APPLICABILITY OF NEURAL NETWORKS TO ROTOR MISALIGNMENT DETECTION

A Dissertation
Presented to
The Academic Faculty

by

Tara Chan

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
May 2022

COPYRIGHT © 2022 BY TARA CHAN

APPLICABILITY OF NEURAL NETWORKS TO ROTOR MISALIGNMENT DETECTION

Approved by:

Dr. Christopher Saldana, Advisor
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Katherine Fu
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. Thomas Kurfess
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 28, 2022

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Saldana, for helping me navigate through my master's journey and providing valuable insights on my thesis and related projects. By familiarizing me with the world of academia, you have expanded my way of thinking which will help me overcome future endeavors. I would also like to express gratitude to Dr. Fu and Dr. Kurfess for serving on my committee and for providing helpful advice and resources. I would also like to thank my friends and EPICS labmates, particularly Pierrick, for answering my many questions and creating unforgettable memories during my time here. I would like to give special thanks to my wonderful family for supporting me from the very beginning and always encouraging me. And finally, I would like to thank my boyfriend, Martin, for not only providing me computing power but also for being my rock throughout this process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	Artificial Neural Network
ANOVA	ANalysis Of VAriance
CBM	Condition Based Maintenance
CNN	Convolutional Neural Network
FFT	Fast Fourier Transform
MAE	Mean Absolute Error
MSE	Mean Squared Error
PCA	Principal Component Analysis
RBF	Radial Basis Function
RMS	Root Mean Square
SHAP	SHapley Additive exPlanation
SMOTE	Synthetic Minority Oversampling TEchnique
SNR	Signal to Noise Ratio
SVM	Support Vector Machine
TBM	Time Based Maintenance
UA	Underhung Axial Axis
UR	Underhung Radial Axis
UT	Underhung Tangential Axis

SUMMARY

Planning and executing efficient maintenance in a large production environment is a necessary but convoluted endeavor. In rotor systems, the ability to quantify and classify misalignment would reduce effort in diagnosis and planning. To aid in this process, the thesis investigates the applicability of three machine learning algorithms to bearing accelerometer time domain data in their ability to preemptively identify and quantify misalignment, a vibration fault responsible for over 70% of issues in rotating equipment. The performances of artificial neural networks (ANN), convolutional neural networks (CNN) and support vector machines (SVM) were compared on increasing misalignment levels from the MaFaulDa database, and their performance was evaluated with the introduction of noise and low sampling rates to determine their transferability into a realistic environment. For classification of misalignment type, the CNN model outperformed the other models with a 95.1% accuracy and 93.9% F1-score. In quantifying the severity of misalignment, the ANN model was superior with a 16.8% MAE and 71.4% F1-score. Furthermore, it was determined that these models are robust against noise; however, sampling frequencies below 25 kHz significantly reduce performance up to 20%. The deployment of these models requires minimal knowledge of the machine and will equip users with a tool to determine precisely when to fix equipment as to minimize time spent in maintenance.

CHAPTER 1. INTRODUCTION

1.1 Motivation

As the manufacturing industry rapidly expands, the quantity and complexity of machines continues to evolve. The consequences of equipment breakdown can be catastrophic, not only from the perspective of operator safety but also from a cost and time standpoint as shown by a case study on an unsuccessful single two-part toothed rim production process that resulted in repair costs amounting to PLN 17000 (~4000 USD) and a seventeen-day setback [1]. Although much time is spent on the design of the equipment, machines continuously experience loading and breakdown is inevitable. For companies that must maintain many such machines, an efficient process is necessary to minimize delays and optimize profits. Addressing the machine breakdowns as they occur is referred to Corrective Maintenance which, as demonstrated by [1] can often lead to large losses for the company. Its counterpart, known as Preventative Maintenance, is a strategy that allows for more frequent maintenance meant to impede catastrophic machine failure. Under Preventative Maintenance exist two popular strategies, Time Based Maintenance (TBM) and Condition Based Maintenance (CBM). TBM planning utilizes probabilities of failure based on the bathtub model to determine set time intervals of maintenance. When using CBM, machine behavior is continuously monitored, and maintenance is performed based on potential signs of failure. In practice, CBM is the preferred method as the bathtub curve assumption used in TBM reliability models is unrealistic and unpredictables breakdowns may occur due to lack of sufficient data [2]. CBM however, is more likely to inhibit catastrophic failure as certain telltale indicators predate 99% of machine breakdowns [3].

A drawback of CBM is that it can be expensive as thorough examination requires immense continuous data collection and processing of machine health conditions. Additionally, if a machine fault is caught late, there may be insufficient time to adequately plan and perform maintenance. But often, the equipment downtime for precautionary maintenance is considerably less than repairing a sudden failure which, due to its unexpected nature, can grow into larger complications. This has led to much research done in implementing CBM and attempting to predict when machine faults will occur to preemptively address potential issues. However, an excess of proactive maintenance can also add to profit loss, encouraging the implementation of Just in Time (JIT) philosophy to reduce maintenance waste [4]. The ability to accurately identify the type and severity of a machine fault is valuable in determining exactly when to alert operators of a fault. And even greater details about a potential fault better advises decisions on maintenance scheduling and planning which can significantly cut down on maintenance waste [5]. This holds true particularly for equipment with stochastic repair time as demonstrated by a production-inventory model created by Widyadana and Wee which foretells that the cost difference between stochastic and fixed repair times is very high [6].

There are infinitely different operations that occur in a manufacturing facility, and many of them can be performed with the help of rotating shafts. Rotor systems are commonly used in manufacturing processes, from spinning rolls of paper to transporting products along an assembly line belt. Thus, being able to efficiently perform CBM on rotor systems can significantly increase profit margins. The second most common vibration fault after imbalance, responsible for around 70% of issues in rotor systems, is misalignment [7, 8]. This type of vibration fault is nearly always present as perfect shaft coupling alignment is practically impossible to achieve, so it is critical to determine if there is severe enough misalignment to warrant maintenance. Furthermore, a better understanding of the misalignment fault will enable operators to plan for the intensity and duration of required maintenance. Thus, this thesis focuses on developing a model to assist operators in planning out CBM by better informing them about the nature of potential issues specifically targeting rotor misalignment faults.

1.2 Problem Statement

Methods to identify the presence of misalignment have been heavily investigated; however, there is much less research done in methods to support quantification of the amount and type of rotor misalignment that would assist in performing more cost-efficient maintenance. This work aims to develop a model to understand the degree to which supervised machine learning approaches can determine the amount and type of misalignment based on accelerometer data. Three machine learning algorithms were evaluated: artificial neural networks (ANN), convolutional neural networks (CNN), and support vector machines (SVM). The effectiveness of the created models was judged by the following research questions:

1. What is the ability of the models to characterize the nature of misalignment?
    a. What minimum combination of sensor axes is necessary to generate useful results?
2. How robust would the models be in various real-world environments?

Using a dataset of various misalignment levels and types, a study of the relevant accelerometer measurements was performed with baseline models. The models were then optimized and compared for both classification of misalignment type and regression to predict misalignment amount. The performance of the fully developed models was tested with various levels of noise and sampling frequencies. The benefits of the models are discussed based on overall performance and practical use considerations.

1.3 Thesis Structure

The following sections elaborate on the considerations and results of this work. Chapter 2 reviews background and existing work done in the area of misalignment detection. Chapter 3 presents methodology and the dataset being used in this study. Chapter 4 explains the analysis performed on the misalignment data and results from the models. Discussion of the model performance is covered in Chapter 5. And finally in Chapter 6, the study is summarized in the conclusion.

CHAPTER 2. BACKGROUND

2.1 Misalignment

Rotor misalignment is a fault that occurs at the coupling between two shafts that are not co-axial. There are three high level types of misalignment: parallel, angular, and combined parallel and angular misalignment as shown in Figure 1A. Parallel misalignment occurs when the shaft centerlines are offset from each other. Angular misalignment is when the centerlines are at an angle to each other (not parallel). And within those types of misalignments, one can have vertical or horizontal misalignment in which the shafts shift in the y or x axes respectively, given that the z-axis is defined as along the length of the shaft (Figure 1B).

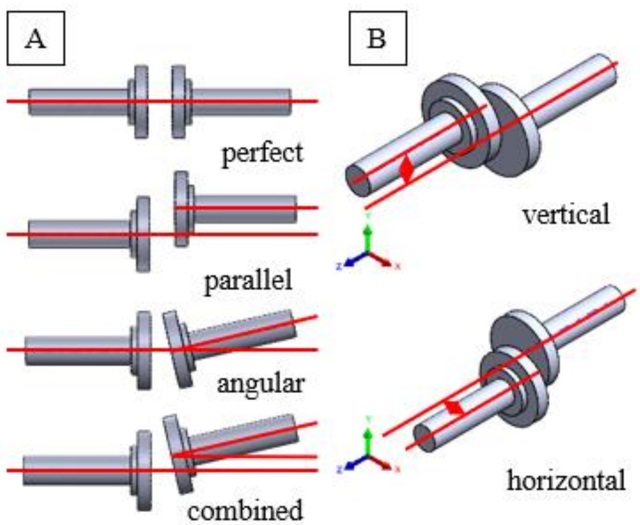

Figure 1. Types of shaft misalignment (A) perfect, parallel, angular, and combined and (B) vertical vs horizontal parallel misalignment

Vibration analysts can often recognize the presence of misalignment by examining the time and frequency domain of accelerometer data. By looking at the Fast Fourier

Transforms (FFT) of waveforms generated by the accelerometers, the presence of harmonics is a good indicator that there may be misalignment present. The first peak is always the shaft frequency, but the relative height of the harmonics varies based on the type of misalignment as discovered by Dewell and Mitchel [9]. For instance, in the case of parallel misalignment, it is expected that that second harmonic has the highest peak while for angular misalignment, the first and second harmonic are dominant as shown in Figure 2A. The second key indicator of misalignment is the presence of "M" and "W" shapes visible in the 2.0 mm misaligned time series waveform as compared to a nominal (0 mm) case in Figure 2B.

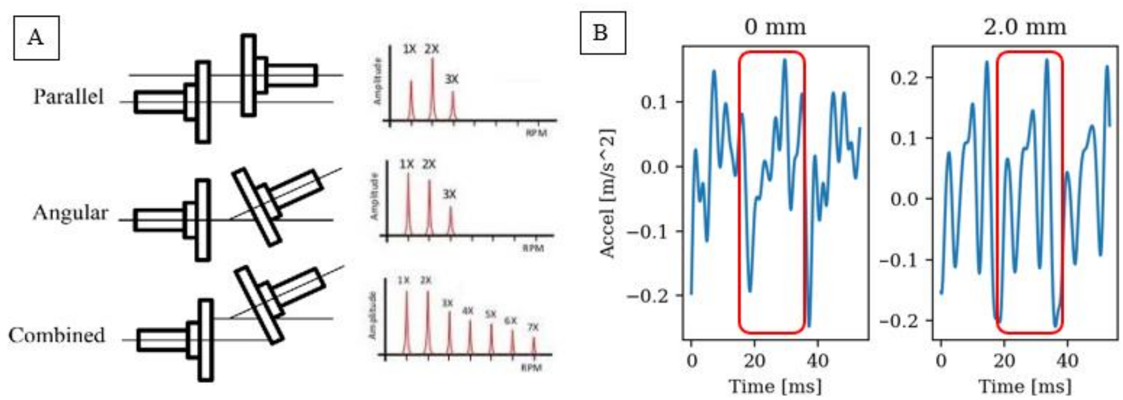

Figure 2. (A) Harmonics of different types of misalignment and (B) 0 mm waveform vs M-shape in a 2 mm misaligned system

2.2 Existing Work

In the field of vibration, there has been a great deal of focus on trying to better understand and identify various vibration faults including misalignment. And while the main goals of the identification methods differ, there is a similar set of tools used for all vibration analysis. For instance, without any specific fault type in mind, Kalkat et al. [10]

successfully utilized an ANN to determine the vibration parameters of a rotor system, demonstrating that machine learning models could learn its behavior. Below details similar tools used for the purpose of identifying and characterizing misalignment.

2.2.1 Theoretical Models and Understanding Misalignment

The mechanics of misaligned shafts have been examined by Dewell and Mitchell [9] and building on their work, Xu and Marangoni [11] have generated theoretical models to predict the vibration response of misaligned and imbalanced rotor systems. From these models, it was discovered that systems containing misalignment were distinguishable via peak harmonics at 2X and 4X the shaft rotation frequency and that it was common to observe additional peaks at even multiples [12]. This was further confirmed in another theoretical model for a dual motor system by Wang and Jiang [13] who also found that the peaks increase in amplitude with more angular and parallel misalignment. However, it has been established from the theoretical models [12] and an ANSYS parallel misalignment simulation by Hariharan and Srinivasan [14] that occasionally the 2X peak may not be visible if that harmonic is not within close range the system's natural frequencies. Even still, the frequency domain has been used in multiple misalignment identification methods with proven success.

In an effort to more comprehensively capture the behaviour of misalignment, data from multiple axes have been combined and inspected. Orbital plots of the FFTs from multiple axes have been studied by Patel and Darpe [15]. Their findings demonstrated that 1X and 3X peak harmonics are common in angular misalignment and that backwards whirling is an additional indicator of misalignment. A dynamic model using multiple axes created by Lee and Lee [16] generated orbit plots that inform on how bearing stiffness and increasing misalignment are positively correlated. Sinha and Elbhbah [17] also combined the waveforms of multiple axes into a composite bispectrum (calculated using a double Fourier Transform) and visually identified differences between the bispectrums of healthy and misaligned systems. These findings demonstrate that the frequency spectrum contains useful, distinguishable features to characterize misalignment and that multiple axes may provide key information. In an application, Wu and Chung [18] used a hybrid EEMD and EMD method which employs the Hilbert-Huang Transformation (another frequency spectrum transformation that can capture non-stationary signals) to determine relative increased shaft misalignment based on increased shaft amplitude modulation.

However, the frequency domain is a simplification of the vibration data and there is potential loss of critical features. Without using the frequency spectrum, a mathematical model of imbalance and misalignment was created by Jalan and Mohanty [19] to predict the resulting forces on the rotor system. From this, the location and severity of the fault could be detected based on how the experimental data lines up with the analytical results. Simply using time domain is another approach which has been vetted by Tahir et al. [20] in distinguishing between imbalance and misalignment faults. Although there is likely to be excessive data when using the time domain, it decreases the risk of losing of potentially useful information. Extracting time domain features can also reduce the concern of using counterproductive data. Additionally, vibration analysts in industry look for the "M" and "W" shapes which are only visible in the time domain. With a deeper understanding of misalignment and ways to represent it, a closer look at the using various machine learning models to characterize misalignment can be taken.

2.2.2 SVM Methods

SVMs are a machine learning method commonly used for classification. The backing principle is that for features in space, there should be a way to divide the features into their corresponding classes. To demonstrate this in a simple linearly separable 2D representation in Figure 3A, the algorithm attempts to draw a line that will maximize the distance between the features and the dividing line. This idea can be extended to multiple dimensions and even nonlinear separations using kernels (Figure 3B) [21].

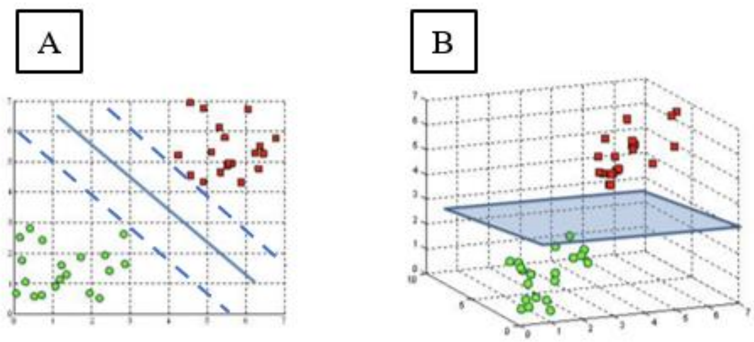

Figure 3. SVM visualizations for (A) 2D space showing the dividing line and margins and (B) 3D space [21]

This tool has been used extensively in vibration fault diagnosis [22]. For example, an SVM created by Tahir et al. [20] can distinguish between imbalance and misalignment using time domain features. Energy entropy has been used as a feature in an SVM that detected the presence of misalignment in wind turbines [23]. Lee [24] applied an SVM to determine the presence of misalignment with an accuracy of 98.8% after using principle component analysis (PCA) on the raw data. A benefit of this method is that it does not require a large amount of data to train on and works exceedingly well for target classes with no feature overlap.

2.2.3 ANN Methods

The Artificial Neural Network is the most basic of neural network architectures. A neural network is a machine learning algorithm that functions akin to a human brain. Without diving too deep into specifics, an ANN is made up of layers of connected nodes that make decisions in the output layer based on the input layer shown in Figure 4. The benefit of an ANN is that it can model nonlinear situations because all the nodes are interconnected, and it has been proven to quickly learn how to approach a given problem [25].

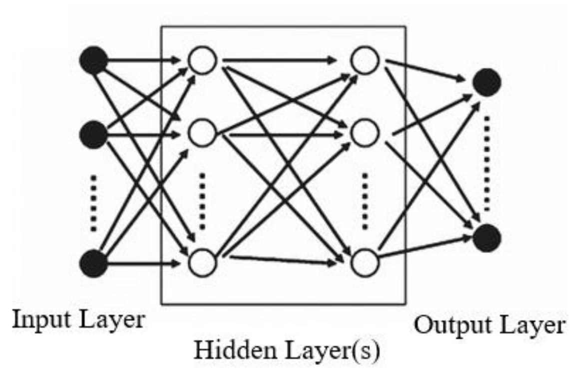

Figure 4. Structure of an ANN [25]

The use of ANNs for bearing faults detection has been validated [26]. For misalignment, Kuropatwinski et al. [27] attempted to implement an ANN to identify the amount of misalignment based on power spectral densities, but they discovered that it was not sufficient alone. Yet, ANN and SVM methods have been compared by Jack and Nandi [28] and found that both could identify the presence of a vibration with 100% accuracy but noted that the ANN was faster to train and implement than the SVM. A successful implementation by Umbrajkaar et al. [29] fed wavelet transformations in both ANN and SVM models to determine the amount and type of misalignment for both parallel and angular misalignment with an average error of 2.28%. Saridakis et al. [30] used bearing properties such as eccentricity, attitude angle, and minimum film thickness in an ANN to predict values of misalignment amount and bearing wear. It seems that ANN models are the preferred method in finding severity of misalignment, though they have also exhibited success in classifying different vibration faults, namely imbalance, misalignment, and nominal cases [31, 32].

2.2.4 CNN Methods

Convolutional Neural Networks are yet another type of neural network, although their main use is in classifying images. The same basic principle of layers of connected nodes applies, but the architecture is more involved. In addition to the layers from the ANN, a CNN has key convolutional layers as seen in Figure 5 that allows the neural network to look at smaller sections of the input. This is beneficial because the model is now able to identify features within an isolated section and take into consideration spatial features [33].

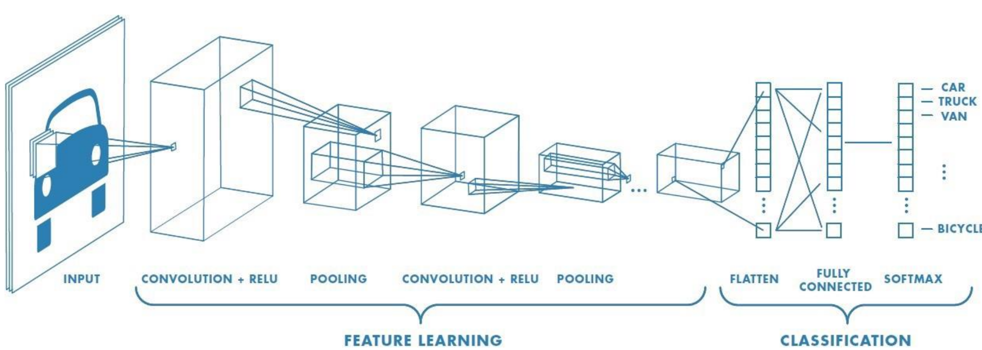

Figure 5. CNN general architecture [34]

Although waveforms are not typically thought of as images, CNNs have been widely used, covering even radar identification and demonstrating that they are robust against noise [35, 36]. In the field of vibration, it is often used to classify fault types. Zhao et al. [37] utilized a CNN on features from vibration signals decomposed by variable mode decomposition and PCA to classify misalignment and crack faults and also finds that CNNs are not easily susceptible to noise. Souza [38] created a CNN model for fault classification with 99.6% accuracy, though it is a rather complex model taking almost two hours alone to generate. Continuous wavelet scalograms and dot patterns have also been used as the input for a CNN to classify vibration faults [39, 40].

2.3 Research Gap

It is evident that misalignment characterization has been thoroughly investigated and there exists promising results from the above studies. However, the majority of these models only serve to identify the presence of misalignment or to differentiate misalignment from other vibration faults. Approaches to quantify the severity of misalignment are relatively less researched and most of the existing models only use ANNs. Furthermore, none closely examine the capabilities of the time domain. Additionally, most of the existing models require either a large amount of knowledge about the machine or extensive expertise about machine learning to implement them. This limits their application to only a few machines, and in a vast manufacturing facility, a more generalizable model would be desirable. Thus, this study investigates the applicability of the above models to a more straightforward method to characterize misalignment using the time domain.

CHAPTER 3. METHODOLOGY

To characterize the nature of misalignment, different levels of parallel and horizontal misalignment are analysed from the MaFaulDa database [41].

3.1 MaFaulDa Database

The MaFaulDa database contains a complete set of vibration data for seven vibration faults, though this paper will only be examining the nominal, horizontal misalignment, and vertical misalignment cases. The data was collected using a rigid coupling on the experimental apparatus below known as the Machinery Fault Simulator from SpectraQuest which allows for fine adjustment of induced vibration faults [42].

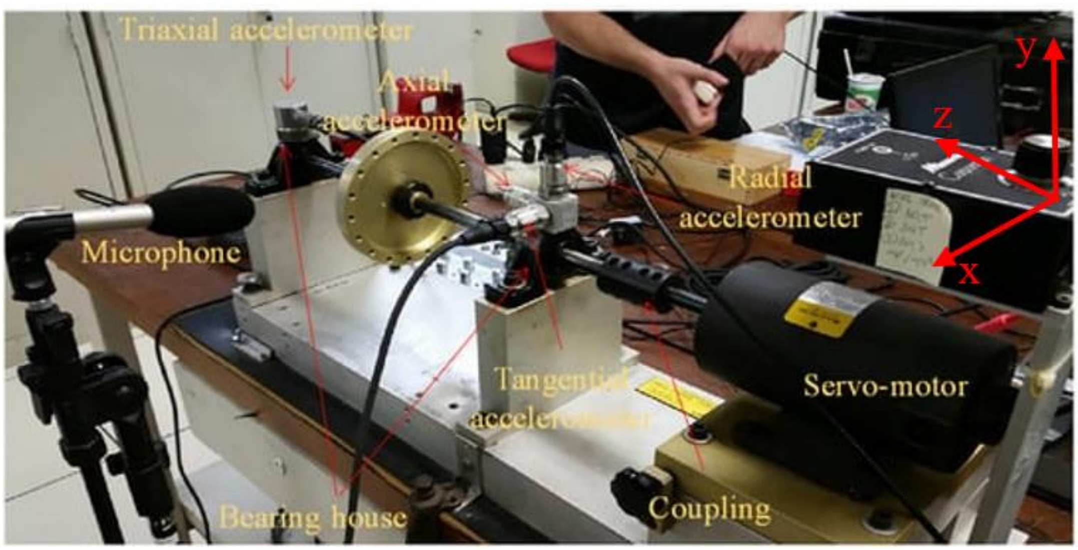

Figure 6. MaFaulDa data experimental set up [42]

Accelerometers mounted to the two bearings stands surrounding the rotor provided data in the three orthogonal directions. The data from the bearing stand between the motor and the rotor is referred to as the “underhung” bearing and the bearing stand on the outside of the rotor is “overhung”. Acceleration was measured in the x, y, and z directions as labeled in Figure 6. These directions will be defined as the tangential, radial, and axial directions, respectively. All the data was collected with a duration of 5 seconds and sampling frequency of 50 kHz. The shaft rotation frequency was also varied from 737 to 3686 rpm with intervals of approximately 60 rpm such that for each level of misalignment shown in Table 1, there were nearly 50 samples. Thus, from all the relevant data including the nominal condition, there are a total of 547 samples for each overhung and underhung bearing.

Table 1. MaFaulDa Data Misalignment Levels

Vertical [mm]	Horizontal [mm]
0.51	0.5
0.63	1.0
1.27	1.5
1.40	2.0
1.78	-
1.90	-

The MaFaulDa database has been used in several papers, mainly for classification of the provided fault types [38, 43-47]. A closer look at the MaFaulDa misalignment data by Ganeriwala et al. [48] revealed that simple spectral analysis would not be sufficient for misalignment detection but noted higher and more frequent harmonic peaks generally meant more misalignment. Zigang et al. [47] used this database to investigate how uncertain rotor parameters in angular misalignment affect the response predictions, concluding that it did affect the response amplitudes but not the vibration characteristics. Hence, using time series data may retain greater valuable information without requiring a high level of familiarity with the machine.

3.2 Data Preparation

Time series data was used to as the input to the three discussed models. To reduce the effect of noise, a low pass filter was applied before the waveforms were split into sections containing two periods of shaft rotation. The resulting short signals were then all normalized with respect to the maximum range to assist model generalizability.

To prepare the data for training the machine learning models, the data was resampled since the nominal case has less than a quarter of the horizontal and vertical case samples. Resampling was only performed for misalignment classification as the regression target values were relatively evenly distributed. The upsampling method used was SMOTE (Synthetic Minority Oversampling TEchnique) which upsamples not simply by duplicating existing samples but by creating new samples by interpolating between features of existing samples [49]. This method has been proven to help model accuracy in many classification problems and has been used with the MaFaulDa dataset by Ali et al. [45] who saw an increase in accuracy from 81 to 96% for imbalance classification. Note that the resampling was performed only on the training set as to not bias the testing set. The resampled waveforms were then fed into baseline ANN, CNN, and SVM models to determine the optimal data source. Suitability of overhung and underhung bearing sensors, velocity and acceleration values, and axes combinations were compared with each other.

Finally, features were selected and extracted using tsfresh, a python library that performs features extraction for time series data [50]. In addition to the implemented feature selection algorithm in the package that only eliminates minimal features with the exact same score, additional feature selection was performed using the Kendall correlation metric [51]. While the Pearson correlation metric is the most commonly used to rank features, it is only appropriate for normally distributed features. And upon testing the features for normality using the Shapiro-Wilk, D'Agostino's K^2, and Anderson-Darling test, it was discovered that very few of the features had a gaussian distribution. Hence, the Kendall correlation metric was chosen as it is a non-parametric test (disregards distribution of data) that can identify features with monotonic relationships. The Spearman metric has similar properties as Kendall, except that is specialized for ordinal data and the waveform features have no relative ranking. The Kendall correlation metric, τ, is calculated in Eqn. (1), where n_c and n_d are the number of concordant and discordant values of two features and n is the feature sample size.

$$\tau = \frac{n_c - n_d}{\frac{1}{2} * n * (n-1)} \tag{1}$$

The last step of feature down-selection was choosing the most relevant features from the reduced set. Using the sklearn feature selection library [52], the top 75 features were chosen based on the built-in function to calculate ANOVA (ANalysis Of VAriance) values that determine the significance of each of the features with respect to how they influence the target prediction.

3.3 Model Optimization

All three machine learning model types were used for misalignment classification, but only the ANN and CNN models were used for regression. The decisions to use either acceleration or velocity data and either overhung or underhung bearing data was determined by several baseline models using a single and double axis. Then the hyperparameters of each of the models listed in Table 2 were finely tuned using single, double, and triple axes to obtain the ideal models for classification and regression.

Table 2. All models to optimize

Model Goal	Input Data	ML Algorithm
Classification	Waveform	ANN
		CNN
		SVM
	Feature	ANN
		CNN
		SVM
Regression	Waveform	ANN
		CNN
	Feature	ANN
		CNN

The classification and regression models were evaluated using different performance metrics due to the nature of the resulting data types. The classification models calculated accuracy, precision, recall and F1-Score in Eqns. 2-5 for each class and averages of those scores were used to rank the models. These metrics are calculated using true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) which are easily visualizable in a confusion matrix. Accuracy reveals how many predicted values are correct, but it can fail to capture the severity of misclassification. Precision fills that gap by determining how many predicted values have been misclassified and the recall score conveys which classes have difficultly being identified. The F1-score is a comprehensive metric that combines the precision and recall values. The accuracy and F1-score were mainly used in final model evaluations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad \textbf{(2)}$$

$$Precsion = \frac{TP}{TP + FP} \quad \textbf{(3)}$$

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(4)}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad \textbf{(5)}$$

The regression models used the mean absolute error (MAE) as the chosen loss function in Eqn. 6. In this equation, n is the total number of samples, y_i is the actual value, and $\hat{y_i}$ is the predicted value. MAE is chosen over the Mean Squared Error (MSE) to reduce the influence of outliers on the regression model as the waveforms can vary somewhat widely.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \quad \textbf{(6)}$$

Each of the models went through k-fold cross validation with five folds so that each model was trained five separate times on the training set, which is 80% of the data [53]. This method of validation not only ensured that the model was not overfitting, but that the resulting average performance metric was not an outlier. Once the hyperparameters were selected via cross validation, the final models were evaluated on the testing data which make up of 20% of the overall data.

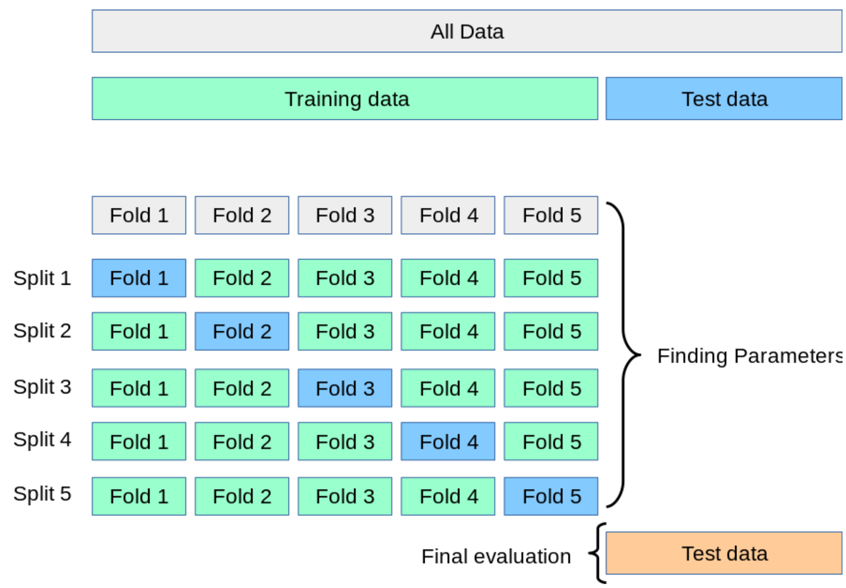

Figure 7. K-Fold Cross Validation [53]

3.4 Model Robustness Evaluation

The resulting models were tested for practical use in a realistic setting by adding noise and lower sampling frequencies to the MaFaulDa data. Noise levels were determined by the signal to noise ratio (SNR) which was calculated using Eqn. 7. SNR calculated in units of voltage requires that the common log of the root mean square (RMS) of the signal, S, over noise, N, to be multiplied by 10. Data was provided as acceleration, and because the acceleration is linearly proportional to the sensor voltage, the two values could simply be divided.

$$SNR = 10 * \log\left(\frac{S}{N}\right) \quad \textbf{(7)}$$

To obtain reasonable noise levels for testing, the spectral noise specification of the accelerometer from the experiment, Model 601A01 [54], was used as the basis. A frequency of 1 kHz was the maximum provided spectral noise rating and was relevant considering noise tends to decrease with an increase in sampling frequency. Hence while this was the most appropriate specification to use, the noise was likely less than what was calculated. This makes the noise tests conservative as greater noise was added. With a rating of 0.7 $\frac{\mu g}{\sqrt{Hz}}$ and the knowledge that the MaFaulDa data, sampled at 50 kHz, can reach a signal RMS of 0.1 gs, a noise level of 43 dB for the given data can be calculated. Thus, the added noise levels of 20, 15 and 5 dBs were used since a lower SNR is a noisier signal.

The sampling frequencies tested were 25, 10, and 5. This is to more accurately simulate a potential manufacturing facility that may purchase more common accelerometers. Sampling frequency values between 10 and 30 kHz are commonly seen in accelerometers used for vibration, and 5 kHz, the lowest tier, is when the filtered signal visibly degraded.

CHAPTER 4. ANALYSIS AND RESULTS

4.1 Signal Processing

To reduce the effect of noise on the model, the signals were run through a low pass filter that eliminated noise above five times the shaft frequency. Velocity and acceleration were both initially considered as inputs to the model and their filtered forms are displayed in Figure 8.

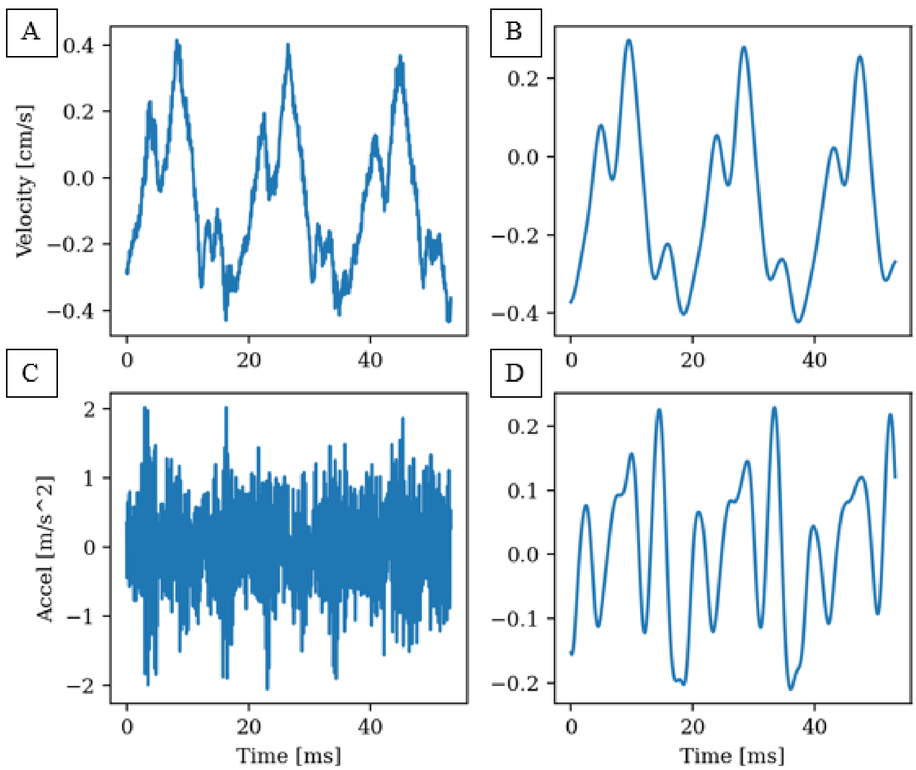

Figure 8. Raw (left) and filtered (right) waveforms for (A, B) velocity and (C, D) acceleration

Before inputting the waveforms into the models, they were split into sections containing two periods of shaft frequency as shown in Figure 9. With the split waveforms, not only have 48,582 samples been created from the given 547 samples provided by the MaFaulDa database, but the model also had a greater chance of recognizing the "W" and "M" shapes. In the case of inputs with combined axes, waveforms from each axis were padded at the ends with zeros and placed end to end in a single waveform. The padding ensured that the model learned to recognize the multiple sections within the waveform.

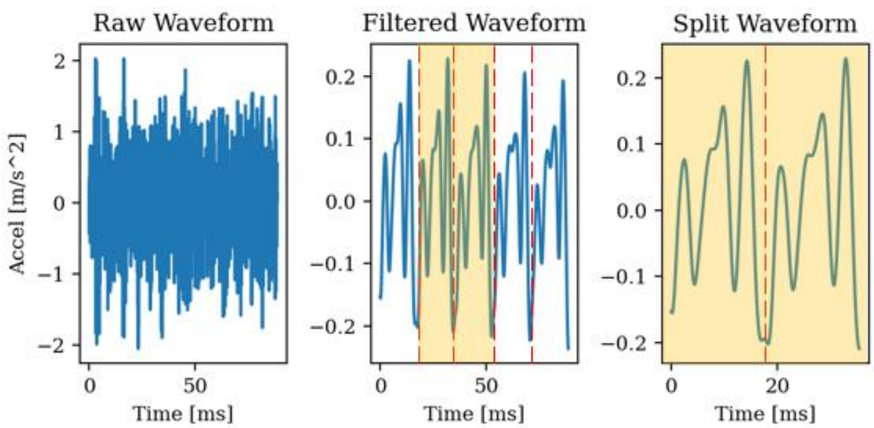

Figure 9. Process for splitting the waveform into short sections

4.2 Data Refinement and Selection

First and foremost is the issue of unbalanced data in the MaFaulDa database. There was a much higher quantity of vertical than horizontal or nominal samples because there were more steps of vertical misalignment in the experiment – 21,635, 13,764, and 3499 samples respectively. To amend this, the benefits of data resampling was investigated on baseline ANN and CNN classification models to determine whether to include upsampling, downsampling, or both. As mentioned previously, upsampling was performed using SMOTE and downsampling was performed with random selection. Higher F1-scores correspond to using both upsampling and downsampling as shown in Table 3. A higher F1-score indicates there will less degradation in performance transitioning from training to testing data, which aligns considering that without upsampling and downsampling, the model would tend to overfit on the larger number of vertical samples.

Table 3. Upsampling and downsampling study using baseline classification models

Upsampling	Downsampling	Model Type	Accuracy	Precision	Recall	F1-Score
N	N	ANN	89.3	97.5	65.9	78.6
Y	N	ANN	89.3	95.6	91.1	93.3
Y	N	ANN	84.5	84.3	76.8	80.4
N	Y	ANN	82.0	88.4	86.0	87.2
Y	Y	ANN	89.2	94.1	88.0	91.0
Y	Y	ANN	86.3	93.8	88.2	90.9
Y	N	CNN	91.7	92.8	90.4	91.6
Y	N	CNN	83.2	83.6	72.5	77.7
Y	Y	CNN	91.8	96.5	86.2	91.0
Y	Y	CNN	92.3	94.8	90.6	92.7

Many sources of data are provided in the MaFaulDa database, so a study was conducted to perhaps possible to eliminate excessive data. With less sensor placement there was reduced computation and set up times. All the available data was run with baseline ANN, CNN, and SVM classification models and performance between velocity and acceleration data and overhung and underhung bearing sensor placement was compared in Figure 10. The box and whisker plots revealed that acceleration and underhung bearings should be used because not only were the average values of the accuracies noticeably higher than their counterparts, but there was less variation. This implies that these data sources better transfer over to real world environments with more consistency.

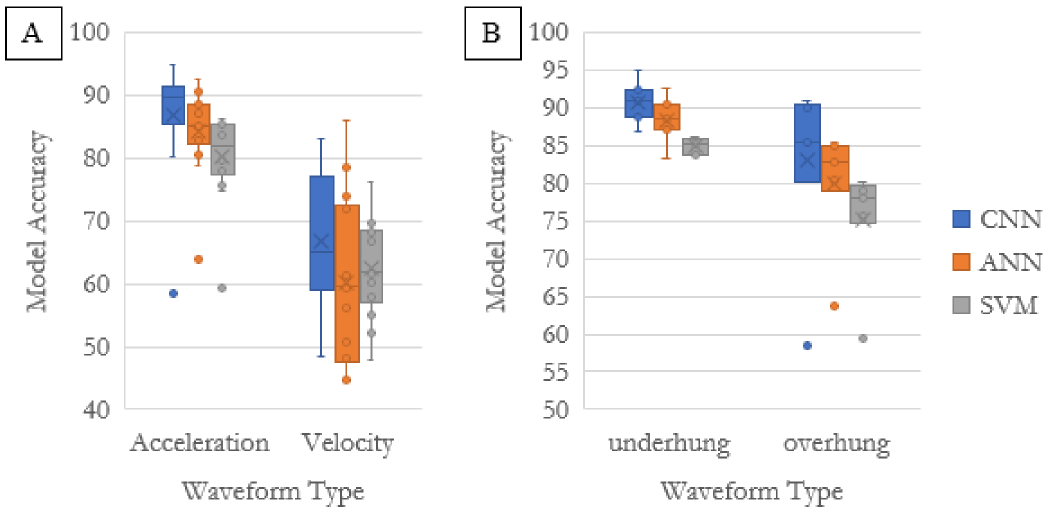

Figure 10. Baseline model performance comparing (A) acceleration and velocity and (B) overhung and underhung sensor placement

The filtered velocity and accelerations waveforms were compared for each class and for three levels of misalignment (Figure 11). Although the velocity waveforms were cleaner, there was little visible difference between classes of misalignment and the waveform amplitudes did not increase with greater misalignment. Acceleration on the contrary had signals that were visually distinguishable and had amplitudes that increased nonlinearly with misalignment. It should be noted that in industry, it is more common to visually review the velocity waveforms for misalignment diagnosis. However, from the raw waveforms, it is evident that identifying patterns is difficult without the low pass filter. The goal of this study was to acquire more details about the nature of the misalignment rather than simple fault detection, and so it follows that the acceleration and underhung bearing waveforms are more practical in this scenario.

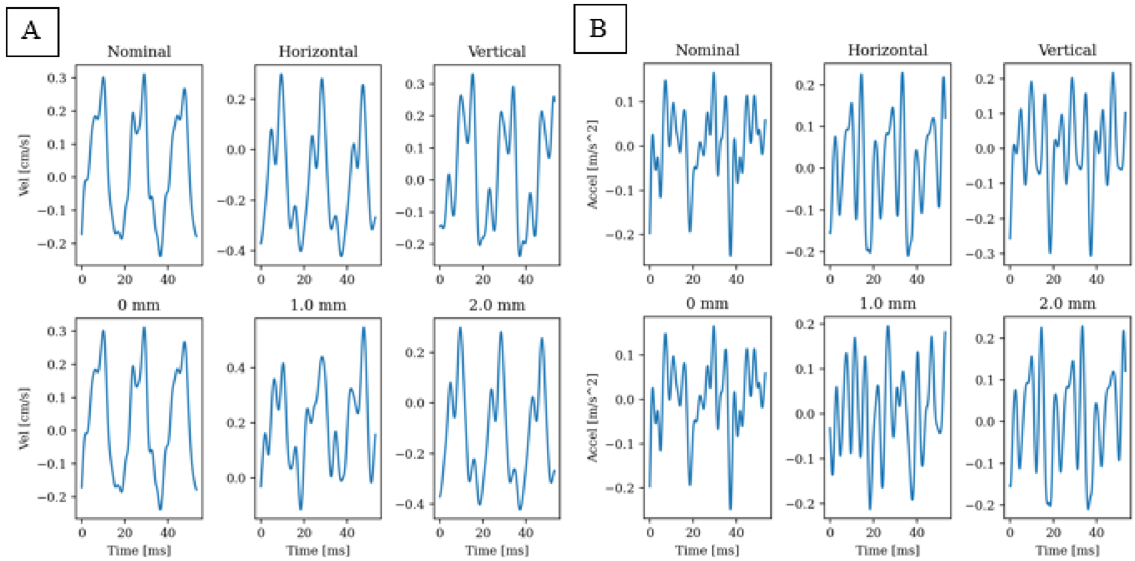

Figure 11. Comparing waveforms for each class and three levels of misalignment for (A) Velocity and (B) Acceleration

Another study on the axes considered performance in the radial (R), tangential (T), axial (A) directions. The box and whisker plots in Figure 12 demonstrate that between single axis inputs, the accelerometer data in the tangential direction outperformed the axial and radial directions. And for double axis inputs, the difference was not as clear but based on the average accuracies of 74.6%, 74.7%, and 74.2%, the combined axial and tangential directions were slightly more consistent than the other combinations. The decision to move forward with the axial and tangential direction stems from two more considerations. The first being that the radial and tangential directions had very similar waveforms once gravity was removed since they are both perpendicular to the shaft axis, so the inclusion of the axial direction would provide a more varied set of data. Secondly, the CNN model already appeared to surpass the other models, so the higher CNN accuracy for the axial and tangential directions was given greater consideration.

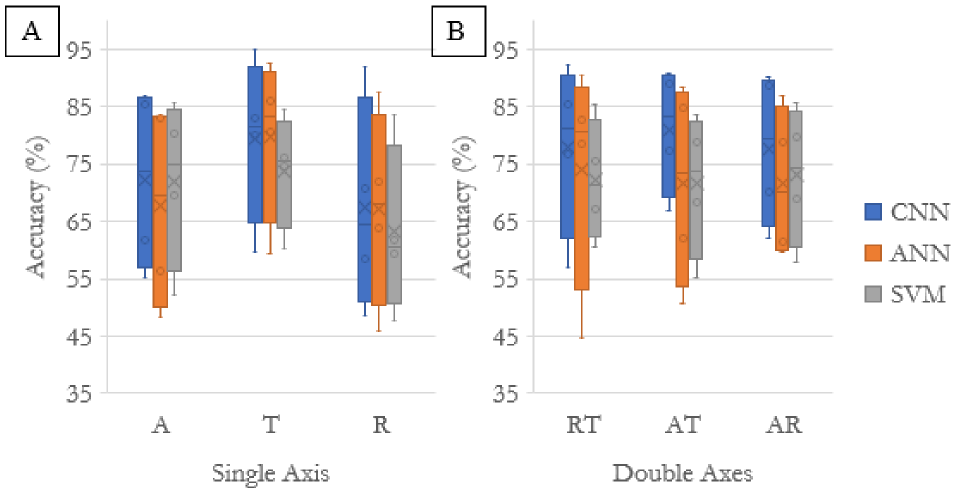

Figure 12. Comparisons of baseline models for (A) single and (B) double axes

Accuracy for single, double, and triple axes was also examined in Figure 13. At this stage, it is not definite which combination performs better, thus all three were used in further studies. Yet, it should be noted that a reduction in the required number of axes makes data collection simpler, so a slight improvement in accuracy from a greater number of sensors may not be weighted as heavily.

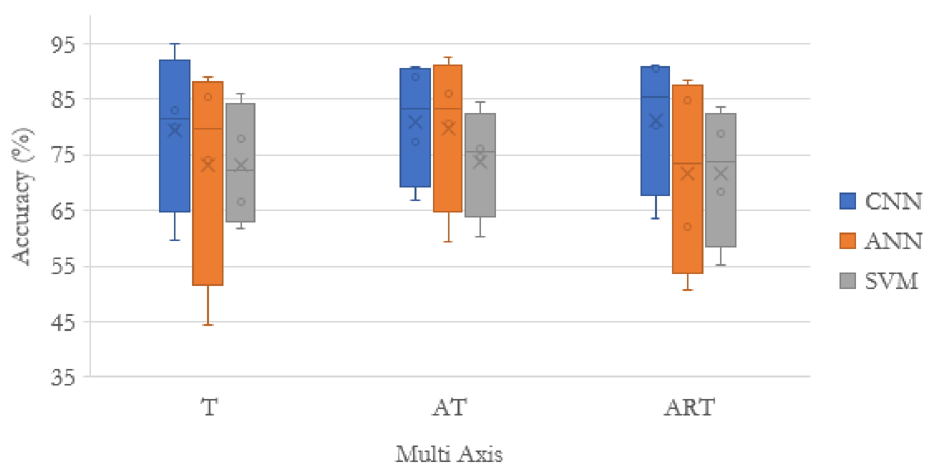

Figure 13. Comparison between single, double, and triple axes

To summarize data refinement and selection, the samples from the MaFaulDa data were initially separated into an 80/20 training and testing split. The training set was then both upsampled and downsampled to prevent against overfitting. With a refined data set, the data sources were studied, and it was determined that acceleration and underhung data would be used for the remainder of this study. Additionally, the optimal axes combinations were identified for single, double, and triple axes which all were evaluated with the models.

4.3 Feature Extraction

The tsfresh library extracts over 700 features, and if all the features were to be utilized, the required computation time for feature extraction would be unreasonable. Thus, the number of features was reduced not only by a method within the tsfresh library, but also by using the Kendall correlation metric as mentioned above. In Figure 14, it can be seen that many features were closely monotonically related to each other based on the blue and yellow sections of the heat map. A value of 1 indicates a positive monotonic relationship and -1 indicates a negative relationship. If features are closely correlated, those similar features are likely redundant. In the interest of reducing the number features, all the feature pairs with a Kendall correlation of over 0.8 or under -0.8 were eliminated. Using the tsfresh feature selection algorithm, the number of features was brought down to 530 and then with Kendall, down further to 126 features.

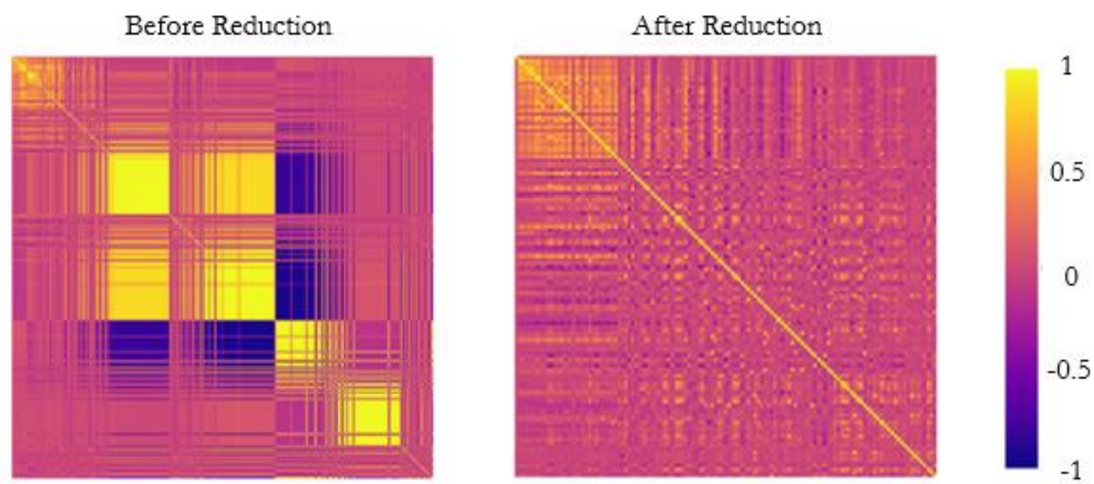

Figure 14. Feature reduction using the Kendall correlation metric

Still, 126 features require a substantial amount of time and memory to extract from each sample, more than the laptop in use can physically compute with 8.0 GB of RAM. Thus, the top 75 features were extracted using the ANOVA metric which determines how well the feature correlated with the target label. A full table of all the extracted features for the underhung tangential axis are found in Appendix A which include various metrics with different parameter combinations. A few of the more significant features were the "matrix profile" feature which uses the Tukey's Five Number Set to summarize the data distributions, and the "change quantiles" feature which examines the change seen in a specific quantile of the data [50]. This feature selection process was performed for all three axes combinations and for both regression and classification such that only 75 features total were used for each model. For example, if only a single axis was used, all 75 features would be used as the input. If three axes were used, the top 25 features from each axis would be used.

4.4 ANN Models

The neural networks were implemented using TensorFlow [55]. The structure of the artificial neural networks used was rather simple, comprising of an input layer, hidden layer, and an output layer. As mentioned, rough hyperparameters were examined in an initial optimization of a baseline classification ANN model. The "dropout" dictates how many nodes in a layer to randomly eliminate during the training phase to reduce the risk of overfitting. This regularization technique was applied to both the input and hidden layer. The "Dense #" hyperparameter in Table 4 was the number of nodes in each layer of the ANN. All combinations of hyperparameters were tested and the average loss and accuracy ("Acc") of each hyperparameter is listed. The "UT" and "UA" abbreviations refer to "underhung tangential" and "underhung axial" and are used throughout the rest of the tables this thesis. It is evident from the values that changing these hyperparameter values did not drastically affect model performance. Yet there is quite a separation between the single and double axes models, obtaining accuracies differences of at least 4% consistently. From this, the range of dropout values to test was modified to [0.05, 0.1, 0.2], but the "Dense #" range remained the same.

Table 4. Initial hyperparameter optimization of classification ANN

		UT		UA+UT	
		Avg Loss	Avg Acc	Avg Loss	Avg Acc
Dropout	0	25.6	92.2	38.1	87.5
	0.1	26.0	91.5	33.8	87.9
	0.2	27.2	91.0	37.0	86.4
Dense #	64	22.3	92.4	35.7	87.6
	128	30.2	90.6	37.6	86.9
	256	26.3	91.7	35.7	87.2

The new set of hyperparameters was evaluated using the same method as above with 10 epochs for each of the ANN models, resulting in the best performing axes for each model highlighted in Table 5. The detailed values used in this analysis can be found in Appendix B. It is interesting to note that the same types of axes and hyperparameters are common for both classification and regression models respectively.

Table 5. ANN models' final hyperparameters

ANN	Axes	Dropout	Dense #	Loss	Accuracy	F1-Score
Classification - Waveform	UT	0.05	64	25	91.7	90.3
	UA+UT	0.1	64	36.4	86.8	86.1
	UA+UR+UT	0.15	64	49.5	83.1	85.2
Regression - Waveform	UT	0.1	256	21.9	N/A	63.6
	UA+UT	0.1	256	24.1	N/A	59.3
	UA+UR+UT	0.05	256	16.8	N/A	71.4
Classification - Features	UT	0.05	64	33.8	86.7	84.7
	UA+UT	0.05	128	64.8	70.1	75.7
	UA+UR+UT	0.15	64	47.6	80.1	80.4
Regression - Features	UT	0.05	128	35.1	N/A	44.4
	UA+UT	0.05	64	37.8	N/A	41.6
	UA+UR+UT	0.05	256	27.5	N/A	59

Using the final ANN model hyperparameters, each of the four models were optimized using 30 epochs so they fully converged. The training and validation scores are plotted for all four models in Figure 15. In the legend, “T” indicates the training set and “V” is the validation set. Because there was no significant drop in model performance from the training to validation set, there was no extreme overfitting occurring in the model training. Furthermore, between the waveforms and features results, it appears that the features provided more consistent results as the curve seemed to be smoother as it converged from epoch to epoch for the validation set. However, for the regression results in Figure 16 the opposite was true in that the waveforms produced a smoother convergence curve. This may speak to the better performance of the classification models.

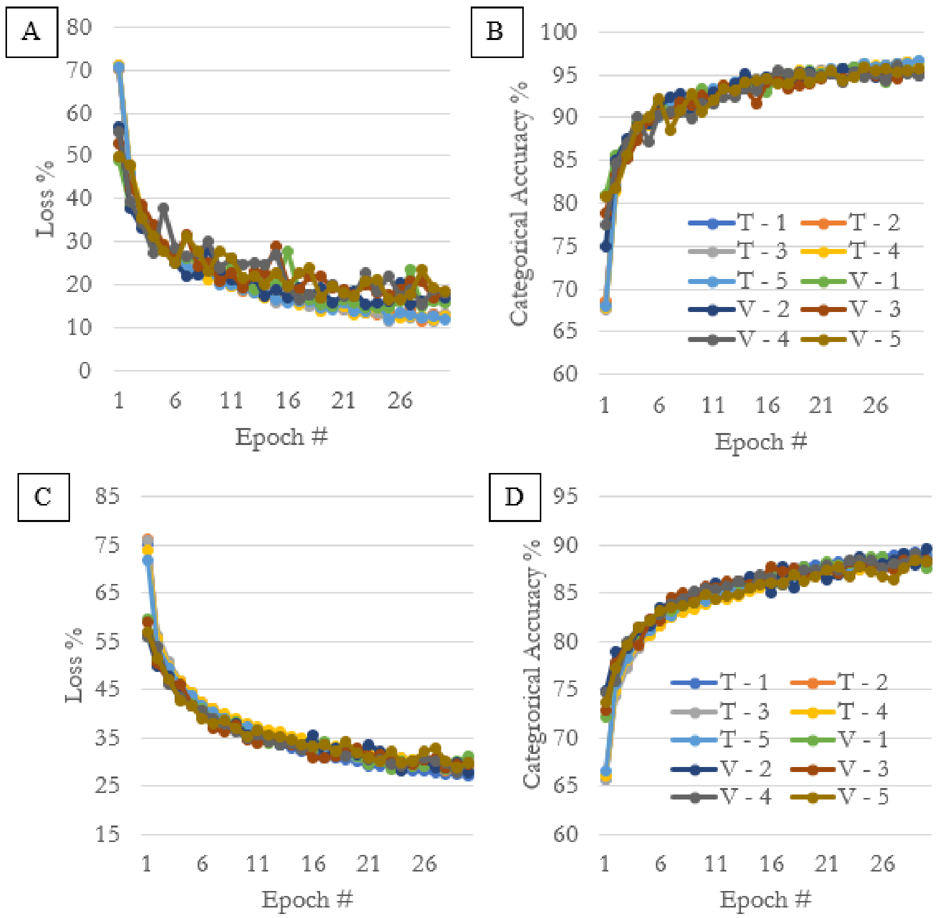

Figure 15. 5-fold cross validation convergence for ANN classification models, displaying (A) loss using waveforms (B) accuracy using waveforms (C) loss using features and (D) accuracy using features

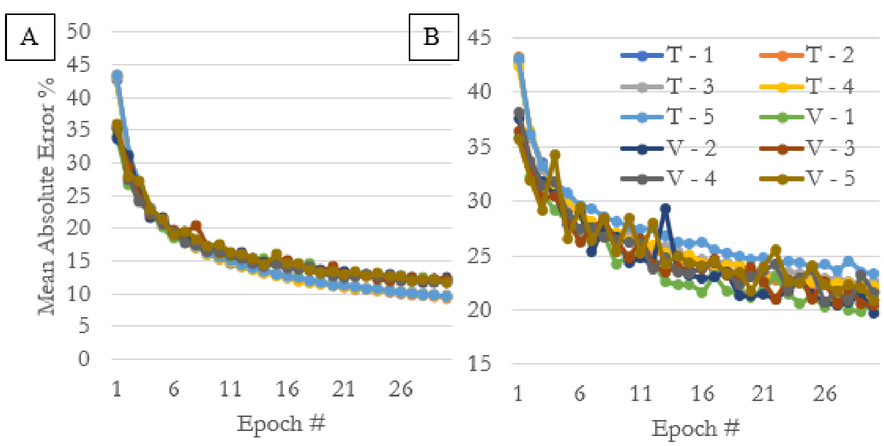

Figure 16. 5-fold MAE results for ANN regression models for (A) waveforms and (B) features

The final results for the ANN classification models are displayed in Figure 17 and regression results are displayed in Figure 18. The confusion matrices are normalized across the true labels such that a 1 on the diagonal would mean that all the samples were classified correctly. To create a confusion matrix for regression, the samples have been grouped into bins of 0.5 mm such that a bin labeled “0.0” encompasses [0, 0.5) mm values. From the darkness of the confusion matrix squares, it can be seen that even though some values are misclassified, nearly all of the values are within 1 mm of the intended value. Note that there were no true labels “>2”, so if it is included in the confusion matrix, it implies that there were false predicted values greater than two. However, in nearly all of the following confusion matrices, the values were rounded to zero.

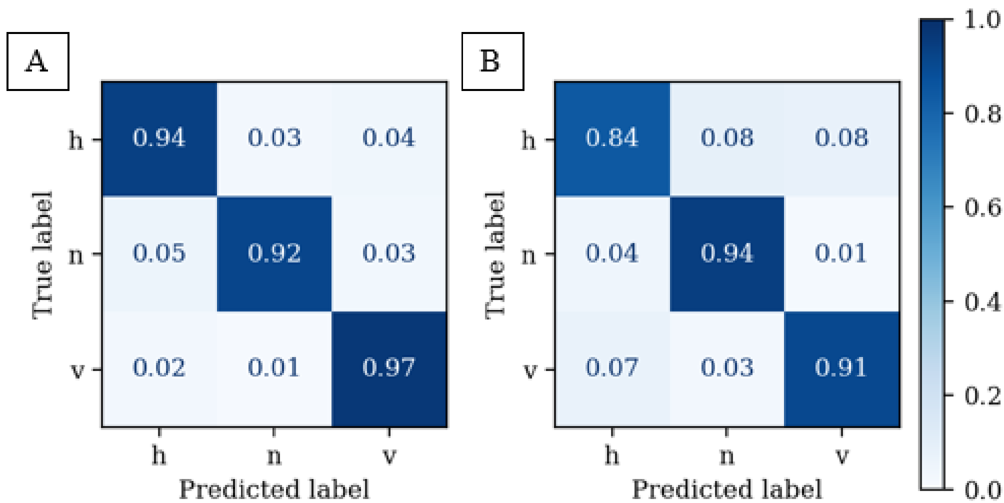

Figure 17. Confusion matrix for optimized ANN classification model using (A) waveform and (B) features

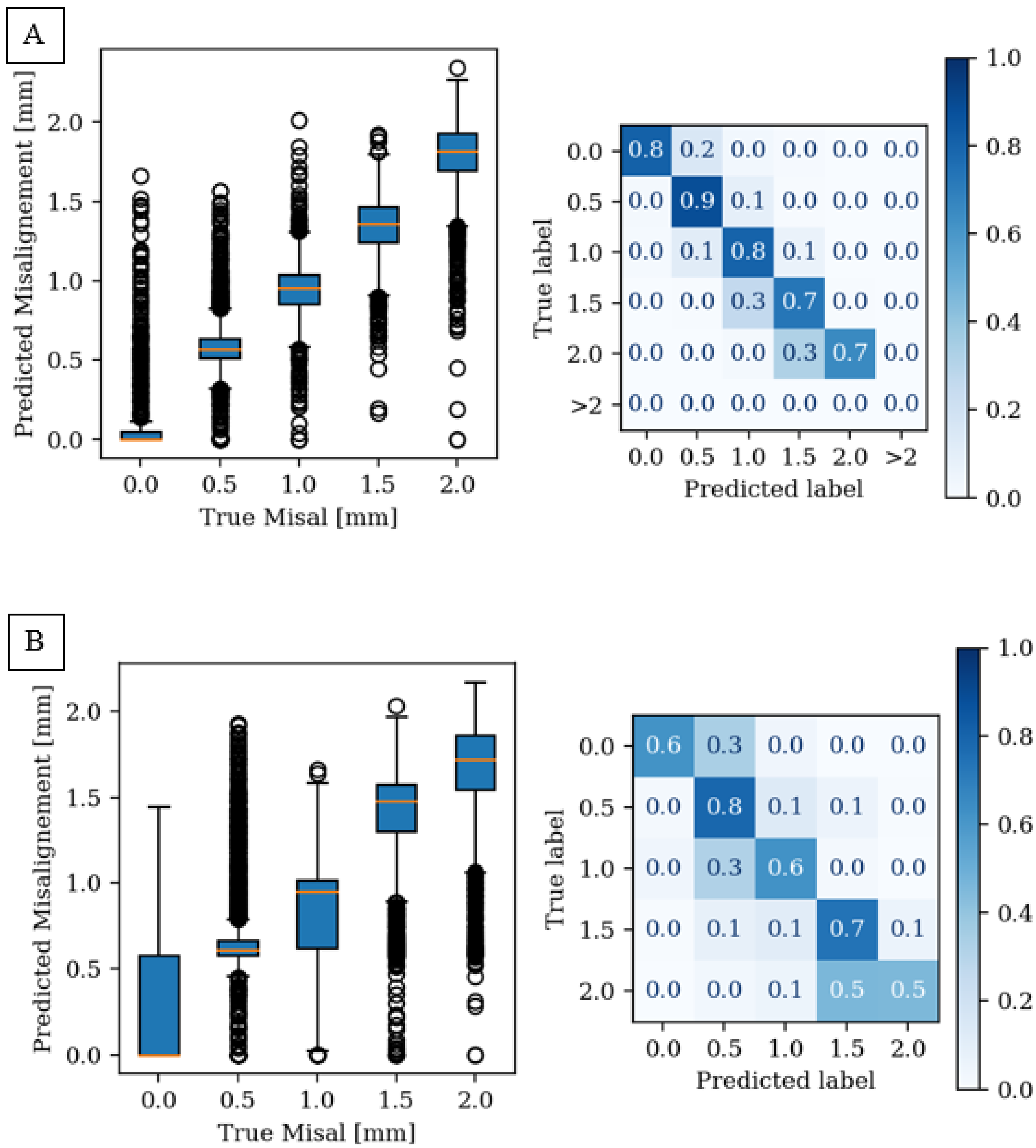

Figure 18. Box and whisker plots (left) and confusion matrices (right) for ANN regression models using (A) waveforms and (B) features

4.5 CNN Models

The convolutional neural network models had two convolutional layers, a max pooling layer, and a dense layer before the output layer. The purpose of the max pooling layer was to select the most important feature in a window to condense the data and provide less confusion for the model. The “Dropout” and “Dense #” hyperparameters are the same as in the ANN, but the “Filter” and “Kernel” hyperparameters are unique to CNNs. In the convolutional layer, the filter value is the number of times the CNN applied a feature transformation to the incoming data before stepping through it. Kernel size is essentially the size of the window used to step through the incoming data. Thus, a smaller kernel was better for detecting small features and vice versa. Based on the initial hyperparameter tuning results in Table 6, the Dense # and Filter values were the only sets changed to [128, 256, 512] and [32] respectively. The average accuracy values of the “Dense #” parameters indicated that a higher value might be preferable. The average accuracies of the filter values were very similar; hence 32 was used going forward as model computation times for 64 are double 32.

Table 6. Initial hyperparameter optimization of classification CNN

		UT		UA+UT	
		Avg Loss	Avg Acc	Avg Loss	Avg Acc
Dropout	0	19.5	94.2	37.2	88.9
	0.1	18.4	94.3	36.5	88.6
	0.2	18.9	94.1	34.8	88.6
Dense #	64	22.0	93.1	36.9	87.3
	128	18.6	94.2	35.3	89.1
	256	16.1	95.3	36.3	89.7
Filter	32	19.0	94.2	35.4	88.7
	64	18.9	94.3	36.9	88.7
Kernel	3	19.8	93.8	37.6	87.7
	7	18.1	94.4	37.8	88.4
	11	18.9	94.4	33.1	90.0

The final optimized hyperparameters and axes are below with 10 epochs, and in contrast to the ANN, the CNN was less consistent and the final axes and hyperparameters vary for each model. Particularly for the regression model using features, the double and triple axes did not produce usable results because everything was being classified as 0 mm and for the single axis, the number of features had to be reduced from 75 to 25 to produce meaningful results. This is likely because CNNs are proficient at examining sections of images and finding features within those to utilize. When the data is transformed into a feature set, there are no "W" or "M" shapes to examine as they have already been boiled down to features.

Table 7. CNN models' final hyperparameters

CNN	Axes	Dropout	Dense #	Kernel	Loss	Acc	F1-Score
Classification - Waveform	UT	0.2	256	11	16.5	95.1	93.9
	UA+UT	0.2	256	11	30.5	91.1	88.6
	UA+UR+UT	0.1	256	11	19.7	94.5	93.2
Regression - Waveform	UT	0	512	11	22.7	N/A	58.1
	UA+UT	0.1	256	11	20.1	N/A	56.8
	UA+UR+UT	0	256	11	17.1	N/A	69.9
Classification - Features	UT	0	512	7	15.9	94.9	94.5
	UA+UT	0	512	7	49.8	78	83.8
	UA+UR+UT	0	512	3	39	83.7	85.9
Regression - Features	UT (25 features)	0.2	256	11	28.9	N/A	59.7
	UA+UT	-	-	-	-	N/A	-
	UA+UR+UT	-	-	-	-	N/A	-

Using the final CNN model hyperparameters, each of the four final CNN models were optimized using 30 epochs. Unlike the features used for the ANN classification models, it appears that the features have a smoother convergence curve in Figure 19. And perhaps the most prominent observation was that the CNN feature regression model in Figure 20 there are folds that completely failed to predict the data, meaning that it was unreliable. Furthermore, the difference between the training and validation set appeared larger. This might perhaps be explained by the lack of dropout used in the final models.

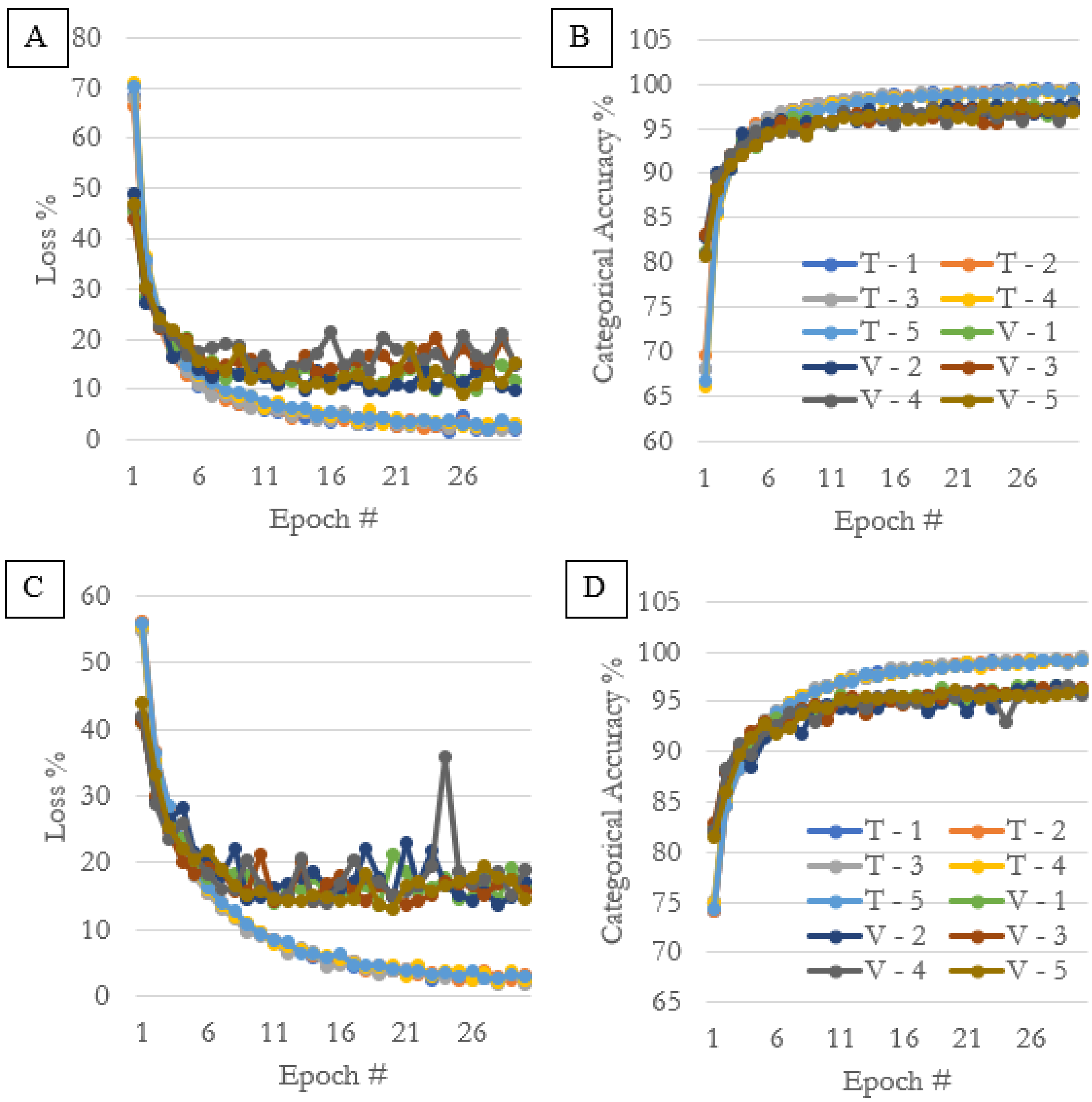

Figure 19. 5-fold cross validation convergence for CNN classification models, displaying (A) loss using waveforms (B) accuracy using waveforms (C) loss using features and (D) accuracy using features

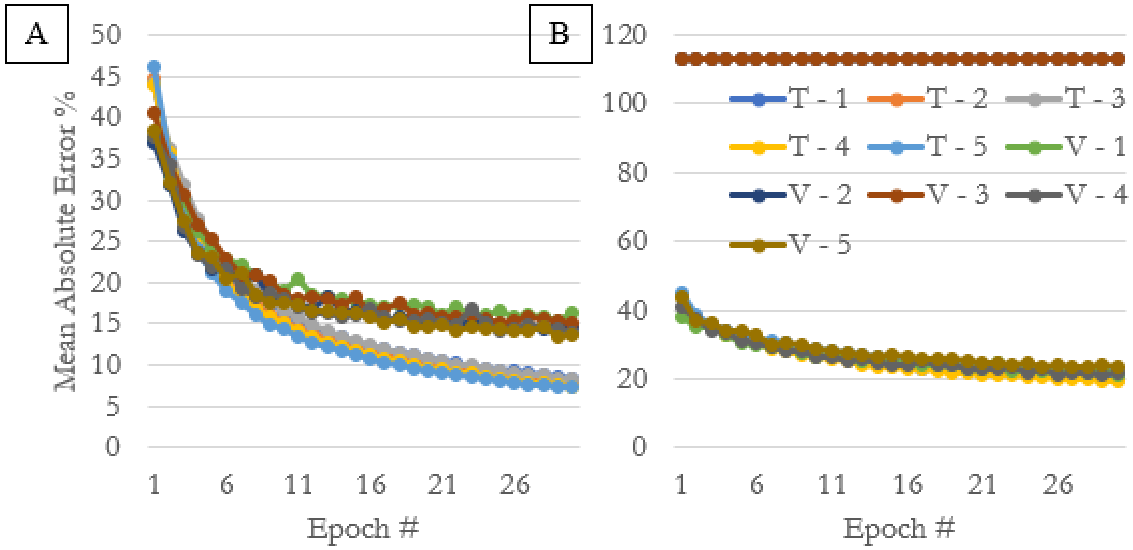

Figure 20. 5-fold MAE results for regression CNN models for A) waveforms and (B) features

The results for the CNN models are below, using the same confusion matrices and box and whisker plots as the ANN model results. Again for both Figure 21 and Figure 22, the waveforms performed better than the features and even more noticeably so for regression.

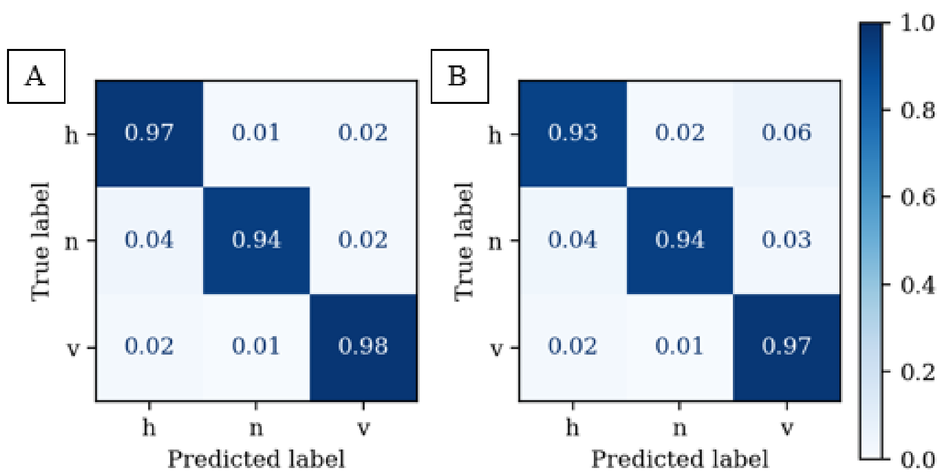

Figure 21. Confusion matrix for optimized CNN classification model using (A) waveform and (B) features

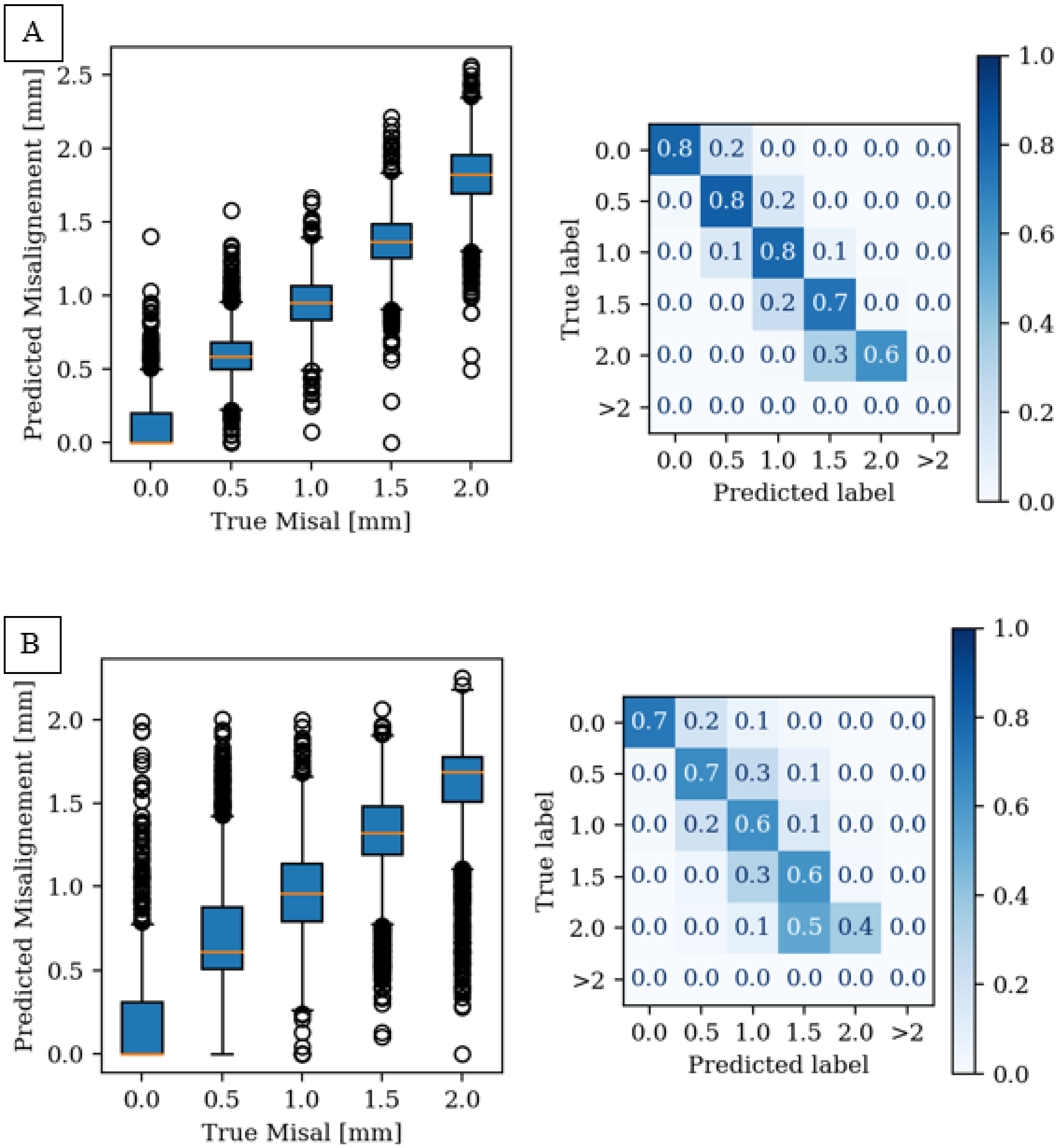

Figure 22. Box and whisker plots (left) and confusion matrices (right) for CNN regression models using (A) waveforms and (B) features

4.6 SVM Model

In the support vector machine initial hyperparameter selection, there were two main items considered: C-value and the Kernel. The C-value is the regularization technique commonly used for SVMs that controls the margin width of the decision boundary line; a smaller C leads to a large margin which and vice versa. A large margin tends to generate better results for clearly separable data because otherwise it would cover overlapping features and result in more misclassification. The kernel in the context of an SVM is the way that the data is transformed so that it can be linearly separable. This was useful here because the waveform acceleration data was certain to overlap with more than one class. The RBF kernel stands for “radial basis function” and is the most generalizable kernel. Polynomial kernels vary based on the degree of the polynomial which is another hyperparameter altered in this optimization. From Table 8, the performance between C-values of 0.001 and 1 was rather large, but between 1 and 100 there was much less difference, so the new range was determined to be [1, 10, 100]. And between the two kernel types, although a polynomial kernel of degree four performed the best of the polynomial kernels, the RBF kernel still exceeded it and is henceforth used. It should be noted that perhaps the reason the fourth degree polynomial worked best was because the waveform “W” and “M” shapes most closely resembled fourth degree polynomials.

Table 8. Initial hyperparameter optimization of classification SVM

		SVM	
		UT	UA+UT
		Avg Acc	Avg Acc
C-Value	0.001	37.1	32.7
	1	65.6	72.3
	100	72.2	73.8
Kernel	rbf	72.6	74.1
	poly	54.7	56.0
Poly Degree	0	36.6	36.6
	3	62.8	61.4
	4	60.6	63.4
	5	58.9	62.6

The final hyperparameter optimization for the SVM models indicated that a higher C-value was more suited for misalignment type classification, likely due to many of the features overlapping. The difference in performance between the waveform and feature inputs was also much greater than the previous ANN and CNN classification models.

SVM	Axes	C-value	Accuracy	F1-Score
Classification - Waveform	UT	100	91	90.3
	UA+UT	100	90.6	88.3
	UA+UR+UT	100	88.2	88.6
Classification - Features	UT	100	87.4	88.1
	UA+UT	100	70.1	74.6
	UA+UR+UT	100	77.8	79.8

The SVM model was the only model in which the accuracies for any class improved from waveforms to features. In Figure 23, the features actually classified the nominal case with better accuracy; however, the drop in the horizontal and vertical cases did decrease a greater amount.

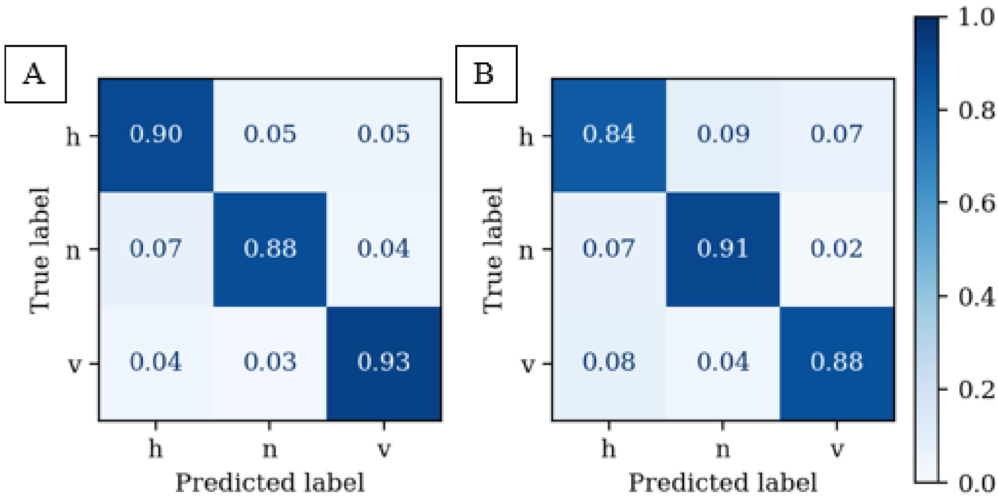

Figure 23. Confusion matrix for optimized SVM classification model using (A) waveform and (B) features

CHAPTER 5. DISCUSSION

5.1 Model Performance Comparison

All the performance metrics are combined in Figure 24. While all classification models perform with above 80% accuracy, the CNN models performed the best, with over 95% accuracy for both waveform and feature inputs. For regression, the ANN models performed most consistently, with a mean absolute error of 11.8% with waveform inputs and 21.6% for feature input. In both classification and regression, the full waveform input provided superior results though it is more evident in the regression models.

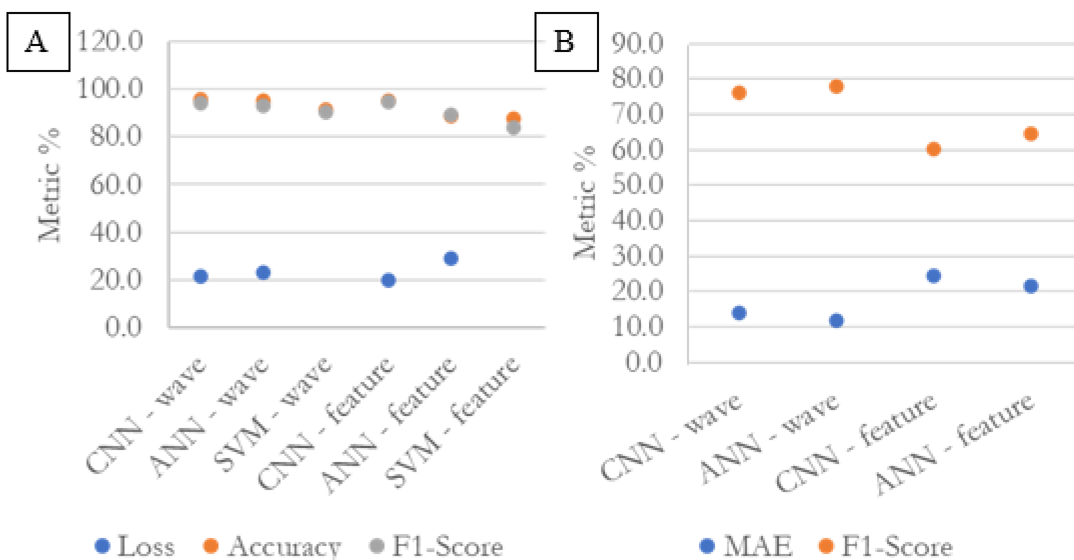

Figure 24. All model performance metrics for (A) classification and (B) regression

5.1.1 SHAP values

SHAP values, which stands for "SHapley Additive exPlanation", are quantities assigned to features of any algorithm that inform the user on the more relevant features for any model [56]. One can observe each model's perspective on the same waveform and how it classified it. In Figure 25, a vertical misalignment waveform sample from the tangential axis was classified by CNN and ANN models. The plots display the SHAP values for each class, but it is clear based on the higher maximum SHAP value in the vertical cases that both models classified it correctly. Looking at the locations of the higher, yellow values, it appears that the models were looking for the peaks give it information about the waveforms. Interestingly, the CNN looked at the positive peaks and the ANN focused on the negative peaks.

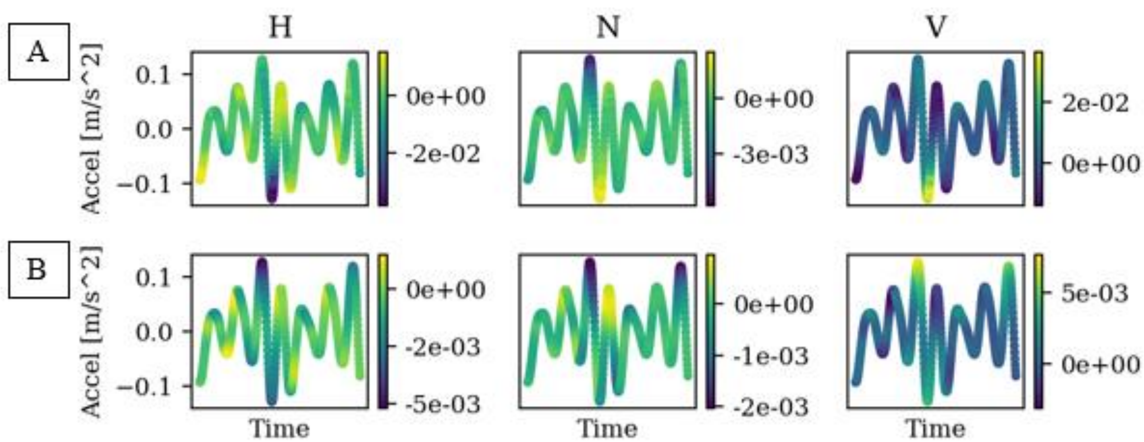

Figure 25. SHAP values of an (A) ANN and (B) CNN classifying a vertical sample

Looking at a horizontal sample in Figure 26, the models once again scrutinized at the peaks of the waveform to classify the misalignment type. This perhaps supports the notion that the models are looking for peaks from the "M" and "W" shapes. However, for the CNN classification, it was able to classify the waveform even though the maximum value is lower likely because there is a larger amount of yellow highlighted in the center of the waveform. A consistent behavior between the models is that they tended to give importance to the centers of the waveforms.

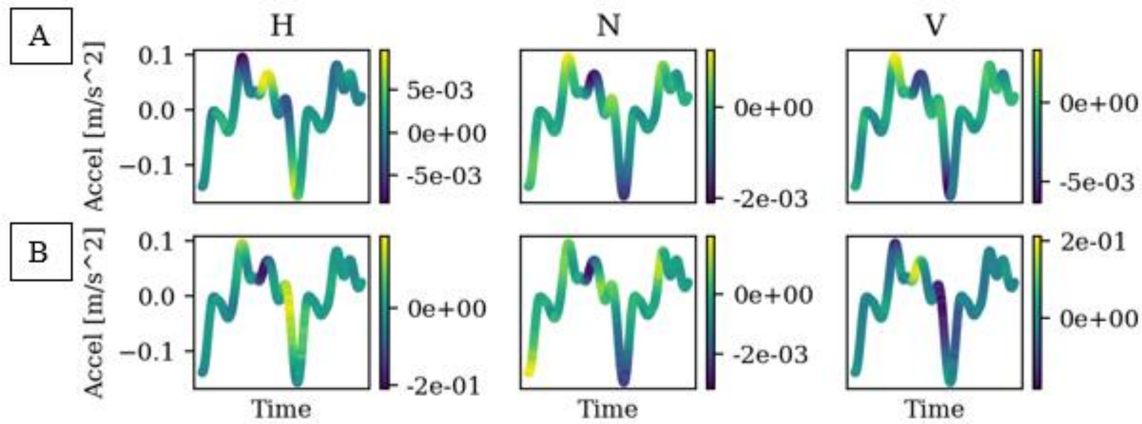

Figure 26. SHAP values of an (A) ANN and (B) CNN classifying a horizontal sample

SHAP values are more typically used in feature selection to determine which features are contributing to the model. Below in Figure 27, the top twenty features and their SHAP values are displayed for a vertical sample, and it is curious that the ANOVA rankings were not consistent in practice, although the first ranked value was certainly the most useful value. The feature numbers correspond to the table in Appendix A. However, it does support that using 75 features was crucial since the lower ranked features ranked high in the plot and although not shown, the ranking was different across various samples.

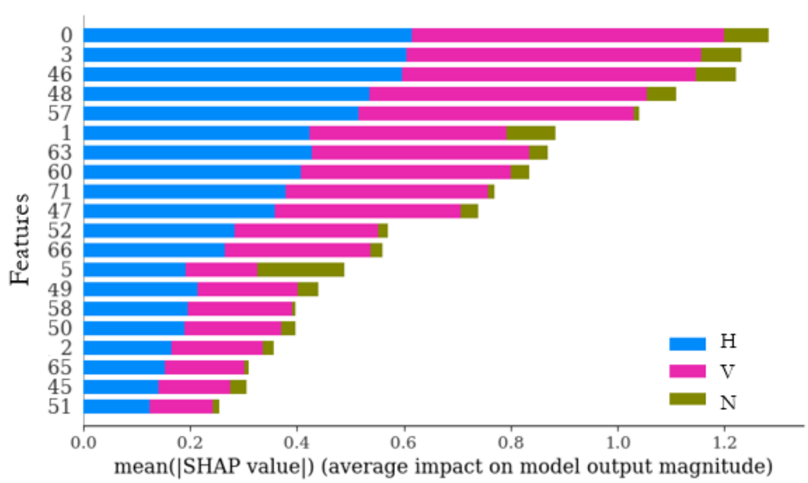

Figure 27. SHAP values for features for an ANN classification model

5.1.2 Computation Times

Computation times were also monitored for each of the final models, and the total times are listed on the bars in Figure 28. The items in the legend were performed in sequence and when summed, made up the entire training or testing process. The training times in seconds were for the entire training set of 38,864 samples. The testing times however, are in milliseconds and represent the computation time for a single sample in the testing set. The time for each sample was determined as the average for all 9,718 testing samples. The data is presented differently because in practice, the training sequence will always be performed over many samples, but the testing sequence can be applied to one or many samples at a time. The most evident takeaway is that feature extraction added a significant amount of time for each model, in most cases more than quadrupling that of the waveforms; however, it did reduce the model evaluation time. Comparing model types, it appears that the SVM took significantly longer to evaluate than both the neural networks which were very similar in computation time, although the CNN takes slightly longer. The device running these tests used an Intel® Core™ i5-7300HQ CPU, and an NVIDIA GeForce GTX 1050 GPU which was used by the neural networks with the aid of NVIDIA's cuDNN library [57]. For further details about the components of the computation times, refer to Appendix C.

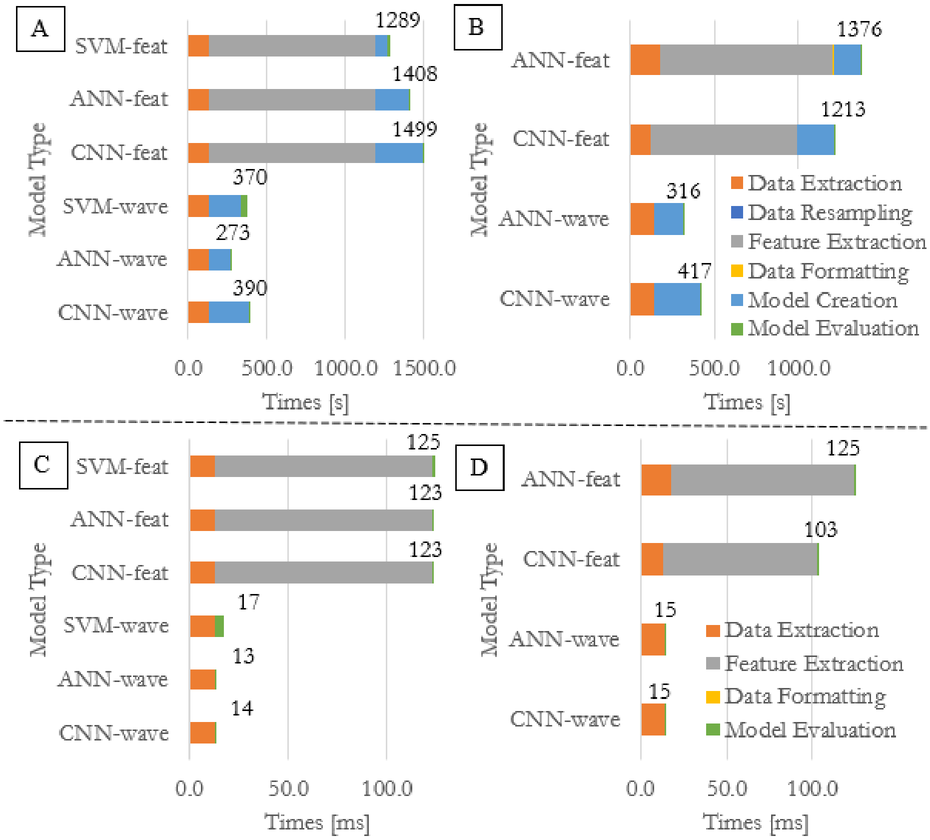

Figure 28. Cohesive training (top) and testing (bottom) computation times for (A, C) classification and (B, D) regression

Because the feature extraction times made up a majority of the overall computation time, a short study was done to determine if decreasing the number of features used would improve model time. However, looking at Figure 29, there was no clear trend between computation time and the number of features used as the input layer of the model. On the contrary, looking at it with respect to the number of features used per axis, there was a somewhat linear trend showing an increase in computation time as the number of features per axis increased. Thus, adding axes did not drastically affect the feature extraction time, only the number of features one wishes to extract from the axes.

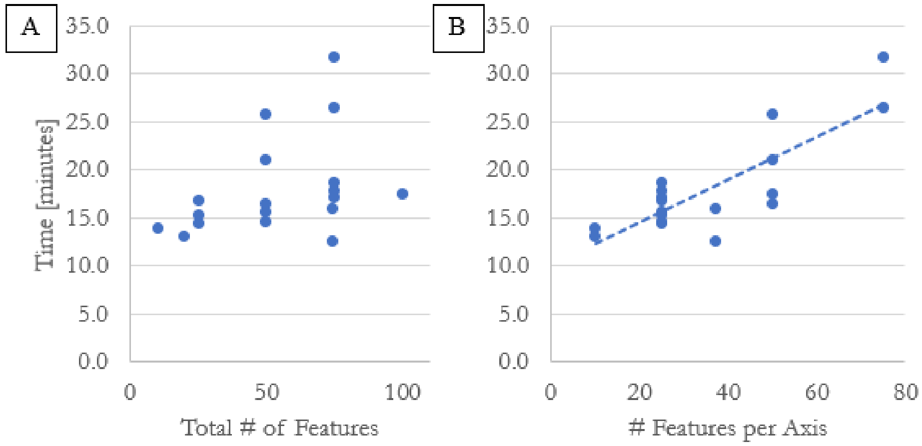

Figure 29. Feature computation times with respect to (A) the total number of features for a model and (B) total number of features per axis

Regarding the model hyperparameters, a brief study was conducted on how each parameter affected model computation time. Results are located in Appendix C.

5.2 Robustness

The MaFaulDa data was collected in ideal conditions, with a high sampling frequency and little to no environmental factors to contribute to noise levels. Thus, it would be imperative to evaluate the model performance in more realistic environmental conditions.

5.2.1 Noisy Data

The noise levels tested had SNR levels of 20, 15, and 10 dB. A low pass filter was used in the beginning of the signal processing to prevent against noise and was found to satisfy its purpose. Based on the heat maps in Figure 30, the models performed slightly better with no noise; however, the model type had a greater impact than the change in noise level on the results.

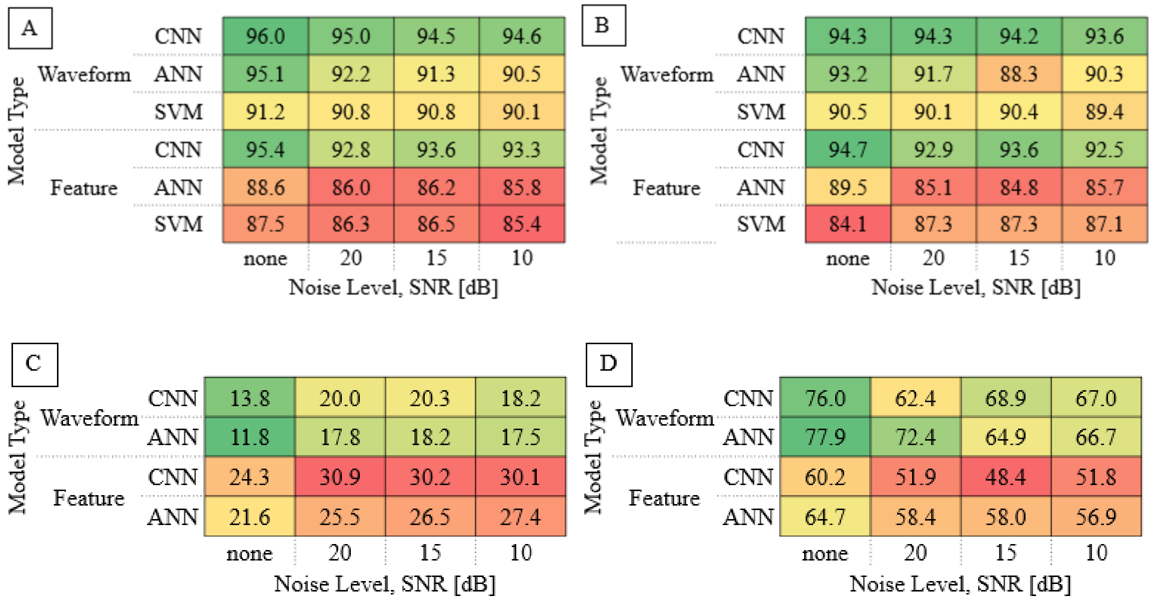

Figure 30. Effects of noise level on the classification (top) and regression (bottom) models via the (A) accuracy, (B, D) F1-score, and (C) MAE of each model

The small decrease in model performance is visually depicted in Figure 31. Although the model performance did fall, because of the low pass filter, the difference was minimal between the extreme levels on noise – only a 3% drop. This was perhaps because when using the acclerometer data, the waveform already appeared extremely noisy and thus the low pass filter served its purpose as originally intended. An interesting result is that ANN models with waveforms increased in performance at 10 dB which was perhaps a result of having too many counterproductive features at a lower noise level.

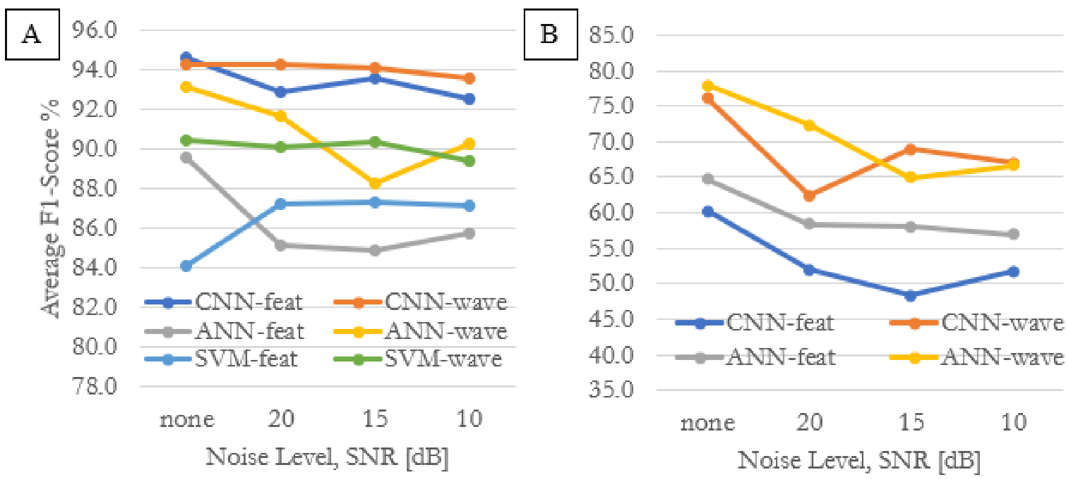

Figure 31. Effect of noise level on F1-score of (A) classification and (B) regression models

5.2.2 Sampling Rates

The sampling rates was originally 50 kHz and for the purpose of this study was decreased to 25, 10, and 5 kHz. The resulting waveform from 25 kHz was not visibly different from 50 kHz, but the 10 kHz and 5kHz waveforms showed clear deterioration of the signal. Although it was not clearly visible from the raw waveforms, looking at the filtered waveforms it was evident that the “M” and “W” patterns were absent.

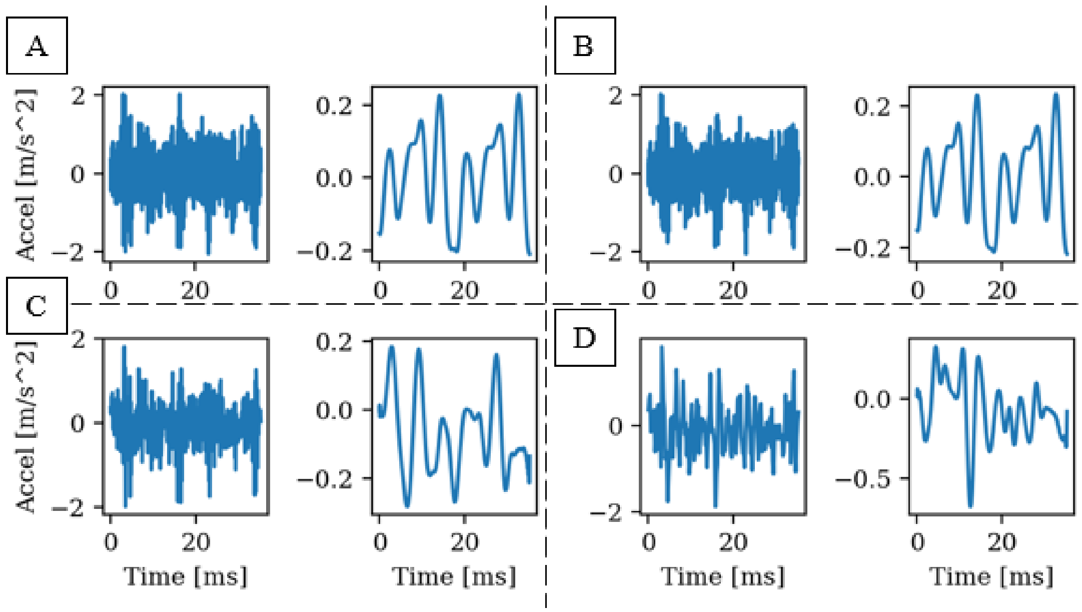

Figure 32. Raw and filtered waveforms for (A) 50kHz (B) 25kHz (C) 10kHz and (D) 5kHz

Consequently, the results from the drop in sampling frequency were more severe compared to the effect from the noise level. This may also be due to effects from aliasing which is a phenomenon in which the waveforms are misinterpretted due to a low sampling frequeny. The key to prevent this, if given the sampling frequency required to visualize the desired features, is to use the Nyquist frequency, calculated by doubling that desired frequency. Clearly, 5 and 10 kHz fall below the Nyquist frequency and thus does not capture the "M" and "W" shapes.

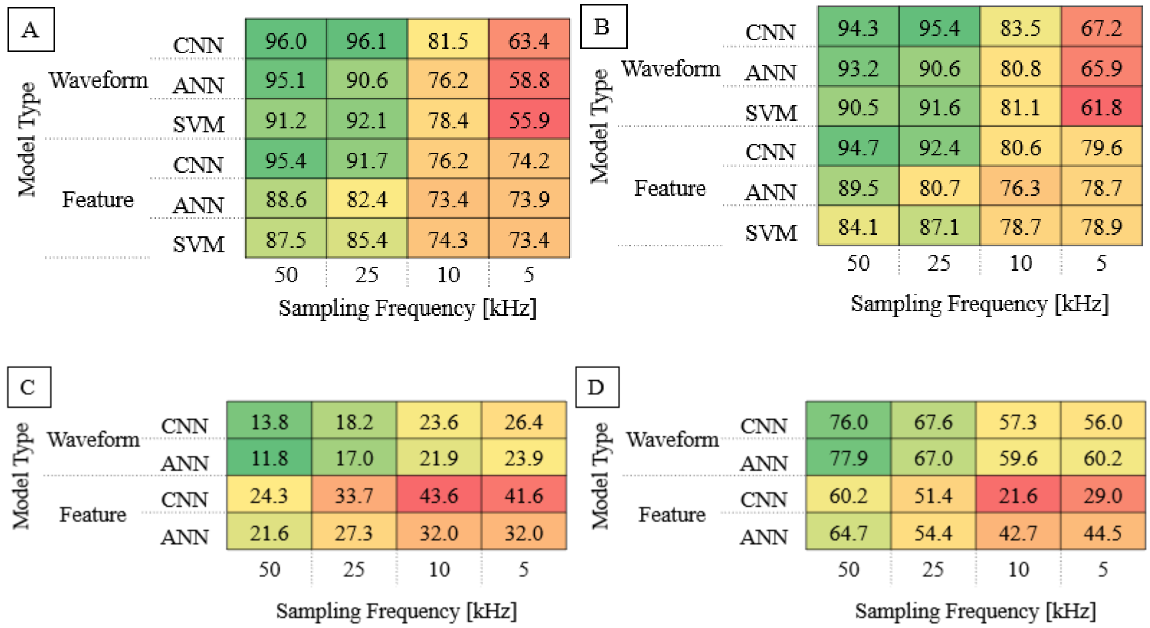

Figure 33. Effect of sampling frequency on the classification (top) and regression (bottom) model via the (A) accuracy, (B, D) F1-score, and (C) MAE of each model

The large difference in performance was more easily observed in Figure 34. Though evidently, the reduction in sampling frequency affected the classification model much more so than the regression model. In the classification models, F1-scores curiously increased at 25 kHz before falling dramatically while scores for the regression model drop consistently between 25 and 10 kHz then plateau. Because of the results of the classification model, the features input might be recommended for systems that can only afford 5 kHz. This dramatic decrease aligns with Sokolovksy's work [46] in which he performed vibration fault classification using the MaFaulDa database and discovered that decreasing the sampling frequency from 50kHz to 1 kHz dramatically reduced the accuracy of his model (using logistic regression and wavelet inputs), particularly for the nominal and horizontal misalignment cases.

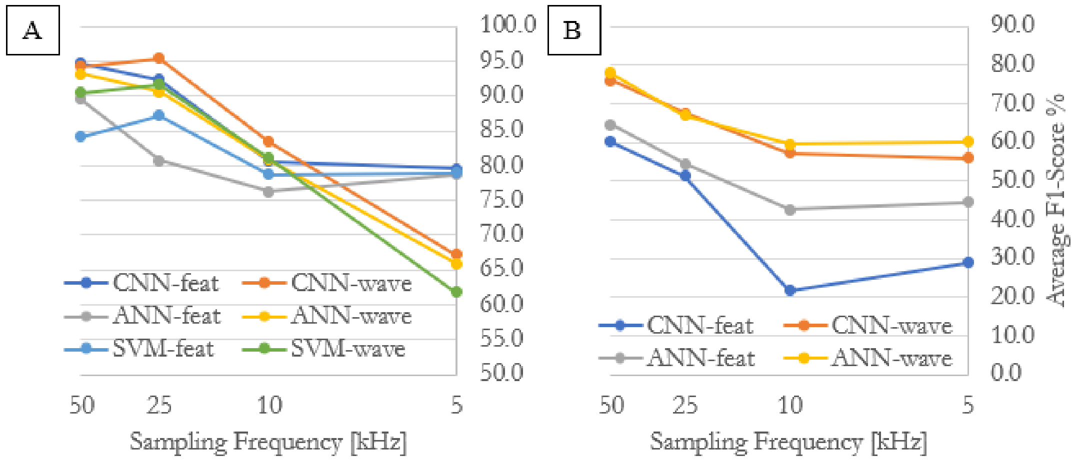

Figure 34. Effect of sampling frequency on F1-score of (A) classification and (B) regression models

5.2.3 Misalignment Amount Levels

While the regression models were able to predict somewhat consistently in 0.5 mm buckets of misalignment, some operators may not demand such granularity. Two different sets of levels were examined in Figure 35, where the regression results from the ANN waveform regression model were sorted in three and two bins respectively. In the three tiers, the low (L) bin is misalignment less than or equal to 0.7, the medium (M) bin is greater than 0.7 and less than or equal to 1.4, and the high (H) bin is anything greater than 1.4. For the two tier levels, low is classified as anything less than 1 mm and high is everything else. From the confusion matrices, it was clear that the regression model had difficulty predicting the higher levels of misalignment. This may be due to some class imbalance in the training set as there were a larger range of values on the lower side of the spectrum from the nominal case which had misalignment amounts exclusively of 0 mm.

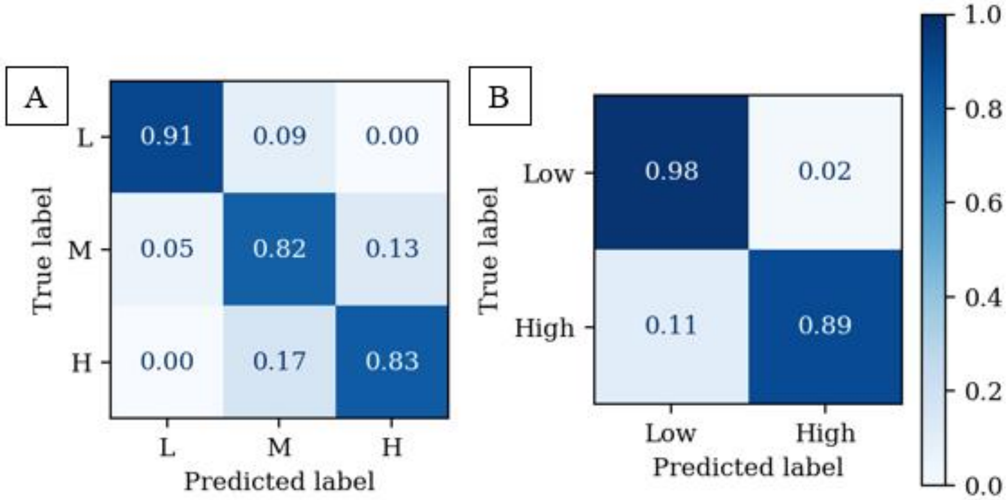

Figure 35. Levels of misalignment using the ANN regression waveform model for (A) three and (B) two levels

5.3 Recommendation

With the final results of the waveform and feature inputs models in Table 9, a few recommendations can be made for implementation of these models.

Table 9. Summary of final model results

	Model	Axes	Loss	Accuracy	F1-Score
Classification - Waveform	CNN	UT	16.5	95.1	93.9
Regression - Waveform	ANN	UA+UR+UT	16.8	N/A	71.4
Classification - Features	CNN	UT	15.9	94.9	94.5
Regression - Features	ANN	UA+UR+UT	27.5	N/A	59

The CNN model using the underhung tangential axis is the clear choice for classifying misalignment type at 95% accuracy. For both features and waveforms, it outperformed the ANN and SVM models. The other two algorithms were similar in performance, both ANN and SVM models being approximately 4% less accurate. Although the ANN models were slightly faster in computation time than the CNNs and the 5-fold convergence plots in Figure 15 and Figure 19 demonstrated that the training and validation results of the CNN diverge more so than the ANN, the relatively large difference in performance outweighed that consideration. This difference may be explained by examining the SHAP values in Figure 25 and Figure 26, which indicated the CNN is better at interpreting the "M" and "W" shapes in misalignment by identifying the peaks. This application is aligned with the idea that CNNs are suited for classification based on images [35, 36], as shown particularly for vibration data [37-40]. A potential benefit of using an SVM is that in the event of less available training data, it should be able to maintain its performance, but the SVM models took inarguably longer to compute, even longer than the ANN [27]. A drawback of the CNN is that it took longer during training; however, it was only 5% longer than an SVM and 42% longer than the ANN, and in the scope of 390 seconds for nearly 40,000 samples, it is inconsequential. Furthermore, in practice, the model will only be training intermittently – whenever this is an apparent change in the data set or after a fixed time interval set by the user.

The regression results are not as promising as the classification results; nevertheless, the ANN regression model using all three underhung axes with an MAE of 16.8% is recommended for generating the most accurate misalignment amounts. It was determined that all three axes were necessary at the cost of more sensors because the difference in performance of the ANN models between not just the single and triple axes but also the double and triple axes was rather significant at 8% and 12% respectively for waveform inputs. This was seen again using features, increasing 15% and 17%

respectively. The selection of the ANN over the CNN model was more difficult than the axis selection as it is only marginally better in numbers than the CNN by 0.3% less loss, 1.5% increase in F1-score, and 25% quicker training computation time. Although they are close in performance, this decision is backed by a few considerations. When looking back to the 5-fold validation results in Figure 16 and Figure 20, the ANN metrics converged in a much smoother curve with less of a change between the training and validation sets, meaning that it will likely be more successful in various environments across different types of machines. This is also supported by the box and whisker plots in Figure 18 and Figure 22 which exhibited less variation in the ANN results. Furthermore, many studies have demonstrated the effectives of the ANN [26-32].

In a lab environment, the waveforms inputs are certain to provide better results. However, in a situation where accelerometers have a sampling frequency of less than 10 kHz, the classification model would greatly benefit from using feature inputs even though the computation times are significantly longer. The use of features removes excess information and makes the model more generalizable as clearly shown by the effects of sampling frequency in Figure 34. In a similar use case to identify the presence of misalignment, Lee et al. [24] demonstrated that it was crucial to preprocess the data by using PCA, improving the raw data performance from 49.79% to PCA 98.8%. The implementation of features also supports the chosen CNN model. Among classification models, the CNN exhibited a lower decline in performance from noise and retained a higher F1-score with reduced sampling frequency even though the ANN and SVM feature models dropped less overall. As for regression in the context of a less idealistic environment, the ANN demonstrated its superior robustness by falling only 20% in F1-Score compared to the CNN which plunged 30% from 50 to 5 kHz. The ANN's steadfast performance against noise had also been repeatedly demonstrated by Zhao and others [35, 36].

The increased computation time of feature extraction does not have as much of an impact if the models are only being used to classify misaligned rotor systems as, ideally, a facility would not have many misalignment cases at once and each sample only takes 125 milliseconds using the CNN. On the other hand, the regression model should utilize the waveforms directly as it will be run more frequently to distinguish the severe and minor misalignment cases. In this high use application, the difference between 125 and 15 milliseconds for the ANN regression model is quite significant on top of the consideration that the results from waveform inputs drastically exceed those from features.

In summary, for classification of misalignment types, the CNN model using waveforms is recommended unless sensors with lower sampling frequencies are used, then features should be extracted. To predict the amount of misalignment, it is always recommended to use the ANN regression model with waveform inputs. These models using the time domain require essentially no knowledge of the machines or machine learning to implement and therefore should be easier to integrate into manufacturing facilities.

CHAPTER 6. CONCLUSION

In this thesis, machine learning methods were compared to develop a tool to better understand misalignment in rotor systems. Using accelerometer data from the MaFaulDa database, waveforms from various vertical and horizontal misalignment amounts ranging from 0 to 2 mm were input into a classification and regression model to classify the type and predict the amount of misalignment respectively. To achieve this, the waveforms were filtered and split into smaller sections containing two shaft periods such that “M” and “W” shapes which are common in misalignment could be clearly identified. A study to determine the minimum MaFaulDa data established that only the underhung bearing accelerometer was necessary and for classification, only the tangential axis needed to be used while for regression, all three axes were needed. Features were also extracted and tested against the filtered waveforms to identify the preferable input to the model. In the classification models, three main algorithms were investigated: ANN, CNN, and SVM. The results of the optimization demonstrated that a CNN is best used to classify the type of misalignment, lending to the idea that CNNs are able to identify patterns in images, namely the “M” and “W” shapes. For regression, ANN and CNN based models were compared, and it was found that the ANN model was able to predict misalignment amounts with greater accuracy. An additional study on the robustness of the models was conducted by adding in noise and reducing the sampling frequency, two phenomenon that may likely emerge in a large manufacturing facility. It was found that with the low pass filter, noise had very little effect on the output of either model although accuracy did decrease by around 3%. However, reducing the sampling rate to 5 kHz had a much more prominent effect and dropped the F1-Score of the classification model by nearly 30% using waveforms. Thus, if these models were to be used in a manufacturing facility limited by sensors with low sampling frequencies, features may be more practical even though they take much longer to compute. But if the accelerometers can obtain sampling frequencies of above 10 kHz, the filtered waveforms would produce more reliable results. Hence, easy to implement classification and regression models using the time domain have been developed to characterize misalignment and assist operators in executing more efficient CBM.

6.1.1 Limitations and Future Work

While the models do have over a of 95% categorical accuracy and 16% MAE with the MaFaulDa database, they should be benchmarked with other misalignment sets from various systems and sensors before implementing them in industry. This would help verify these models and increase the confidence in their generalizability. It would also be worthwhile to examine the effectiveness of these models on angular and combined misalignment which are also very common.

The regression model for predicting misalignment amount could certainly be improved as it was only accurate within 1 mm of misalignment. An improvement to the current model would be an implementation of regression balancing or checking for improvement when trained on datasets curated with more evenly distributed misalignment as the results clearly favor the higher misalignment amounts. Additionally, Kalkat [10] determined that increasing shaft rotation frequency yielded better results for ANN models, so it might of interest to observe the performance this regression model with respect to shaft frequency. Another approach to improving the utility of the misalignment amounts model is in a use case where extremely accurate predictions of misalignment are not necessary and tiers are acceptable, then perhaps a classification model would return improved performance as it has been demonstrated that using bins does yield better results. Additionally, the use of the frequency domain may be more applicable as it was determined that higher peaks in the FFT correlate to higher levels of misalignment [13]. This was implemented by Umbrajkaar et al. [29] who successfully predicted misalignment with an ANN with 0.02 mm increments although with a different data set. His work utilized SYM8 wavelets and required a more signal processing knowledge to implement, but it was able to achieve very accurate results with an average MSE of 2.28%. He also used three axes of accelerometer data, but it may be worthwhile to investigate an accurate method that is able to transform the accelerometer so that a model only requires one axis.

APPENDIX A. FEATURE SELECTION

Table 10. Extracted features from the UT axis for classification

Feature Name	Score
matrix_profile__feature_"75"__threshold_0.98	607.5
matrix_profile__feature_"max"__threshold_0.98	558.4
matrix_profile__feature_"min"__threshold_0.98	448.3
augmented_dickey_fuller__attr_"usedlag"__autolag_"AIC"	214.8
ar_coefficient__coeff_7__k_10	182.6
spkt_welch_density__coeff_8	107.2
change_quantiles__f_agg_"var"__isabs_True__qh_0.8__ql_0.4	93.4
agg_linear_trend__attr_"stderr"__chunk_len_5__f_agg_"var"	86.5
change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.2	84.7
agg_linear_trend__attr_"stderr"__chunk_len_50__f_agg_"var"	77.1
quantile__q_0.3	76.9
agg_linear_trend__attr_"intercept"__chunk_len_50__f_agg_"var"	75.1
change_quantiles__f_agg_"var"__isabs_True__qh_0.4__ql_0.2	73.8
quantile__q_0.7	72.8
quantile__q_0.4	69.4
cwt_coefficients__coeff_10__w_5__widths_(2, 5, 10, 20)	68.6
spkt_welch_density__coeff_2	68.5
change_quantiles__f_agg_"mean"__isabs_True__qh_0.6__ql_0.4	67.9
cwt_coefficients__coeff_4__w_10__widths_(2, 5, 10, 20)	67.3
quantile__q_0.1	66.1
cwt_coefficients__coeff_8__w_5__widths_(2, 5, 10, 20)	65.5
quantile__q_0.8	65.4
cwt_coefficients__coeff_14__w_2__widths_(2, 5, 10, 20)	63.2
cwt_coefficients__coeff_1__w_2__widths_(2, 5, 10, 20)	62.9
fft_coefficient__attr_"abs"__coeff_5	62.4
cwt_coefficients__coeff_12__w_2__widths_(2, 5, 10, 20)	61.9

APPENDIX B. HYPERPARAMETER OPTIMIZATION TABLES

Table 11. Hyperparameter optimization of ANN classification model using waveforms

		UT			UA+UT			UA+UR+UT		
		Loss	Acc	F1-Score	Loss	Acc	F1-Score	Loss	Acc	F1-Score
Dropout	0.05	26.4	91.7	90.4	36.8	86.8	86.1	44.8	84.8	86.3
	0.10	27.9	90.8	90.1	35.5	86.9	85.5	42.8	84.5	87.0
	0.15	27.3	91.1	89.8	37.3	86.3	85.1	39.8	85.6	87.3
Dense #	64	27.0	91.2	90.5	34.5	87.9	86.0	38.9	86.2	87.2
	128	27.8	90.8	90.0	36.3	86.8	85.7	42.1	85.0	87.1
	256	26.8	91.5	89.8	38.7	85.3	85.0	46.4	83.8	86.2

Table 12. Hyperparameter optimization of ANN regression model using waveforms

		UT		UA+UT		UA+UR+UT	
		MAE	F1-Score	MAE	F1-Score	MAE	F1-Score
Dropout	0.05	23.6	61.2	53.7	46.8	18.2	67.2
	0.10	23.1	61.0	24.5	56.5	18.3	68.5
	0.15	24.0	59.0	25.6	52.8	18.2	65.4
Dense #	64	25.3	57.3	25.7	53.8	19.5	65.3
	128	23.9	60.6	54.7	45.7	18.4	66.6
	256	21.5	63.2	23.4	56.7	16.8	69.3

Table 13. Hyperparameter optimization of ANN classification model using features

		UT			UA+UT			UA+UR+UT		
		Loss	Acc	F1-Score	Loss	Acc	F1-Score	Loss	Acc	F1-Score
Dropout	0.05	36.1	85.2	85.1	62.2	72.3	74.9	49.2	78.8	80.4
	0.1	37.1	84.5	85.1	59.0	73.9	74.1	49.5	78.6	80.2
	0.15	38.6	84.1	84.7	63.1	72.3	74.0	48.7	78.7	80.5
Dense #	64	34.6	86.2	85.0	63.6	71.7	74.4	47.6	79.7	80.3
	128	37.3	84.6	85.0	61.5	72.5	74.6	48.4	79.2	81.4
	256	39.9	83.1	84.9	59.3	74.4	74.1	51.4	77.3	79.4

Table 14. Hyperparameter optimization of ANN regression model using features

		UT - 25		UA+UT		UA+UR+UT	
		MAE	F1-Score	MAE	F1-Score	MAE	F1-Score
Dropout	0.05	34.0	47.2	37.4	41.1	26.9	58.8
	0.1	34.4	50.9	38.3	37.6	27.5	58.1
	0.15	35.4	51.2	40.4	34.5	28.9	56.4
Dense #	64	36.0	49.2	37.8	39.1	28.5	56.3
	128	34.2	50.7	40.4	38.9	27.4	58
	256	33.6	49.3	38.0	35.2	27.5	59

Table 15. Hyperparameter optimization of CNN classification model using waveforms

		UT			UA+UT			UA+UR+UT		
		Loss	Acc	F1-Score	Loss	Acc	F1-Score	Loss	Acc	F1-Score
Dropout	0	18.7	94.6	93.4	42.4	88.3	87.5	41.4	88.8	89.8
	0.1	18.2	94.7	92.7	37.6	89.4	88.0	36.9	88.9	90.0
	0.2	16.8	94.9	93.5	34.4	89.7	88.4	39.0	88.0	89.7
Dense #	128	20.1	93.8	92.7	40.5	87.5	87.1	38.4	88.1	89.6
	256	16.8	95.1	93.4	36.8	89.7	88.0	35.2	89.4	90.3
	512	17.0	95.2	93.5	37.1	90.2	88.7	43.7	88.2	89.6
Kernel	3	17.3	94.8	93.2	39.1	87.9	87.0	40.4	87.3	88.9
	7	19.0	94.4	92.7	39.7	88.9	87.7	38.8	88.7	89.9
	11	17.6	94.9	93.7	35.5	90.6	89.1	38.1	89.7	90.8

Table 16. Hyperparameter optimization of CNN regression model using waveforms

		CNN - regression					
		UT		UA+UT		UA+UR+UT	
		MAE	F1-Score	MAE	F1-Score	MAE	F1-Score
Dropout	0	25.3	54.3	36.1	48.6	20.0	63.2
	0.1	27.0	47.5	27.9	45.8	31.0	55.1
	0.2	29.0	44.1	29.7	41.0	32.2	51.5
Dense #	128	29.0	47.2	29.6	43.6	42.6	50.1
	256	26.6	48.6	27.4	47.6	20.5	59.5
	512	25.6	50.1	36.8	44.2	20.1	60.2
Kernel	3	27.6	48.5	29.9	42.6	23.0	55.1
	7	27.2	47.1	37.7	43.3	30.6	56.3
	11	26.4	50.3	26.2	49.5	29.7	58.3

Table 17. Hyperparameter optimization of CNN classification model using features

		UT			UA+UT			UA+UR+UT		
		Loss	Acc	F1-Score	Loss	Acc	F1-Score	Loss	Acc	F1-Score
Dropout	0	18.2	93.6	93.1	48.8	79.0	82.6	38.3	83.9	84.6
	0.1	18.5	93.4	93.2	49.6	78.4	82.0	43.9	81.3	83.1
	0.2	18.4	93.3	93.0	49.4	78.5	81.6	40.9	82.4	83.6
Dense #	128	19.8	92.7	92.5	50.0	78.4	81.2	41.9	82.2	83.5
	256	18.7	93.3	92.9	50.4	78.2	82.2	42.7	81.9	83.4
	512	16.7	94.3	93.9	47.4	79.3	82.9	38.6	83.6	84.4
Kernel	3	19.4	92.9	92.1	49.8	78.7	82.3	39.0	83.9	85.7
	7	17.8	93.8	93.7	49.3	78.4	82.1	39.4	83.3	84.7
	11	17.9	93.7	93.5	48.7	78.8	81.9	44.7	80.5	80.9

Table 18. Hyperparameter optimization of CNN regression model using features

		UT - 25	
		MAE	F1-Score
Dropout	0	65.6	44.0
	0.1	56.9	46.6
	0.2	29.6	55.4
Dense #	128	48.6	46.5
	256	47.0	51.6
	512	56.6	47.8
Kernel	3	56.7	44.1
	7	66.3	42.7
	11	29.1	59.2

Table 19. Hyperparameter optimization of SVM classification model using waveforms

		UT		UA+UT		UA+UR+UT	
		Acc	F1-Score	Acc	F1-Score	Acc	F1-Score
C-value	1	80.5	80.5	81.2	81.3	79.7	82.5
	10	87.0	87.0	88.7	87.7	86.8	87.8
	100	91.0	90.3	90.6	88.3	88.2	88.6

Table 20. Hyperparameter optimization of SVM classification model using features

		UT		UA+UT		UA+UR+UT	
		Acc	F1-Score	Acc	F1-Score	Acc	F1-Score
C-value	1	77.9	79.3	66.4	68.7	68.4	71.3
	10	83.5	84.2	67.8	71.1	73.2	75.8
	100	87.4	88.1	70.1	74.6	77.8	79.8

APPENDIX C. MODEL COMPARISON ANALYSIS

Table 21. All classification model metrics

				F1-Scores			
Input Type	Model Type	**Loss**	**Accuracy**	**Avg**	**H**	**N**	**V**
Waveform	CNN	21.3	96.0	94.3	93.2	94.0	95.7
	ANN	23.2	95.1	93.2	92.6	92.7	94.2
	SVM	-	91.2	90.5	89.1	90.4	91.9
Features	CNN	19.8	95.4	94.7	93.6	95.7	94.7
	ANN	29.0	88.6	89.5	86.1	91.8	90.7
	SVM	-	87.5	84.1	89.0	89.5	89.7

Table 22. All regression model metrics

			F1-Scores					
Input Type	Model Type	**MAE**	**Avg**	**0**	**0.05**	**0.1**	**0.15**	**0.2**
Waveform	CNN	13.8	76.0	87.3	76.1	71.3	69.3	76.2
	ANN	11.8	77.9	87.8	82.2	73.0	68.2	78.5
Features	CNN	24.3	60.2	82.5	61.8	54.3	52.0	50.3
	ANN	21.6	64.7	73.3	61.7	63.4	65.0	59.9

Table 23. Final classification model computation times for training

	Classification					
	Waveform			Features		
	CNN	ANN	SVM	CNN	ANN	SVM
# Axes	1	1	1	1	1	1
Data Extraction	125.6	125.6	125.6	127.0	127.0	127.0
Data Resampling	0.9	0.9	0.9	0.9	0.9	0.9
Feature Extraction	-	-	-	1062.8	1062.8	1062.8
Data Formatting	2.5	2.5	2.5	2.5	2.5	2.5
Model Creation	256.8	140.6	200.7	301.5	212.7	79.2
Model Evaluation	4.0	3.0	41.2	4.5	2.9	17.4
Total [s]	389.8	272.6	370.8	1499.2	1408.7	1289.8

Table 24. Final classification model computation times for testing

	Regression			
	Waveform		Features	
	CNN	ANN	CNN	ANN
# Axes	3	3	1	3
Data Extraction	137.4	137.4	126.1	175.8
Data Resampling	-	-	-	-
Feature Extraction	-	-	869.0	1030.8
Data Formatting	2.5	2.5	2.5	2.5
Model Creation	273.5	173.4	212.7	164.1
Model Evaluation	3.9	2.7	3.1	2.7
Total [s]	417.3	316.0	1213.4	1375.9

Table 25. Final regression model computation times for training

	Classification					
	Waveform			Features		
	CNN	ANN	SVM	CNN	ANN	SVM
# Axes	1	1	1	1	1	1
Data Extraction	12.9	12.9	12.9	13.1	13.1	13.1
Feature Extraction	-	-	-	109.5	109.5	109.5
Data Formatting	0.3	0.3	0.3	0.3	0.3	0.3
Model Evaluation	0.4	0.3	4.2	0.5	0.3	1.8
Total [ms]	13.6	13.5	17.4	123.3	123.1	124.6

Table 26. Final regression model computation times for testing

	Regression			
	Waveform		Features	
	CNN	ANN	CNN	ANN
# Axes	3	3	1	3
Data Extraction	14.2	14.2	13.0	18.1
Feature Extraction	-	-	89.5	106.2
Data Formatting	0.3	0.3	0.3	0.3
Model Evaluation	0.4	0.3	0.3	0.3
Total [ms]	14.8	14.7	103.1	124.8

Table 27. ANN hyperparameter effects on average computation time

		Training [s]	Testing [s]
Dropout	0	54.2	2.2
	0.1	60.8	2.4
	0.2	62.3	2.4
Dense #	64	56.4	2.3
	128	60.8	2.3
	256	60.2	2.3

Table 28. SVM hyperparameter effects on average computation time

		Training [s]	Testing [s]
C-value	0.001	510.9	99.1
	1	344.2	70.7
	100	366.1	60.2
Kernel	rbf	428.2	116.1
	poly	401.8	66.8
Poly Degree	0	472.0	88.0
	3	390.1	59.3
	4	359.2	59.2
	5	385.8	60.8

Table 29. CNN hyperparameter effects on average computation time

		Training [s]	Testing [s]
Dropout	0	188.7	5.2
	0.1	198.8	5.1
	0.2	203.9	5.5
Dense #	64	186.6	5.2
	128	197.2	5.3
	256	207.6	5.3
Kernel	3	176.7	4.8
	7	201.3	5.4
	11	213.5	5.5
Filter	32	151.8	4.5
	64	242.5	6.0

REFERENCES

[1] W. Biały and J. Ruzbarský, "Breakdown Cause and Effect Analysis. Case Study," Management Systems in Production Engineering, vol. 26, no. 2, pp. 83–87, Jun. 2018, doi: 10.1515/mspe-2018-0013.

[2] A. H. C. Tsang, "Condition-based maintenance 3 Condition-based maintenance: tools and decision making," Journal of Quality in Maintenance Engineering, vol. 1, no. 3, pp. 3–17, 1995.

[3] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," Computers and Industrial Engineering, vol. 63, no. 1, pp. 135–149, 2012, doi: 10.1016/j.cie.2012.02.002.

[4] S. Phogat and A. K. Gupta, "Expected maintenance waste reduction benefits after implementation of Just in Time (JIT) philosophy in maintenance (a statistical analysis)," Journal of Quality in Maintenance Engineering, vol. 25, no. 1, pp. 25–40, Mar. 2019, doi: 10.1108/JQME-03-2017-0020.

[5] S. Janasekaran and S. H. Lim, "Reduction of Non Added Value Activities During Machine Breakdown to Increase Overall Equipment Efficiency: Surface Mounting Technology Production Case Study," in Lecture Notes in Mechanical Engineering, 2020, pp. 51–56. doi: 10.1007/978-981-13-8297-0_7.

[6] G. A. Widyadana and H. M. Wee, "Optimal deteriorating items production inventory models with random machine breakdown and stochastic repair time," Applied Mathematical Modelling, vol. 35, no. 7, pp. 3495–3508, Jul. 2011, doi: 10.1016/j.apm.2011.01.006.

[7] S. R. Bognatz, "Alignment of critical and non critical machines," Orbit, vol. 4, pp. 23–25, 1995.

[8] G. N. D. S. Sudhakar and A. S. Sekhar, "Coupling Misalignment in Rotating Machines: Modelling, Effects and Monitoring," Noise & Vibration Worldwide, vol. 40, no. 1, pp. 17–39, 2009.

[9] D. L. Dewell and L. D. Mitchell, "Detection of a Misaligned Disk Coupling Using Spectrum Analysis," Journal of Vibration, Acoustics, Stress, and Reliability in Design, vol. 106, pp. 9–16, Jan. 1984, [Online]. Available: http://vibrationacoustics.asmedigitalcollection.asme.org/

[10] M. Kalkat, S. ¸ Yıldırım, and I. Uzmay, "Rotor Dynamics Analysis of Rotating Machine Systems Using Artificial Neural Networks," International Journal of Rotating Machinery, vol. 9, pp. 255–262, 2003, doi: 10.1080/10236210390202874.

[11] M. Xu and R. D. Marangoni, “Vibration Analysis of a Motor-Flexible-Coupling-Rotor System Subject to Misalignment and Unbalance Part I: Theoretical Model and Analysis,” Journal of Sound and Vibration, vol. 176, no. 5, pp. 663–679, 1994.

[12] M. Xu and R. D. Magaroni, “Vibration Analysis of a Motor-Flexible-Coupling-Rotor System Subject to Misalignment and Unbalance Part II: Experimental Validation,” Jounal of Sound and Vibration, vol. 176, no. 5, pp. 681–691, 1994.

[13] N. Wang and D. Jiang, “Vibration response characteristics of a dual-rotor with unbalance-misalignment coupling faults: Theoretical analysis and experimental study,” Mechanism and Machine Theory, vol. 125, pp. 207–219, Jul. 2018, doi: 10.1016/j.mechmachtheory.2018.03.009.

[14] V. Hariharan and PSS. Srinivasan, “Vibration Analysis of Misaligned Shaft-Ball Bearing System,” Indian Journal of Science and Technology, vol. 2, no. 9, 2009.

[15] T. H. Patel and A. K. Darpe, “Experimental investigations on vibration response of misaligned rotors,” Mechanical Systems and Signal Processing, vol. 23, no. 7, pp. 2236–2252, Oct. 2009, doi: 10.1016/j.ymssp.2009.04.004.

[16] Y.-S. Lee and C.-W. Lee, “Modelling and Vibration Analysis of Misaligned Rotor-Ball Bearing Systems,” Journal of Sound and Vibration, vol. 224, no. 1, pp. 17–32, 1999, [Online]. Available: http://www.idealibrary.comon

[17] J. K. Sinha and K. Elbhbah, “A future possibility of vibration based condition monitoring of rotating machines,” Mechanical Systems and Signal Processing, vol. 34, no. 1–2, pp. 231–240, Jan. 2013, doi: 10.1016/j.ymssp.2012.07.001.

[18] T. Y. Wu and Y. L. Chung, “Misalignment diagnosis of rotating machinery through vibration analysis via the hybrid EEMD and EMD approach,” Smart Materials and Structures, vol. 18, no. 9, 2009, doi: 10.1088/0964-1726/18/9/095004.

[19] A. K. Jalan and A. R. Mohanty, “Model based fault diagnosis of a rotor-bearing system for misalignment and unbalance under steady-state condition,” Journal of Sound and Vibration, vol. 327, no. 3–5, pp. 604–622, Nov. 2009, doi: 10.1016/j.jsv.2009.07.014.

[20] M. M. Tahir, A. Hussain, S. Badshah, A. Q. Khan, and N. Iqbal, “Classification of unbalance and misalignment faults in rotor using multi-axis time domain features,” Jan. 2017. doi: 10.1109/ICET.2016.7813273.

[21] N. Kumar, “Introduction to Support Vector Machines,” Marktechpost, Mar. 2021. https://www.marktechpost.com/2021/03/25/introduction-to-support-vector-machines-svms/ (accessed Mar. 29, 2022).

[22] A. Widodo and B. S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis,” Mechanical Systems and Signal Processing, vol. 21, no. 6, pp. 2560–2574, Aug. 2007, doi: 10.1016/j.ymssp.2006.12.007.

[23] Y. Xiao, N. Kang, Y. Hong, and G. Zhang, "Misalignment fault diagnosis of DFWT based on IEMD energy entropy and PSO-SVM," Entropy, vol. 19, no. 1, 2017, doi: 10.3390/e19010006.

[24] Y. E. Lee, B. K. Kim, J. H. Bae, and K. C. Kim, "Misalignment Detection of a Rotating Machine Shaft Using a Support Vector Machine Learning Algorithm," International Journal of Precision Engineering and Manufacturing, 2021, doi: 10.1007/s12541-020-00462-1.

[25] J. Zou, Y. Han, and S.-S. So, Artificial Neural Networks, vol. 458. Humana Press, 2008.

[26] G. N. Marichal, M. Artés, and J. C. García-Prada, "An intelligent system for faulty-bearing detection based on vibration spectra," JVC/Journal of Vibration and Control, vol. 17, no. 6, pp. 931–942, May 2011, doi: 10.1177/1077546310366264.

[27] J. J. Kuropatwinski, S. Jesse, J. W. Hines, A. Edmondson, and J. Carley, "Prediction Of Motor Misalignment Using Neural Networks," 1970. [Online]. Available: https://www.researchgate.net/publication/2717436

[28] L. B. Jack and A. K. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," Mechanical Systems and Signal Processing, vol. 16, no. 2–3, pp. 373–390, 2002, doi: 10.1006/mssp.2001.1454.

[29] A. M. Umbrajkaar, A. Krishnamoorthy, and R. B. Dhumale, "Vibration Analysis of Shaft Misalignment Using Machine Learning Approach under Variable Load Conditions," Shock and Vibration, vol. 2020, 2020, doi: 10.1155/2020/1650270.

[30] K. M. Saridakis, P. G. Nikolakopoulos, C. A. Papadopoulos, and A. J. Dentsoras, "Identification of wear and misalignment on journal bearings using artificial neural networks," in Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology, Jan. 2012, vol. 226, no. 1, pp. 46–56. doi: 10.1177/1350650111424237.

[31] A. A. de Lima et al., "On fault classification in rotating machines using fourier domain features and neural networks," 2013. doi: 10.1109/LASCAS.2013.6518984.

[32] C. C. Wang, Y. Kang, P. C. Shen, Y. P. Chang, and Y. L. Chung, "Applications of fault diagnosis in rotating machinery by using time series analysis with neural network," Expert Systems with Applications, vol. 37, no. 2, pp. 1696–1702, Mar. 2010, doi: 10.1016/j.eswa.2009.06.089.

[33] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, Mar. 2018, vol. 2018-January, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[34] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks ," Dec. 15, 2018. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed Mar. 29, 2022).

[35] C. Wang, J. Wang, and X. Zhang, "Automatic Radar Waveform Recognition Based on Time-Frequency Analysis and Convolutional Neural Network," in ICASSP, 2017, pp. 2437–2441.

[36] X. Zhang, Z. Yin, and Q. Dong, "An experimental study of axial misalignment effect on seizure load of journal bearings," Tribology International, vol. 131, pp. 476–487, Mar. 2019, doi: 10.1016/j.triboint.2018.11.014.

[37] W. Zhao, C. Hua, D. Dong, and H. Ouyang, "A novel method for identifying crack and shaft misalignment faults in rotor systems under noisy environments based on CNN," Sensors (Switzerland), vol. 19, no. 23, Dec. 2019, doi: 10.3390/s19235158.

[38] R. M. Souza, E. G. S. Nascimento, U. A. Miranda, W. J. D. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," Computers and Industrial Engineering, vol. 153, Mar. 2021, doi: 10.1016/j.cie.2020.107060.

[39] S. Guo, T. Yang, W. Gao, and C. Zhang, "A novel fault diagnosis method for rotating machinery based on a convolutional neural network," Sensors (Switzerland), vol. 18, no. 5, May 2018, doi: 10.3390/s18051429.

[40] X. Zhu et al., "Rotor fault diagnosis using a convolutional neural network with symmetrized dot pattern images," Measurement: Journal of the International Measurement Confederation, vol. 138, pp. 526–535, May 2019, doi: 10.1016/j.measurement.2019.02.022.

[41] F. Ribeiro, "MAFAULDA: Machinery Fault Database," Signals, Multimedia, and Telecommunications Laboratory. http://www02.smt.ufrj.br/~offshore/mfs/page_01.html (accessed Mar. 29, 2022).

[42] "SpectraQuest Machinery Fault Simulator," SpectraQuest. https://spectraquest.com/machinery-fault-simulator/details/mfs/ (accessed Mar. 29, 2022).

[43] F. Ribeiro, M. Marins, S. Netto, and E. Silva, "Rotating machinery fault diagnosis using similarity-based models," in XXXV Simposio Brasileiro de Telecomunicacoes e Precessamento de Sinais, 2017, pp. 277–281. doi: 10.14209/sbrt.2017.133.

[44] D. Pestana-Viana, R. Zambrano-Lopez, A. A. A. de Lima, T. de M. Prego, S. L. Netto, and E. A. B. da Silva, "The Influence of Feature Vector on the Classification of Mechanical Faults Using Neural Networks," in LASCAS 2016 - 7th IEEE Latin American Symposium on Circuits and Systems, Apr. 2016, pp. 115–118. doi: 10.1109/LASCAS.2016.7451023.

[45] M. Awais Ali, A. Ameen Bingamil, A. Jarndal, and I. Alsyouf, “The Influence of Handling Imbalance Classis on the Classification of Mechanical Faults Using Neural Networks,” 2019.

[46] A. Sokolovsky, D. Hare, and J. Mehnen, “Cost-Effective Vibration Analysis Through Data-Backed Pipeline Optimisation,” 2021.

[47] L. Zigang, J. Jun, and T. Zhui, “Non-linear vibration of an angular-misaligned rotor system with uncertain parameters,” JVC/Journal of Vibration and Control, vol. 22, no. 1, pp. 129–144, Jan. 2016, doi: 10.1177/1077546314525432.

[48] S. Ganeriwala, S. Patel, and H. A. Hartung, “The Truth Behind Misalignment Vibration Spectra of Rotating Machinery.” [Online]. Available: www.spectraquest.com

[49] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.

[50] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package),” Neurocomputing, vol. 307, pp. 72–77, Sep. 2018, doi: 10.1016/j.neucom.2018.03.067.

[51] H. Abdi, “The Kendall Rank Correlation Coefficient,” in Encyclopedia of Measurement and Statistics, Thousand Oaks, CA: Sage, 2007, pp. 508–510. [Online]. Available: http://www.utd.edu/

[52] Pedregosa et al., “Scikit-learn: Machine Learning in Python”, JMLR 12, 2011, pp. 2825-2830

[53] “Cross-validation: evaluating estimator performance,” Scikit-Learn.

[54] “Model 601A01 Platinum Low-Cost Industrial ICP® Accelerometer Specifications”, *PCB Piezotronics*. https://www.pcb.com/products?m=601A01

[55] Abadi, M et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 2015, Available: www.tensorflow.org.

[56] Lundberg, S. M., & Lee, S.-I. “A Unified Approach to Interpreting Model Predictions,” in 31st Conference on Neural Information Processing Systems (NIPS). 2017, https://github.com/slundberg/shap

[57] Vingelmann, P., & Fitzek, F. H. P., NVIDIA, *CUDA, release: 10.2.89*. 2020, https://developer.nvidia.com/cuda-toolkit