "The team" followed by "finished their project early.", it knows the second part comes after the first.

To help BERT in classifying or comprehending the text's overall meaning, the [CLS] token is inserted at the beginning of the input. In the input, the [SEP] token denotes the end of a sentence or divides it into multiple sentences.

Figure 4.5 shows how BERT uses these techniques to improve its understanding of sentences. In BERT's training process, it learns not only from predicting masked words (MLM) but also from predicting if sentences are connected (NSP). This dual approach has made BERT one of the most advanced models for understanding language in various

applications. Through NSP, BERT learns not just individual words but how sentences fit together, making it highly effective in tasks like answering questions and analyzing text for meaning and in our case for unbalance detection in machines.

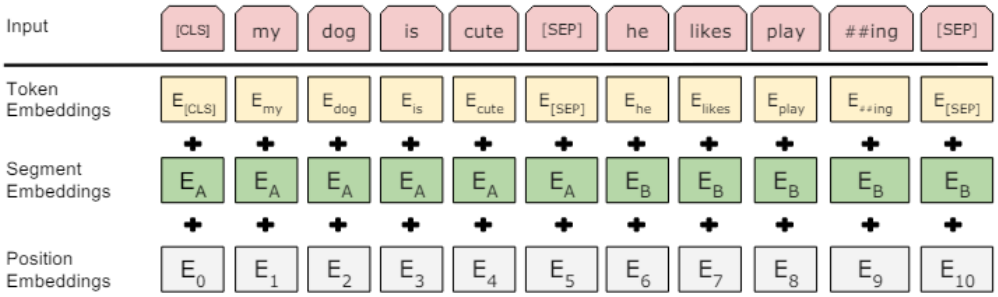

Figure 4.5: This image shows how BERT represents its input embeddings, which help the model understand the structure of sentences. [4].

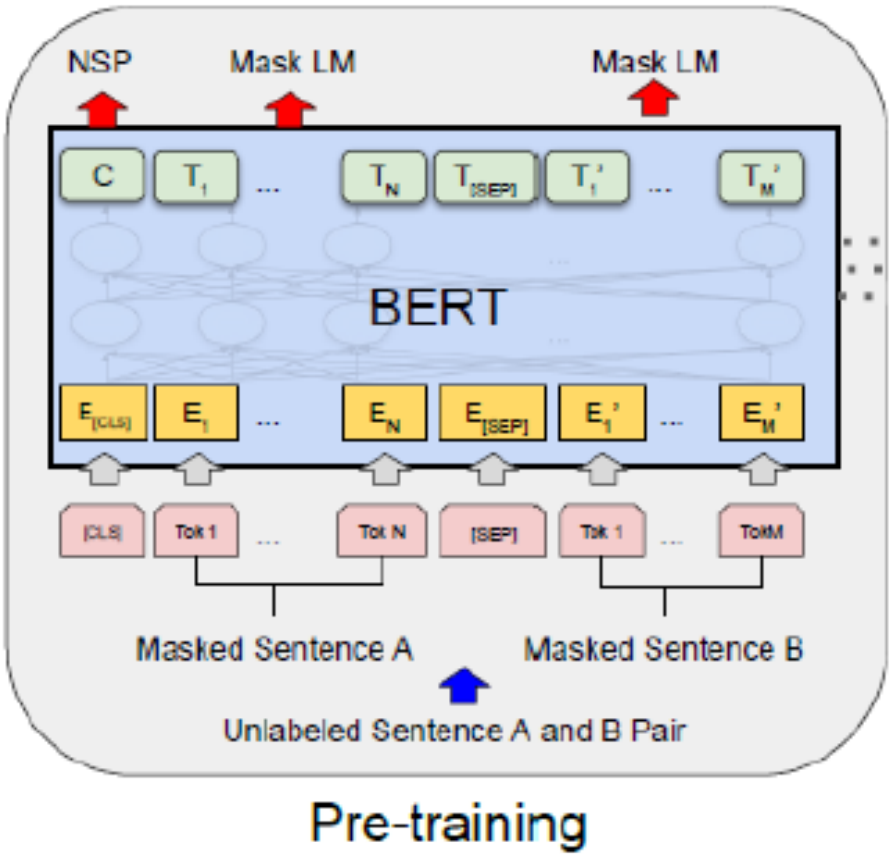

Figure 4.6: This diagram illustrates the pre-training process of BERT. [4].

5. Data Preprocessing

5.1 Steps Involved

The process starts with taking raw vibration data from a sensor as shown in figure 5.1. This data is loaded from kaggle platform. The data is likely stored in a suitable format like a CSV file.

After the data is loaded, it is preprocessed. This preprocessing step involve filtering the data to remove noise or irrelevant information and normalizing the data. It may also involve transforming the data into a format that is more suitable for machine learning algorithms. In this case, the data is segmented into windows. Each window represents a short period of time, such as one second. The features from each window are then extracted. These features are numerical values that summarize the important information in the window. For example, the features might include the mean, standard deviation, and frequency of the vibration data in the window.

Now coming to data labeling the preprocessed data is then labeled. In this case, the labels indicate whether the data is from a healthy or faulty machine. This labeling process can be done manually by a human expert, or it can be done automatically using a supervised learning algorithm.

The labeled data is then used to train a machine-learning or deep-learning model. A machine learning model is an algorithm that is trained on a set of data in order to make predictions on new data. In this case, the machine learning model is a binary classification model, which means that it is trained to predict one of two possible classes: healthy or faulty. The training process involves feeding the labeled data into the machine learning model. The model learns from this data and develops an internal representation of the data that allows it to distinguish between healthy and faulty data.

Now training and validation data's features have been extracted. A function is created to extract features from vibration data for each class in the training set (data with the

suffix 'D'). The extracted features and labels for each class are stored in separate variables. Similarly, the function is used again for each class in the validation set (data with the suffix 'E') to create separate feature matrices and label arrays. Separate feature matrices and label arrays for training and validation data are concatenated(stacked vertically). This results in combined feature matrices and label arrays for training and validation purposes.

Next is the Data Transformation block, This section includes applying a Fast Fourier Transform (FFT) along the time axis to the combined training and validation data. FFT can be used to convert the time-series data into the frequency domain, which might be useful for certain machine learning models. The combined and potentially transformed vibration data is converted into a list of comma-separated strings. Each string represents the features of a single window of data. Once the model is trained, it is tested on a separate set of data that the model has not seen before. This test data is also labeled, and the model's predictions are compared to the actual labels. The accuracy of the model is evaluated based on how well its predictions match the actual labels. Once the model is trained and evaluated, it can be used to make predictions on new, unseen data. In this case, the new data is vibration data from a machine that the model has not seen before. The model predicts whether the new data is from a healthy or faulty machine.

Next, the bert trainer class was developed and the fine-tuning process was done which is explained in detail in the next chapter (Section 6.2).

The prediction from the model is then used to assess the health of the machine. If the model predicts that the machine is faulty, then maintenance can be scheduled to address the problem. This can help to prevent machine failures and improve overall machine reliability.

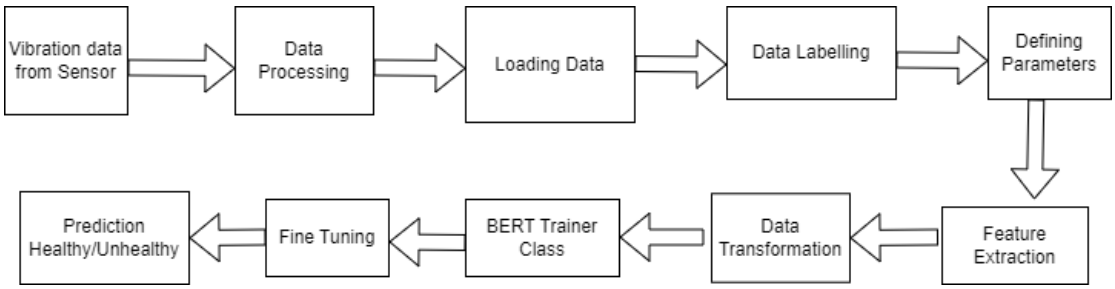

Figure 5.1: Flow Chart

In summary, the block diagram in Figure 5.1 shows a process for using machine learning or deep learning to monitor the health of machinery using vibration data. The

process involves loading and preprocessing the data, labeling the data, training a machine learning model, testing and evaluating the model, and using the model to make predictions on new data. The predictions from the model are then used to assess the health of the machinery.

5.2 Performance Evaluation

The performance of the models is evaluated by using the following techniques

- Accuracy
- Confusion Matrix
- Error Rate (ER)
- Precision
- Recall
- True Positive Rate
- F1 Score
- False Positive Rate

5.2.1 Accuracy

Accuracy is measured by taking the number of correct predictions and dividing it by the total number of predictions. A model with higher accuracy performs better, as it makes more correct predictions relative to the total number of attempts.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.2.2 Confusion Matrix

It is an easy way to assess how well classification models work. The number of classes that were correct and those that were wrongly predicted is shown in the matrix. It is used to determine the number of classes that were successfully classified by comparing

the estimated model's result with the class outcome. Following are the key term used in confusion matrix (Figure 5.2)

- **True Positive (TP)**: Accurately estimated the class to be "positive" when it is positive.
- **False Positive (FP)**: When a class is negative, it is incorrectly predicted to be "positive".
- **True Negative (TN)**: Predicted the class correctly as "negative," while the real class is also negative.
- **False Negative (FN)**: When a class is actually positive, it is incorrectly predicted to be "negative."

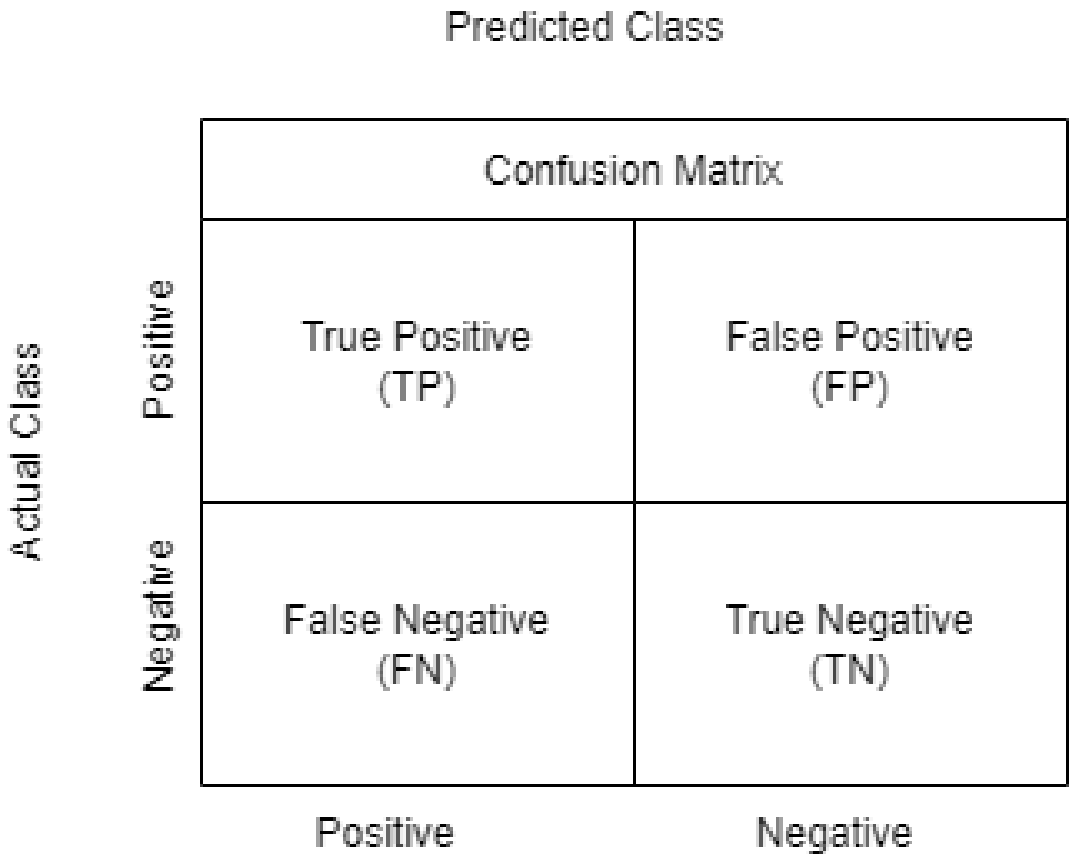

Figure 5.2: Confusion Matrix

5.2.3 Error Rate (ER)

The error rate is determined by dividing the number of incorrect predictions by the total number of predictions made on the dataset. It ranges from 0 to 1, where 0 means the model made no mistakes, and 1 indicates the worst possible performance.

$$\text{ER} = \frac{FP + FN}{TP + TN + FP + FN}$$

The error can also be calculated as

$$\text{ER} = 1 - \text{Accuracy}$$

5.2.4 Precision

It is a classification model's capacity to find only the appropriate information points. The number of true positives divided by the number of true positives + the number of false positives is the mathematical definition of accuracy.

$$\text{Precision} = \frac{TP}{FP + TP}$$

5.2.5 Recall

It is the capacity of a model to search through a data set and locate every significant case. Recall can be defined mathematically as the ratio of true positives to the sum of true positives and false negatives.

$$\text{Precision} = \frac{TP}{FN + TP}$$

5.2.6 True Positive Rate (TPR)

It helps to evaluate a model's capacity to identify true values. Another name for it is sensitivity or recall.

$$\text{TPR} = \frac{TP}{FN + TP}$$

5.2.7 F1-Score

For balance classes, accuracy is a good way to assess the model's performance; however, this method is ineffective for imbalance classes. An improved option for assessing the performance of unbalanced datasets is the F1 score. The model performs better the greater the value of the F1 score. The value of the F1 score ranges from 0 to 1.

$$\text{F1 Score} = \frac{2.recall.precision}{recall + precision}$$

5.2.8 False Postive Rate (FPR)

In FPR, negative cases were incorrectly categorized as positive cases. The FPR of the good model is low.

$$\text{FPR} = \frac{FP}{FP + TN}$$

5.3 Dataset Plots

5.3.1 Time-domain Plot

The dataset plot in Figure 5.3 is the time domain plot between the Vibration amplitude and the time. The plot in the first row represents the plot of the healthy machine with no unbalance (i.e. 0D.csv file) and the second-row plots represent the unhealthy machine with the highest unbalance (i.e. 4D.csv file). Both the healthy and unhealthy plots are from all three vibration sensors placed as shown in the measurement setup Figure 3.1.

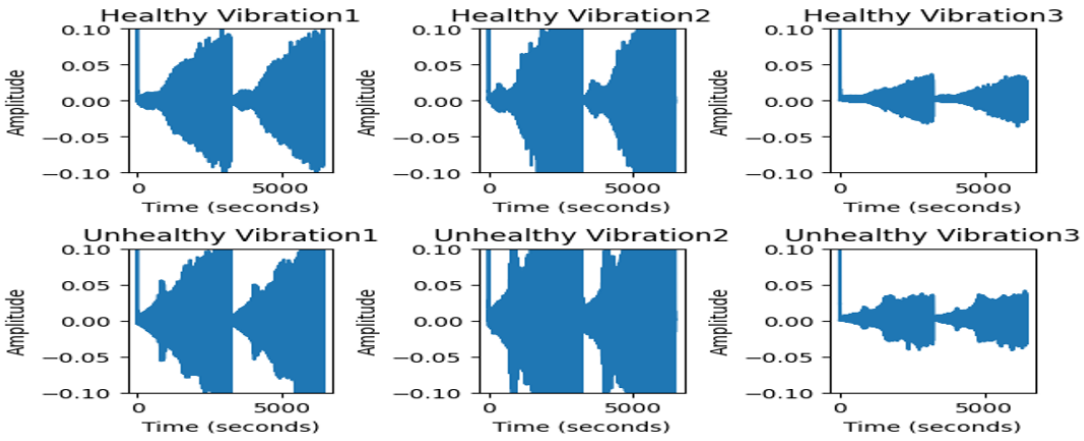

Figure 5.3: Time domain Plot of 0D and 4D dataset

Healthy Vibration Signals:

The amplitude in the graphs for the healthy condition ranges between -0.1 and 0.1, which suggests a somewhat stable operational state. With time, the waveforms in these plots show a more regular and consistent pattern, which is an indication of a motor that is operating smoothly. These signals' periodicity or repetition indicates that there aren't any noticeable abnormalities or interruptions to the motor's normal operation.

Unhealthy Vibration Signals:

Plots for the unhealthy conditions similarly show an amplitude range that is comparable to the signals in the healthy conditions, but it is more irregular and inconsistent. These signals fluctuate more over time and are less consistent. The waveform's anomalies and sudden fluctuations may be a sign of underlying problems that impair the motor's performance, such as imbalance, misalignment, mechanical wear, or other flaws. These

patterns point to possible instability and performance decline in the motor.

Detailed Analysis:

The small discrepancies between the healthy and unhealthy signals, even having identical amplitude ranges, show how difficult it is to identify defects by visual inspection alone. For the duration, the healthy signals exhibit consistent patterns, signifying a stable operating state. The unhealthy signals, on the other hand, show more frequent fluctuations and abnormalities, which are frequently early warning signs of motor issues.

High-level analytical methods are required to distinguish between these states. The complex nature of the changes may make a simple time-domain analysis insufficient. The frequency components of the signals can be analyzed using techniques like the Fourier Transform to spot changes in the harmonic content that might point to defects. Transient features that are not visible in the time-domain representation can be found via wavelet analysis.

5.3.2 FFT Plot

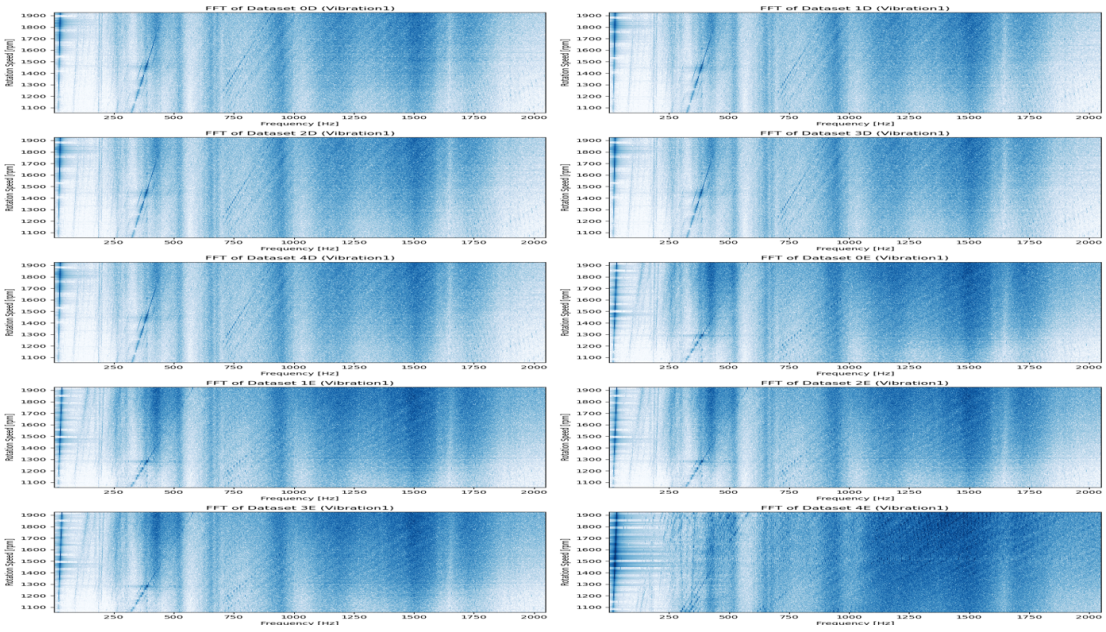

Figure 5.4: FFT Plots

The Fast Fourier Transform (FFT) spectrograms for the various datasets identified as 0D to 4D and 0E to 4E are displayed in Figure 5.4. These spectrograms represent the frequency content of vibration data gathered from a machine under various unbalanced

situations. The color intensity in each plot shows the vibration signal amplitude at each frequency and speed. The x-axis shows the frequency in Hertz (Hz), the y-axis the rotational speed in revolutions per minute (rpm).

In the development datasets (0D to 4D), 0D denotes an unbalanced machine in operation. Smooth operation is shown by the spectrogram for 0D, which has a comparatively consistent frequency content and no noticeable peaks. The spectrograms show that the degree of unbalance rises on going from 1D to 4D. As the imbalance increases, the machine's natural frequencies are aroused, which leads to increasingly noticeable and strong frequency peaks, especially at lower frequencies. The more intense colors in the spectrograms at higher imbalance levels (3D and 4D) indicate more complex patterns and larger amplitude vibrations.

The development datasets and the evaluation datasets (0E to 4E) exhibit comparable patterns, demonstrating the effectiveness of the methodology. The datasets' spectrograms are similar to those of their development counterparts: 0E and 0D exhibit homogeneous frequency content, while 1E and 4E display progressively more complicated and intense vibration patterns with greater imbalance.

The occurrence and severity of frequency peaks are, in essence, the main distinctions between the unhealthy (1D to 4D, 1E to 4E) and healthy (0D, 0E) states. In contrast to unhealthy states, which display complex, irregular patterns with distinct peaks and higher amplitudes, healthy states display simpler, more uniform patterns. This analysis provides important insights for predictive maintenance and early fault diagnosis by demonstrating how unbalance affects machine vibrations. By identifying these vibration patterns, operational efficiency can be improved and machine breakdowns can be avoided promptly.

Now let us take a close look at the FFT plots of 0E (No unbalance) and 4E (Highest unbalance) datasets.

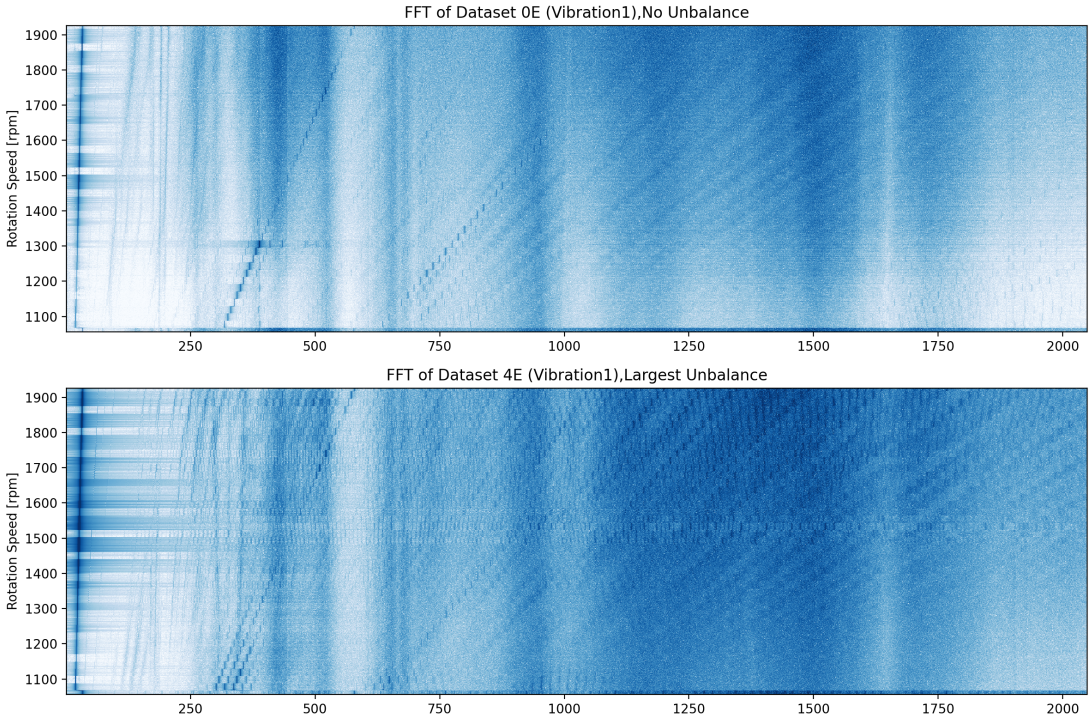

Figure 5.5: FFT Plot of 0E and 4E

The vibration data from two datasets—Dataset 0E, which represents a balanced condition with no unbalance, and Dataset 4E, which represents the state with the greatest unbalance—are shown in the Fast Fourier Transform (FFT) charts that are provided. The horizontal axis in both figures shows the frequency of vibrations in Hertz (Hz), which ranges from 0 to 2000 Hz, while the vertical axis shows the machine's rotation speed in revolutions per minute (rpm), which ranges from 1100 to 1900 rpm. The color intensity in the plots indicates the amplitude of the vibration signal; a darker shade indicates a larger amplitude.

The frequency distribution of the Dataset 0E (No Unbalance) plot looks pretty uniform with fewer prominent peaks, indicating a balanced system with fewer vibration amplitudes. This reduced amplitude indicates minimal vibrations and steady, effective performance.

On the other hand, the scenario depicted in the plot of Dataset 4E (Largest Unbalance) is very different. The darker shades on this plot represent larger amplitudes, and there are more noticeable peaks at particular frequencies. This implies that the system is resonating at specific frequencies due to the imbalance, producing louder vibrations. If

the imbalance is not corrected, it might result in mechanical wear, more maintenance requirements, and possibly catastrophic failure. The increasing vibration amplitude is a clear indicator of the imbalance.

5.3.3 Time-Domain Plots measured at 1000 RPM

The below plots in Figure 5.6 display vibration signals from two different datasets recorded at approximately 1000 RPM, focusing on scenarios with no unbalance (Dataset 0E) and the largest unbalance (Dataset 4E). These are time-domain plots showing the amplitude of vibrations over a specific period.

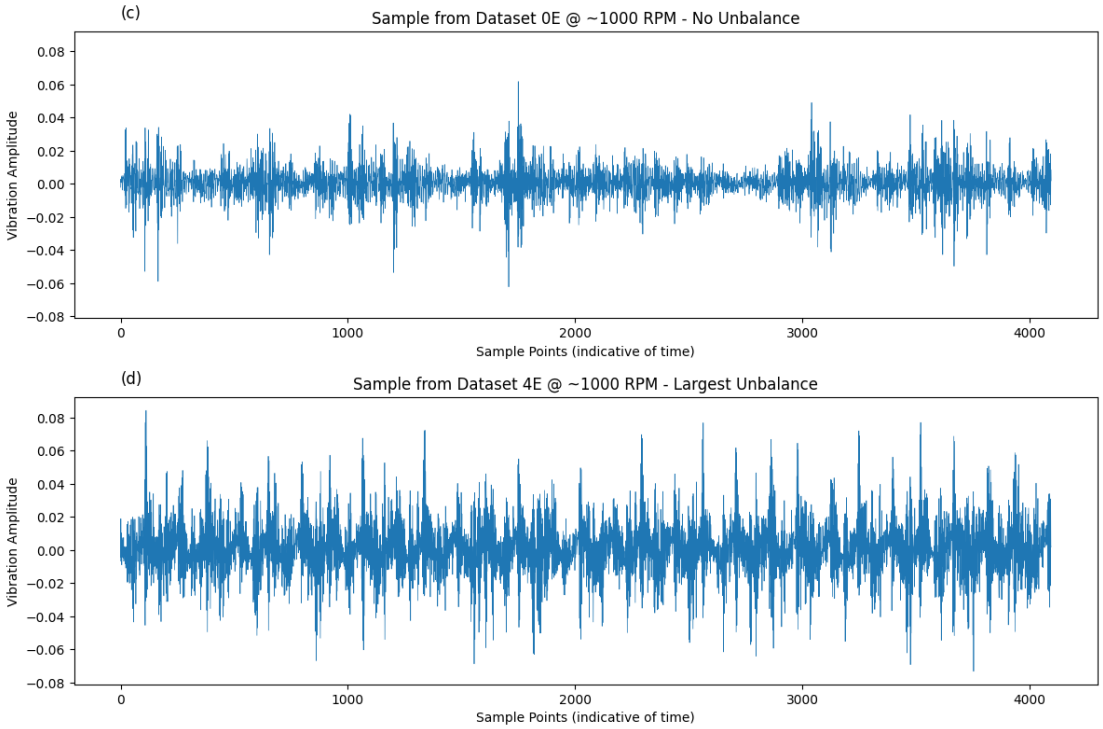

Figure 5.6: Plots of dataset recorded by vibration sensor 1 under no unbalance and highest unbalance at 1000 rpm

Dataset 0E (No Unbalance): The first plot represents the vibration signal from Dataset 0E, where the system was operating at 1000 RPM without any unbalance. The horizontal axis, labeled "Sample Points (indicative of time)," spans 4096 sample points, indicative of the time over which the data was collected. The vertical axis shows the "Vibration Amplitude," ranging between -0.08 and 0.08. In this plot, the vibration signal is relatively stable with lower amplitudes, indicating a balanced system with minimal vibrations. The signal appears as a dense cluster of small oscillations around the zero

line, signifying that the machine operates smoothly with no significant disturbances. The low amplitude of vibrations demonstrates that the system is in a stable state with minimal mechanical stress and wear.

Dataset 4E (Largest Unbalance): The second plot displays the vibration signal from Dataset 4E, recorded under the condition of the largest unbalance at a rotational speed of 1000 RPM. The horizontal axis represents sample points over time, similar to the first plot, and the vertical axis shows vibration amplitude, also ranging from -0.08 to 0.08. In this plot, the vibration signal is characterized by higher amplitudes and more pronounced oscillations compared to the first plot. The increased amplitude and variability of the signal indicate significant unbalance in the system, leading to stronger vibrations. These strong vibrations suggest that the system is experiencing mechanical stress and instability, which could result in increased wear and potential failure if not addressed.

Major Differences: The primary difference between the two datasets is the amplitude of the vibrations. Dataset 4E exhibits significantly higher vibration amplitudes compared to Dataset 0E, highlighting the effect of unbalance. The balanced system (Dataset 0E) displays a more stable vibration profile with lower overall amplitudes, while the unbalanced system (Dataset 4E) shows instability with higher and more variable amplitudes. These differences are visually represented by the varying color intensities in the plots, where darker shades indicate higher vibration amplitudes.

Amplitude and Color Intensity: The color intensity in the plots represents the amplitude of the vibration signal at different frequencies and rotation speeds. Darker shades indicate higher amplitudes, signifying stronger vibrations at those frequencies. This visual representation helps in identifying the frequency components where the system experiences the most significant vibrations, which are crucial for diagnosing potential issues and understanding the impact of mechanical imbalances.

Conclusion: These time-domain plots effectively illustrate the impact of unbalance on the vibration characteristics of a rotating machine. The balanced system operates with minimal vibrations, ensuring stability and longevity, while the unbalanced system exhibits high vibrations at certain frequencies, indicating the need for corrective measures to prevent mechanical failure and ensure efficient operation. The provided datasets specifically highlight the vibration sensor 1 readings obtained during a full measurement, with a focus on the scenario of no unbalance at 1000 RPM and the highest unbalance at the same rotational speed. This comparison underscores the importance

of maintaining balance in rotating machinery to prevent mechanical wear and potential failure, making these plots valuable for diagnosing and addressing mechanical issues in rotating systems.

5.3.4 Time-Domain Vibration Signals: Comparative Analysis of No Unbalanced and Largest Unbalance

Figure 5.7 presents time-domain plots comparing vibration sensor data from two different datasets: Dataset 0E (with no unbalance) and Dataset 4E (with the largest unbalance). The y-axis of both plots represents the vibration amplitude measured by the sensor, while the x-axis represents the sample index, scaled to reflect a 1-second window based on the sampling frequency of 4096 Hz.

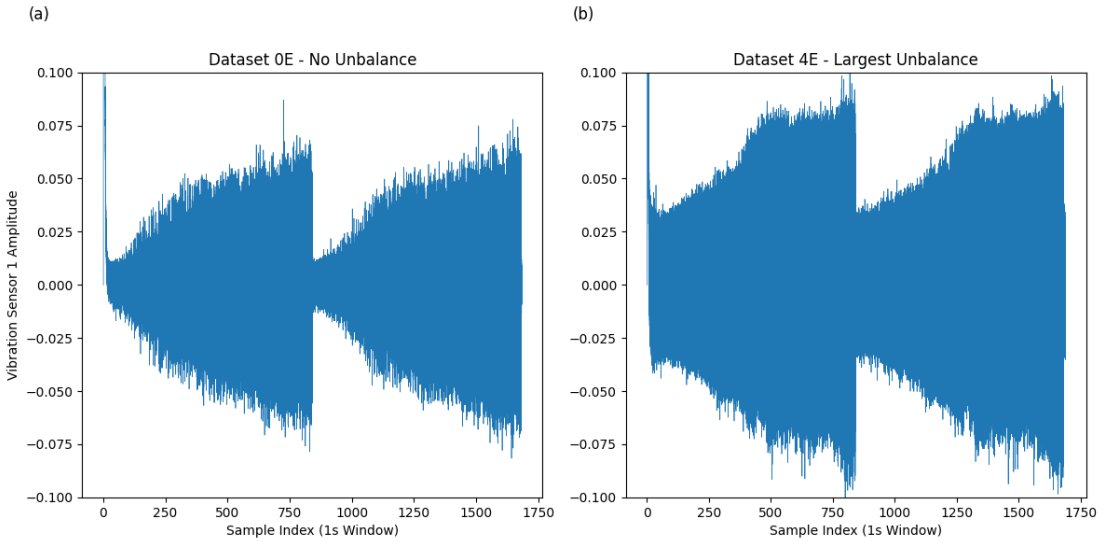

Figure 5.7: Time domain Plot under 1 sec window

Dataset 0E - No Unbalance

In the first subplot (labeled "(a)"), the data from Dataset 0E, which represents the condition of the machine with no unbalance at approximately 1000 RPM, shows relatively low and consistent vibration amplitude. The waveform appears stable, indicating that the machine operates smoothly without significant disruptions or spikes in vibration amplitude.

Dataset 4E - Largest Unbalance

The second subplot (labeled "(b)"), illustrates the vibration data from Dataset 4E, where

the machine has the largest unbalance at the same RPM. In this plot, the vibration amplitude is noticeably higher and more erratic compared to Dataset 0E. The presence of higher peaks and troughs indicates increased vibration and instability, likely due to the unbalance.

Analysis

The comparison between the two datasets demonstrates the impact of unbalance on the vibration characteristics of the machine. Dataset 0E's plot, with its lower and more uniform amplitude, signifies stable operation with minimal vibrational disturbances. Conversely, Dataset 4E shows significant unbalance, evidenced by higher and more variable vibration amplitudes, suggesting that unbalance introduces instability and higher vibration magnitudes in the machine's operation.

These time-domain plots are crucial for diagnosing machine conditions, where a steady and low-amplitude signal represents normal operation, and deviations from this norm indicate potential issues like unbalance, which can lead to further mechanical failures if not addressed.

5.3.5 Comparative Analysis of Healthy and Unhealthy Spectrograms

Figure 5.8 presents spectrogram comparisons of vibration sensor data from two distinct datasets: Dataset 0E, characterized by no unbalance, and Dataset 4E, featuring the largest observed unbalance. Each plot illustrates the intensity distribution over frequency (y-axis) and time (x-axis), providing insights into the vibration characteristics observed.

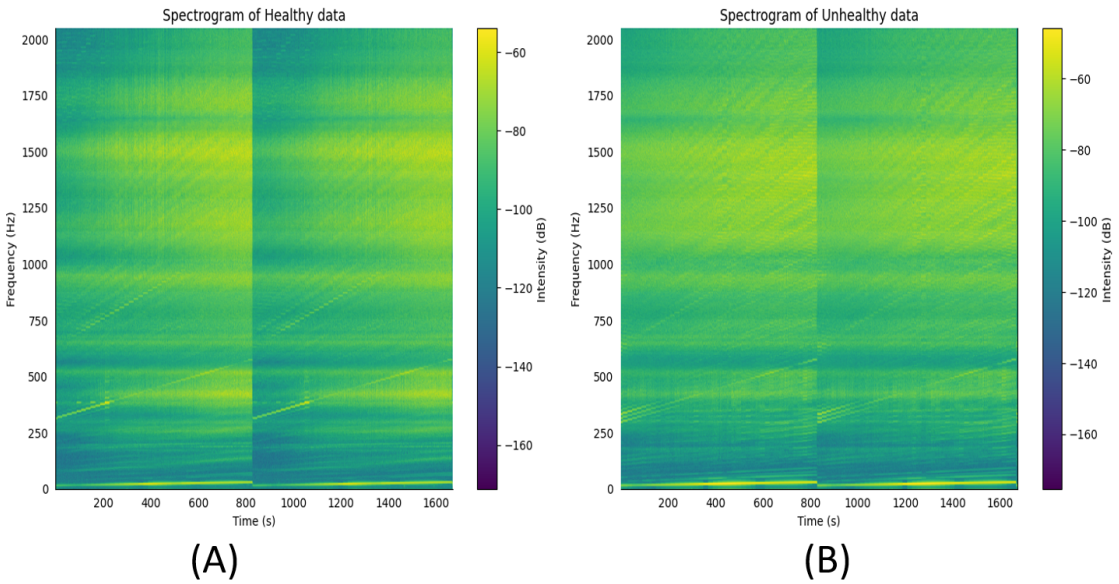

Figure 5.8: Spectograms of Healthy and Unhealthy data

In comparing the spectrograms of healthy and unhealthy vibration data, distinct differences emerge across frequency bands, intensity distribution, and temporal stability. The healthy data (Figure 5.8 (A)) reveals clear, stable frequency bands at around 250 Hz, 500 Hz, 750 Hz, and up to 2000 Hz, characterized by concentrated, distinct horizontal bands with consistent vibration patterns. Intensity levels range from -160 dB to -60 dB, showing uniformity and predictable higher intensity regions, indicating a stable vibration signature that repeats predictably over time, such as around 800 seconds.

Conversely, the spectrogram of unhealthy data (Figure 5.8 (B)) displays notable variability and less distinct horizontal bands in similar frequency ranges. Frequencies like 250 Hz, 500 Hz, and 750 Hz exhibit more fluctuations and less concentration, accompanied by scattered and irregular intensity levels ranging similarly from -160 dB to -60 dB. Higher-intensity regions appear less uniform and more dispersed, reflecting irregular vibration patterns. Temporally, the unhealthy data shows greater variability and periodic disturbances, indicating an unstable vibration signature prone to fluctuations in both intensity and frequency over time.

Overall, while intensity ranges are comparable, the unhealthy condition introduces additional harmonics, irregularities, and variability in frequency components above 1000 Hz. This complex and unstable vibration pattern suggests issues like unbalanced and irregular mechanical behavior, contrasting sharply with the stable patterns in healthy data. Spectrogram analysis is crucial for detecting abnormalities, aiding proactive maintenance, and ensuring optimal machinery performance.

5.3.6 Confusion Matrix

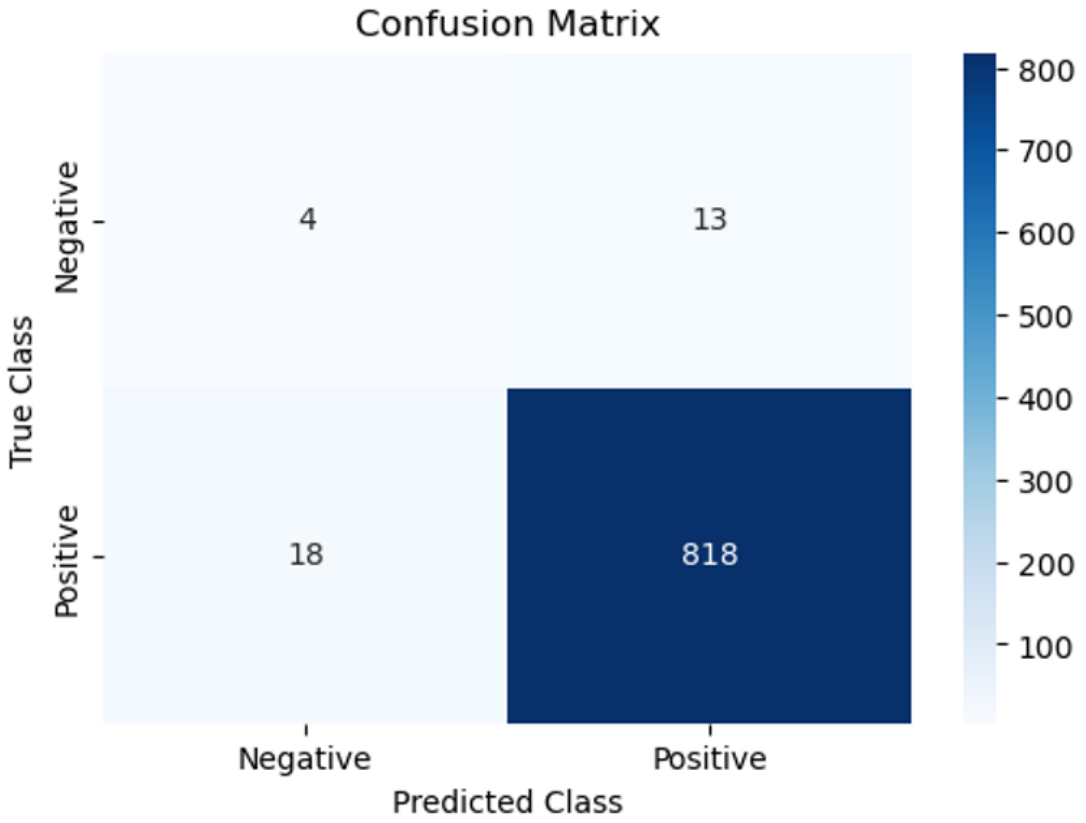

Figure 5.9: Confusion Matrix

The confusion matrix shown in Figure 5.9 evaluates the performance of a binary classification model, with the true class labels on the vertical axis and the predicted class labels on the horizontal axis. It consists of four key sections: true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP). The top-left cell (TN = 4) represents the instances where the model correctly predicted the negative class, while the top-right cell (FP = 13) shows the instances where the model incorrectly predicted the positive class instead of the negative class. The bottom-left cell (FN = 18) indicates the instances where the model incorrectly predicted the negative class when it was positive, and the bottom-right cell (TP = 818) shows the instances where the model correctly predicted the positive class. Thus, the matrix reveals that the model accurately identified 4 negative instances and 818 positive instances, but made errors by predicting 13 negative instances as positive and 18 positive instances as negative.

6. Results and Discussions

6.1 Fine Tuning

For this thesis, the BERT (uncased) model which is pre-trained and then fine-tuned has been used, according to our problem. BERT Architecture has already been explained in Chapter 4. It utilizes vibration data of the rotating shaft from Kaggle for data evaluation and fine-tuning. Preprocessing to the raw data has been applied shown in Chapter 5. To apply our method, Fast Fourier Transform (FFT) has been computed for each segment of the first vibration sensor stream, "Vibration 1", which consists of 4096 values or one-second windows. This process yielded 2048 Fourier coefficients per window, aligning with the Shannon-Nyquist sampling theorem's principles, ensuring relevance for classification purposes.

Next, the data is split into training, testing, and validation sets using an 80/10/10 ratio respectively. For fine-tuning the model, the following hyperparameters have been specified: Binary Cross Entropy Loss (BCE) is utilized as our loss function, which is ideal for binary classification tasks [8]. The Sigmoid Activation Function was employed to produce probabilities for binary classification outputs. Our learning rate was set to 5e-6, chosen to balance training stability and convergence speed. During training, a batch size of 8 is used, while for testing and validation, a batch size of 2 has been employed to efficiently evaluate model performance. Our model underwent training for 30 epochs using the AdamW optimizer algorithm [9], which is effective in optimizing neural networks by adjusting learning rates dynamically. Due to hardware constraints, specifically using an Nvidia T400 GPU with 4GB of memory, it is not possible to increase the batch size or extend the number of training epochs beyond 30. These limitations guided our decisions to ensure efficient use of computational resources while fine-tuning our model effectively.

The model has been evaluated on the validation set at the end of each epoch and uses

accuracy as the performance metric for the training process which was explained in Chapter 5. Figure 6.1 is a perfect illustration of this.

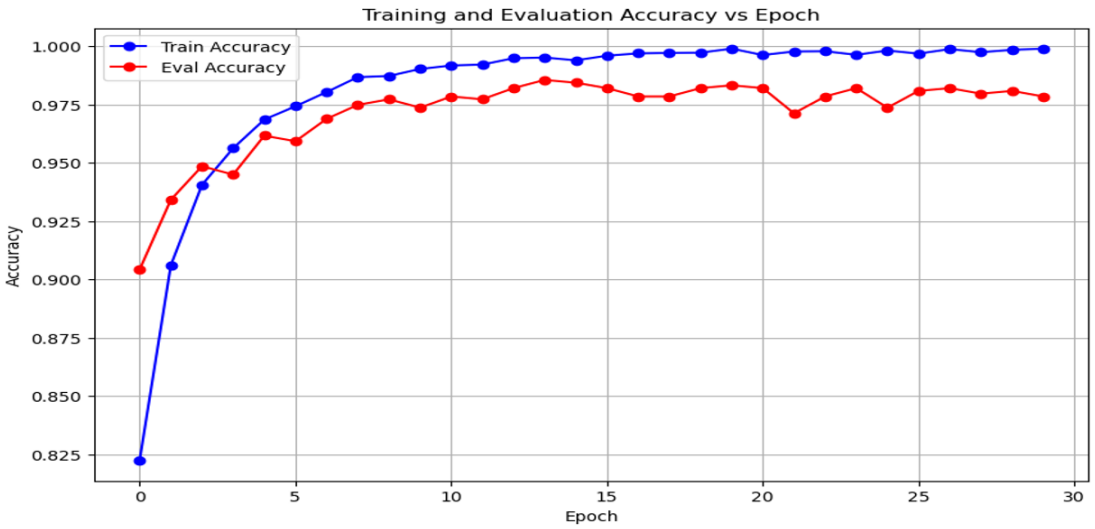

Figure 6.1: For both the Training and Validation Sets, accuracy vs the epoch curve

As expected, large improvement in performance in the first six epochs with an impressive increase in training accuracy is observed. Following the sixth epoch, there is a slight rising tendency toward the end of the accuracy, which stabilizes in the range of 97% to 99%. Figure 6.2, which shows the loss of the training and validation dataset throughout every epoch shows the model understands most during the late stages. The train set's loss is continuously and monotonously decreasing. In the first three to four epochs, there is the greatest drop in this instance, resulting in the largest performance improvement.

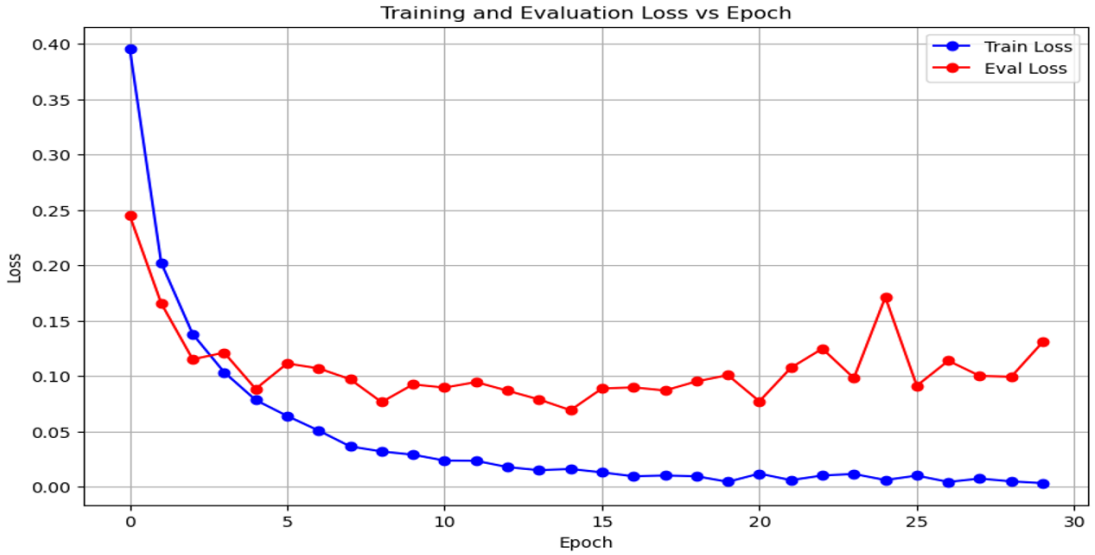

Figure 6.2: Epoch curve against loss for training and validation sets

However, both the train loss and accuracy curves show a consistent improvement in the model overall. On closer inspection, however, the evaluation loss curve shows that it continues to decline as the training loss does. After the 23rd epoch, the model's performance degraded as the loss increased, but after the 24th epoch, it reverted to its previous level. Overall, if, after a certain epoch, with the decrease in training loss if there is an increase in evaluation loss, it has been concluded that the model has started to overfit. However, in this case, no overfitting is observed up to 30 epochs.

6.2 Implementation

The BertTrainer class is designed to make the fine-tuning and evaluation of BERT-like models in PyTorch seamless and efficient. When an instance of BertTrainer is created, it includes several crucial components: a pre-trained BERT model, a tokenizer, data loaders for both training and evaluation, and key training parameters such as epochs, learning rate, batch size.

During the training phase, the process iterates through the specified number of epochs. For each epoch, batches of data from the training loader are processed. Using the tokenizer, text inputs are converted into token IDs, which are then passed through the model. The model's outputs, or logits, are used to compute the loss against the true labels using the binary cross-entropy loss function. If in training mode, the optimizer (AdamW) updates the model's weights based on the gradients computed during back-propagation. Throughout this process, performance metrics like loss, accuracy, precision, recall, and F1-score are tracked, providing insights into how well the model is learning and performing.

When evaluation is enabled, the model switches to evaluation mode and processes data from the evaluation loader. In this mode, the model's parameters remain fixed, allowing a focus solely on calculating evaluation metrics to assess how effectively the model generalizes to unseen data. These metrics are printed out after each epoch, giving a clear view of the model's progress and performance.

Furthermore, the BertTrainer includes functionality to save the model. By setting the save parameter to True, the model's state is saved whenever the evaluation loss for an epoch is lower than any previously recorded loss. This ensures that the best-performing

version of the model is always retained.

To use the BertTrainer, a pre-trained BERT model and tokenizer are loaded from the Hugging Face library. Training and evaluation data loaders are then prepared. Once the BertTrainer is set up with the chosen model, tokenizer, and training settings, the training process is started by calling the train method with evaluation enabled. This method handles both training the model on the data and evaluating its performance. This approach makes it easier for researchers and developers to fine-tune transformer models, simplifying the complex tasks involved in it.

After initializing the BertTrainer class with the BERT base model, tokenizer, and setting parameters like the number of epochs to 30 and learning rate to 5e-6 for training, it was used to train and evaluate the model on the Kaggle datasets. The trained model was saved to the path provided by the user. Subsequently, the weights of a previously fine-tuned model (best weight) were loaded into BERT base to continue working with those weights. For evaluation purposes, the trainer was re-initialized with the same model and tokenizer but with save set to False since there was no need to save the model again. The model was then evaluated on the test set to assess its ability to classify machine health based on vibration data. These steps illustrate a practical workflow for using the BertTrainer class outside traditional NLP tasks, demonstrating its adaptability to different classification challenges such as machine health monitoring through sensor data analysis.

6.3 Results

The final result coming out on the testing dataset is the accuracy of **98.44%**.

Performance Metrics	Training	Validation	Testing
Accuracy	99.9%	98.56%	98.44%
F1	0.9984	0.9778	0.9761
Recall	0.9985	0.9778	0.9792
Precision	0.9982	0.9778	0.973

Table 6.1: Final Result

6.4 Comparsion with Existing Work

Result from the research paper O. Mey et al., "Machine Learning-Based Unbalance Detection of a Rotating Shaft Using Vibration Data". [11]

Method Used	Accuracy Achieved
CNN	94% and 93.6% (With 2 and 4 convolution blocks respectively)
Random Forest	91.6%
FCNN	93.2%
HMM	95%

Table 6.2: Performance of Existing Paper

6.5 Conclusion

The primary objective of this thesis was to determine whether a machine is in a healthy or faulty condition, with the overarching aim of accurately classifying the health of machines. This work is significant in the context of predictive maintenance, a crucial aspect of industrial machinery management that can prevent costly repairs and minimize downtime. By leveraging deep learning techniques, specifically the BERT Transformer Model, this research sought to enhance the accuracy of machine health classification compared to existing methods.

In the research paper [11] which used the same dataset used in this thesis, the highest accuracy achieved in classifying machine health was 94% using CNN. This served as a benchmark for this work. Recognizing the potential of advanced deep learning models, the application of the BERT Transformer Model, traditionally pre-trained for natural language processing (NLP) tasks, was explored in the domain of vibration data analysis and machine health classification. This innovative approach represents a novel application of BERT, demonstrating its versatility beyond its typical use cases.

Experiments utilized an open-source dataset from Kaggle, which contained vibration data essential for training and testing the model. By fine-tuning the BERT model on this vibration data, a maximum accuracy of 98.6% in classifying machine health was achieved. This significant improvement over the original benchmark underscores the

efficacy of using advanced deep learning models for predictive maintenance.

Several factors contributed to this success. Firstly, the BERT model's ability to understand complex patterns and relationships in data played a critical role. Despite being designed for NLP tasks, BERT's architecture is highly adept at capturing intricate features in sequential data, analogous to the temporal patterns found in vibration signals. Training BERT on this data leveraged its powerful feature extraction capabilities to distinguish between healthy and faulty machine states with high precision.

Moreover, extensive hyperparameter tuning was conducted to optimize the BERT model's performance. This involved setting appropriate learning rates, determining the optimal number of epochs, and selecting suitable batch sizes. Fine-tuning these hyperparameters ensured that the model was trained effectively, achieving the best possible performance on the vibration data.

The architecture of the BERT model, with its attention mechanisms, allowed it to focus on the most relevant parts of the vibration signals, improving its decision-making process. The self-attention mechanism enabled the model to weigh the importance of different parts of the input data, leading to more accurate classifications.

The implications of these findings are substantial.Achieving an accuracy of 98.6% means that predictive maintenance systems can now more reliably identify potential faults before they lead to machine failures. This not only reduces maintenance costs but also extends the service life of machinery and prevents unexpected production interruptions. Industries that rely heavily on rotating machines, such as manufacturing, oil and gas, and power generation, stand to benefit significantly from these advancements.

In conclusion, this thesis has demonstrated the potential of applying the BERT Transformer Model to the field of predictive maintenance. By achieving a higher accuracy than previously reported methods, it has been shown that advanced deep learning models can be effectively adapted to new domains, offering significant performance improvements. The successful application of BERT to vibration data sets a new benchmark for machine health classification and opens up new avenues for research and development in predictive maintenance technologies. This work highlights the transformative impact that cutting-edge deep learning techniques can have on industrial applications, opening the way for more reliable and efficient maintenance strategies in the future.

REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, PMLR, May 2013.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*, 2018. https://arxiv.org/abs/1810.04805.

[5] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What Does BERT Look At? An Analysis of BERT's Attention. *arXiv:1906.04341*, 2019. https://arxiv.org/abs/1906.04341.

[6] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv:1506.06724*, 2015. https://arxiv.org/abs/1506.06724.

[7] Mohammad Taher Pilevar and José Camacho-Collados. Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning. *Synthesis Lectures on Human Language Technologies*, 2020.

[8] Qi Li, Xiaoxuan Jia, Jie Zhou, Li Shen, and Jinyang Duan. Rediscovering BCE Loss for Uniform Classification. *arXiv:2403.07289*, 2024. https://arxiv.org/abs/2403.07289.

[9] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv:1711.05101*, 2017. https://arxiv.org/abs/1711.05101.

[10] G. Öcalan and İ. Türkoğlu. Fault diagnosis of rotating machines using raw vibration signals and deep learning. In *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–7, Elazig, Turkey, 2021. 10.1109/ASYU52992.2021.9599062.

[11] O. Mey, W. Neudeck, A. Schneider, and O. Enge-Rosenblatt. Machine learning-based unbalance detection of a rotating shaft using vibration data. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1610–1617, Vienna, Austria, 2020. 10.1109/ETFA46521.2020.9212000.

[12] J.-H. Han, D.-J. Choi, S.-K. Hong, and H.-S. Kim. Motor fault diagnosis using CNN based deep learning algorithm considering motor rotating speed. In *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 440–445, Tokyo, Japan, 2019. 10.1109/IEA.2019.8714900.

[13] H. Matthew, A. D. Ayu, I. H. Suherman, A. Subiantoro, and B. Kusumoputro. Deep learning neural networks diagnosis of power transformer through its DGA data. In *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 292–296, Jakarta, Indonesia, 2022. 10.23919/EECSI56542.2022.9946518.

[14] X. Cheng, M. Zhao, J. Zhang, J. Wang, X. Pan, and X. Liu. TransNILM: A transformer-based deep learning model for non-intrusive load monitoring. In *2022 International Conference on High Performance Big Data and Intelligent Systems (HDIS)*, pages 13–20, Tianjin, China, 2022. 10.1109/HDIS56859.2022.9991439.

[15] P. Shan, P. Hou, H. Ge, L. Yu, Y. Li, and L. Gu. Image feature-based for bearing health monitoring with deep-learning method. In *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, pages 1–6, Qingdao, China, 2019. 10.1109/PHM-Qingdao46334.2019.8942972.

[16] R. Fan. Transformer-based deep learning method for the prediction of ventilator pressure. In *2022 IEEE 2nd International Conference on Information Communication and Software Engineering (ICICSE)*, pages 25–28, Chongqing, China, 2022. 10.1109/ICICSE55337.2022.9828926.

[17] Y. L. Karnavas, S. Plakias, and I. D. Chasiotis. A multi-scale deep learning attention-based feature method for rolling elements bearing fault detection in industrial motor drives. In *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–4, Thessaloniki, Greece, 2021. 10.1109/MOCAST52088.2021.9493397.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2019.

[19] Chuang Chen, Z. H. Zhu, Jiantao Shi, Ningyun Lu, and Bin Jiang. Dynamic predictive maintenance scheduling using deep learning ensemble for system health prognostics. *IEEE Sensors Journal*, PP:1–1, October 2021. 10.1109/JSEN.2021.3119553.

[20] K. Gnana Sheela and Anu Rose Varghese. Machine Learning based Health Monitoring System. *Materials Today: Proceedings*, vol. 24, pp. 1788–1794, 2020. 10.1016/j.matpr.2020.03.603.

[21] Bharati Ainapure, Prajakta Patrikar, Aanchal Mehta, Aniket Jain, and Sanika Shah. Machine Learning based Intelligent Health Monitoring System. In *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, pages 183–188, 2022. 10.1109/ICIEM54221.2022.9853117.

[22] Abdul Sttar Ismail and Falah Amer Abdulazeez. Leveraging Machine Learning Algorithms for Predictive Maintenance in Internet of Things (IoT) Systems. In *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART), pages 271-274, 2023.* 10.1109/SMART59791.2023.10428136

[23] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," in IEEE Transactions on Industrial Electronics, vol. 65, no. 2, pp. 1539-1548, Feb. 2018, doi: 10.1109/TIE.2017.2733438.

[24] O. Janssens, R. Van de Walle, M. Loccufier and S. Van Hoecke, "Deep Learning for Infrared Thermal Image Based Machine Health Monitoring," in IEEE/ASME Transactions on Mechatronics, vol. 23, no. 1, pp. 151-159, Feb. 2018, doi: 10.1109/TMECH.2017.2722479