APPLICATION OF DOMAIN INVARIANT FEATURES IN CONDITION MONITORING OF ROLLING ELEMENT BEARINGS UNDER CHANGING OPERATING CONDITIONS

ESTHER WANGUI GITUKU

DOCTOR OF PHILOSOPHY

(Mechatronic Engineering)

JOMO KENYATTA UNIVERSITY

OF

AGRICULTURE AND TECHNOLOGY

2023

Application of Domain Invariant Features in Condition Monitoring of Rolling Element Bearings Under Changing Operating Conditions

Esther Wangui Gituku

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mechatronic Engineering of the Jomo Kenyatta University of Agriculture and Technology

2023

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signature..... Date...../...../.....

Esther Wangui Gituku

This thesis has been submitted for examination with our approval as the University Supervisors:

Signature..... Date...../...../.....

Dr.-Ing. James K. Kimotho

JKUAT, Kenya

Signature..... Date...../...../.....

Dr.-Ing. Jackson G. Njiri

JKUAT, Kenya

DEDICATION

For mum, dad and Karis.

ACKNOWLEDGEMENTS

I would first like to thank my family for their support in this long journey: I did not know it would take so long. I also acknowledge my supervisors Dr.-Ing. James Kimotho and Dr.-Ing. Jackson Njiri in a very special way. They gave me a second chance. I would also like to recognize the friendship of Ms. Lydia Ehaba who gave somewhere to sit and begin work on this research. I cannot forget the SoMMME family. Everyone one of you who supported me through some challenging times in the course of these studies; you hold a special place in my heart. I also acknowledge the members of the department of Mechatronics who went out of their way to enable me focus on this research both times; I thank you most sincerely.

I also acknowledge all the students who are undertaking their PhDs in our departments - you can do this.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CBM	Condition Based Monitoring
CORAL	Correlation Alignment
CNN	Convolutional Neural Network
CWRU	Case Western Reserve University
DA	Domain Adaptation
DAE	Denoising Auto Encoder
DANN	Domain Adversarial Neural Networks
DBN	Deep Belief Network
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FRB	Fuzzy Rule Based
HFRT	High Frequency Resonance Technique
IR	Inner Race
LSTM	Long Short Term Memory
ML	Machine Learning
MMD	Maximum Mean Discrepancy
OR	Outer Race
RCMFE	Refined Composite Multiscale Fuzzy Entropy
RKHS	Reproducing Kernel Hilbert Space
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
REB	Rolling Element Bearing
RMS	Root Mean Square
RMSE	Root Mean Square Error
TCN	Time Convolutional Network

ABSTRACT

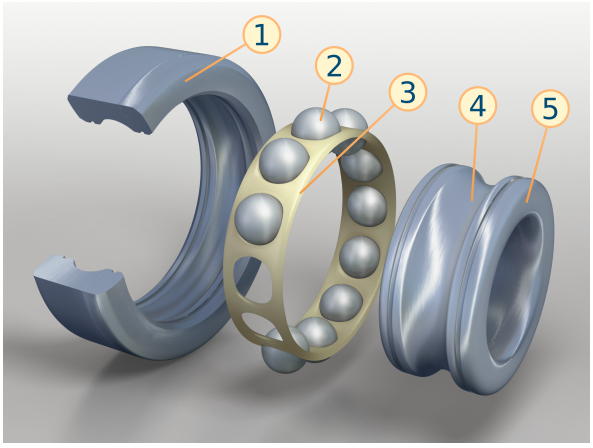
Data driven condition based monitoring of bearings has gained a lot popularity in recent times especially due to the fact that physics based models of equipment are difficult to formulate fully and accurately because of the complexity of machines. The low cost of sensors that has availed a large amount of data from operating machinery has further propelled the use of data-driven condition monitoring. Data driven models are heavily reliant on the domain of data they are trained on (source data). This means that they suffer in performance when applied to test data (target data) from a different domain. The most widely used techniques to counter this drop in performance are domain adaptation methods which seek to reduce the discrepancy between the two datasets. A key challenge with domain adaptation methods is the requirement for target data during training as a reference for the amount of discrepancy that exists. The other associated challenge is that the adaptation method has to be reconfigured or completely overhauled for each new test data because adaptation methods have varying capacity depending on the magnitude of the domain shift. This in turn means that models have to be retrained each time new test data are acquired. The goal of this work was to find and apply domain invariant features to the development of models so as to remove their dependence on target data but still be able to perform condition monitoring of REBs across domains. Publicly available datasets were used in the study: the Case Western Reserve and Ottawa Universities bearing datasets were used for diagnosis while the FEMTO-ST bearing dataset was used for prognosis. The Refined Composite Multi-scale Fuzzy Entropy Feature (RCMFE) was found to be a domain invariant feature for diagnosis. RCMFE had excellent fault detection ability, correctly detecting fault 100% of the time in different operating conditions. With the training data prepared such that each class of fault had a mixture of fault diameter sizes, RCMFE could easily differentiate inner race fault from ball and outer race faults with an average accuracy above 95%. However, the average accuracy for differentiating between ball and outer race fault fell to about 80%. With the training data arranged such that each fault type and size constituted a single class, RCMFE could isolate the three types of fault with an average accuracy of about 97%. Thus, this feature was able to achieve cross-domain diagnosis without domain adaptation. The hazard functions of kurtosis and shape factor were found to be trendable domain-invariant features for use in prognosis. Because the training data are full lifetime data, the challenge of low performance on test bearings with longer remaining useful life was overcome by supplementing the training data with its truncated versions. The combination of hazard functions and augmented training data enabled successful prognosis in changing operating conditions.

CHAPTER ONE

INTRODUCTION

1.1 Background

Rolling element bearings (REBs) are integral components in machinery where their primary task is transmitting axial and/or radial loads from the rotating parts to the structure as well as minimizing frictional losses. Ball bearings consist of inner and outer rings, the balls/rolling elements and a separator/cage as shown in Figure 1.1.



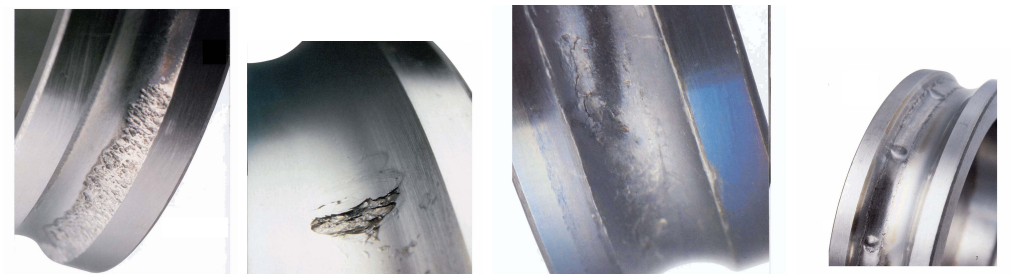
(a) Parts of an REB (1) Outer Ring (2) Rolling element (3) Separator (4) Inner raceway (5) Inner ring

(b) Assembled REB

Figure 1.1: Rolling Element Bearings

Bearings have been identified as being responsible for 30-50% of the failures experienced by machinery (Leturiondo, 2016; Zhou et al., 2020; Cao et al., 2020). Bearing defects can be categorized according to the damaged part i.e. inner race, outer race, rolling elements or cage fault. The defects may also be categorized by their causes e.g. faults due to assembly errors, due to lubrication issues or due to excessive operating conditions such as overloading (Cubillo et al., 2016). Defects can also be categorized according to their type i.e. distributed versus localized faults (Tandon & Choudhury, 1999). Distributed defects include surface roughness, waviness misaligned races and off-size rolling elements. These faults are caused by improper installation, abrasive wear and

manufacturing error. Analyzing the response of a monitoring signal due to distributed defects is important not only for condition monitoring of the bearing but also for quality inspection. Localized faults include spalls, cracks and pits in the rolling elements and/or in the races. Spalling is the most dominant localized failure and usually occurs when a fatigue crack begins below the surface and propagates upwards until a piece of metal breaks away to leave a small pit or spall. Figure 1.2 illustrates some of the common defects encountered in REBs (Barden Precision Engineering, N.D.).



(a) Spalling 1 **(b)** Spalling 2 **(c)** Lubricant failure **(d)** Brinelling

Figure 1.2: Faults that can affect REBs (Barden Precision Engineering, N.D.)

The spalling in Figure 1.2a is as a result of overloading which causes premature fatigue while that in Figure 1.2b is caused by normal fatigue due to gradual wear. Brinelling occurs when loads exceed the elastic limit of the material of inner ring and is identifiable as indentations on the inner race as shown in Figure 1.2d. Static overloading and/or impacts can cause brinelling as is the case when a hammer is used to remove or install bearings, when bearings are dropped or the outer ring is forcefully pressed to mount the bearing onto a shaft. Lubricant failure can be surmised when the balls and/or the races have a blue/brown discolouration as seen in Figure 1.2c. The result of this failure is excessive wear of the balls, races and cage which in turn causes overheating and at worst, eventual catastrophic failure. The Lubricant typically fails because of restricted flow and excessive temperatures that degrade its properties.

The maintenance task associated with detecting, isolating and identifying fault is known as diagnosis. On the other hand, prognosis may be defined as the ability to

perform early detection of the precursor of fault or incipient fault and furthermore have the capability of tracking the fault upto failure. The most commonly encountered prognostic measure is the Remaining Useful Life (RUL) and has to do with predicting how much time remains before failure occurs given the current machine condition or age and the historical operation profile. An alternative measure predicts the probability that a component functions without fault until a certain future time (e.g. the next inspection point) given its current condition and past profile (Jardin et al., 2006).

Prognosis is more useful from a maintenance perspective but it is usually a challenging process to get right as it depends on events yet to happen. Coupled with the fact that not all faults are predictable makes diagnosis the more studied process.

Just as with other components and equipment, REB maintenance has evolved from corrective to preventative to reliability-based and currently to Condition Based Maintenance (CBM) (Kimotho, 2016). Condition based maintenance relies on evaluation of continuously collected data to determine the state of components and make decisions on if/when replacements should be done through its diagnostic and prognostic procedures.

Both diagnostic and prognostic methods can be broadly classified into data-driven and physics-based (Ding, 2009). The former approach relies on training models where data is collected for healthy and faulty states to learn distinguishing characteristics. The mathematical models learned are abstract and devoid of any physical meaning in relation to the actual system. When applied to new data (testing data), the model is able to determine the component's health.

Physics-based methods utilize mathematical models derived from first principles of the actual physical system. If the mathematical model well estimates the physical system, the response features from the system and model agree. When faults occur in the actual system, the measurements of select response features from the model and system will differ i.e. result in a residual. When the residuals rise above a set threshold, fault is

detected (Ding, 2009). Physics-based models require knowledge of the different failure mechanisms that can occur to accurately construct the mathematical model. Provided the model and actual system agree, these methods are very accurate. The biggest challenge is the development of an adequate model especially if the system is complex. The model must also be validated because of the assumptions and linearizations made to make it tractable (Dawn et al., 2015).

Because state of the art machinery are quite complex and components can undergo multiple failure mechanisms simultaneously, sufficient physical models have become much more difficult to develop thus raising the popularity of data-driven models. The availability of credible and publicly available bearing data-sets have also propelled research in data based CBM (NASA, 2022b; Case Western Reserve University Western Bearing Data Center, 2018).

Although, the research of data-driven CBM of bearings is mature, many of the published works are based on the assumption that data on which the learned model will be applied (testing data) is drawn from the same domain as the data used to develop the model (training data) (Wei et al., 2014; J. Liu et al., 2017; Al-Raheem & Abdul-Karem, 2010; M. He & He, 2017; Qi et al., 2019; Kimotho & Sextro, 2014; Duoung et al., 2018). The domain is composed of data and its distribution.

In practice, the distribution and/or label space rarely remain the same between training and implementation phases. The discrepancy arises due factors such as changing operating conditions between the training and testing phases, testing on bearings from similar (but not the same) equipment, testing on data with different fault severity from training data and class imbalance where the labels in the test data are not equal to those in the training data. Class imbalance can occur when the test data is drawn from a newly installed machine which can only provide healthy data. Operating conditions change with the task at hand even for the same equipment: for instance, in a machine shop, operating conditions will vary depending on the properties of the material be-

ing worked. Another key cause of domain and/or task shift is the use of artificially generated data for successful training due to lack of sufficient data collected in actual operation. The shortage is usually due to the long lifetime of components nowadays but can also be caused by factors such as safety regulations. In sensitive applications such as in the operation of high speed trains or in aircraft, components are replaced after a set number of operating cycles and not due to aging or breakdown. It then becomes difficult to collect enough healthy and faulty data in actual deployment of equipment. Consequently, when training machine learning models, researchers are forced to fall back on the use of artificial data which can be generated in the required quantities. Unfortunately, bearing damage is a complex phenomenon with development of fatigue damage or damage caused by solid particles being randomly influenced (Lessmeier et al., 2016). Thus, true fault evolution remains largely unknown and production of artificial faults even at varying levels of severity may not wholly reflect real fault damage. It is then conceivable that the domain of artificial data used for training and that of testing data drawn from actual operation and on which the trained model is to be applied are going to be different.

Data driven models are inextricably linked to the underlying distribution of the training data and hence a model trained in one distribution and applied to data from another distribution will experience a drop of performance shaped by the amount of discrepancy. This is because the learning process optimizes performance of the model for the training data and data of similar characteristics i.e. data drawn from the same distribution. When the distribution changes, the accuracy of the trained model will fall unless steps are taken to in some way incorporate the characteristics of the new data space in the model. The domain of data encapsulates data and its distribution such that the source domain is the data space of the training data and the target domain is that of the data the model is to be applied to in the field.

Clearly, in order to promote the adoption of data-driven CBM models in the field,

there is need to consider the issue of domain and/or label space shift during their development. Currently, the most pursued method of dealing with this shift is that of transforming the source domain before training in order to maximize performance in the target domain in a process known as Domain Adaptation (DA) (Weiss et al., 2016; Zhuang et al., 2020). The goal of DA is to reduce the discrepancy between the two domains as much as possible such that the learned model will perform acceptably in the target domain. Unfortunately, the biggest drawback of DA methods is the requirement of at least some data from the target domain to train the model or update an already learned model. Several complications arise from this requirement: 1) A model can only be trained when target domain data is obtained. The training data cannot be used alone to create a stand-by model. 2) A new model must be learned for each target data e.g. for each new operating condition. This is because the amount and/or type of discrepancy is most likely going to be different for various target domains. 3) If the amount and/or type of discrepancy between distribution changes e.g. the divergence widens, a previously successful method may no longer be effective. 4) The initial amount of data from the target domain used for training may be too little to capture the actual distribution of the target domain accurately. For instance, in the early stages after installation, only healthy data may be available from a new machine. The limited data will not represent the target distribution fully and the learned model will still perform poorly in the target domain.

In this research work, an alternative avenue of tackling domain shift will be explored- using domain invariant features. Recent works have shown that models trained with either conventional handcrafted features or features autonomously learned from deep networks do not perform well in domains other than the training domain. A question still remains as to whether there are features that have been previously used for various diagnosis and/or prognosis tasks (not necessarily bearings related) that have hinted at the capability of being domain invariant - even though they are yet to be explored for the same. The domain invariance may be suggested by their definition or inadvertent

exposition in literature. If they exist, such features would be greatly advantageous in the dealing with domain shift in CBM because training models will once again be a process independent of target data. As such, once a model is trained and fine- tuned using training data, it is ready for application to any relevant target domain without alteration. The possibility that of the existence of such features forms the thesis on which this research work is based.

1.2 Problem Statement

Data driven CBM algorithms suffer in performance when the domain of target data is different from that of training data. Practically, these domains are likely to vary due to factors such as changing operating conditions, different fault severity between the two datasets or testing on data from a different (even if similar) machine. Currently, the most popular trend in tackling this shift is to employ Domain Adaptation (DA) methods which seek to narrow down the discrepancy between the source and target data before training the model. In this way, the model is guaranteed to perform well in the target domain. Although there has been tremendous success with DA methods, they are not off the shelf i.e. the models have to be retrained each time a new target is encountered. Furthermore, even for the same target data, enough of it has to be available during training to accurately represent the domain gap between itself and the source data. Otherwise, re-training will still have to be redone.

This work seeks the development of off-the-shelf models that can perform diagnosis and prognosis of REBs across different domains. Such models need only be trained once and are thereafter applicable to any target data. This goal is only achievable if suitable domain-invariant features for both diagnosis and prognosis can be found.

1.3 Objectives

The main objective of this work was to find and apply domain invariant features for data-driven condition monitoring of rolling element bearings under changing operat-

ing conditions. In order to achieve this goal, the work was split into three specific objectives.

1. To investigate a variety of diagnostic features from literature for their ability to remain domain invariant in bearing diagnosis.

2. To examine a variety of prognostic features from literature for trendability as well as their ability to remain domain invariant in bearing prognosis.

3. To develop a framework for the implementation of diagnosis and prognosis using the domain invariant features.

1.4 Outline of Thesis

The rest of thesis is organized as follows. Chapter 2 presents a theoretical background of the concepts that are the basis of this work including how machine learning models learn as well as the definition and construction of entropy and hazard based features. Chapter 3 goes through all the procedures used for training both diagnosis and prognosis models including a description of the experimental data. In chapter 4, the results are presented and discussed. Finally, the concluding chapter summarizes the work as well as suggesting recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter is divided into three sections where first a review of the current practices in the CBM of REBs is given. Later, two domain invariant features for diagnosis and prognosis are described in detail. Finally, the algorithms to map these features to the relevant outputs are briefly discussed.

2.2 Review of Related Works

In this section, a quick recap of features that are common in CBM of REBs will be presented. A discussion on the learning process of data-based models will follow from which the impact of changing distributions (domain shift) on the performance of the model will be made clear.

2.2.1 Commonly used features

The two most common signals used to study the condition of REBs are vibration and acoustic emissions (El-Thalji & Jntunen, 2015) with vibration being more frequently used. Vibration and noise generation in bearings occurs due to varying compliance or defects. Varying compliance is a phenomenon caused by the periodic variation of the stiffness of the system. The position of radial rolling elements in the load zone changes with rotation of the bearing hence changing the stiffness of system periodically. The variation of stiffness produces vibrations known as varying compliance vibrations (Tandon & Choudhury, 1999). Presence of a defect significantly raises the baseline vibration level because interacting with a defect causes abrupt changes in the contact stresses at the interface resulting in a pulse of short duration which gives rise to vibration and noise in the structure.

Practically, it is difficult to find sufficient defective data in order to study the fault signature of bearings. The shortage is usually due to the long lifetime of components but can also be caused by factors such as safety regulations. In sensitive applications such as the operation of high speed trains or in aircraft, components are replaced after a set number of operating cycles and not due to aging or breakdown. It then becomes difficult to collect enough healthy and faulty data in actual deployment of equipment. Consequently, researchers are forced to fall back on the use of artificial data which can be generated in the required quantities. There are two main ways of generating artificial data. One option is to run the bearings till failure and study the vibration response throughout the lifetime. Alternatively, defects can be intentionally introduced by either acid etching, spark erosion, scratching, mechanical indentation etc. Afterwards, the vibration response of the faulty bearings is compared to that flawless bearings.

A feature vector carrying the health signature of the bearing is extracted from raw data and should be of the lowest dimension possible; however, raw vibration data is a time series of very high dimension. It cannot be easily used to directly determine health status because high-dimensional input spaces require much more data for training (Worden et al., 2011). Thus, some selected characteristics of the data are extracted as the low dimensional features to be analyzed in place of the raw data. Also, raw vibration data is noisy and this contributes to masking the fault signature.

The features used to study vibration response can be divided into time domain, frequency, time-frequency features and time-scale features. In the time domain, either summary statistics (e.g. Root Mean Square (RMS) and crest factor) (*Detecting faulty rolling–element bearings*, N.D.) or moments of data are used (Tandon & Choudhury, 1999). The optimal measurement of kurtosis (4^{th} moment) has been investigated in both the time and frequency domains (Dyer & Stewart, 1978; Antoni, 2006; Sikora, 2016, 2016; Gustafsson & Tallian, 1962).

The vibrations generated as a result of impulsive loading due to defects excite some structural resonances- the shock pulse method of diagnosis is based on this phenomenon and is widely used in the industry (Yang et al., 2014).

More sophisticated time domain features can be obtained from time series modeling. The idea is to propose a model structure for the unknown underlying dynamics of a system and estimate the model parameters from the available time data. In the context of damage detection, if the model parameters are determined when the bearing is healthy, change of model parameters at some future time may be indicative of damage. The determined model can be used to predict the system's response and these predictions should compare well with the actual system outputs i.e the difference between the predictions and the actual output (residuals) should have low variance. If the residuals suddenly have large variance, it is an indication that the dynamics of the system are changing as a result of some fault (Worden et al., 2011). System identification and parameter estimation can also be approached from a non-linear perspective justified by the fact that instantaneous variation in loading conditions, damping and friction make electric machines exhibit non-linearity (El-Thalji & Jantunen, 2015). Some non-linear parameter identification techniques include correlation dimension and complexity which is defined as the degree of regularity in a signal (Yan & R.X., 2007; Yan & Gao, 2004). Analysis of the complexity measure shows that initiation and the growth of faults is connected with the changes in the complexity value (Yan & Gao, 2004).

Frequency domain features have also been successfully exploited for CBM. The short pulses that occur as a result of striking defects may excite the natural frequencies of the bearing elements and housing structures. A diagnosis method that monitors an increase in the energy level of the high frequency range of the spectrum where the natural frequencies reside has been proposed (Yang et al., 1989). Measures such as the arithmetic mean, geometric mean and correlation can be used to quantify the difference

in spectra of good and faulty bearings.

Each component in an REB has a characteristic frequency that can be determined by its dimensions and the speed of rotation. In theory, whenever there is fault in one of the components, there should be an increase in energy at the particular characteristic frequency. This increase in energy can be monitored as a diagnosis tool (Taylor, 1980).

Unfortunately, because the characteristic frequencies are in the low frequency range which is contaminated by vibrations from other sources, signal processing tools such as envelop detection and the High Frequency Resonance Technique (HFRT) have been used to extract the characteristic defect frequency (Prasad et al., 1985; Voronkin et al., 1988; Segla et al., 2012; Mishra et al., 2021; Antoni & Randall, 2006; Chatterton et al., 2014; Barszcz & Jablonski, 2011).

Another common implementation of frequency features is to transform the time signal into the frequency domain and analyse the full spectrum using Fourier analysis. The idea behind the analysis is that any function including the function describing the vibration time series data can be expressed as a sum of sines and cosines i.e. a change of basis to trigonometric coordinates composed of orthogonal sines and cosines. Once the conversion is done, only the frequencies that have significant coefficients are picked and these fewer frequencies form a low dimensional representation of the original signal. If at a future time the Fourier analysis of the signal yields different dominant spectral components, it would be an indication that the system has changed e.g. acquired a fault. The frequency components are found by taking the Discrete Fourier Transform (DFT) of a sampled signal; the equivalent for a continuous signal is the Fast Fourier Transform (FFT) (Worden et al., 2011).

Time-frequency methods such as the Short-Time-Fourier-Transform, Gabor transform and wavelet analysis aim to address the weaknesses of extracting features purely from the time or frequency domains (Kutz, 2013): the time domain ignores all frequency information while the frequency domain loses time localization.

Generally, data gathered from condition monitoring sensors is usually noisy and it is beneficial to try and remove as much of the noise as possible and retain the part of the signal carrying the desired information. A common method of denoising are averaging i.e. passing the data through smoothing filters such as the Weiner and Savitzky-Golay filters. Another popular method is that of decomposing the data into wavelet coefficients and truncating off the high level coefficients which are associated with noise or reducing the wavelet components that are above a threshold by a certain amount (Worden et al., 2011). Another important signal processing step is normalization of data which removes the effect of different scales between sensor measurements. The removal of outliers should also be considered as they can bias the model learned from data. However, their removal should be approached carefully because an outlier could be the a sign that the system is entering another state.

In machine learning, the commonly used tools for mapping the input feature vector $\boldsymbol{x}$ into the output $\boldsymbol{y}$ include artificial neural networks, convolutional neural networks, support vector machines and fuzzy logic networks (Jiang & Wang, 2018; Sadhu et al., 2017; Zhiqiang et al., 2017; Al-Raheem & Abdul-Karem, 2010; M. He & He, 2017; Hoang & Kang, 2018). There are extensions to these tools such as the Long Short Term Memory (LSTM) neural network and Recurrent Neural Network (RNN) that are especially suited to prognosis as they consider the historical dependence of the current data point to the previous one (Malhi et al., 2011; Yuan et al., 2016; Zhao et al., 2016).

2.2.2 Machine Learning

Machine learning (ML) aims to solve prediction problems where the mapping of inputs to outputs cannot be easily or completely formulated as a concise mathematical function. If the outputs are discrete, the prediction task is known as classification; if the outputs are continuous, then it is a regression task while if the output is a probability, the task is that of probability estimation. Machine learning models makes use of data

collected in pairs of inputs and outputs to learn the underlying structure that relates the inputs to outputs. Usually, the models are not directly trained with raw data as it is often very noisy and may result in an ML model with numerous parameters to train. It is therefore commonplace for experts in a particular field to extract relevant descriptive features from the raw data and use these as inputs to train the ML model.

During training, the probability of a class $\pi_j = Pr(y = j)$ and the marginal distribution of data in the class $p_j(\boldsymbol{x})$ are available for each class j, and this information comprises the learned model. The probability of any pair of input and output $(\boldsymbol{x}, y)$ i.e. $Pr(\boldsymbol{x}, y)$ is their overall joint distribution computed as the probability of the label $Pr(y)$ multiplied by the marginal probability of seeing the data point $\boldsymbol{x}$ under that label i.e. $Pr(\boldsymbol{x}|y)$. But $Pr(y) = \pi_y$ and $Pr(\boldsymbol{x}|y) = p_y(\boldsymbol{x})$. Concisely,

$$Pr(\boldsymbol{x}, y) = Pr(y)Pr(\boldsymbol{x}|y) = \pi_y p_y(x). \tag{2.1}$$

The categorization of a new test point $\boldsymbol{x}_{te}$ amounts to finding the label that results in the largest conditional probability $P(y|\boldsymbol{x}_{te})$. In the generative approach, the conditional probability is obtained from the joint probability using Bayes' rule which makes use of class priors $\Pr(y) = \pi_y$ and the individual class marginals $\Pr(\boldsymbol{x}|y) = \mathrm{P}_y(\boldsymbol{x})$. The new test point $(\boldsymbol{x}_{te})$, is assigned the label that maximizes the joint distribution $\Pr(\boldsymbol{x}_{te}, y)$ i.e. the label with the largest value out of $\pi_1 P_1(\boldsymbol{x}_{te})$, $\pi_2 P_2(\boldsymbol{x}_{te})$ and $\pi_3 P_3(\boldsymbol{x}_{te})$ is picked.

Therefore, the prediction ability of the model directly depends on how it learns the individual marginal distributions of the classes and class priors from the training data. The combination of data and its distribution is known as the domain of data.

In reality, the domain from which test data is drawn from is often different from the training domain. In condition monitoring of bearings, the equipment in which they are installed for instance in a mechanical workshop, are usually run in varying conditions (e.g. in terms load-torque and motor speed) depending on the material being machined.

This change of operating condition has already been shown to adversely affect the classification performance. Further, training data may be drawn from an existing machine but the test data comes from a newly installed machine. The domain of data from the two similar machines will be different. In image recognition, marginal distribution shift would be experienced when training images are hand-drawn but testing images are photos. In sentiment analysis, there will be a difference in the marginal distribution if a model is trained to classify sentiments from one topic/product but applied to a different topic/product. Apart from a change in the marginal distribution, the space of features itself can change. For instance, where an image classification model is trained with grayscale images but tested on images of color. Another case of dissimilar feature spaces is in natural language processing (NLP) where training is done in one language but the model applied to another language.

Domain change is not the only factor that can mismatch the distributions of training and test data. Label spaces can change between the two sets of data. For instance, in the condition monitoring of equipment, both healthy and faulty data may be available for training especially if the monitoring signals are being collected from existing equipment, but the test data will realistically only consist of healthy data especially in the case of new equipment. Thus the training data may have more labels than the test data. Conditional distribution of data ($P(y|x)$) can also change between the training and test sets. Such an occurrence can cause the same image to be classified differently in the training and testing spaces depending on some prior knowledge about the data e.g in the case where there is class imbalance in one of the domains.

Thus, a change in the domain or task between the training and testing domains will deteriorate the performance of machine learning models. Formally, the domain of data, $\mathcal{D}$, comprises the data/features and its distribution i.e. $\mathcal{D} = \{\boldsymbol{x}, P(\boldsymbol{x})\}$ while the task $\mathcal{T}$ consists of the labels and the prediction function learned from the data i.e. $\mathcal{T} = \{y, P(y|\boldsymbol{x})\}$. In conformity with existing literature, the data space from which

training data is obtained will be known as the source domain while that of the test data will be known as the target domain (Pan & Yang, 2009).

Although the research of data-driven CBM of bearings has been successful, much of the published works are based on the assumption that the domain and task remain the same between the training and testing sets (Wei et al., 2014; J. Liu et al., 2017; Al-Raheem & Abdul-Karem, 2010; M. He & He, 2017). In reality, this is seldom the case due to factors such as changing operating conditions, testing on bearings from similar equipment and class imbalance especially where the test data comes from a new machine currently producing only healthy data. Thus in order to make prediction models adaptable in the field, there is need to consider the issue of domain and/or task shift. The process of transforming the source domain to enable transfer of knowledge such that prediction performance in the target domain is maximized is known as domain adaptation (DA) (Weiss et al., 2016; Zhuang et al., 2020). The goal of DA is to bring the distributions of the source and target domains as close together as possible.

In domain adaptation literature, this transfer learning is categorized in several ways depending on the availability or lack of labels for the target data (Zhuang et al., 2020). Herein, source data is always labeled and supervised learning describes full availability of target labels while semi-supervised learning applies to the presence of a limited number of target labels. In unsupervised learning, no target labels are available. It is useful to bear in mind that in a practical industrial setup, the target labels are usually not available with the possible exception of new equipment where generated data are initially known to be healthy (semi-supervised learning).

In the CBM of REBs, the most common DA techniques can be categorized as either divergence or adversarial based with automatic generation of features in place of handcrafted features. This is because the definition of handcrafted features is fixed and the applied DA technique may fail to bring the distributions closer. In automatic generation, the feature extractor can update its weights during training to generate new

features with each run until the domain and/or conditional distribution of the source and target are similar.

In divergence based methods, the difference between the source and target domains is measured by some distance or metric. The definition of the distance is captured in an objective function that is then minimized by finding a transformed data space where divergence between the two distributions is minimal. In other words, during training, the weights of the feature extractor are updated to produce domain invariant features. One popular distance is the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) which defines the distance between two distributions as the average of the distances between all the moments of the two distributions in some embedded space (Maheshwari, 2021). In order to compute all the moments of a distribution, the MMD maps the feature space $\mathbf{X}$ to an infinite dimensional space known as a Reproducing Kernel Hilbert Space (RHKS). The RHKS is a vector space that is endowed with an inner vector product and allows use of the kernel trick such that the distance between distributions can be computed without computing their densities in some latent high dimensional space (Ng, 2018). In its original form, MMD assumes that only a marginal distribution shift exists between the source and target domains while the conditional distribution remains the same. However, it is notable, that both the source and a target dataset must be available to counter domain invariance by minimizing MMD.

Li et al. (W. Li et al., 2020) used a modified multi-kernel MMD loss to improve classification accuracy in changing load conditions with the Case Western Reserve University (CWRU) data. A convolutional neural network (CNN) was used as the feature extractor. Kang et al. (Kang et al., 2019) utilized a convolutional Gaussian-Bernouli deep belief network (DBN) to extract features from frequency-domain amplitudes the CWRU data in changing load conditions. This feature extraction method was found to be moderately beneficial in countering the effect changing loads instead of only using a DBN. Domain adaptation was achieved by incorporating two MMD losses -

the first one to reduce marginal distribution and the second one to reduce the conditional distribution in the individual classes of data in what was called joint distribution adaptation. Often, in deep networks, the feature extractor is a CNN and the classifier consists of a few fully connected layers and it is in the last of these latter layers that MMD is applied. Smith and Shibatani (G. Smith & Shibatani, 2020) applied MMD to all layers from the CNN feature extractor to the fully connected layers of the classifier when classifying gearbox faults in changing operating speeds. Che et al. (Che et al., 2019) also use make use of multi-kernel MMD in addition to a model based technique to fine tune a classifier initially trained on source data to perform well on target data for changing loads in the CWRU data. In the study, the CWRU data was perturbed with white Gaussian noise to simulate the noisy nature of real world data. A denoising autoencoder (DAE) was used for dimension reduction and denoising while a CNN was used for automatic feature generation.

Some of shortcomings using MMD are that the computational cost increases exponentially with the number of training samples (J. He et al., 2021) and that as many target samples as there are source domain samples are required to accurately define discrepancy in the domains (Z.-H. Liu et al., 2019). Otherwise, the classifier will still not generalize well in the target space.

In place of MMD that measures discrepancy by checking infinite-order moments, the CORrelation ALignment (CORAL) distance measures discrepancy by only considering second order statistics i.e. covariances. He et al. (J. He et al., 2021) found that using the CORAL loss outperformed using MMD for the CWRU data.

The other popular method of dealing with domain and conditional distribution shifts in the CBM of REBs is deep domain adaptation using adversarial learning e.g. by using Domain Adversarial Neural Networks, (DANN). The architecture of a DANN is shown in Figure 2.1.

The feature extractor, composed of CNNs is used to generate descriptive features from

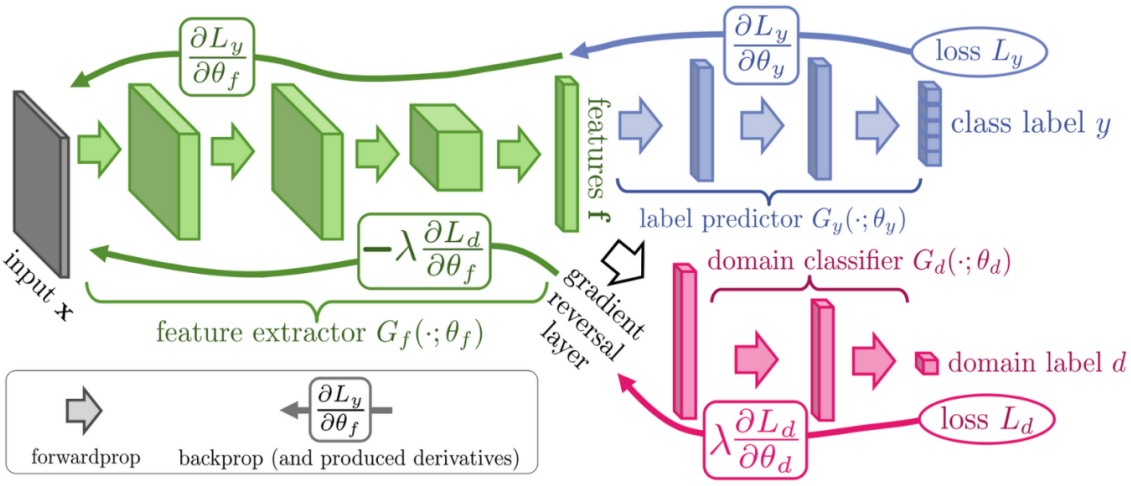


Figure 2.1: The architecture of a DANN (Ganin et al., 2016)

data instead of using handcrafted features. Both the source and target data pass through the feature extractor but only the source data is passed along to the label predictor keeping in mind that only the source data is labeled. The extracted features from both domains are also passed to a domain classifier whose job is to identify from which domain a set of features is drawn. The essence of a DANN is adversarial learning; minimize the classifier loss while maximizing the domain loss. The domain loss is maximized by reversing the gradients from the domain classification section during back-propagation by adding a gradient reversal layer as shown in Figure 2.1. The gradients from the classification layer are back-propagated with their original signs in the usual manner. The desired effect is that the feature extractor produces features that are good for label classification but indistinguishable in terms of domain of origin. Liu et al. (Z.-H. Liu et al., 2019) successfully utilized a DANN for CWRU data in changing operating conditions. Wang, Liu and Zhao (X. Wang et al., 2020) used a DANN to tackle the harder distribution shift problem where the source and target data are drawn from different machines. In the study, the source and target data were obtained from several different datasets i.e. the CWRU data, the IMS data (NASA, 2022a) and JiangNan University. By incorporating batch nuclear-norm maximization regularization, the authors were able to handle class imbalance and the issue of few

available target samples which causes violation of low density assumption of semi-supervised learning (Chapelle & Zien, 2005). Mao et al. (Mao et al., 2021) utilized a Time Convolutional Network (TCN) and a DANN to mine representative features and identify incipient fault in bearings for online application using lifetime data. da Costa et al. (da Costa et al., 2019) used a Long-Short Term Neural Network (LSTM) to extract features and a DANN to predict the remaining useful life of bearings using the FEMTO lifetime data (NASA, 2012) resulting in lower RMSE scores than when not using any DA method. Zao and Liu (Zao & Liu, 2022) applied a CNN for feature extraction and a DANN for RUL prediction with the FEMTO data and similarly recorded lower RMSE than without using any DA technique. Ragab et al. (Ragab et al., 2021) used a contrastive loss in addition to the usual adversarial loss term to ensure that target specific information was preserved thus improving performance over the normal adversarial training in estimating RUL for the C-MAPPS engine dataset (NASA, 2020).

A particular challenge with DANNs is that they can be harder to train than standard neural networks because of the two sets of gradients with reverse signs. Several techniques such as replacing the maximized part in the objective function with its dual formulation may be used to stabilize the training process (Kouw & Loog, 2021). It is to be noted that DANN still requires the presence of target data during training.

Use of MMD and adversarial learning are by no means the only DA approaches in the CBM of REBs. If the source and target domains share a common subspace but contain domain specific noise, a method known as subspace alignment may be implemented for DA. A specific number of principal components from each domain are computed and then a linear map that aligns the source to target components is found. A classifier is then trained on the transformed source data. Zhang et al. (B. Zhang et al., 2017) used subspace alignment for the CWRU data.

Zhang et al (R. Zhang et al., 2017) successfully applied parameter transfer to CWRU

data by utilizing a few labeled target data during training when label spaces in the source and target are not the same i.e. different number of classes in the two domains. Other documented DA methods are extensively employed in the fields of image recognition and NLP (Pan & Yang, 2009; Weiss et al., 2016; Zhuang et al., 2020).

From the foregoing literature on DA in data-based CBM of REBs, one of the most noticeable differences with initial machine learning methods is that automatic feature generation using either CNNs or autoencoders for diagnosis and LSTMs for regression is preferred over handcrafted features. The reasoning here is obvious - in autonomous generation, new features are re-created at to promote domain invariance by updating the weights of the feature extractor during training. On the hand, the possibility of re-defining hand-crafted features is not available and the only remaining degree of freedom looking for mapping of the features where the difference in distribution is reduced; such a space may not exist especially if the discrepancy is large.

Traditional handcrafted features in the time, frequency and time-frequency domains have been shown not perform well in domain and/or conditional distribution shift (Zhou et al., 2020). However, the use of automated feature extraction implies that the features used are specific to a particular configuration of source and target data i.e. depending on the amount of discrepancy which can differ even if the same machine is being monitored. For instance, in the foregoing literature using CWRU data, the data is arranged such that the data from one loading condition e.g. condition 1 forms the training/source data while any of the other three remaining conditions form the target data (Z.-H. Liu et al., 2019; X. Wang et al., 2020). In some cases, only fault size is considered (Z.-H. Liu et al., 2019; X. Wang et al., 2020) and in other cases all the different fault sizes as well as the normal samples each form a label for each operating condition (Kang et al., 2019).

In regard to prognostics, the FEMTO-ST dataset (NASA, 2012), most of the research is done on only one operating condition. Qi et al.(Qi et al., 2019) propose a novel spectral

based entropy condition indicator and use a particle filter to estimate the parameters of an exponential model used to fit the indicator. The RUL is estimated at each time step by systematically progressing the condition indicator to a threshold marking end of life using the exponential model. The authors were able to get predictions within ±20% error of the actual RUL for the bearings studied. However, the study only considered data from a single operating condition of the FEMTO data (condition 1).

Kimotho and Sextro(Kimotho & Sextro, 2014) used auto-correlation to improve the trendability of commonly employed time, frequency time-frequency features for use in prognostics. An Extreme Learning Machine (ELM) network was then used to select the best features to use RUL prediction which gave very good performance on the FEMTO data with less than 9% absolute error. However, only data from operating condition 1 was used for both training and testing.

Duong et al. (Duoung et al., 2018) derive a novel RMS based indicator with high monotonicity, robustness and trendability. Although the feature outperforms statistical model based prediction tools in RUL prediction accuracy, it was only deployed for the first operating condition in the FEMTO dataset.

Soualhi et al. (Soualhi et al., 2014) used the Hilbert Huang Transform to create features which were used not only for prognostics but also for detecting the faulty component. The features were highly accurate in diagnosing the fault as well as prognosis using one-time step ahead prediction. However, the features were only applied to data from operation condition 1 in the FEMTO dataset.

There are a few authors who have tackled domain shift using FEMTO-ST data. Hinchi and Tkiouat (Hinchi & Tkiouat, 2018) developed used a convolutional neural network to extract features and a long-short-term-memory network to predict RUL. All the six datasets for training (Table 3.3) were used for learning and the rest of the data used for testing. The performance of the model was moderate with an average absolute error of 46%.

From the foregoing literature,where cross-domain condition monitoring has been done, it was mostly accomplished using deep learned features and domain adaptation methods which require target data to be present during the training process. This is turn means that the data driven models have to be re-trained each time new target data is encountered - a process which is usually complex to set up and may additionally be computationally expensive.

2.3 Domain Invariant Features

In this section, two features selected from literature for their domain-invariability potential are discussed in detail. These features behave similarly across domains enabling mapping from inputs to outputs of CBM models without requiring domain adaptation.

2.3.1 Entropy for diagnosis

Monitoring signals collected from rotating machinery have non-linear characteristics due to instantaneous variations in loading conditions, friction and damping (El-Thalji & Jantunen, 2015). As such, non linear models can be fitted to the time series data and the parameters such as entropy used as features for CBM. As a bearing deteriorates, the vibration signal will increase in amplitude most of which will constitute an elevated noise floor. As such, the signal's regularity (complexity) will decrease thus increasing its entropy value (Yan & R.X., 2007).

In this section, entropy is justified as a good candidate for domain invariance. A short example of the computation of entropy will be presented later and will illustrate the feature's independence of the distribution of data. It has been used for the analysis of biomedical signals (Pincus et al., 1991; Azami et al., 2017).

The randomness/regularity of a series can be quantified by the amount of repeated patterns present in the data or in other words how well the data can be compressed; if it is composed of repeating patterns it is more compressible. Entropy of a time series

is the probability of the amount of surprise that is revealed by the data. The more random a series is, the higher its entropy.

Approximate entropy (ApEn) as proposed by Pincus (Pincus, 1995; Pincus et al., 1991; Pincus & Goldberger, 1994) is able to quantify the regularity of practical data which is characterized by noise and finiteness by checking for presence of vectors of repeating patterns. ApEn breaks down a time series into $m-$dimensional template vectors where the elements are composed of m consecutive points in the series. Each template is compared with all the other vectors (including itself) to check if the distance between all corresponding elements is within some tolerance r; all vectors that satisfy the distance criteria are labeled as "possible" vectors for the particular template (Delgado-Bonal & Marshak, 2019).

Next, the same series is divided into vectors of dimension $m+1$ and for each template vector, the proximity of each of its "possible" vectors is confirmed by computing the distance between the $(m+1)^{th}$ elements of the template vector and each of its "possible" vectors. If the distance between the $(m+1)^{th}$ elements is still less than r, then, the particular possible vector is now labeled a "match".

The steps of ApEn for a fixed m and r are enumerated as follows (Pincus, 1995).

a. From a time series U with N data points $(u(1), u(2), u(3), ..., u(N))$,

   create m-dimensional vectors $\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(N-m+1)$ where

   each $\boldsymbol{x}(i)$ is defined as $\boldsymbol{x}(i) = [u(i), u(i+1), u(i+2), ..., u(i+m-1)]$.

b. Compute the Chebyshev distance between vectors $\boldsymbol{x}(i)$ and $\boldsymbol{x}(j)$ i.e.

$$d[\boldsymbol{x}(i), \boldsymbol{x}(j)] = max_{k=1,2,...,m}(|u(i+k-1) - u(j+k-1)|). \tag{2.2}$$

c. Compute the sum $C_i^m(r)$ using the "possibles" for $\boldsymbol{x}(i)$

$$C_i^m(r) = \frac{1}{N-m+1} \sum_{j=1}^{N-m+1} \text{number of times that } d[\boldsymbol{x}(i), \boldsymbol{x}(j)] \leq r. \tag{2.3}$$

d. Compute the normalized sum $\phi_i^m(r)$ using the logarithm of $C_i^m(r)$ as

$$\phi^m(r) = \frac{1}{N-m+1} \sum_{j=1}^{N-m+1} \log C_i^m(r) \tag{2.4}$$

e. Repeat steps a to d for template vectors of dimension $m+1$ to compute $\phi^{m+1}(r)$. For each template, only its "possible" vectors are compared. Therefore the quantity $C_i^{m+1}(r)$ is the sum of all the "matches" for the template vector.

f. Finally, compute ApEN as

$$\begin{aligned} \text{ApEN}(m, r, N) &= \phi^m(r) - \phi^{m+1}(r) && (2.5) \\ &= \frac{1}{N-m+1} \sum_{j=1}^{N-m+1} \log C_i^m(r) - \frac{1}{N-m} \sum_{j=1}^{N-m} \log C_i^{m+1}(r) && (2.6) \end{aligned}$$

Since $\phi_i^m(r)$ and $\phi_i^{m+1}(r)$ are logarithmic quantities, the negative value of ApEn can also be interpreted as a normalized sum of the ratios of "matches" to "possibles" for all vectors $\boldsymbol{x}(i)$. The ratio is the conditional probability that vectors $\boldsymbol{x}(i)$ and $\boldsymbol{x}(j)$ which are close in the vector space $\Re^m$, remain close in the $\Re^{m+1}$ vector space by satisfying the distance criterion.

The concept of ApEN is intuitively understood by observing that steps a to e are very similar to the steps of reconstructing the phase space of a deterministic, stationary dynamical system using Takens' embedding theory (Takens, 1981). Consider a dynamical system fully represented by k state variables such that at any point in time the state of the system is represented by the vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_k,]$ in the $k-$dimensional state space. The states $\boldsymbol{x}$ are also known as phase points such that the state space is

also called the phase-space. The variation of the states in time is depicted by motion along some path in the state space known as the phase trajectory. This path indicates the long-term behaviour/tendency of the system.

Often it is not possible to observe $\boldsymbol{x}$ in its entirety for practical systems, and hence an observation function g only measures a single state variable i.e. $g : R^k \rightarrow R^1$ is assumed. Let the scalar value measured be denoted by $g(\boldsymbol{x})$ where $g(\cdot)$ is considered a projection of the phase space trajectory onto a single coordinate of the $k-$dimensional space (Navarrete, 2018). Takens (Takens, 1981) proved that it is possible to use the delay vector given in equation 2.7

$$v(\boldsymbol{x}, \tau, d_e) = [g(\boldsymbol{x})_t, g(\boldsymbol{x})_{t-\tau}, g(\boldsymbol{x})_{t-2\tau}, ..., g(\boldsymbol{x})_{t-(d_e-1)\tau}, g(\boldsymbol{x})_{t-d_e\tau}] \tag{2.7}$$

as a surrogate for the phase point $\boldsymbol{x}$ and hence reconstruct a topologically similar phase-space for a suitably chosen time delay τ and embedding dimension d_e .

Indeed, one criteria used in selecting a good embedding dimension d_e is that of embedding the scalar time series $g(\boldsymbol{x}_t)$ in increasing higher dimensions and checking at each stage if vectors that are nearest neighbors in $\Re^{d_e}$ are still close in $\Re^{d_e+1}$. Assuming there is a large amount of data available (i.e. a sufficiently long time series), the Euclidean norm between two nearest vectors in the reconstructed space should be small (Small, 2005). Points that are close due to system dynamics and not due to an inaccurate embedding dimension should separate slowly and hence remain close in $\Re^{d_e+1}$.

Thus, ApEN represents the dynamics of a system in a low dimensional space (m is usually equal to 2 or 3) and computes the long-term trajectory by plotting the delay vectors $\boldsymbol{x}_i$. Then, ApEN measures how much the shape of the trajectory is distorted in m versus $m+1$-dimensional space due to template vectors losing neighbours in the latter space.

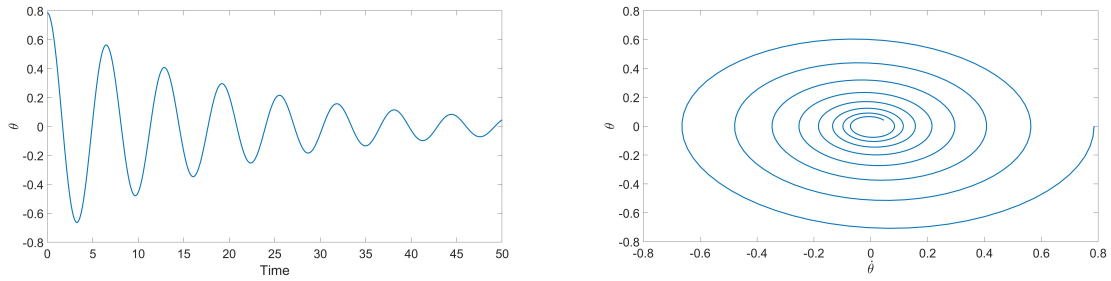
A depiction of the "possibles" and "matches" for two well known as systems (the simple

pendulum and Lorenz attractor) are now presented.

The linearized dynamics of a simple pendulum are given in equation 2.8 (Zill & Cullen, 2009)

$$\ddot{\theta} + \frac{g}{l}\theta = 0 \tag{2.8}$$

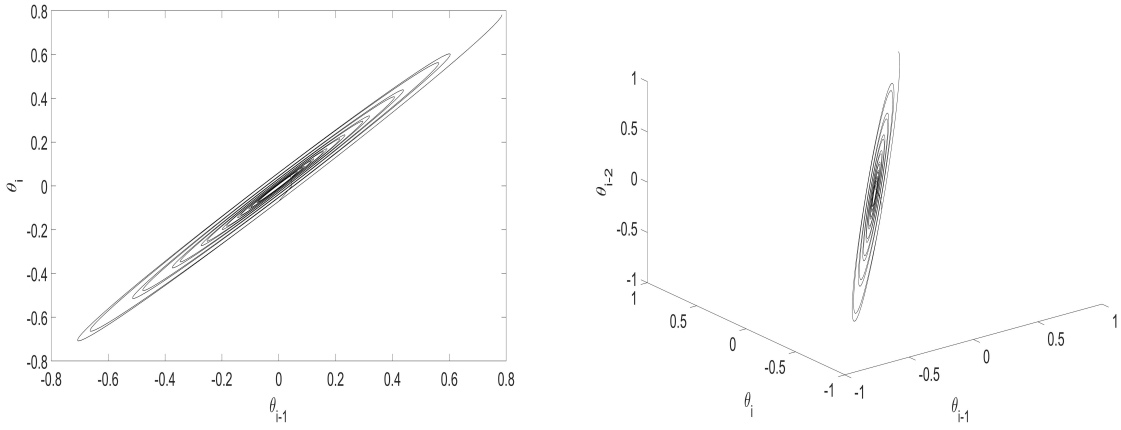
Figure 2.2 shows the angular position of a pendulum with time and its phase space (θ vs $\dot{\theta}/\omega$).



(a) Angular position **(b)** Phase space

Figure 2.2: Dynamics of the Simple Pendulum

Figure 2.3 shown the vectors of ApEN in $m = 2$ and $m = 3$ dimensions for the simple pendulum.



(a) m=2 **(b)** m=3

Figure 2.3: Dynamics of the Simple Pendulum

It is clear that the reconstructions in Figure 2.3 closely mirror the phase space of Figure 2.2.

The Lorenz attractor is a more complicated non-linear system defined by the dynamics (Sparrow, 2012),

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{2.9}$$

The 2-D phase space with any pair of the coordinates x, y or z show that the system is characterized by switching between two lobes as seen in Figure 2.4 .

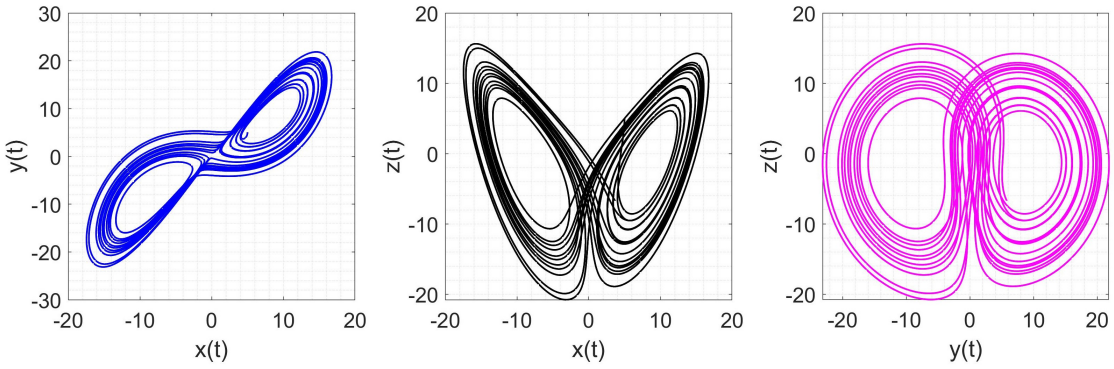
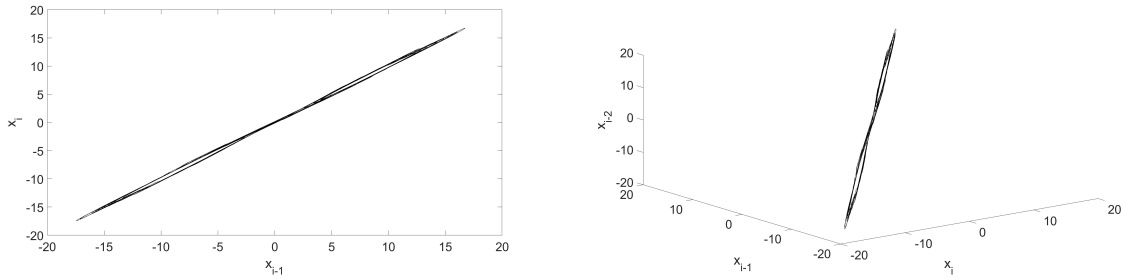


Figure 2.4: Phase Space - Lorenz Attractor ($\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$)

Figure 2.5 shows a the ApEN vectors for $m = 2$ and $m = 3$ using the coordinate x.

Once again, the two lobe structure of the 2-D phase space in Figure 2.4 is evident in Figure 2.5.

The fact that ApEN seems to be reconstructing the phase space may explain why this feature is suitable in domain shift. It is well known that the phase space captures the underlying dynamics of a system (Zill & Cullen, 2009). All possible starting points



(a) m=2 **(b)** m=3

Figure 2.5: ApEN vectors for the Lorenz attractor

chart similar trajectories/solution vectors in the phase plane as guided by the eigen vectors of the system. Healthy bearings have the same general underlying dynamics while bearings with a particular fault ought to have the same dynamics; if it is supposed that data from different domains (e.g. different operating conditions) is analogous to various initial conditions, then the shape of the solution vectors in the phase space will always be similar for a particular set of system dynamics. Therefore, it is expected that the entropy feature is invariable in different operating conditions.

Pincus recommends that m takes on a low value usually $m \in \{2, 3\}$ (Pincus et al., 1991; Pincus, 1995; Pincus & Goldberger, 1994). The choice of r must be large enough than the noise in the data; initial experiments showed that an r value three times the mean noise amplitude performs well. r must also be large enough for there to be a sufficient number of conditioning vectors for any $\boldsymbol{x}_i$ i.e. a sufficient number of "possible" vectors such that the conditional probability estimates are reasonable. r values between 0.1 to 0.25 of the standard deviation of a series were found to give results for a series assumed to have slow dynamics such as heart-rate (Pincus et al., 1991). However, the exact choice of r in the recommended range is an issue if the series is not consistent. Two series $\boldsymbol{A}$ and $\boldsymbol{B}$ series are consistent if

$$\text{when ApEN } (m_1, r_1)(\boldsymbol{A}) \leq ApEN(m_1, r_1)(\boldsymbol{B})$$

$$\text{then ApEN } (m_2, r_2)(\boldsymbol{A}) \leq ApEN(m_2, r_2)(\boldsymbol{B}) \text{ for any } r_2 \geq r_1.$$

Further discussion on other choices of r for different types of time series data is presented by Delgado-Bonal (Delgado-Bonal & Marshak, 2019).

Due to the particular formulation of ApEN, for the quantity $C_i^m(r)$ (see (2.3)) to be finite i.e. avoid computing log(0), then the template vector must also self count such that there is always at least one "possible" and one "match" for any template. This self counting introduces a bias which becomes more severe when the number of points in a series N are few. The bias implies more regularity than there is in reality.

Richman and Moorman defined a related regularity statistic called Sample Entropy (SampEN) (Richman & Norman, 2000) to eliminate the bias of ApEN by reformulating the quantities used to compute the conditional probabilities. Similar to ApEN, the "possibles" for each template are those vectors that are with a distance r without counting itself in $m-$dimensional space.. However, in contrast with ApEN, the "matches" are all the vectors within the same distance r of the template in $(m+1)$-dimensional space. Hence, SampEN uses all the vectors to find both the "possibles" and the "matches".

The "possibles" for any template are known as $A_i^m(r)$ while the sum of all "possibles" for all the templates is known as $A^m(r)$. The two quantities are computed as

$$\begin{aligned} A_i^m(r) &= \frac{1}{N-m-1} \times \sum_{j=1, j\neq i}^{N-m} p_i \\ \text{where } p_i &= \begin{cases} 1 & \text{if } d[|x_m(j) - x_m(i)|] < r] \\ 0 & \text{if } xd[|x_m(j) - x_m(i)|] \geq r] \end{cases} \end{aligned} \tag{2.10}$$

$$A^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} A_i^m(r) \tag{2.11}$$

Similarly, the "matches" for any template are known as $B_i^m(r)$ while the sum of all "matches" for all the templates is known as $B^m(r)$.

$$B_i^m(r) = \frac{1}{N-m-1} \times \sum_{j=1, j\neq i}^{N-m} p_i \tag{2.12}$$

$$B^{m+1}(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} B_i^m(r) \tag{2.13}$$

SampEn is computed as

$$SampEn(m, r, N) = -log\left(\frac{B^{m+1}(r)}{A^m(r)}\right) \tag{2.14}$$

Comparing equations (2.6) and (2.14), it is seen that for ApEN to be defined, it requires that all templates find a "match" while SampEN only requires any template to find a match for it to be defined. This is because of the re-positioning of the logarithm function in SampEN. Compared to its counterpart, ApEN is dependent on the size of series, N as indicated by the normalization factors $\frac{1}{N-m-1}$ and $\frac{1}{N-m}$ that appear in the computation of $\phi_i^m(r)$ and $\phi_i^{m+1}(r)$ respectively.

It is seen that SampEn, compares how much the shape of the long-term trajectory of the system represented by the scalar time series changes as the dimension migrates from m to $m+1$.

SampEn also quantifies the distortion of the long-term trajectory of the system by summing up changes in the neighborhood of each template vector/phase point between m and $m+1$ dimensional space.

For both ApEN and SampEN, whether a vector qualifies as "possible" or a "match" is based on a binary decision rule; if the distance criterion is satisfied, the quantities

$C_i^m(r)$, $C_i^{m+1}(r)$, $A_i^m(r)$, $B_i^{m+1}(r)$ are incremented by 1 otherwise they are incremented by 0 (see equations (2.3), (2.10) and (2.10)). This is akin to using the Heaviside function to decide when to add 1 or 0. Chen et al (Chen et al., 2007) modified SampEN to use a fuzzy function μ_c instead of the Heaviside function. The Fuzzy function, first introduced by Zadeh (Zadeh, 1965) brings in the concept of a membership degree where each point is associated with a real number in the range [0,1]. In the case of SampEN this real number quantifies how far apart a vector is from the template vector based on the limit r; a real number closer to 1 indicates that the distance between the two vectors is quite small. This modified statistic is known as Fuzzy Entropy (FuzzyEN) and the exponential function was adopted as the fuzzy function. The exponential family of functions is continuous as well as being convex such that self-similarity yields the maximum.

The algorithm of FuzzyEn closely resembles that of SampEN with a few modifications as follows.

i) The m dimensional vectors $\boldsymbol{x}_i$ are created while subtracting a baseline x_i^0 from each of the elements.

$$\begin{aligned} \boldsymbol{x}_i^m &= \{u_i, u_{i+1}, u_{i+2}, \cdots, u_{i+m-1}\} - u_i^0; \\ &\text{for } i = 1, 2, 3, \cdots, N \end{aligned} \tag{2.15}$$

x_i^0 is the average of the m elements in vector $\mathbf{x}_i^m$.

ii) The distance between any two vectors $\mathbf{x}_i^m$ and $\mathbf{x}_j^m$ is computed using Chebyshev's distance.

$$d_{ij}^m = \max_{k=1,2,\ldots,m} \left(|x(i+k-1) - x(j+k-1)|\right). \tag{2.16}$$

iii) A new quantity, the degree of similarity D_{ij}^m of $\mathbf{x}_i^m$ and $\mathbf{x}_j^m$ is now computed using the fuzzy function $\mu(d_{ij}^m, n, r)$.

$$D_{ij}^{m} = \mu(d_{ij}^{m}, n, r) \tag{2.17}$$

where $\mu(d_{ij}^{m}, n, r)$ is the exponential function with parameters n and r determining its shape/boundary.

$$\mu(d_{ij}^{m}, n, r) = e^{(-(d_{ij}^{m})^{n}/r)} \tag{2.18}$$

iv) ϕ^{m} is then defined as

$$\phi^{m} = \frac{1}{N-m}\sum_{i=1}^{N-m}\left(\frac{1}{N-m-1}\sum_{j=1,j\neq i}^{N-m} D_{ij}^{m}\right). \tag{2.19}$$

v) Steps i-iv are repeated for template vectors of length $(m+1)$ such that ϕ^{m+1} is similarly given by

$$\phi^{m+1} = \frac{1}{N-m}\sum_{i=1}^{N-m}\left(\frac{1}{N-m-1}\sum_{j=1,j\neq i}^{N-s} D_{ij}^{m+1}\right). \tag{2.20}$$

vi) For a finite series, FuzzyEN is obtained as a function of the natural logarithm

$$\text{FuzzyEn}(m, n, r, N) = \ln(\phi^{m}(n, r)) - \ln(\phi^{m+1}(n, r)) \tag{2.21}$$

The parameters r and n determine the shape of the exponential function with the former affecting the rate of decay and the latter determining how much penalty is associated with the distance d_{ij}^{m}. For instance, a value of $n = 2$ results in a lower value of D_{ij}^{m} than if $n = 3$ for the same value of d_{ij}^{m}; thus, a lower n imposes a higher penalty (Chen et al., 2007). For FuzzyEN, m and r are chosen just as for SampEN while n is chosen to be a low number e.g $n \in \{1, 2, 3\}$.

For the application of FuzzyEN in machine fault detection, it is important to recognize that machines are composed of many interacting components such as bearings, shafts

and gears; therefore condition monitoring data is bound to contain several oscillatory modes due to the interaction and coupling between the components. Thus, analysis at a single scale is insufficient for exploring the data and multi-scale entropy features are usually utilized for fault detection (L. Zhang et al., 2010; S.-D. Wu et al., 2013; Zheng et al., 2017; L. Zhang et al., 2010; K. Zhu & Li, 2015, 2014; Y. Li et al., 2019, 2018).

Costa et al (Costa et al., 2002), first introduced a technique known as coarse graining for obtaining multi-scale series from a single time series. A coarse grained time series $\mathbf{y}_\tau$ is obtained from the original time series for any scale τ by calculating the arithmetic mean of τ neighboring values without overlapping.

$$\mathbf{y}_\tau(j) = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} \mathbf{x}(i); \quad 1 \leq j \leq \frac{N}{\tau} \tag{2.22}$$

This process is depicted in Figure 2.6.

The fuzzy entropy at a scale τ is FuzzyEn$(\mathbf{y}_\tau, m, n, r)$.

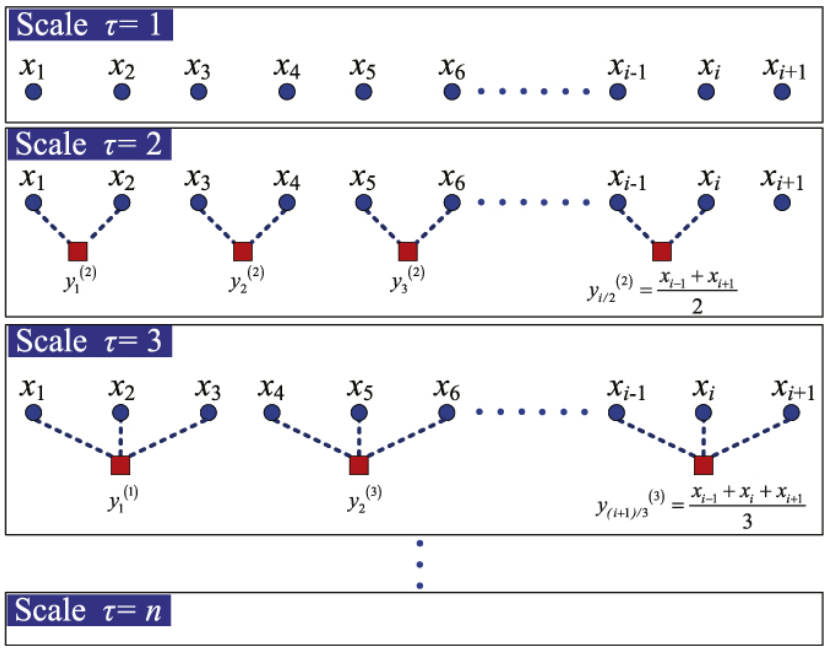


Figure 2.6: Coarse graining a time series at various scales(Y. Li et al., 2019)

There are two main drawbacks of the aforementioned method of obtaining multi-scale

data (Azami et al., 2017; S. Wu et al., 2014). The first is that entropy thus computed lacks symmetry in its dependence on the original series. For instance, looking at Figure 2.6, it can be seen that at scale 2, y_1^2 is composed of x_1 and x_2 while y_2^2 is composed of x_3 and x_4. However, at scale 3, y_1^3 is composed of x_1, x_2 and x_3 while y_2^3 is composed of x_4, x_5 and x_6. Thus, FuzzyEN is bound to be inconsistent across the different scales. This inconsistency may be interpreted as unintended smoothing action which may change the entropy value eventually changing the category of a particular dataset. Secondly, consider that for an N point time series, the length of the coarse grained series at scale factor τ is equal to N/τ; this new length may not be sufficient for accurate calculation of fuzzy entropy when τ is large. In addition, for a large τ there may be no matching templates meaning Equation (2.21) is undefined. Inaccurate and undefined fuzzy entropy compromises reliability of the method.

Improved methods such as the composite and refined composite coarse graining have subsequently been put forward. Refined composite coarse graining for calculation of multiscale sample entropy was introduced in (S. Wu et al., 2014) and extended for fuzzy entropy in (Azami et al., 2017). The composite coarse graining procedure is depicted in Figure 2.7 where it can be seen that coarse graining at any scale results in more than one new time series. For instance at scale 2, two coarse grained series are obtained while at scale 3, three coarse grained series are created.

To compute the fuzzy entropy at a scale τ, the functions ϕ^m and ϕ^{m+1} are computed for all the coarse grained time series in that scale i.e. at scale $\tau = 2$; $\phi_{\tau,1}^m$, $\phi_{\tau,2}^m$, $\phi_{\tau,1}^{m+1}$ and $\phi_{\tau,2}^{m+1}$ are computed and their averages $\overline{\phi}_\tau^m$ and $\overline{\phi}_\tau^{m+1}$ are also calculated. The fuzzy entropy computed at scale τ is now referred to as the Refined Composite Multiscale Fuzzy Entropy (RCMFE)

$$\mathrm{RCMFE}(\mathbf{x}, \tau, m, n, r) = -\ln\left(\frac{\overline{\phi}_\tau^{m+1}}{\overline{\phi}_\tau^m}\right) \tag{2.23}$$

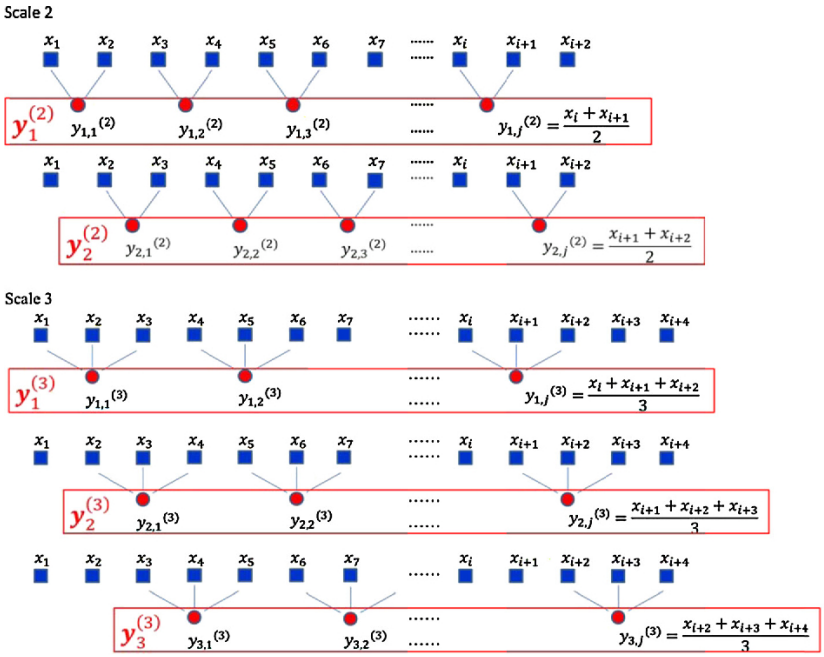


Figure 2.7: Composite coarse graining of a time series (S. Wu et al., 2014)

Equation (2.23) is always defined unless all $\phi_{\tau,j}^{m}$ and all $\phi_{\tau,j}^{m+1}$ are zeros. The length of the original data series N is chosen as 4000 which is sufficient for RCMFE calculation (Zheng et al., 2017).

2.3.2 Hazard function features for prognosis

Bearing lifetime data can be viewed from a survival analysis point of view in which case survival time is defined as the length of time taken from the start of bearing operation up to failure (Tableman & Kim, 2004; Nadler & Zurbenko, 2013). If T is a non-negative random variable representing time until failure, then the survival function $S(t)$ which estimates the probability of surviving past time t is defined as

$$S(t) = 1 - F(t) = Pr(T > t) \quad \text{for} \quad t > 0, \tag{2.24}$$

where $F(t)$ is the cumulative density function of T. At the beginning of accelerated aging experiments, the bearings are generally fault free and hence the survival function

decreases monotonically. The hazard function characterizes the rate of change of the survival function with time. Thus, as survival is steadily decreasing in bearing lifetime data, the hazard increases with time. The hazard function $h(t)$ thus indicates the instantaneous risk of failure at time t given that failure has not yet occurred.

$$h(t) = \lim_{\Delta t \to 0} \quad \frac{P(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} \tag{2.25}$$

Although there are several families of survival distributions in use, the Weibull distribution is popularly employed as a purely empirical model even where there is insignificant theoretical justification (Paterno, 2004). This is because it makes no assumptions about the distribution of the base hazard function and can also model both increasing and decreasing hazards. It is thus suitable for modeling the time to failure of machine components or the RUL.

The 2-parameter Weibull distribution is commonly characterized by a shape (β) and scale (α) parameter which are learned from data. The Weibull hazard function is given by (Lee & Wang, 2003)

$$h(t) = \beta \, \alpha^{\,\beta} t^{\,\beta-1}. \tag{2.26}$$

The process of modeling degradation of components and predicting RUL greatly relies on the proper choice of health indicators (HIs). Some of the metrics used to rate the HIs in this regard include monotonicity, robustness and trendability (Lei et al., 2018; Duoung et al., 2018). Unfortunately, commonly used time domain features such as the Root Mean Square (RMS) and kurtosis perform poorly with regard to the mentioned metrics and are thus not able to properly track fault and hence reliable predict the RUL. In contrast, the hazard functions of these features manifest the desired metrics in an excellent fashion and most importantly for this work, they behave in the same manner across different operating conditions i.e. are domain invariant.

By recalling their definition, it is clear that hazard functions are monotonically increasing and thus always trending. The definition also guarantees that hazard functions are similar regardless of the domain of data used to characterize the Weibull distribution. Thus, hazard functions satisfy both trendability and invariability across operating conditions.

The hazard functions will be computed for two commonly used time-domain health indicators in bearing prognostics i.e. Root Mean Square (RMS), and kurtosis (Lei et al., 2018; N. Li et al., 2018; Duoung et al., 2018; Mahamad et al., 2010). Kurtosis is known to be highly sensitive to incipient fault (N. Li et al., 2018) and hence it is used to detect the change-point of bearing state i.e. where fault is first detected and from which the bearing progressively deteriorates to failure. RMS is an indicator of the energy level of the system which should increase as the fault worsens due to increased vibration. It is thus commonly used to track fault (J. Zhu et al., 2014).

For a vector $\boldsymbol{x} = x_1, x_2, \ldots, x_i, x_{i+1}, \ldots x_N$ with a total of N elements, the RMS is computed as

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2}. \tag{2.27}$$

The shape factor (SF) which is a scaled version of the RMS, can also be used.

$$SF = \frac{RMS}{\left(\frac{1}{N}\sum_{i=1}^{N}|x_i|\right)}. \tag{2.28}$$

On the other hand, kurtosis can be thought of as describing the shape of the amplitude distribution of a signal. It is a measure of the size of the tails of a distribution compared to the tails of a normal/Gaussian distribution of the same variance (J. Zhu et al., 2014)

$$kurtosis = \frac{N * \sum_{i=1}^{N}(x_i - \bar{x})^4}{\left(\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^2}. \tag{2.29}$$

2.4 Mapping Algorithms

In this section, the machine learning algorithms used for mapping features to outputs are outlined.

2.4.1 Self Organizing Fuzzy Classification for Diagnosis

In this section, the self organizing fuzzy (SOF) classifier that was used for classification in the diagnosis is described. This classifier was adopted because it groups by a process analogous to clustering rather than using decision boundaries which could be helpful in promoting successful cross domain categorization. Fuzzy set theory, first introduced by Lofti A. Zadeh (Zadeh, 1965), generalizes the classical crisp sets where a member either belongs or does not belong to a set. Instead, in fuzzy set theory, a member is allowed to partially belong to a set and hence a fuzzy set $\boldsymbol{A}$, is characterized by a membership function $\mu_{\boldsymbol{A}}(x)$ that can take on any value in the closed interval [0,1]. Membership functions are selected by experts with domain knowledge and can be Gaussian, trapezoidal, triangular, etc. The parameters of the membership functions are determined by offline optimization or are handcrafted based on practical experience (Angelov & Gu, 2019). Fuzzy sets have been extended to Fuzzy rule-based (FRB) systems which enable an intuitive and powerful method of making decisions using rules encapsulated in IF...THEN statements. It has been theoretically proven that fuzzy systems have the universal approximator property and hence are able to reasonably approximate any continuous non-linear function (L. Wang & Mendel, 1992) even though a large number of fuzzy sets and rules may be required. There have been three broad categories of FRB systems in fuzzy systems all of which have an antecedent (IF) section followed by

a consequents (THEN) section (Angelov & Gu, 2019).

1. Zadeh-Mamdani (Mamdani & Assilian, 1975): This was the original system and consists of several linguistic If...Then rules e.g.

$$\text{IF } (x_1 \; is \; LT_{i,1}) \text{ AND } (x_2 \; is \; LT_{i,2}) \; \cdots \text{ AND } (x_N \; is \; LT_{i,N})$$
$$\text{THEN}(y_i \; is \; LT_{i,out})$$

   where $\boldsymbol{x} = [x_1, x_2, x_3, \cdots, x_N]^T$ is the feature vector; $LT_{i,j} (i = 1, 2, 2, \cdots, M, j = 1, 2, 3, \cdots, N)$ is the linguistic term of the j^{th} fuzzy set of the i^{th} fuzzy rule; M is the number of fuzzy rules; $LT_{i,out}$ is the linguistic term of the output of the i^{th} fuzzy rule; y_i is the output of the i^{th} fuzzy rule. Linguistic terms are defined by membership functions and may be terms such as low, high, medium etc as applied to features such as speed, temperature, income etc.

2. Tagaki-Sugeno (Takagi & Sugeno, 1985): This FRB is similar to the previous one with the exception that the consequent is a regression model.

$$\text{IF } (x_1 \text{ is } LT_{i,1}) \text{ AND } (x_2 \; is \; LT_{i,2}) \; \cdots \text{ AND } (x_N \text{ is } LT_{i,N})$$
$$\text{THEN}(y_i \text{ is } \overline{\boldsymbol{x}}^T \boldsymbol{a}_i)$$

   where $\overline{\boldsymbol{x}} = [1, \boldsymbol{x}^T]^T$; $\boldsymbol{a}_i = [a_{i,0}, a_{i,1}, a_{i,2}, \cdots, a_{i,N},]^T$ is a vector consequent parameters for the i^{th} fuzzy rule.

3. AnYa systems invented by Angelov and Yager (Angelov & Yager, 2012) have an antecedent that is based on a prototype instead of a fuzzy set and a consequent that is a function.

$$\text{IF } (x_i \text{ is } \sim p_i) \text{ THEN}(y_i \text{ is } \overline{\boldsymbol{x}}^T \boldsymbol{a}_i)$$

where p_i is the prototype of the i^{th} fuzzy rule and $\sim$ denotes similarity or typicality.

The design of the Zadeh-Mmadani and Tagaki-Sugeno FRB systems consists of three main steps

a. Selection of the membership function i.e. triangular, Gaussian etc.

b. Definition of the linguistic terms (terminology, quantity)

c. Determination of the parameters of the membership functions

On the other hand, AnYa systems are simpler to design and require only that the peak of the membership function be specified i.e. that prototypes are chosen. Prototypes are focal points in data clouds which attract neighbourhood samples. This AnYa fuzzy classifier will henceforth be referred to as the self organizing fuzzy (SOF) classifier.

The empirical approach to data characterization is formulated as "Given observations of outcomes of real processes/experiments alone, estimate the ensemble properties of the data and, furthermore, estimate these for any feasible outcome" (Angelov & Gu, 2019). These ensemble properties include data clouds/clusters and their prototypes (peaks). This approach does not require any prior knowledge about the problem and neither does it make any assumptions about the data generation model which is a key consideration in this work where the generating distributions in either the domain and/or task are changing. Data characterization is done by evaluating several non-parametric ensemble functions such as cumulative proximity, eccentricity, data density and typicality.

1. Cumulative proximity: This is the sum of square distances of a sample from all other samples. This metric provides information about the centrality of a sample.

$$q(\mathbf{x_i}) = \sum_{j=1}^{K} d^2(\mathbf{x_i}, \mathbf{x_j}); \qquad j = 1, 2, 3, \cdots, K \tag{2.30}$$

where $\mathbf{x_i}, \mathbf{x_j} \in \{\mathbf{x}\}_K$ and $d(\mathbf{x_i}, \mathbf{x_j})$ is a function computing the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$. The shorter this distance metric is, the more similar two samples are. $\mathbf{x}$ can be of any dimension i.e. a sample can have any number of features. K is the number of samples in the dataset.

2. Eccentricity: This is the normalized cumulative proximity which is useful in capturing the properties of samples at the tail ends of a distribution i.e. samples far away from the peak.

$$\xi(\mathbf{x_i}) = \frac{2q(\mathbf{x_i})}{\frac{1}{K}\sum_{j=1}^{K} q(\mathbf{x_j})}; \qquad j = 1, 2, 3, \cdots, K \tag{2.31}$$

The $\frac{1}{K}$ in the denominator is used to standardize eccentricity to prevent tendency to zero as K grows large.

3. Unimodal data density is a measure describing mutual proximity of samples and is thus the inverse of eccentricity.

$$D(\mathbf{x_i}) = \frac{1}{\xi(\mathbf{x_i})} = \frac{\sum_{j=1}^{K} q(\mathbf{x_j})}{2Kq(\mathbf{x_i})} = \frac{\sum_{l=1}^{K}\sum_{j=1}^{K} d^2(\mathbf{x_l}, \mathbf{x_j})}{2K\sum_{j=1}^{K} d^2(\mathbf{x_i}, \mathbf{x_j})}; \qquad j = 1, 2, 3, \cdots, K \tag{2.32}$$

From the denominator of Equation (2.32) , it is seen that this measure is inversely proportional to the sum of distances between a sample and all others. Obviously, the closer a sample is to the global mean, the higher its data density i.e. more samples surround it.

4. Multimodal density. In the data, a sample may be observed repeatedly and hence the set of unique samples is defined as $\{\mathbf{U}\}_\mathbf{T} = \mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \cdots, \mathbf{u}_T$ where each distinct value is only recorded once and T is the number of unique samples. The corresponding frequencies of the unique samples in the data are denoted by

f_i. Multimodal density is then defined for each unique sample $\mathbf{u_i}$ in $\{\mathbf{U}\}_T$ and is the product of the unimodal density of the unique sample and its frequency f_i.

$$D^{MM}(\mathbf{u_i}) = f_i \times \frac{\sum_{j=1}^{K} q(\mathbf{x_j})}{2K \times q(\mathbf{u_i})}; \qquad i = 1, 2, 3, \cdots, T. \tag{2.33}$$

The described data-centric measures are then used to find the local modes in a given dataset. In the case of classification, the modes are learned individually for each class c.

2.4.2 Neural Networks for Prognosis

In this section, a brief description of the very commonly used Neural Network (NN) that will be used as the prediction model for the hazard function based HIs is given.

Neural networks are non-linear models that establish functional relationships between inputs and outputs and where the parameters defining the relationship have to be adjusted for optimal performance. The parameters are determined by exposing the network to a set of examples (composed of inputs and corresponding outputs) and observing the response from the network. The parameters are continuously adjusted to minimize the error from the networks "answers".

In this work, a basic single-hidden layer neural network (NN) will be used to map the condition indicators to the output i.e. the RUL (see Figure 2.8) (*Function fitting a neural network*, 2010).

In expanded form, the NN's outputs can be expressed mathematically as

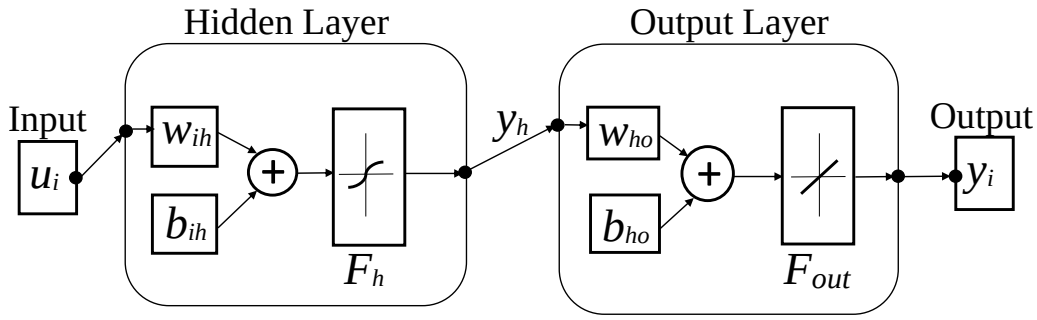


Figure 2.8: A single hidden layer neural network

$$\begin{aligned} \mathbf{y}_h &= F_h\left(\mathbf{w}_{ih} \times \mathbf{u}_i + \mathbf{b}_{ih}\right) \\ \mathbf{y}_i &= F_{out}\left(\mathbf{w}_{ho} \times \mathbf{y}_h + \mathbf{b}_{ho}\right) \end{aligned} \tag{2.34}$$

where $\mathbf{y}_h$ is an intermediate output and $\mathbf{y}_i$ is the final output corresponding to input $\mathbf{u}_i$. $\mathbf{w}_{ih}$ is the matrix of weights connecting each dimension of of input $\mathbf{u}_i$ to all the nodes in the hidden layer while weights $\mathbf{w}_{ho}$ connect all dimensions of the intermediate output $\mathbf{y}_h$ to the nodes of the output layer. $\mathbf{b}_{ih}$ and $\mathbf{b}_{ho}$ are the biasing terms in the hidden and output layers receptively. Functions $F_h(\cdot)$ and $F_{out}(\cdot)$ are the activation/transfer functions that map respective inputs to outputs in the desired range of values. The dimension of input $\mathbf{u}_i$ is usually greater than one which indicates the use of more than one health indicator. Because RUL prediction is a regression task, the output is a single number in the chosen units e.g. RUL in terms of the percentage of life remaining or time in hours.

2.5 Summary

From the foregoing literature, key areas addressed by this research are derived from the fact that domain adaptation requires

1. target data during training,

2. re-training of the models each time new target data is acquired.

This requirements cannot always be met: for instance

a. target data may completely unavailable as in the case of a newly installed machine

b. the available target data may not be sufficient such that the discrepancy in distributions is not accurately captured

Underlying the concept of DA is that model must always be re-trained; the training process usually requires expertise to set up and may be computationally expensive. As such, it is very desirable to develop strategies that only require training once. This goal is achievable if domain invariant features for both diagnosis and prognosis exist since then the model only need be trained with once. The two domain invariant features proposed are the Refined Composite Multiscale Fuzzy Entropy (RCMFE) for diagnosis and hazard features for prognosis.

CHAPTER THREE

DESIGN METHODOLOGY

In this section, the methodology designed to carry out domain invariant diagnosis and prognosis is outlined in detail.

3.1 Diagnostics

There were two datasets used for diagnostics. The first dataset used is courtesy of the Case Western Reserve University (CWRU) bearing data center(Case Western Reserve University Western Bearing Data Center, 2018). It consists of vibration data from normal/healthy bearings and as well as from faulty bearings with artificially seeded single point faults. Although faulty bearing data is sampled at both 12 kHz and 48 kHz, only the former set from the drive end was used as the baseline healthy data was only available at the 12 kHz sampling rate (Y. Li et al., 2019). Of particular interest in this work is that the CWRU dataset was recorded for four operating conditions related to motor load and speed thus constituting four different data domains. Further, the faults are seeded at four different diameters which also introduce a shift in data distribution. There are three types of fault in the data for all the operating conditions i.e. Outer Race (OR), Inner Race (IR) and ball faults, all of which have data corresponding to the four fault sizes. The exception is the OR data which lacks the 0.711 m diameter. The organization of data into operating conditions and fault sizes is listed in Table 3.1. In addition, although OR data is collected in three different positions relative to the bearings' load zone, only measurements at the 6 o'clock position were used since there was no mechanism to convert the dynamometer torque into radial load borne by the bearings, and thus the only effective radial load was gravity (W. A. Smith & Randall, 2015).

The second dataset consists of vibration data collected under different time varying speed conditions courtesy of the University of Ottawa (Huang & Baddour, 2018). The

Table 3.1: CWRU data description

Tags	Specifications	Fault diameters (dia)
CWRU 1	0 hp/1797 rpm	IR = 0.177 m, 0.355 m, 0.5334 m and 0.711 m.
CWRU 2	1 hp/1772 rpm	Ball = 0.177 m, 0.355 m, 0.5334 m and 0.711 m.
CWRU 3	2 hp/1750 rpm	OR = 0.177 m, 0.355 m and 0.5334 m.
CWRU 4	3 hp/1730 rpm	

operating conditions in this dataset are time-varying throughout the recording of the vibration signal. The variations are as follows: increasing speed (Cond 1), decreasing speed (Cond 2), increasing then decreasing speed (Cond 3) and decreasing then increasing speed (Cond 4). The speed ranges spanned are given the dataset's documentation. The data is categorized into healthy, OR and IR classes. For each operating setting and class, three trials are conducted to increase authenticity resulting in a total of 36 files.

3.1.1 Fault diagnosis using Refined Composite Multiscale Fuzzy Entropy

Machine learning models require that data is divided into training and testing datasets. The subdivision process and preparation of the data is outlined.

1. Each data file was divided into 20 examples containing 4000 consecutive non-overlapping points.

2. Each example was denoised using a bior 3.7 wavelet of level 3 (Lessmeier et al., 2016) and then normalized. RCMFE was then computed for 20 scales with an embedding factor of 2. The 20 RCMFE values made up the feature vector.

3. Training data was organized according to the four operating conditions i.e. for source domain 1, the training data was obtained from CWRU 1/Cond 1 for the CWRU and Ottawa datasets respectively. Similarly, training data for source domain 2 was drawn from CWRU 2/Cond 2, source domain 3 data was drawn

from CWRU 3/Cond 3, and finally source domain 4's data was drawn from CWRU 4/Cond 4.

4. The files used from the CWRU dataset were as follows: for the fault detection stage, healthy data vs 0.177 m dia IR, OR and ball data was used. For the initial stage of fault isolation where a classifier is trained on the three fault classes, the training data consisted of 0.177 m dia, 0.355 m dia and 0.711 m dia IR, 0.177 m dia, 0.355 m dia and 0.533 m dia OR and 0.177 m dia, 0.355 m dia and 0.711 m dia ball fault data. The data was mixed in the described manner to ensure the largest fault data available for each class was used for training. Thus, because the 0.711 m dia is unavailable for OR from the daaset repository, one fault class (0.533 m dia) was skipped from the IR and ball classes to ensure class imbalance did not occur. For IR vs Ball classifier, 0.177 m dia, 0.355 m dia, 0.533 m dia and 0.711 m dia were used for both the IR and ball classes. In the ball vs OR classifier, 0.177 m dia, 0.355 m dia and 0.711 m dia was used for the ball class while 0.177 m dia, 0.355 m dia and 0.533 m dia was used for the OR class.

5. The files used from the Ottawa dataset were as follows: for the first stage i.e fault detection, the classifiers were trained with H-A-1 vs I-A-1/O-A-1 for source domain 1, H-B-2 vs I-B-2/O-B-2 for source domain 2, H-C-3 vs I-C-3/O-C-3 for source domain 3 and H-D-1 vs I-D-1/O-D-1 for source domain 4. In the second stage of fault isolation the classifiers were trained with I-A-1 vs O-A-1 for source domain 1, I-B-2 vs O-B-2 for source domain 2, I-C-3 vs O-C-3 for source domain 3 and I-D-1 vs O-D-1 for source domain 4. The suffixes 1,2 and 3 in this dataset indicate the trial number. The middle letter is indicative of the operating condition i.e A for Cond 1, B for Cond 2, C for Cond 3 and D for Cond 4.

6. During the training phase, all classes consisted of an equal number of samples to avoid class imbalance.

7. The testing/target data for each of the source domains consisted of every other

sample not in the training data i.e. all remaining examples from the rest of the conditions were combined into the test set.

8. For each of the four source domains, SOF classifiers were trained and used to categorize the test samples.

A summarized flowchart of the steps followed in the diagnostic procedure is given in Figure 3.1.

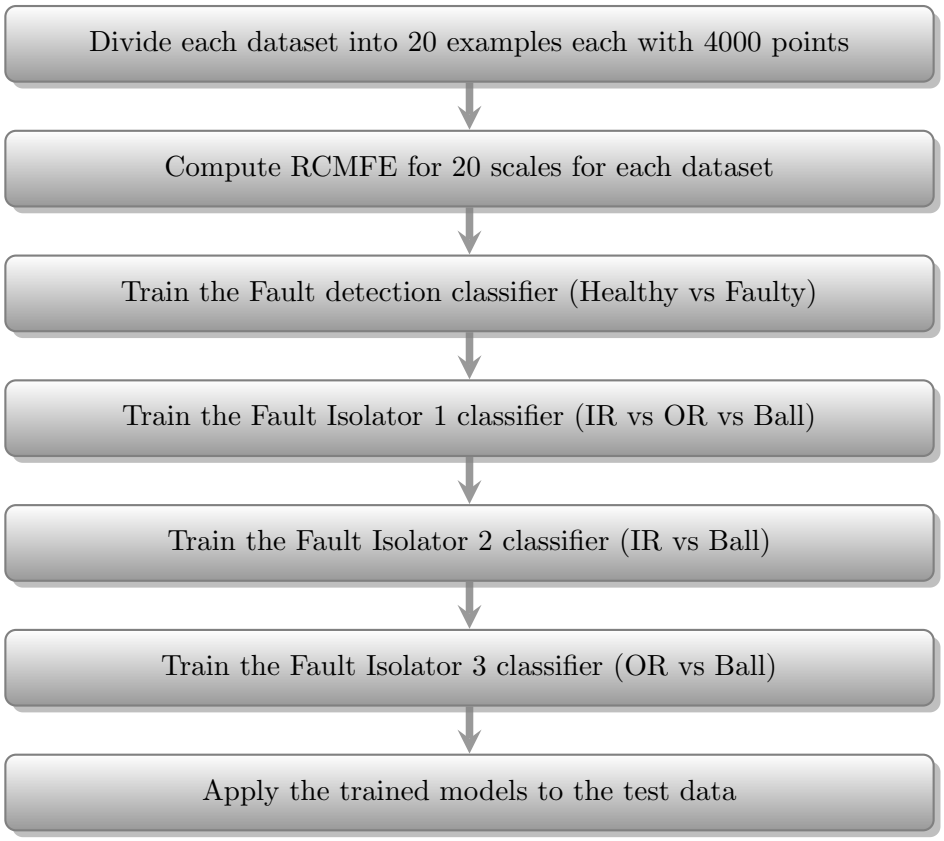


Figure 3.1: General steps followed for diagnostics

3.1.2 Assignment of classes by the Self Organizing Fuzzy Classifier

The self organizing fuzzy (SOF) classifier uses the data characteristics described in section 2.4.1, to group samples into classes.

Let the K^c training samples of the c^{th} class ($c = 1, 2, 3, \cdots, C$) be denoted by $\{\mathbf{x}\}^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \mathbf{x}_3^c, \cdots, \mathbf{x}_{K^c}^c\}$; $\{\mathbf{U}\}_T^c = \{\mathbf{u}_1^c, \mathbf{u}_2^c, \mathbf{u}_3^c, \cdots, \mathbf{u}_T^c\}$ be the set of all unique samples of

the c^{th} class and $\{\mathbf{f}\}_T^c = \{f_1^c, f_2^c, f_3^c, \cdots, f_T^c\}$ the corresponding frequency of occurrence set. The local modes and their centers for any class are found based on multimodal densities as follows.

1. Calculate the multimodal densities $D^{MM}(\mathbf{u_i})$ of all unique samples $\mathbf{u}_i^c$ using Equation (2.33).

2. Find the unique sample with the highest multimodal density, remove it from $\{\mathbf{U}\}_T^c$ and place it as the first element $\mathbf{r}_1$ in a ranking list $\{\mathbf{r}\}$.

3. From the remaining samples in $\{\mathbf{U}\}_T^c$, remove the sample with the shortest distance to $\mathbf{r}_1$ and let it be the second element $\mathbf{r}_2$ in the ranking matrix $\mathbf{r}$. Similarly, $\mathbf{r}_3$ is the element with the shortest distance to $\mathbf{r}_2$. This process is repeated until $\{\mathbf{U}\}_T^c$ is empty.

4. Initial prototypes $\{\mathbf{p}\}_0^c$ are found from the local maxima of multimodal densities ranked in $\mathbf{r}$ according to **Condition 1**.

Condition 1:

$$\text{IF } \left(D^{MM}(\mathbf{r_i}) > D^{MM}(\mathbf{r_{i+1}})\right) \text{ AND } \left(D^{MM}(\mathbf{r_i}) > D^{MM}(\mathbf{r_{i-1}})\right) \quad \text{THEN } \left(\mathbf{r}_i \in \{\mathbf{p}\}_0^c\right) \tag{3.1}$$

5. Once the initial prototypes are found, they are used to attract the data samples surrounding them forming data clouds which represent the local modes present in the data. The clouds are analogous to clusters except for the fact that they are non-parametric and do not conform to any particular shape. A point is assigned to the closest prototype by

$$\text{closest prototype } = \underset{\mathbf{p}\in\{\mathbf{p}\}_0^c}{\text{argmin}}\left(d(\mathbf{x_i}, \mathbf{p})\right); \qquad \mathbf{x_i} \in \{\mathbf{x}\}^c \tag{3.2}$$

6. The initial data clouds must then be filtered to retain only the larger and more representative clouds in order to improve the generalization ability of the model. This corresponds to increasing the level of granularity. The higher the granularity, the more information about the local modes the model is able to capture. The very first prototypes computed are at the zeroth level of granularity hence the subscript 0 in $\{\mathbf{p}\}_0^c$. The filtering process proceeds as follows.

   a. Calculate the centers of all the data clouds $\mathbf{z_i} \in \{\mathbf{z}\}_0^c$.

   b. Calculate the multimodal density at the centers $\mathbf{z_i}$ as the product of the unimodal density weighted by the support (number of elements) of the particular cloud, S_i.

$$D^{MM}(\mathbf{z_i}) = S_i \times D(\mathbf{z_i}); \qquad \mathbf{z_i} \in \{\mathbf{z}\}_0^c \tag{3.3}$$

   c. Calculate the average radius of local influence of each protoype at granularity level L as

$$G^{c,L} = \sum_{\mathbf{x},\mathbf{y}\in\{\mathbf{x}\}^c,\, \mathbf{x}\neq\mathbf{y},\, d^2(\mathbf{x},\mathbf{y})\leq G^{c,L-1}} d^2(\mathbf{x},\mathbf{y})/Q^{c,L}; \qquad G^{c,0} = \overline{d^c} \tag{3.4}$$

   where $\overline{d^c}$ is the average distance between any two data samples in $\{\mathbf{x}\}^c$

$$\overline{d^c} = \frac{1}{K^2}\sum_{i=1}^{K} q(\mathbf{x_i}) \tag{3.5}$$

   and $Q^{c,l}$ are the number of pairs of data samples in $\{\mathbf{x}\}^c$ between which the distance is smaller than $G^{c,l-1}$

   d. For each data cloud center e.g. the i^{th} one, find elements of the set composed of the centers of its neighboring clouds $\{\mathbf{z}\}_i^{N+}$ through **Condition 2**.

$$\textbf{Condition 2:} \quad \text{IF } \left(d^2(\mathbf{z_i},\mathbf{z_j}) \leq G^{c,L}\right) \quad \text{THEN } \left(\mathbf{z_j} \in \{\mathbf{z}\}_i^{N+}\right) \tag{3.6}$$

   e. The prototypes at the L^{th} granular level for class c are finally chosen as per

Condition 3.

$$\textbf{Condition 3:}\quad \text{IF}\ \left(D^{MM}(\mathbf{z_i}) > \max_{\mathbf{z}\in\{\mathbf{z}\}_i^{N+}}\left(D^{MM}(\mathbf{z})\right)\right)\quad \text{THEN}\ \left(\mathbf{z_i}\in\{\mathbf{p}\}_{L}^{c}\right) \tag{3.7}$$

For a total of N_c prototypes, N_c fuzzy rules corresponding to each are formulated as

$$\text{IF}\ \left(\mathbf{x}\sim\mathbf{p}_1^c\right)\ \text{OR}\ \left(\mathbf{x}\sim\mathbf{p}_2^c\right)\ \text{OR}\cdots\ \text{OR}\ \left(\mathbf{x}\sim\mathbf{p}_{N_c}^c\right)\ \text{THEN}\ \left(class\ c\right) \tag{3.8}$$

$\sim$ is the similarity operator and is analogous to the degree of membership in alternative antecedents.

To classify a sample $\mathbf{x}$ as belonging to one of C classes, the firing strength of the c^{th} class ($c = 1, 2, 3, \cdots, C$) is computed as

$$\lambda^c(\mathbf{x}) = \max_{\mathbf{p}\in\{\mathbf{p}\}^c}\left(e^{-d^2(\mathbf{x},\mathbf{p})}\right) \tag{3.9}$$

where $d(\mathbf{x}, \mathbf{p})$ is the distance or similarity between $\mathbf{x}$ and $\mathbf{p}$ and may be any one of the common distance functions such as Euclidean, Cosine, Mahalanobis etc.

The final label is assigned using a "winner takes all" strategy according to

$$\text{label}\ \mathbf{x} = \underset{c=1,2,3,\cdots,C}{\text{argmax}}\left(\lambda^c(\mathbf{x})\right) \tag{3.10}$$

The RCMFE features were used as input to the SOF classifier.

3.2 Prognostics

The data used for prognostics is described in section 3.2.1 followed by the step by step procedure of estimating RUL from the hazard function health indicators.

3.2.1 Experiments

In this work, data for the 2012 IEEE-PHM prognostics challenge was used. The data is available online from the National Aeronautics Space Administration's (NASA) prognostics center of excellence page (NASA, 2012). The dataset was produced by the prognostics health management research team of the FEMTO-ST Institute in France and will henceforth be referred to as the FEMTO dataset (Nectoux et al., 2012).

The data was generated on the experimental platform PRONOSTIA which enabled characterization of the lifetime degradation of ball bearings in just a few hours. Details of the platform setup can be found in the related publication (Nectoux et al., 2012). Because the bearings were not seeded with fault, the data obtained corresponds very closely to that of normally degraded bearings.

Operating conditions were determined by instantaneous measurements of the radial force applied on the bearing, the speed rotation of the shaft handling the bearing and the torque inflicted on the bearing. The conditions were then grouped into three based on rotation speed and the radial force as listed in Table 3.2.

Table 3.2: Grouping operating conditions

	Speed (rpm)	Radial Force (N)
Condition 1	1800	4000
Condition 2	1650	4200
Condition 3	1500	5000

Degradation was characterized by readings from vibration and temperature sensors i.e. two miniature accelerometers and a Resistance Temperature Detector (RTD) probe. The acceleration measurements were sampled at 25.6 kHz. Out of the two orthogonally placed accelerometers, only data from the horizontal axis accelerometer (Horz) was used. The temperature data was also not used.

There are 17 datasets in the FEMTO data, 6 of which are complete run to failure data for training while the other 11 are truncated for testing efficacy of developed

algorithms.

Table 3.3 enumerates the bearings from which the learning and test data are obtained.

Table 3.3: Training and Test bearings

	Operating Conditions		
Datasets	Condition 1	Condition 2	Condition 3
Training Sets	Bearing1_1	Bearing2_1	Bearing3_1
	Bearing1_2	Bearing2_2	Bearing3_2
	Bearing1_3	Bearing2_3	Bearing3_3
Test Sets	Bearing1_4	Bearing2_4	
	Bearing1_5	Bearing2_5	
	Bearing1_6	Bearing2_6	
	Bearing1_7	Bearing2_7	

3.2.2 Remaining Useful Life Estimation

The objective of this stage was to create a remaining useful life (RUL) prediction model that is accurate over different operating conditions. As a preliminary step, training was performed using data from a single operating condition (B1_1) and the model applied to test sets from all the operating conditions. This model performed well on the test sets from condition 1 but poorly on test sets from the other operating conditions. Thus, a simple scheme was devised to incorporate data from all the training conditions as explained as follows.

1. For training, three sets of data were passed to MATLAB's function **fitnet**

   a. B1_1 as the training dataset.

   b. B2_1 as the validation dataset.

   c. B3_1 as the in-training testing dataset.

   A training dataset is used to estimate the weights of the network while the validation dataset prevents the model from overfitting during learning. The in-training testing dataset is used to obtain an un-biased estimate of the model performance

because it is never used in the actual learning process i.e. in the estimation of model weights. The three datasets were specifically chosen from different operating conditions in an attempt to make model accurate across all the three operating conditions.

2. In preparing the model inputs for prognostic modeling, time is an important component in addition to the other health indicators (Mahamad et al., 2010; "Monotonicity", 2018; "Trendability", 2018; "Prognosability", 2018). Expanding the input by incorporating past feature values has also been shown to be helpful (Mahamad et al., 2010). The ANN input $\mathbf{u}_i$ in Equation (2.34) was therefore a 6-dimensional feature matrix comprising of past (subscript *i-1*) and current (subscript i) values of time and the condition indicators i.e. $\text{time}_{i\text{-}1}$, time_i, $\text{kurtosis-hazard-function}_{i\text{-}1}$, $\text{kurtosis-hazard-function}_i$, $\text{shape-factor-hazard-function}_{i\text{-}1}$ and $\text{shape-factor-hazard-function}_i$.

3. Since the goal of the trained model is to predict RUL, the target/output of the ANN was linearly modeled as normalized life percentage with 100% indicating the start of bearing operation and 0% indicating end of life.

4. In section 2.4.2 the ANN was described as having one hidden layer which could have upto 10 nodes. The ideal number of nodes was selected as follows.

    a. For the first candidate number of nodes i.e node = 1, the model weights were randomly initialized and the model trained. The model and its accuracy on the training, validation and in-training test data set were stored.

    b. The model weights were then re-initialized nine more times for node = 1 and after each training, the model and its accuracy on the training, validation and in-training test data sets was stored.

    c. Steps 4a and 4b were then repeated for each of the remaining candidate number of nodes i.e. from 2 to 10 nodes. MATLAB's **fitnet** function for was used for

training with weights being determined by the default Levenberg-Marquardt algorithm. Notably, even though the weights of the model were re-initialized 10 times in evaluating a particular number of nodes, a random seed was first set before training commenced to ensure repeatability.

d. On completion of steps 4a, 4b and 4c, the model with the least number of hidden nodes and the greatest accuracy on the training, validation and in-training test dataset from the 100 trained models was selected.

5. The best model in step 4d was then used to predict the RUL of all the test bearings labeled *Test Sets* in Table 3.3 in the testing phase.

In both training and testing, the R^2 metric was used for evaluating performance and is computed as (Grus, 2015)

$$R^2 = 1 - \frac{\text{MSE}}{\text{Variance(target)}} \tag{3.11}$$

where MSE is the mean squared error. The MSE is normalized by dividing it with the naive prediction error i.e the error obtained by predicting that the RUL is equal to the mean of the target vector which is the definition of variance.

The R^2 metric indicates how much of the variance the data in question is explained by the model and hence a value as close to 1 as possible is desirable.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter first discusses results from diagnosis followed by those from the prognosis process. In the section on diagnostics, data from the Case Western Reserve University (CWRU) and that from Ottawa University were used. For prognosis, the FEMTO-ST dataset was utilized. Preliminary investigations of the datasets are also given before presenting results from each of the condition monitoring tasks.

4.2 Diagnosis

4.2.1 Preliminary Analysis

Figure 4.1 shows a selection of the raw vibration signals of normal and faulty bearings from the CWRU dataset. The different classes of data are not distinguishable from the raw waveforms. The raw data could not be used to extract any credible information on the condition of the bearing due to the high amount of noise.

A preliminary exploration of the CWRU data was done to confirm the performance of commonly used features for diagnosis in changing operating conditions. Two features commonly used for bearing diagnosis (Kimotho & Sextro, 2014) - root mean square (RMS) and kurtosis were extracted from the data and used as inputs to a basic classification model. Three classes of data (healthy, IR and OR) were arbitrarily chosen from two operating conditions (CWRU 1 and CWRU 3) and a Naive Bayes model used to fit distributions in the 2D feature space.

It was observed that the marginal distributions of the classes in different operating conditions are distinct as shown in Figure 4.2. It was also easy to see that depending on the operating condition a test point was drawn from and the model used for prediction, the assigned label could vary. For instance, looking at the decision boundaries learned

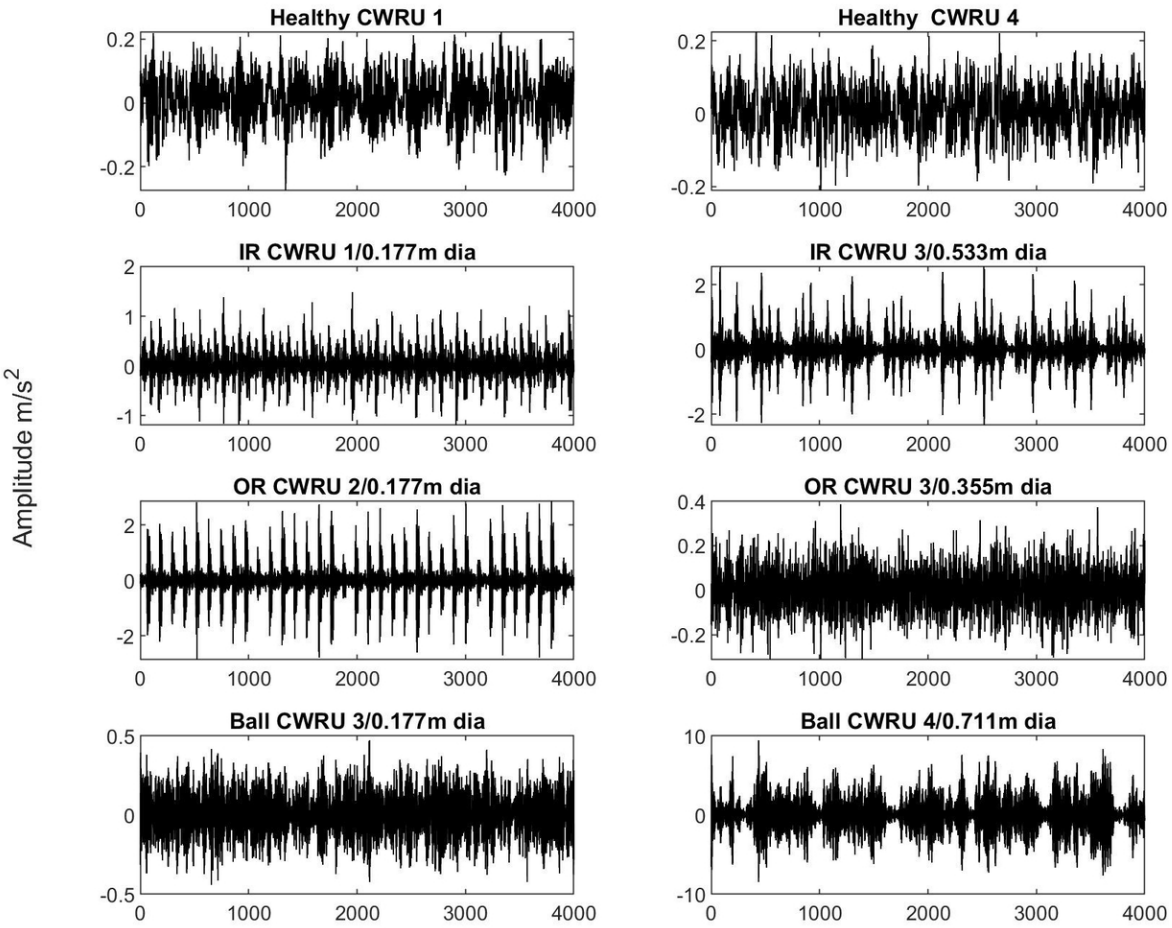


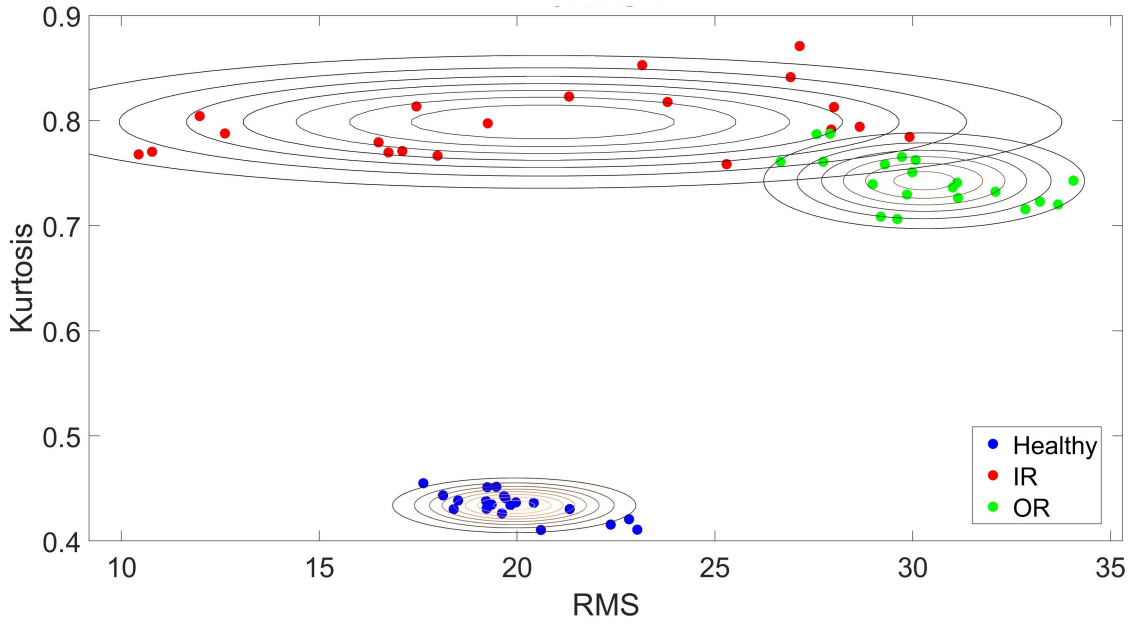
Figure 4.1: Raw waveforms of CWRU data

in Figure 4.3, it was noted that if a test point was drawn from the bottom left half portion of CWRU 3 space, then the model trained with CWRU 3 data will assign the label as OR but the model trained on CWRU 1 data would assign the same test point an IR label.

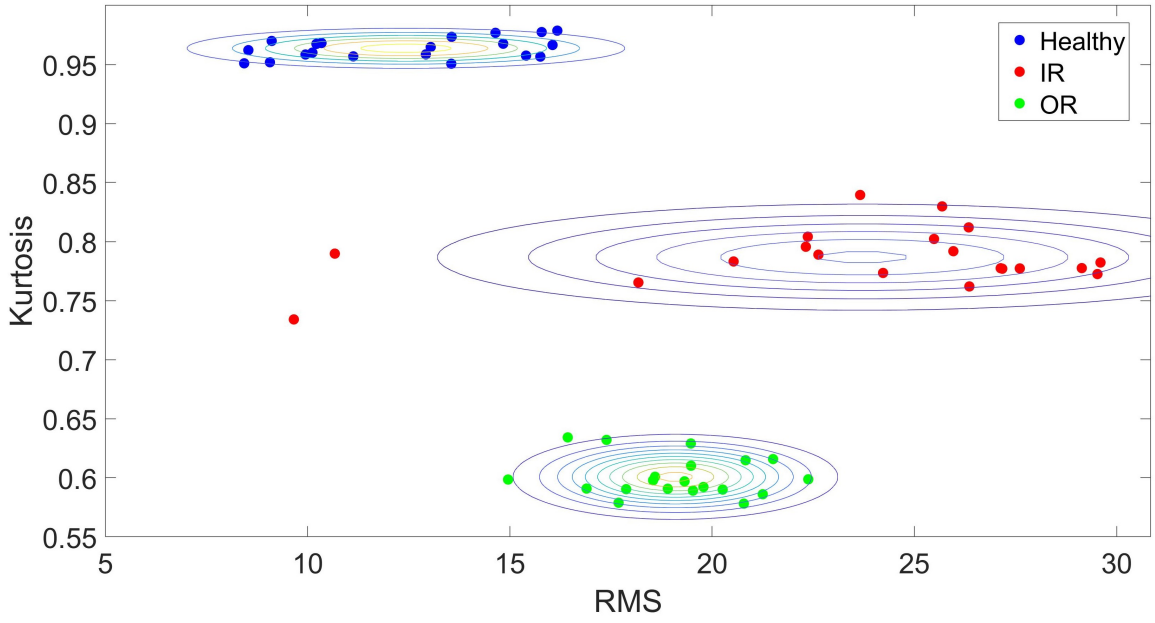
Decision boundaries conceptualize the discriminative approach to prediction where the joint probability of a pair $(\boldsymbol{x}, y)$ is given by $Pr(\boldsymbol{x}, y) = P(y|\boldsymbol{x})P(\boldsymbol{x})$. The marginal distribution $P(\boldsymbol{x})$ is computed by marginalizing out probabilities as

$$P(\boldsymbol{x}) = \sum_{y} p(\boldsymbol{x}, y) = \sum_{i=1}^{y} p(\boldsymbol{x}|y = i)p(y = i).$$

The conditional probability $P(y|\boldsymbol{x})$ is in essence what a discriminative model learns



(a) CWRU 1

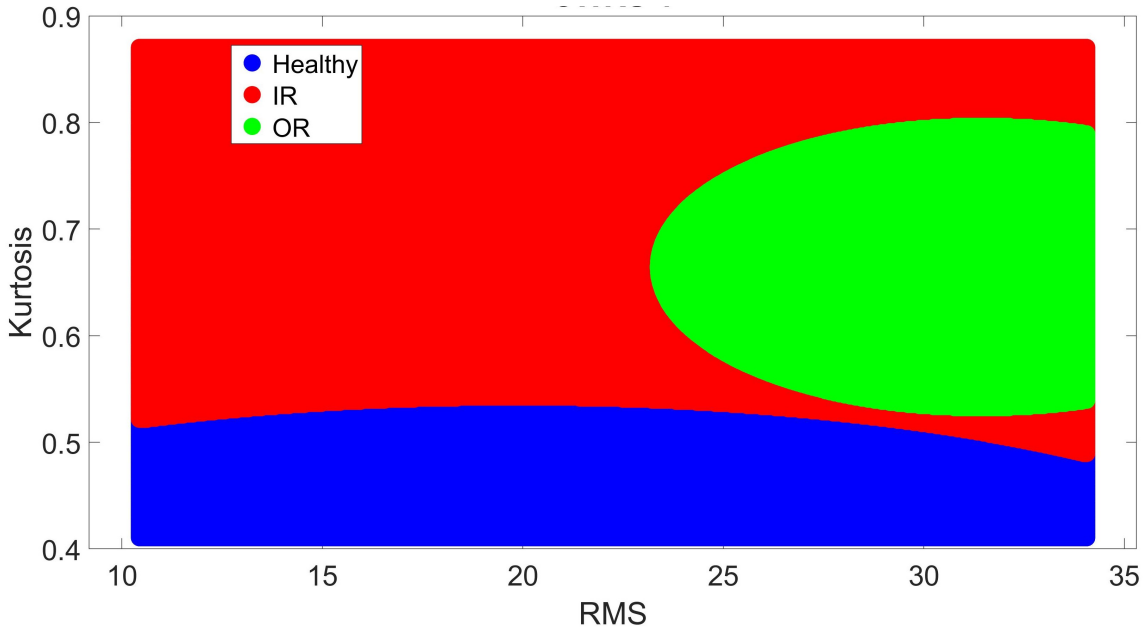


(b) CWRU 3

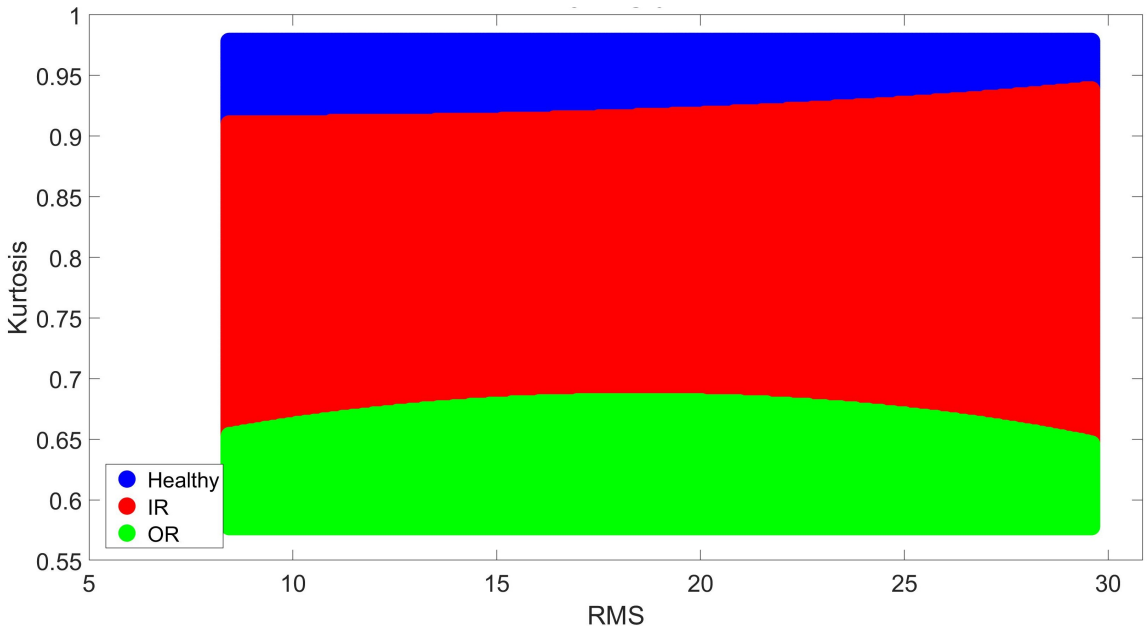
Figure 4.2: Marginal distributions of bearing data in a 2-D feature space in two operating conditions

directly from data and it is aimed at finding thresholds that separate the various classes.

Generally, two features are too few for accurate classification of real-world data and in Figure 4.4, the t-distributed Stochastic Neighbor Embedding (t-SNE) method (van der Maaten & Hinton, 2008) was used to visualize the marginal class distributions of the



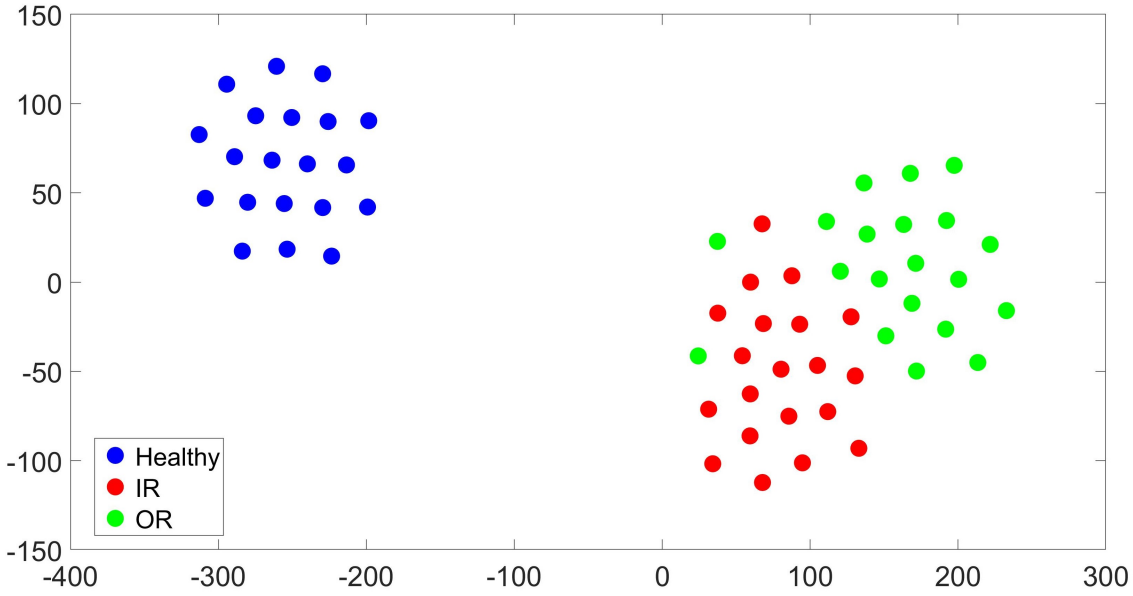
(a) CWRU 1



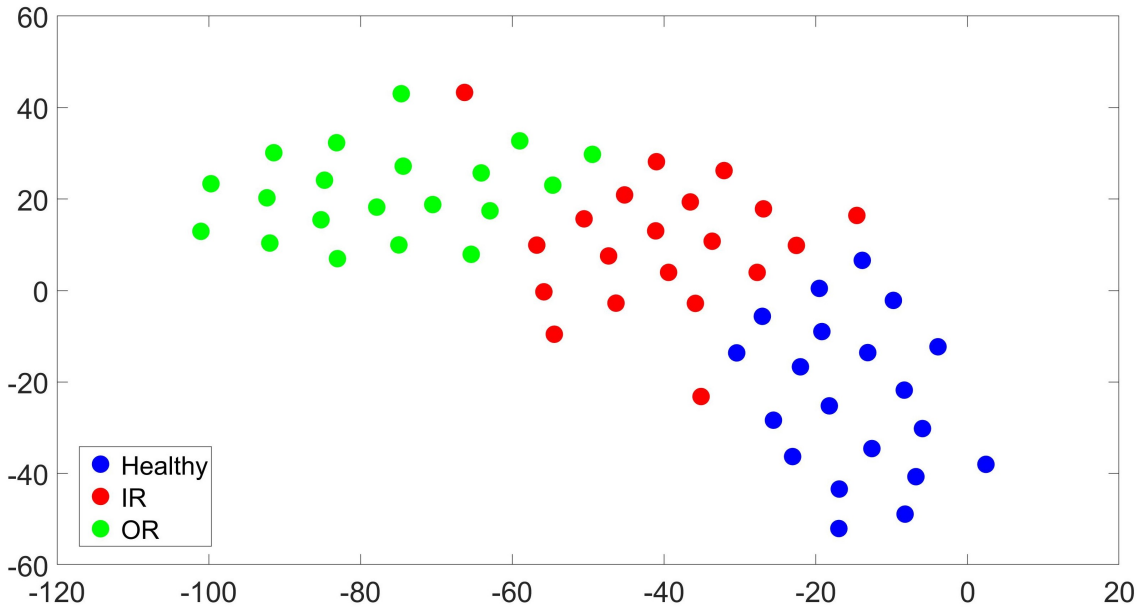
(b) CWRU 3

Figure 4.3: Decision boundaries of bearing data in a 2-D feature space in two operating conditions

CWRU data in a high dimensional feature space ($\Re^{108}$) for the same two operating conditions. The features consist of time, frequency and time-frequency characteristics of the first six intrinsic mode functions of the original denoised raw data (Kimotho & Sextro, 2014).



(a) CWRU 1



(b) CWRU 3

Figure 4.4: Marginal distributions of a high dimensional feature space visualized with t-SNE for two operating conditions of the CWRU data

From Figure 4.4, it was clear that the marginal distributions of the three classes were very different in the two operating conditions which meant that the decision boundaries learned by a model trained with CWRU 1 data would be different from those of a model trained with CWRU 3 data. The combination of data and its distribution is the domain

of data. Again, just like with the reduced feature set of Figure 4.3, the implication is that if the test data is drawn from a domain other than the one used to train the prediction model, there is a higher than random chance that the test point will be mis-classified.

Figure 4.5 shows the RCMFE values of normal and faulty bearings of CWRU data. At the higher scales (see section 2.3.1), the RCMFE of faulty bearings is lower than that of healthy ones which fluctuate around a constant value of 1. This observation is expected as a fault produces periodic impact in the vibration signal every time it is encountered thus increasing the regularity of the data or inversely, reducing its structural complexity and hence the diminishing RCMFE. Clear differences between the four categories of data are visible.

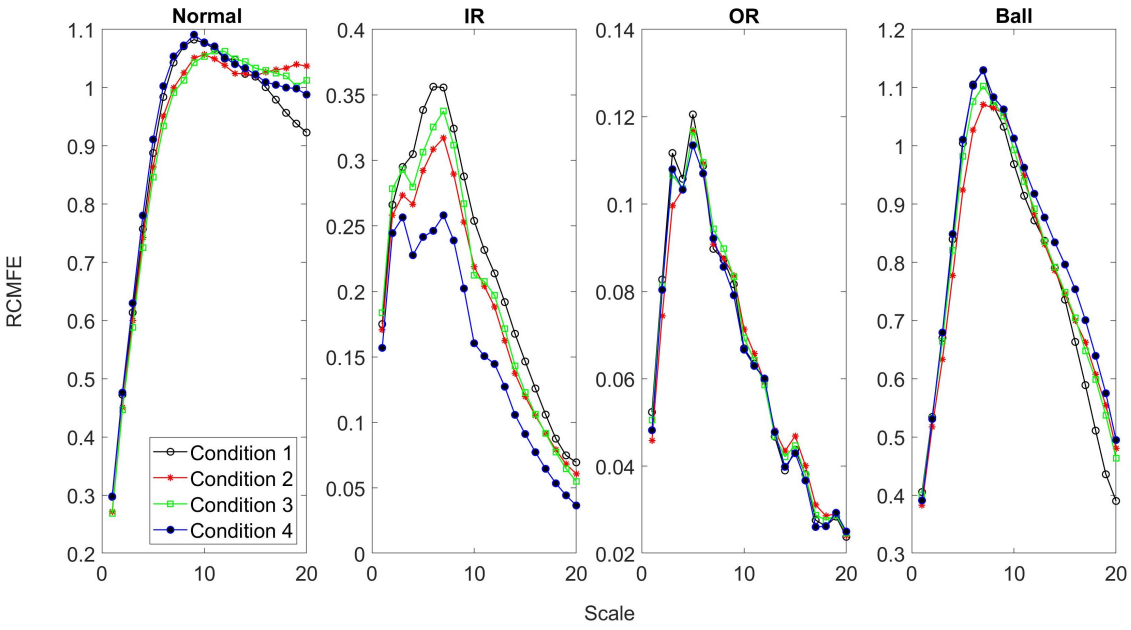


Figure 4.5: RCMFE values for healthy and faulty bearings for CWRU data

On the other hand, a healthy bearing's RCMFE fluctuates around a constant value across the higher scales indicating both a randomness of the data but also a richness in terms of the information contained about the machine. RCMFE behaves consistently across operating conditions indicating its robustness as a feature where conditions are changing.

There is more volatility of the RCMFE values of CWRU data when compared at different fault sizes but the behavior is still consistent scale-wise as seen in Figure 4.6. Because in practice a test sample may have any diameter of fault size within a reasonable range, the classifier's performance will be boosted by having data from all the fault sizes available in the training set.

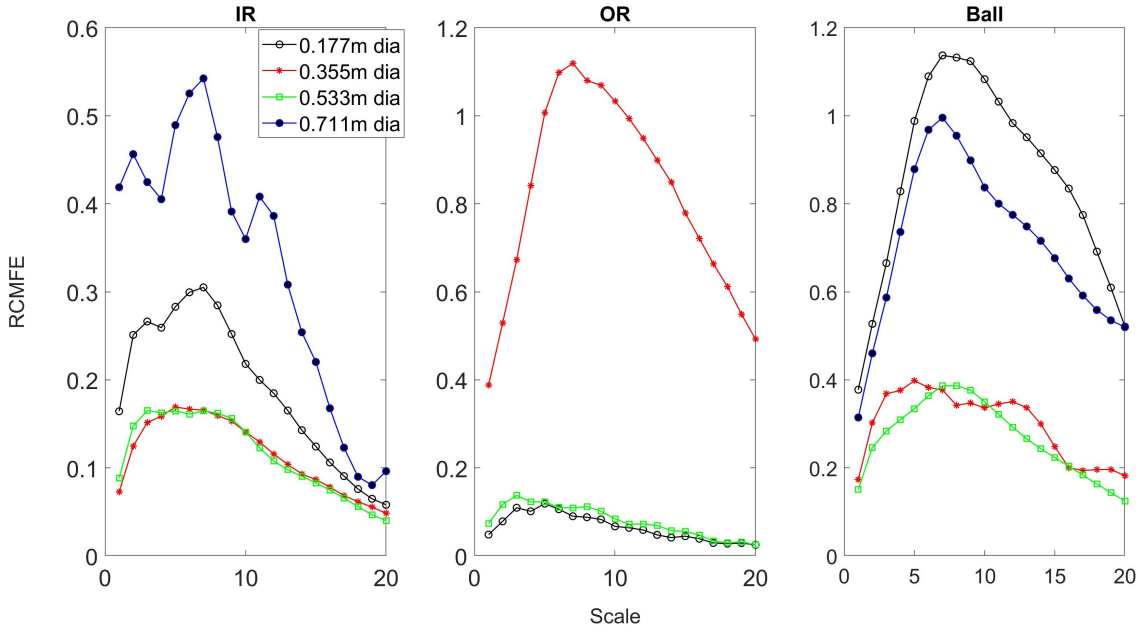


Figure 4.6: RCMFE values for different fault sizes for CWRU data

Figure 4.6 also seems to indicate that although IR fault is readily distinguished from the other two faults, OR and ball fault data are less obviously differentiable. The diagnostic procedure thus organically divided into two stages. The first stage was concerned with fault detection and a binary classifier was trained on healthy and faulty data; Figure 4.5 indicates that the two groups are readily differentiable. The training data was extracted from a single operating condition and tested against RCMFE features from all the other operating conditions. In the second stage where fault isolation was performed, several classifiers were built: the first was a multi-class classifier trained on data from the three faults. From Figure 4.6, it was expected that this classifier would have high recall for IR fault i.e. this fault will be rarely confused for the other two classes and vice versa. The implication is that if a fault was identified as IR fault from this classifier, there

was a high probability that the assignment of an IR fault label was accurate. However, since OR and ball faults were more likely to be confused, once a sample was classified as either of the two, it would be passed through a subsequent binary classifier trained exclusively on ball and OR data in an attempt to raise accuracy.

Figure 4.7 shows the first few milliseconds of some of the raw waveforms of the three classes of data for the Ottawa dataset. Again, there is no obvious distinguishing pattern in the groups.

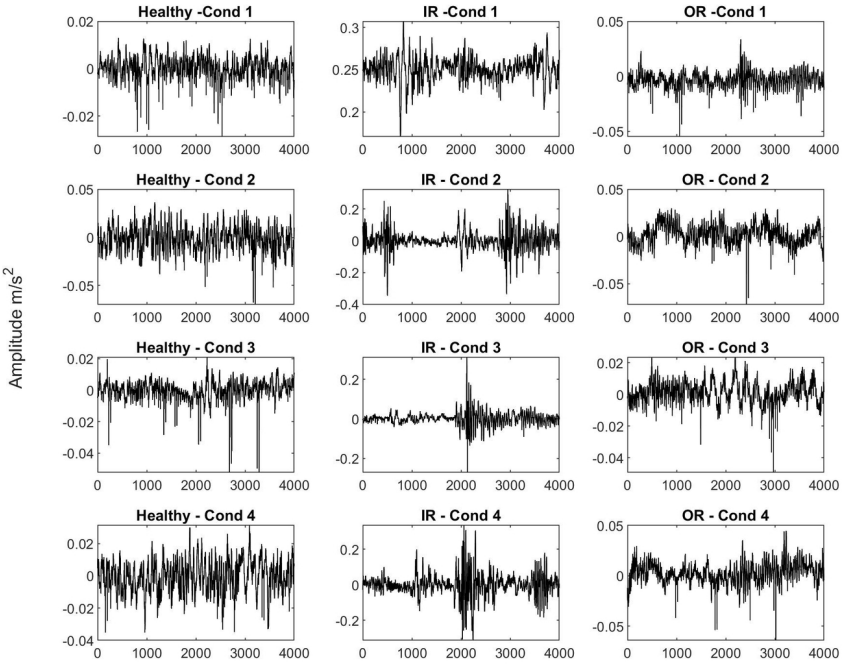


Figure 4.7: Vibration signals from the Ottawa dataset

Figure 4.8 shows distinct patterns of RCMFE values at higher scales with healthy bearing values fluctuating around a generally constant value of 1 while those of faulty data are monotonically decreasing.

Although Figure 4.8 implies good separability of the classes, the two stage approach was still chosen where fault detection was first performed followed by fault isolation.

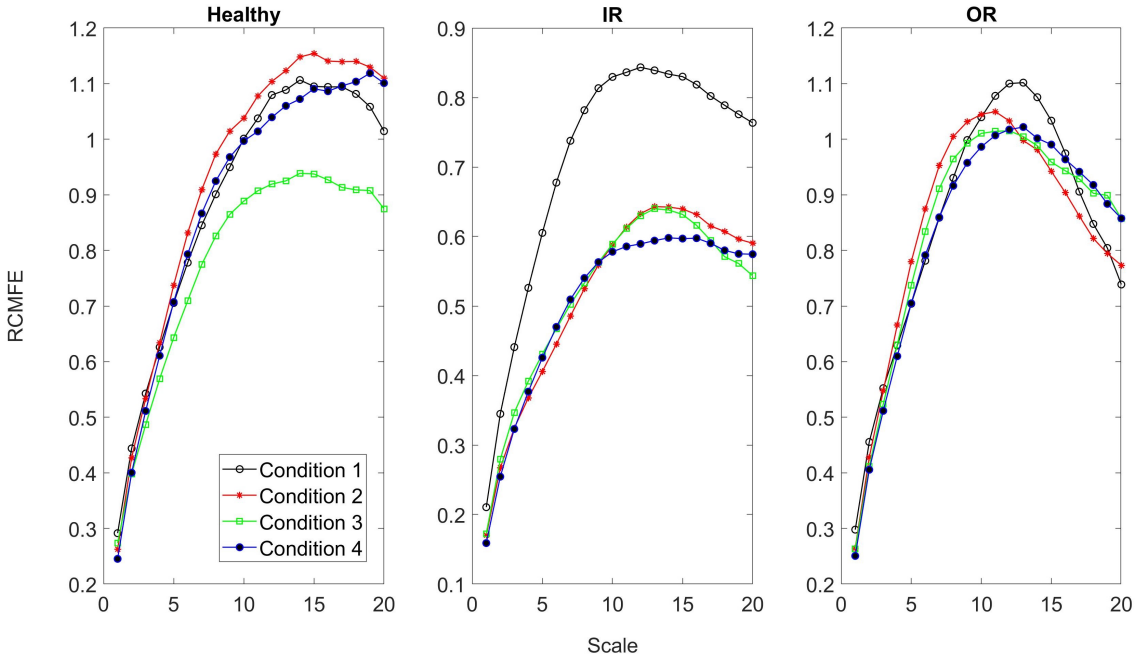


Figure 4.8: RCMFE for the Ottawa dataset

4.2.2 Classification - CWRU data

Table 4.1 shows that in the CWRU dataset, the healthy and faulty data are perfectly separable using RCMFE with a 100% accuracy on all the test data regardless of which source domain was used in training. The column headers indicate the source domain of the training data while test data includes data from all other conditions that were not used for training. Note that accuracy is computed by dividing the sum of correctly labeled samples (samples on the main diagonal) with the total number test samples as indicated in equation 4.1 (Grus, 2015).

$$Accuracy = \frac{\sum_{i=1}^{n} a_{ii}}{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}} \tag{4.1}$$

For fault isolation, a preliminary classifier was trained on all the three fault classes i.e. IR, OR and ball. As a fourth fault size was missing for the ball fault in the online repository, each class consisted of a mixture of three fault sizes only (Case Western

Table 4.1: Confusion matrices for the fault detection stage in the CWRU dataset

	CWRU 1	CWRU 2	CWRU 3	CWRU 4
	Predicted Label			
Actual Label	H: 60, 0 / F: 0, 380	H: 60, 0 / F: 0, 380	H: 60, 0 / F: 0, 380	H: 60, 0 / F: 0, 380
Accuracy	100%	100%	100%	100%

H = Healthy, F=Faulty

Reserve University Western Bearing Data Center, 2018). The operating condition used for training is given as the title of each confusion matrix in Table 4.2. The test data consisted of data from all the other operating conditions not used for training. Once again, the samples of the test data contained a mixture of the fault sizes. Just as indicated by Figure 4.6, the IR fault is perfectly separable from the other two faults. Further more, IR is never confused for OR and vice versa. However, since some ball faults are categorized as IR faults as seen in the last rows of the confusion matrices in Table 4.2, it would be worthwhile to train a binary classifier on IR and ball fault data. The accuracy of the IR/ball classifier is shown in (Table 4.3) where it is seen that the two faults exhibit good separability.

The entries of the matrices of Table 4.2 concerning OR and ball faults show that the two are frequently confused for each other as intimated by Figure 4.6. Thus, samples categorized as either fault should be passed through a binary classifier trained on ball and OR data only. Table 4.4 shows the accuracy of such classifiers. In the end, the accuracy of label assignment to a sample is then an average of accuracies of all the classifiers the samples pass through.

It is probable that the segmentation of fault sizes in the ball and OR fault may be a peculiarity of the CWRU setup and is thus possible that the phenomenon may lessen in another machine.

Table 4.2: Confusion matrices for the preliminary round in fault isolation for CWRU data

	CWRU 1			CWRU 2			CWRU 3			CWRU 4		
	Predicted Label											
Actual Label	IR	OR	Ball	IR	OR	Ball	IR	OR	Ball	IR	OR	Ball
IR	130	0	0	130	0	0	130	0	0	130	0	0
OR	0	63	27	0	81	9	0	77	13	0	71	19
Ball	10	51	69	9	47	74	12	29	89	6	26	98
Accuracy	74.8%			81.4%			84.5%			85.4%		

Table 4.3: Confusion matrices for binary classifiers trained with IR and Ball data for CWRU data

	CWRU 1		CWRU 2		CWRU 3		CWRU 4	
	Predicted Label							
Actual Label	IR	Ball	IR	Ball	IR	Ball	IR	Ball
IR	120	0	120	0	120	0	112	8
Ball	7	113	6	114	11	109	3	117
Accuracy	97%		97.5%		95.4%		95.4%	

Table 4.4: Confusion matrices for binary classifiers trained with OR and Ball data for CWRU data

	CWRU 1		CWRU 2		CWRU 3		CWRU 4	
	Predicted Label							
Actual Label	Ball	OR	Ball	OR	Ball	OR	Ball	OR
Ball	116	4	128	2	130	0	130	0
OR	34	56	32	58	33	57	43	47
Accuracy	81.90%		84.54%		85.0%		80.45%	

4.2.3 Classification - Ottawa dataset

The RCMFE features resulted in high accuracy in the two stages for the Ottawa dataset as seen in the confusion matrices of Tables 4.5 and 4.6. As observed in the CWRU dataset, healthy and fault data are easily distinguishable as well as IR and OR data.

Thus, RCMFE features were consistent in detecting fault and in at least differentiating IR and OR faults as verified with the CWRU and Ottawa datasets. The features also performed well in distinguishing IR and ball faults. However, a slight drop in

Table 4.5: Confusion matrices for fault detection with the classifiers trained with normal vs faulty data for the Ottawa dataset

	Cond 1	Cond 2	Cond 3	Cond 4
	Predicted Label			
Actual Label	H: 108, 2 / F: 2, 218	H: 110, 0 / F: 10, 210	H: 109, 1 / F: 5, 215	H: 107, 3 / F: 5, 215
Accuracy	95.8%	94.1%	95.2%	94.7%

H = Healthy, F=Faulty

Table 4.6: Confusion matrices for fault isolation with the classifiers trained with IR vs OR data for the Ottawa dataset

	Cond 1	Cond 2	Cond 3	Cond 4
	Predicted Label			
Actual Label	IR: 102, 8 / OR: 7, 103	IR: 107, 3 / OR: 14, 96	IR: 103, 7 / OR: 11, 99	IR: 100, 10 / OR: 4, 106
Accuracy	93.1%	92.2%	91.8%	93.6%

performance when differentiating ball and OR faults was observed.

4.2.4 Comparison with related works

As a result of preliminary investigations on the CWRU dataset as shown in Figures 4.5 and 4.6, it is seen that the RCMFE features are robust (similar behaviour) against operating conditions but are less so when the change in domain is obtained from changing fault sizes. A change in fault sizes therefore represents a more severe domain shift than changing operating conditions. For the results presented in section 4.2, each class was constituted of a single fault type but the samples were mixtures of the fault sizes available for use.

In published works that have tackled cross-domain diagnosis with the CWRU dataset, the domains are based on operating conditions. However, each fault size and type is

considered in a separate class with the exception of the 0.711 m diameter fault which is usually omitted as it is missing from the online repository. Thus, there are usually a total of 10 classes; the three fault types each with three fault sizes contributing nine classes and the healthy class bringing the number of categories to 10.

In order to compare the method proposed in this present research work with contemporary works, data was similarly prepared such that a class consisted of a single fault type and single fault size instead of a mixture od fault sizes. Table 4.7 shows some of the results for various fault sizes for this latter arrangement. Once again, the condition shown in the column header is the training domain while the test data was drawn from the other three domains/operating conditions but of the same fault size that used in training.

Table 4.7: Confusion matrices fault isolation considering a single fault size for the CWRU dataset

	CWRU 1 (0.177 m dia)				CWRU 2 (0.355 m dia)				CWRU 3 (0.533 m dia)				CWRU 4 (0.355 m dia)			
	Predicted Label															
Actual Label		IR	OR	Ball		IR	OR	Ball		IR	OR	Ball		IR	OR	Ball
	IR	30	0	0	IR	30	0	0	IR	30	0	0	IR	30	0	0
	OR	0	30	0	OR	0	30	0	OR	0	30	0	OR	0	30	0
	Ball	0	0	30	Ball	1	0	29	Ball	0	0	30	Ball	2	0	28
Accuracy	100%				98.8%				100%				97.7%			

In this alternative arrangement, RCMFE achieves even higher accuracy for fault isolation without utilizing domain adaptation.

In Table 4.8, a comparison with other works that have used CWRU data is given. The comparison is made in terms of the method of feature generation, the domain adaptation technique, the total number of classes considered, the fault size per class, average accuracy and whether the target domain data is required during training. The subscript * in the proposed approach indicates the arrangement of one fault size per class. Total direct comparability is not possible due to some minor differences. For instance, Li et al. (X. Li et al., 2018) only presented results considering CWRU 1 and CWRU 4 as the source and target domains while in the rest of the entries, each

condition was considered as the source in its own turn. In the approach reported here, diagnosis was divided into two stages i.e. fault detection followed by isolation while in the rest of the entries in Table 4.8, the diagnosis process was performed in one stage. Therefore the accuracy of the proposed method is an average of accuracy from the two stages.

Table 4.8: Comparison with other works using the CWRU dataset

Resource	Feature Generation	DA Method	No. of Classes	Fault sizes per class	Avg. Accuracy	Target Required
(X. Li et al., 2018)	CNN	Multi-layer & Multi-kernel MMD	10	1	>90%	Yes
(W. Zhang et al., 2017)	CNN	Adaptive batch normalization	10	1	>90%	No
(X. Li et al., 2019)	CNN GNN	Multi-kernel MMD	10	1	>90	Yes
(Tong et al., 2018)	PCA	MMD	10	1	100%	Yes
(Q. Wang et al., 2019)	CNN	Adversarial learning	10	1	>96%	Yes
*Proposed approach	RCMFE	None	10	1	>97%	No

As is seen in Table 4.8, most researchers have adopted deep learning for feature extraction (CNN) in order to perform cross-domain diagnosis. Also, MMD which is an optimization problem solved by iteration is commonly used to reduce distribution discrepancy. It is notable that generally, DA methods require target domain data during training which means the learning process must be repeated for each new domain of test data. In contrast, the RCMFE features approach proposed in this research work do not require deep learning for feature extraction and neither is any domain adap-

tation technique required. Consequently, the training process which can be expensive computationally need not be repeated each time new target data is acquired. For the generally equivalent performance, RCMFE features allow for cross-domain analysis while avoiding the iterative solving of complex objective functions.

In practical implementation, categorizing each fault size in an individual class does not benefit generalization. This is because, rather than an unseen test sample having a fault size exactly equal to one of those used for training, it is more probable that the fault will be in the range of the fault sizes used for training. Thus, in this work, classes were created based on the type of fault only and in the training data several fault sizes were included for each class. The more challenging nature of this latter arrangement is clear by comparing Figures 4.5 and 4.6 . The results of using RCMFE for domain-invariant diagnosis were compiled for publication under the title "Cross-domain bearing fault diagnosis with refined composite multiscale fuzzy entropy and the self organizing fuzzy classifier"(Gituku et al., 2021)

4.3 Prognosis

4.3.1 Preliminary Analysis

Figure 4.9 shows the raw waveforms of the last samples of a few of the bearings in the FEMTO-ST dataset and as expected, no useful distinct prognostic information is obvious mostly because of how noisy the data is.

The FEMTO dataset is considered a challenging dataset because of several reasons. First, the learning set consists of only two files per condition while the life duration of bearings has a big variance (from 1 to 7 hours). Secondly, the lifetimes and fault characteristics are different even for units operated under the same condition. Looking at Figure 4.10, the RMS patterns from beginning to end of life are very different for the learning sets of the three different conditions(different domains);raw RMS would not make an ideal feature for cross-domain prognosis. The same argument can be made

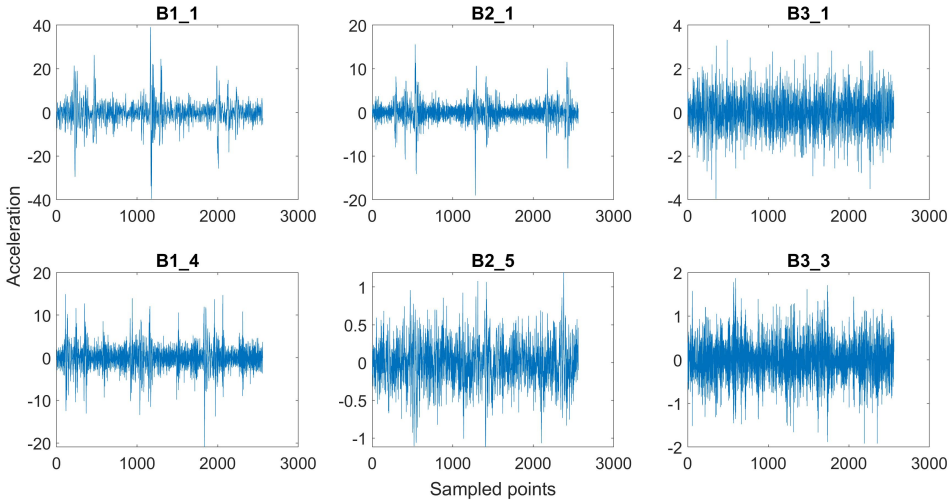


Figure 4.9: Raw Waveforms of select bearings

for raw kurtosis and shape factor as seen in Figures 4.11 and 4.12 complicating the task of obtaining useful information about the degradation of the test bearings from the training bearings (Lei et al., 2018).

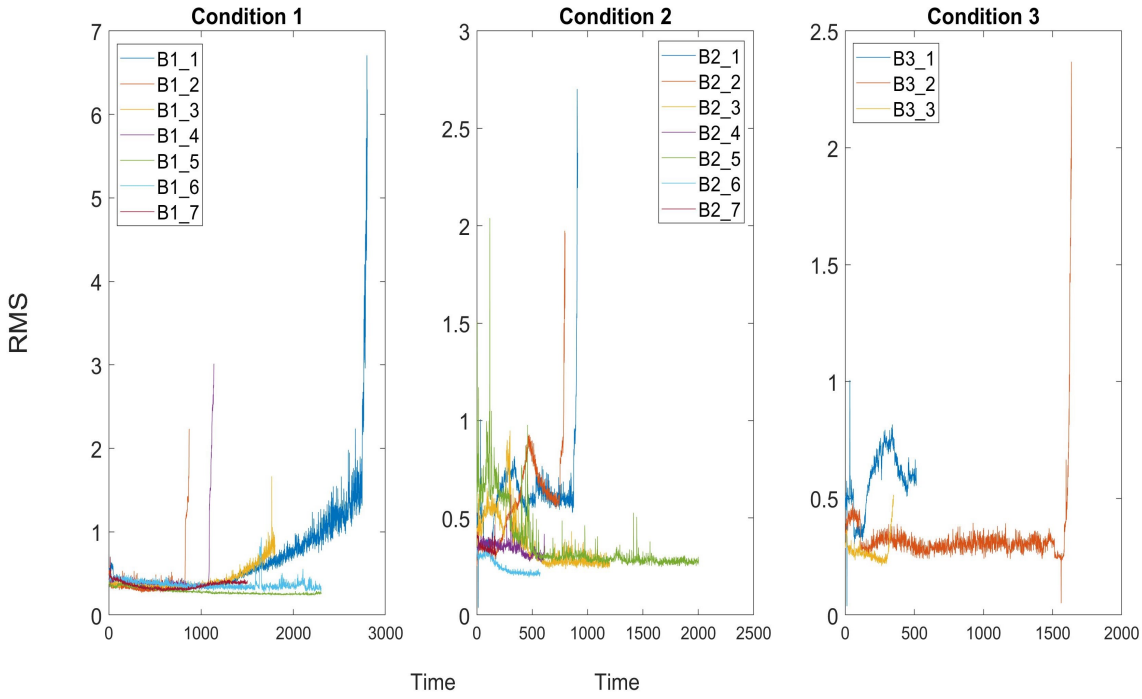


Figure 4.10: RMS of all the bearings in each operating condition

As seen in Figures 4.13,4.14 and 4.15, plots of the select time-domain features i.e RMS, kurtosis and shape factor, do not manifest clear monotonic trends and were thus not

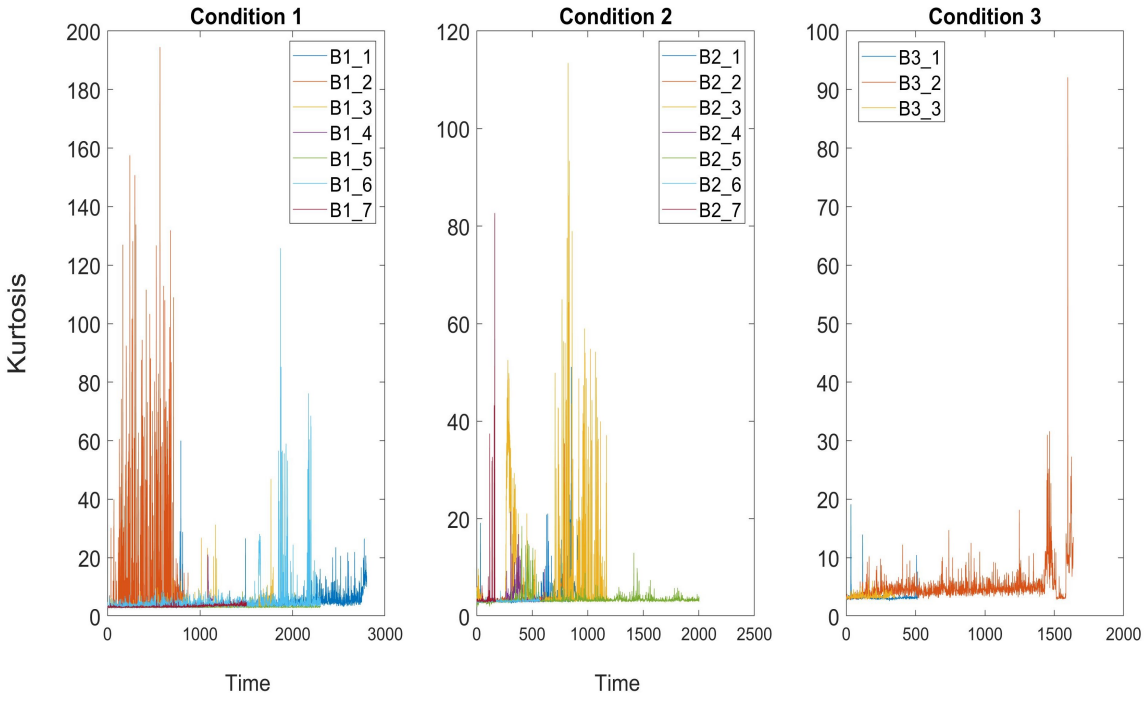


Figure 4.11: Kurtosis of all the bearings in each operating condition

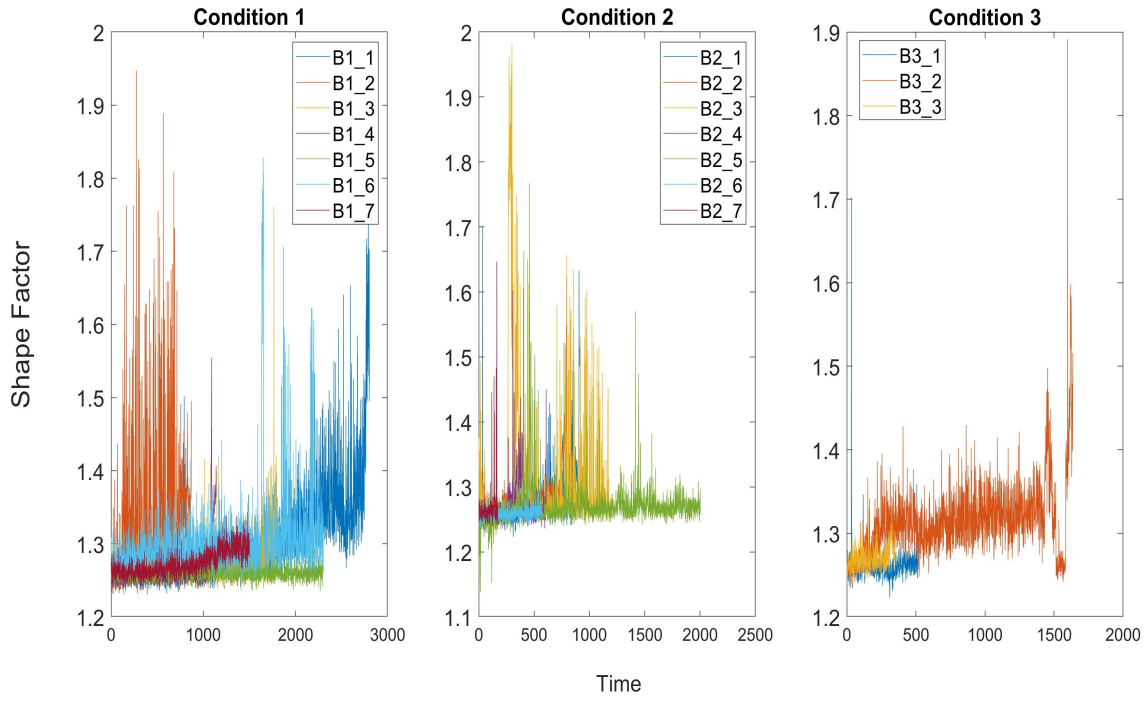


Figure 4.12: Shape factor of all the bearings in each operating condition

ideal for use as condition indicators.

Therefore, instead of using RMS, kurtosis and shape factor directly, their respective

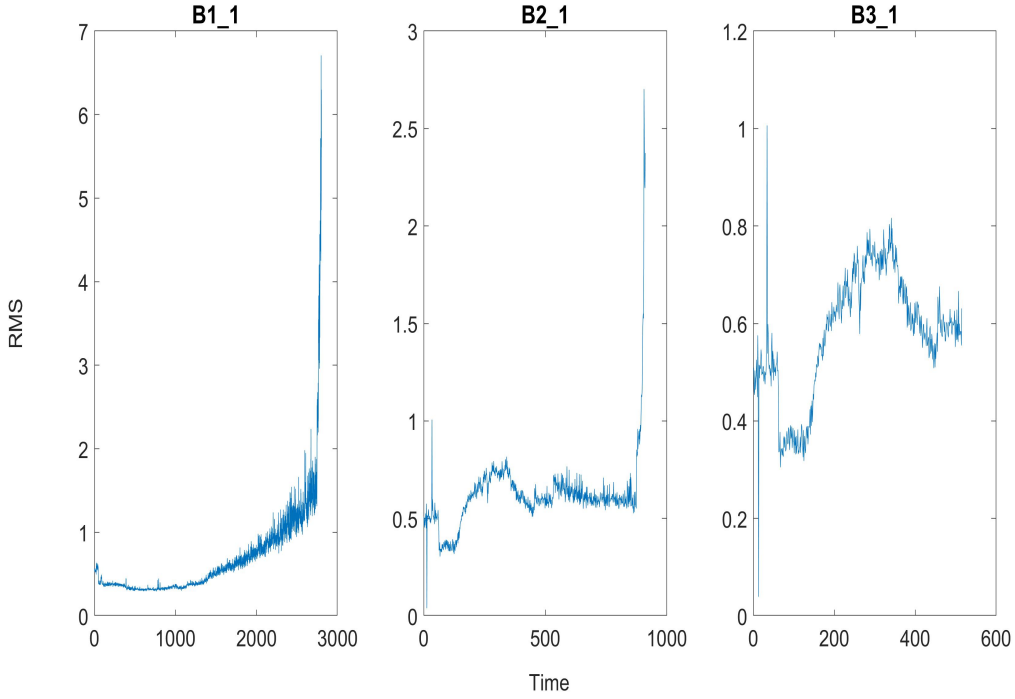


Figure 4.13: RMS of training set bearings

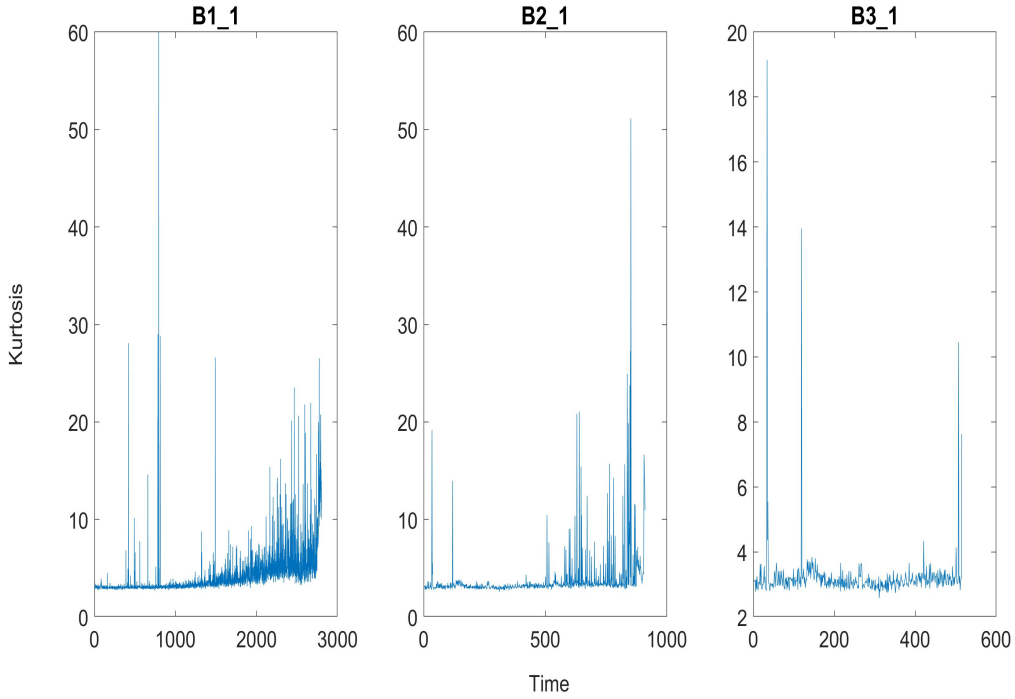


Figure 4.14: Kurtosis of training set bearings

hazard functions were adopted as the new health indicators (see Figure 4.16).

Equation (2.25) was used in computing the proposed health indicators with parameters

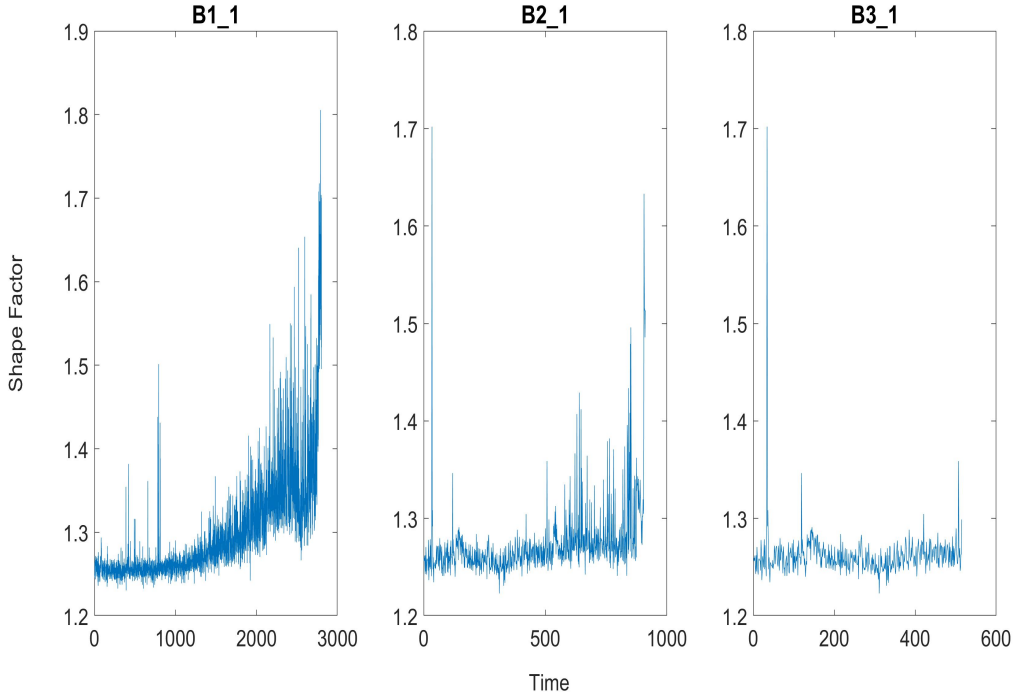


Figure 4.15: Shape Factor of training set bearings

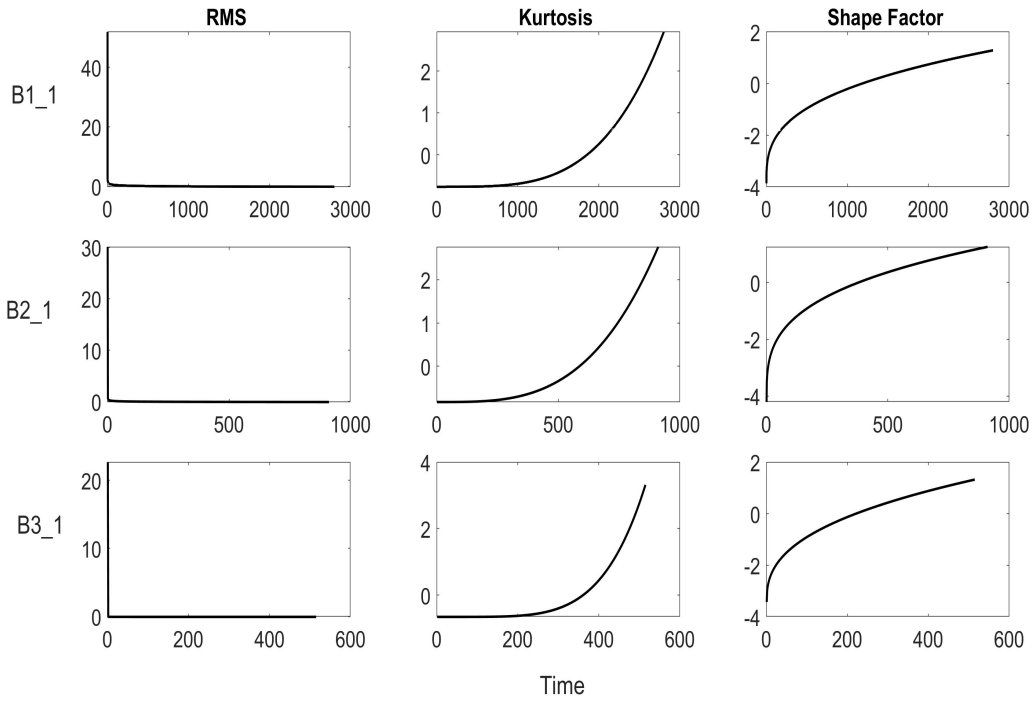


Figure 4.16: Hazard Functions of RMS, kurtosis and the shape factor

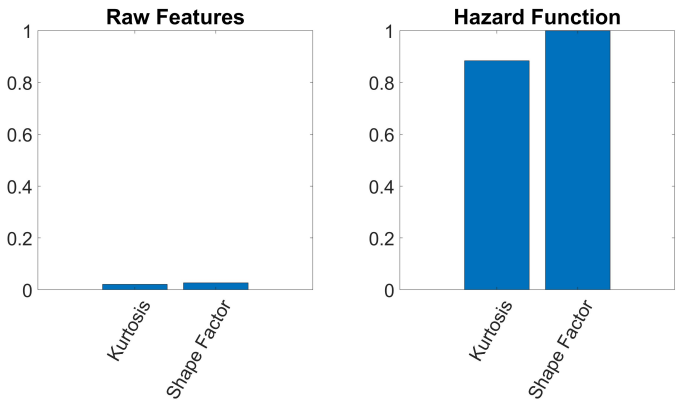
α and β being estimated separately for each raw feature using the *wblfit* function in the scientific computation software MATLAB by MathWorks®.

From Figure 4.16, it is obvious that the monotonic trend of the hazard function health indicators is much better compared to that of the raw features. In addition, the sub-plots show a similar pattern for all operating conditions which is advantageous for creation of an RUL prediction model that is applicable across different operating conditions. Conversion of the raw feature to its hazard function also significantly reduces noise. For this data, the RMS hazard function indicates high probability of early mortality instead of risk that increases with operating time as expected of bearings. For this reason, the RMS hazard function was dropped as a health indicator leaving the kurtosis and shape factor hazard functions.

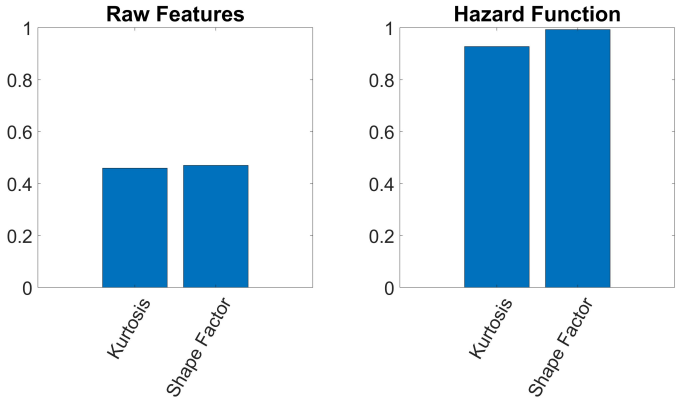
There are several metrics used for quantifying the suitability of health indicators given that their performance has great influence over the success of modeling degradation and RUL prediction accuracy (Lei et al., 2018; Duoung et al., 2018). Three commonly used metrics were contrasted for the direct time-domain features and the features' hazard functions.

In reality, components undergoing degradation are unable to recover or heal themselves and thus an appropriate health indicator should have an increasing or decreasing monotonic trend. This characteristic is known as monotonicity and is an inherent property of the indicator itself. In MATLAB, the metric can computed by calling the *monotonicity* function with appropriately formatted input ("Monotonicity", 2018). A value of 1 indicates perfect monotonicity while 0 indicates non-monotonicity. Figure 4.17a compares the monotonicity of the direct time-domain features versus their hazard function and as can be seen, the hazard function improves monotonicity appreciably.

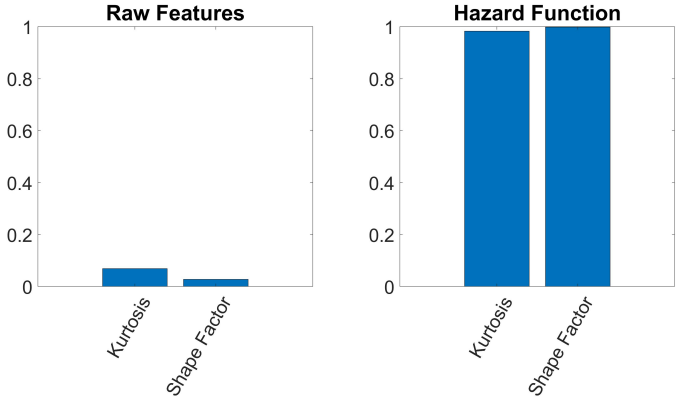
Robustness is yet another commonly used metric and it measures variability of the value of the health indicator at failure based on the paths followed by the feature in multiple run-to-failure experiments ("Prognosability", 2018). There are several reasons why such variability occurs including randomness of the degradation process, sensor noise and more pertinently, changing operating conditions (Lei et al., 2018; Duoung et



(a) Monotonicity



(b) Robustness



(c) Trendability

Figure 4.17: Comparing characteristics of the health indicators for the raw features versus their hazard functions

al., 2018). Data from different operating conditions constituted the multiple run-to-failure experiments in this case. In MATLAB, robustness may be computed by calling the *prognosability* function. A robust feature has less variation at failure measured against the range between its initial and final value("Prognosability", 2018). Figure

4.17b shows the superiority of the robustness of hazard functions compared to the direct time-domain features.

The trendability metric correlates the health indicator with time. Since degradation worsens over time, the evolution of the health indicator is also expected to relate accordingly to the operating time (Lei et al., 2018). A slightly modified definition is used when the *trendability* function in MATLAB is called. In this function, the similarity of the paths a health indicator takes in several run-to-failure experiments is measured ("Trendability", 2018). Figure 4.17c compares the trendability of health indicators from the direct time-domain features against those from the hazard functions. The high trendability of hazard function health indicators means that a single underlying function can be used to accurately explain a particular indicator e.g. the kurtosis hazard function regardless of the operating condition it was drawn from. It is to be recalled that the multiple run-to-failure experiments are derived from lifetime data from the different operating conditions. The high trendability of the hazard function is also inferred by comparing Figures 4.13, 4.14 and 4.15 against Figure 4.16 where it is clearly observed that hazard function health indicators have similar shapes across the three operating conditions. The formulas for the three metrics are given in the appendix.

4.3.2 Remaining Useful Life Prediction

The best model from training (using the training, validation and in-training test sets) was used to predict the RUL of the actual test sets as described in section 3.2.2. The best model had 2 nodes in the hidden layer, an R^2 score of 1 for the training set (B1_1), 0.997 for the validation set (B2_1) and 0.999 for the in-training test set (B3_1). Table 4.9 shows the R^2 scores obtained for each of the test bearings. In the last column the true RUL of each of the test bearings is also given (Linde, 2012).

The model performs well on some of the bearings e.g. B1_4, B1_5, and B1_6 and poorly

Table 4.9: Results from initial training

Bearing Name	R^2 Score	True RUL
B2_3	-0.7	39%
B1_7	-0.09	34%
B2_7	0.51	26%
B1_3	0.59	24%
B2_4	0.77	19%
B3_3	0.77	19%
B2_5	0.89	14%
B1_5	0.97	7%
B1_6	0.98	6%
B1_4	0.99	3%

on others e.g. B2_3 and B1_7 as noted by the respective high and low R^2 scores. The poor performance does not seem to be connected to any particular operating condition indicating the decision to use training data from all the three operating conditions was sound. However, looking at the true RUL column, the selected model performed best on the bearings close to end of life i.e. with a small RUL and worst for the bearings furthest from end of life i.e. with the largest RUL. The bearings are listed in order of decreasing RUL.

The model seems to be overfitting even though validation was performed during training; the issue was most likely connected to the nature of the testing data versus the training data. From this viewpoint, the result is sensible considering the models were trained with full run-to-failure data i.e the RUL progressively decreased from 100% to 0%. The training data was full lifetime data while the testing data was censored/truncated. Thus, the models perform best on bearings nearing end of life as such data was closest to the training data. A second round of training was then performed with all the training datasets expanded. The supplemental data was created by truncating the inputs and corresponding targets at 10% intervals i.e there were additional datasets with inputs and targets from 100% to 90% RUL, 100% to 80% RUL, 100% to 70% RUL, all the way upto 100% to 10% RUL. The original length of the training and target data from 100% to 0% RUL was also included. The supplementation was done

in order to have data sequences in training similar to those that would be encountered in testing. The same learning process described in the methodology was applied to the expanded training dataset and the best model selected. Table 4.10 shows the result of re-training.

Table 4.10: Results from training with the data expanded at 10% intervals

Bearing Name	R^2	True RUL
B2_3	0.73	39%
B1_7	0.90	34%
B2_7	0.99	26%
B1_3	1.00	24%
B2_4	0.98	19%
B2_6	0.98	19%
B3_3	0.98	19%
B2_5	0.95	14%
B1_5	0.87	7%
B1_6	0.86	6%
B1_4	0.81	3%

Now, the general performance is significantly improved over that recorded in Table 4.9 as indicated by the R^2 value. The prediction accuracy of the RUL for bearings furthest from end of life increased while performance on the bearings close to end of life (B1_4, B1_5, and B1_6) noticeably dropped. It is likely that supplementation changed the underlying distribution of the training data and introduced some bias which affected the test bearings with the shortest RULs. The best performance was recorded for bearings with true RUL between 21-30% RUL (B2_7, B1_3), followed by those with true RUL between 11-20% (B2_4, B2_6, B3_3 and B2_5). Overall, the variance of the R^2 was reduced.

In attempting to improve performance further, the model was analysed when the truncated supplemental training data was included at even finer intervals and thus 1% interval data was generated. Because the largest true RUL is 39%, the finer data was added from 50% RUL. In addition to the full lifetime data, the training data now also contained data and targets from 100% to 90% RUL, 100% to 80% RUL, 100% to 70% RUL, 100% to 60% RUL, 100% to 50% RUL, 100% to 49% RUL, 100% to 48%, 100%

to 47% RUL and all the way up to 100 to 1% RUL. Table 4.11 shows the result of the third round of training.

Table 4.11: Performance after the third round of training

Bearing Name	R^2	True RUL
B2_3	0.57	39%
B1_7	0.80	34%
B2_7	0.97	26%
B1_3	0.98	24%
B2_4	1.00	19%
B2_6	1.00	19%
B3_3	1.00	19%
B2_5	0.99	14%
B1_5	0.93	7%
B1_6	0.92	6%
B1_4	0.89	3%

It is clear that the general performance is significantly improved over that recorded in Table 4.9. However, expanding the data using finer truncation intervals seems have an opposite effect to what was recorded in Table 4.10. The performance on bearings nearing end of life rose from what was recorded in Table 4.10 while that of bearings with the largest RUL dropped. The best performance was recorded for bearings with between 11-20% true RUL followed by those with true RUL between 21% to 30% .

Through this initial prognosis steps, three key insights were gained. First, adding supplemental data truncated at 10% resolution improved performance significantly. The best performance was recorded for the bearings with RULs in the mid-range between the largest existing RUL and the smallest RUL i.e between 11% to 20% and between 21% to 30%. Secondly, adding the 1% interval truncated data tipped the prediction accuracy to the opposite of what was obtained with the 10% interval truncated data only (see Tables 4.10 and 4.11). Thirdly, when training with full lifetime data only (Table 4.9), the performance was best for the bearing with the lowest RUL and dropped as the RUL increased.

In view of these three findings, a different approach was taken when adding the data

truncated at 1% intervals. Instead of using a single model, several models were now adopted where each was trained with the 1% interval-truncated data added for specific RUL ranges. The data for training all the models would include the original full lifetime data i.e. with 100% to 0% RUL as well as the 10% interval-truncated data i.e. training inputs and target from 100% to 90% RUL, 100% to 80% RUL, 100% to 70% RUL, 100% to 60% RUL, all the way upto 100% to 10% RUL. This is the same data used to train the model for the results recorded in Table 4.10. Next, the training data for each of the models would be expanded using a specific range of the 1% interval-truncated data. For instance, the training data for the model named Model 6 in Table 4.12 also included 1% interval-truncated data between 50 to 41% RUL i.e the training data for Model 6 had data with 100% to 90 % RUL, 100% to 80 % RUL, 100% to 70 % RUL, 100% to 60 % RUL, 100% to 50 % RUL, from 100 to 49% RUL, 100 to 48% RUL, ..., 100% to 40% RUL, 100% to 30 % RUL, 100% to 20 % RUL, 100% to 10 % RUL and 100% to 0 % RUL. This model is named Model 6 because the 50-41 range is the sixth grouping if the length 100% to 0% was divided into groups at descending 10% intervals. Similarly, Model 7's training data also included 1% interval-truncated data between 40% to 31%, Model 8's data between 30 to 21%, Model 9's data between 20% to 11% and Model 10's data between 10% to 1%. Obviously, Model 1's data should include 1% interval-truncated data between 100% to 91%. Since the longest RUL in the test data is 39%, Table 4.12 only shows the performance of Models 6 to 10 on the test set bearings. In the last column the performance of the original model trained without supplemental data is appended.

The more darkly shaded cells indicate where the interval in which the true RUL of a bearing lies matches the interval of the additional 1% truncated data. For instance, Model 7 was trained with the 1% interval-truncated data between 100 to 40-31% RUL. That is why in Model 7's column the cells for test bearings whose true RUL are within the 40-31% range (i.e. B2_3 and B1_7) are darker than the neighbouring .

Table 4.12: Training successive models with expanded data

Bearing Details		R^2 Scores					
Name	True RUL	Model 6 (50-41%)	Model 7 (40-31%)	Model 8 (30-21%)	Model 9 (20-11%)	Model 10 (10-1%)	Original Model
B2_3	39%	0.97	0.92	0.82	0.61	0.29	-0.70
B1_7	34%	0.99	0.99	0.94	0.83	0.61	-0.09
B2_7	26%	0.92	0.97	1.00	0.97	0.88	0.51
B1_3	24%	0.91	0.96	1.00	0.99	0.92	0.59
B2_4	19%	0.83	0.90	0.96	1.00	0.98	0.77
B2_6	19%	0.83	0.90	0.96	1.00	0.98	0.77
B3_3	19%	0.83	0.90	0.96	1.00	0.98	0.77
B2_5	14%	0.76	0.85	0.92	0.98	1.00	0.89
B1_5	7%	0.64	0.74	0.84	0.92	0.98	0.97
B1_6	6%	0.62	0.72	0.82	0.91	0.97	0.98
B1_4	3%	0.55	0.66	0.76	0.87	0.95	0.99

As seen from Table 4.12, a model performs best on a bearing if the RUL of the bearing falls within the range of the additional 1% interval-truncated data used to train that particular model. Fortunately, designing the framework in this manner also makes the models trained with supplemental 1% interval-truncated data in the ranges surrounding the bearing's true RUL range have high accuracy in predicting that bearing's RUL. This is notable when looking at the lighter shade cells on either side of a darker shade cell. Because Models 6 to 10 show this consistent pattern of behaviour for the available RUL ranges, it is not an unreasonable assumption to conclude that the frame work would work just as well if extended upwards for bearings with RULs bigger than 40%. Once the predicted RUL output falls below 10%, the features data may also be passed to the original model which was predisposed to small RUL due to its exclusive use of full lifetime data in training (last column in Table 4.12).

From the results in Table 4.12, a possible scheme for the successive deployment of the models can be derived as follows. It is assumed that collection of condition monitoring data begins at the start of bearing operation where RUL is 100 %. Thus, the condition monitoring features should be fed to Model 1 whose training data enables it to perform best when RUL is between 90% and 100%. As soon as Model 1 gives an output that is around its lower limit of optimality e.g. 90 % - 89% RUL, incoming condition

monitoring data for the bearing should now be passed to Model 2 which is optimized for the next RUL range i.e. 90 to 81 % RUL. Fortunately, as Models 1 and 3 still perform very well on the data when the RUL is between 90 to 81 % (as shown from the results of Table 4.12), the maintenance engineer may even choose to compute the RUL as a weighted average of Models 1, 2 and 3 with more weight being given to Model 2. Again, when Model 2's output is around its lower limit of optimality i.e. 80 % - 79% RUL, incoming data should be fed to Model 3. Thus, the process would continue until the output of Model 9 is around 10 - 9% RUL where incoming data is fed to Model 10 or the initial model trained with full lifetime data only. At this stage the RUL can also be computed as an average of Model 10 and the initial model. Generally bearings do not self heal and thus RUL decreases monotonically, the deployment scheme just described will work well. This concept of successive deployment of Models 1 through 10 is illustrated in Table 4.13.

Table 4.13: Proposed deployment of the successive models for RUL prediction

Predicted RUL Output									
100-91%	90-81%	80-71%	70-61%	60-51%	50-41%	40-31%	30-21%	20-11%	10-1%
Model 1									
	Model 2								
		Model 3							
			Model 4						
				Model 5					
					Model 6				
						Model 7			
							Model 8		
								Model 9	
									Model 10

The percentage RUL ranges in the second row of Table 4.13 indicate the optimal performance range of a model which coincides with the interval where the 1% interval-truncated training data was added. As was indicated in Table 4.12 the output of each of the models in its optimal range of performance is highly accurate and for the mid-range models (Model 2-9), an averaging scheme may be adopted as already described.

RUL estimation is a process that is changing in both accuracy and uncertainty as time progresses. This successive deployment of models is an attempt to update both these

things by using models trained with apriori information of the current RUL of the bearing.

4.4 Summary

In this section that discussed the results of using features resistant to domain shift, it was found that RCMFE features and hazard function health indicators performed remarkably well though with a few surmountable challenges. Using RCMFE, one is able to detect fault almost all the time (94% accuracy for the Ottawa dataset and 100% accuracy for the CWRU data). Also, OR and IR fault are very distinct using RCMFE such that neither is confused for the other. With regard to isolating among the three types of fault, i.e. IR, OR and ball fault, a hierarchy of classifiers is a good proposition beginning with one trained on all three faults; because IR fault is well distinguishable from ball and OR fault (see Tables 4.2 and 4.6). In a second stage, the test sample may be run through two binary classifiers: one trained on IR versus ball and the other trained on IR versus OR. The purpose of this stage would be to confirm whether test data has IR fault as these first two stages of fault isolation are finely tuned to IR fault. Finally, the third stage classifier should be trained on OR versus ball fault. This hierarchy applies if each class of the training data contains a mixture of fault sizes. As seen from Table 4.7, if the training data is arranged such that each class consists of a single fault type and a single fault size, no hierarchical system is needed and classification can be performed once.

In the FEMTO data, there is a change in distribution because of the different operating conditions. However, even with in the same operating condition, data have different characteristics as can be seen in Figures 4.10, 4.11 and 4.12. Fitting the shape factor and kurtosis to a Weibull distribution and using the hazard function as features eliminates the differences and introduces trendability. The fact that training data is full lifetime data while testing data is truncated/censored also introduces a change is distribution that affects performance. This was mitigated by supplementing the training

data with truncated versions of itself which significantly improved performance of the models when applied to target data.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

This work set out to find and utilize domain invariant features such that target data is not required when training data driven models for condition monitoring of rolling element bearings. To this end, the following contributions were made.

1. Introduction of the Refined Composite Multiscale Fuzzy Entropy feature for domain invariant diagnosis of rolling element bearings.
    a. The concept of RCMFE was demonstrated using two well known systems - the simple pendulum and Lorenz attractor. It was shown that RCMFE attempts to reconstruct the dynamics of the system in a technique similar to time-delay embedding. By this definition, RCMFE remained consistent even in different domains as depicted by changing operating conditions.
    b. RCMFE achieved results as good as those of contemporary works without any domain adaptation technique. Hence, RCMFE avoids re-training of models each time a new target is encountered.
2. Introduction of hazard-based health indicators for domain invariant prognosis.
    a. Hazard-based health indicators were shown to be trendable domain invariant features for use in cross-domain diagnosis.
    b. A training scheme using data from the three different operating conditions of the FEMTO-ST dataset in order to promote domain invariance was also successfully applied.
    c. An approach for supplementation of training data to more closely resemble testing data was proposed and successfully applied.

5.2 Recommendations For Future Work

The work done in this research is by no means exhaustive and there several possible directions that can be pursued.

1. Determination of the best model of for mapping the RUL vector. In this work, a linear model was used but other possible choices such exponential or piece-wise models should be explored.

2. Using multichannel data: Sometimes the setup may be such that is possible to collect more than one stream of data e.g. the CWRU setup incorporated two accelerometers. It may be worthwhile to explore if incorporation of data from multiple channels is beneficial using the developed features.

3. The use of probabilistic algorithms for mapping prognostic data should be explored. Such algorithms automatically generate a prediction interval which is a useful metric in decision making.

4. The success of these features for use with data from similar but not same machines needs to be investigated. For instance, training with data from one publicly available source and testing on another.

REFERENCES

Al-Raheem, K. F., & Abdul-Karem, W. (2010). Rolling bearing fault diagnostics using artificial neural networks based on laplace wavelet analysis. *International Journal of Engineering, Science and Technology*, *2*, 278–290.

Angelov, P., & Gu, X. (2019). *Empirical approach to machine learning.* Springer Nature.

Angelov, P., & Yager, R. (2012). A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, *41*, 163–185.

Antoni, J. (2006). The spectral kurtosis: a useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing*, *20(2)*, 282–307.

Antoni, J., & Randall, R. B. (2006). The spectral kurtosis: application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, *20*, 308–331.

Azami, H., Fernandez, A., & Escudero, J. (2017). Refined multiscale fuzzy entropy based on standard deviation for biomedical signal analysis. *Medical & Biological Engineering & Computing volume*, *55*, 2037–2052.

Barden Precision Engineering. (N.D.). *Bearing Failure: Causes and Cures.*

Barszcz, T., & Jablonski, A. (2011). A novel method for the optimal band selection for vibration signal demodulation and comparison with the kurtogram. *Mechanical Systems and Signal Processing*, *25*, 431–451.

Cao, N., Gao, J., & Cui, B. (2020). Bearing state recognition method based on transfer learning under different working conditions. *Sensors*, *20*, 1–12.

Case Western Reserve University Western Bearing Data Center. (2018). *Seeded fault test data.* (data retrieved from Case Western Reserve University Bearing Data Center Website)

Chapelle, O., & Zien, A. (2005). *Semi-supervised classification by low density separation.*

Chatterton, S., Borghesani, P., Pennacchi, P., & Vania, A. (2014). Optimal frequency

band selection for the square envelope spectrum in the diagnostics of rolling element bearings. In *Proceedings of the 26th conference on mechanical vibration and noise.*

Che, C., Wang, H., Fu, Q., & Ni, X. (2019). Deep transfer learning for rolling bearing fault diagnosis under variable operating conditions. *Advances in Mechanical Engineering*, *11(12)*, 1–11.

Chen, W., Wang, Z., Xie, H., & Yu, W. (2007). Characterization of surface emg signal based on fuzzy entropy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*, 266–272.

Costa, M., Goldberger, A. L., & Peng, C. (2002). Multiscale entropy analysis of complex physiologic time series. *Physical Review Letters*, *89*, 068102-1–068102-4.

Cubillo, A., Perinpanayagam, S., & Esperon-Miguez, M. (2016). A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*, *8(8)*, 1–21.

da Costa, P. R. d. O., Akcay, A., Zhang, Y., & Kaymak, U. (2019). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering and System Safety*, *195*, 1–30.

Dawn, A., Kim, N. H., & Choi, J.-H. (2015). Practical options for selecting data–driven or physics–based prognostics algorithms with reviews. *Reliability Engineering and System Safety*, *133*, 223-236.

Delgado-Bonal, A., & Marshak, A. (2019). Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy*, *21*, 1–37.

Detecting faulty rolling–element bearings. (N.D.). `https://www.bksv.com/media/doc/BO0210.pdf`. (Accessed: 05-04-2021)

Ding, X. (2009). *Fault detection and isolation for railway vehicle suspensions* (PhD Thesis). University of Leeds.

Duoung, B. P., Khan, S. A., Shon, D., Im, K., Park, J., Lim, D.-S., ... Kim, J.-M. (2018). A reliable health indicator for fault prognosis of bearings. *Sensors*

(Bassel), *18*, 3740–3756.

Dyer, D., & Stewart, R. (1978). Detection of rolling element bearing damage by statistical vibration analysis. *Journal of Mechanical Design*, *100(2)*, 229–235.

El-Thalji, I., & Jantunen, E. (2015). A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mechanical Systems and Signal Processing*, *8*, 1–17.

El-Thalji, I., & Jntunen, E. (2015). A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mechanical Systems and Signal Proessing*, *60*, 257–272.

Function fitting a neural network. (2010). `https://www.mathworks.com/help/deeplearning/ref/fitnet.html?searchHighlight=fitnet\&s\_tidsrchtitle`. (Accessed: 05-09-2021)

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *17*, 2096–2030.

Gituku, E. W., Kimotho, J. K., & Githu, J. N. (2021). Cross-domain bearing fault diagnosis with refined composite multiscale fuzzy entropy and the self organizing fuzzy classifier. *Engineering Reports*, *3*, 1–17.

Gretton, A., Borgwadt, K. M., Rasch, M. J., Schoelkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*, 723–773.

Grus, J. (2015). *Data sciencce from scratch.* O'Reilly Media, Inc.

Gustafsson, O. G., & Tallian, T. (1962). Detection of damage in assembled rolling element bearings. *Tribilogy Transactions*, *5(1)*, 197–209.

He, J., Li, X., Chen, Y., Chen, D., Guo, J., & Zhou, Y. (2021). Deep transfer learning method based on 1d-cnn for bearing fault diagnosis. *Shock and Vibration*, *2021*, 1–16.

He, M., & He, D. (2017). Deep learning based approach for bearing fault diagnosis. *IEEE Transactions on Industry Applications*, *53*(3), 3057–3065.

Hinchi, A. Z., & Tkiouat, M. (2018). Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network. *PROCEDIA Computer Science*, *127*, 123 – 132.

Hoang, D.-T., & Kang, H.-J. (2018). A survey on deep learning based bearing fault diagnosis. *Neurocomputing*.

Huang, H., & Baddour, N. (2018). Bearing vibration data collected under time-varying rotational speed conditions. *Data in Brief*, *21*, 1745 – 1749.

Jardin, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*, 1483–1510.

Jiang, J., & Wang, L. (2018). Vmd and hmm based rolling bearing fault diagnosis. In *2018 ieee international instrumentation and measurement technology conference (i2mtc).*

K. Li. (2022). *Mad data set.* `http://mad-net.org:8765/explore.html?t=0.%205831516555847212`. (Accessed: accessed 01 February 2022)

Kang, S., Chen, W., Wang, Y., Xiadong, N., Wang, Q., & Mikulovich, V. I. (2019). Method of state identification of rolling bearings based on deep domain adaptation under varying loads. *IET Science, Measurement & Technology*, *14*, 303–313.

Kimotho, J. K. (2016). *Development and performance evaluation of prognostic approaches for technical systems* (Unpublished doctoral dissertation). Paderborn University.

Kimotho, J. K., & Sextro, W. (2014). An approach for feature extraction and selection from non-trending data for machinery prognosis. In *Proceedings of the second european conference of the prognostics and health management society.*

Kouw, W. M., & Loog, M. (2021). A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*, 766–785.

Kutz, N. (2013). *Data–driven modeling & scientific computation* (O. U. Press, Ed.).

Lee, E. T., & Wang, J. W. (2003). *Statistical methods for survival data analysis*. Wiley Interscience.

Lei, Y., Li, N., Guo, L., Li, N., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834.

Lessmeier, C., Kimotho, J. K., Zimmer, D., & Sextro, W. (2016). Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *Proceedings of the european conference of the prognostics and health management society.*

Leturiondo, U. (2016). *Hybrid modelling in condition monitoring* (Unpublished doctoral dissertation). Lulea University of Technology.

Li, N., Lei, Y., , Lin, J., & Ding, S. X. (2018). An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, *62*, 7762–7773.

Li, W., Yuan, Z., Sun, W., & Liu, Y. (2020). Domain adaptation for intelligent fault diagnosis under different working conditions. In *Proceedings of the matec web of conferences* (pp. 1–5).

Li, X., Zhang, W., & Ding, Q. (2019). Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks. *IEEE Transactions on Industrial Electronics*, *66*, 5525–5534.

Li, X., Zhang, W., Ding, Q., & Sun, J.-Q. (2018). Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Processing*, *157*, 180–197.

Li, Y., Wang, X., Liu, Z., Liang, X., & Si, S. (2018). The entropy algorithm and its variants in the fault diagnosis of rotating machinery: A review. *IEEE Access*, *6*, 66723–66741.

Li, Y., Wang, X., Si, S., & Huang, S. (2019). Entropy based fault classification using the case western reserve university data: A benchmark study. *IEEE Transactions*

On Reliability, *69*, 754–767.

Linde, L. (2012). https://github.com/Lucky\-Loek/ieee\-phm\-2012\-data\-challenge\-dataset. (Accessed: 30-01-2021)

Liu, J., Hu, Y., Wu, B., Wan, Y., & Fengyun, X. (2017). A hybrid generalized hidden markov model–based condition monitoring approach for rolling bearings. *Sensors*, *5*.

Liu, Z.-H., Lu, B.-L., Wei, H.-L., Chen, L., Li, X.-H., & Ratsch, M. (2019). Deep adversarial domain adaptation model for bearing fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*, *10*, 1–10.

Mahamad, A. K., Saon, S., & Hiyama, T. (2010). Predicting remaining useful life of rotating machinery based artificial neural network. *Computers and Mathematics with Applications*, *60*, 1078–1087.

Maheshwari, H. (2021). *Understanding domain adaptation.* https://towardsdatascience.com/understanding-domain-adaptation-5baa723ac71f. (Online; accessed 12 August 2021)

Malhi, A., Yan, R., & Gao, R. (2011). Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, *60*, 703–711.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-machine Studies*, *7*, 1–13.

Mao, W., Sun, B., & Wang, L. (2021). A new deep dual temporal domain adaptation method for online detection of bearings early fault. *Entropy*, *23*, 1–19.

Mishra, K., Shakya, P., Babureddy, V., & Vignesh, A. (2021). An approach to improve high-frequency resonance technique for bearing fault diagnosis. *Measurement*, *178*, 1–24.

Monotonicity [Computer software manual]. (2018). https://www.mathworks.com/help/predmaint/ref/monotonicity.html. (Accessed: 10-11-2021)

Nadler, D. L., & Zurbenko, I. G. (2013). Developing a weibull model extension to

estimate cancer latency. *International Scholarly Research Notices Epidemiology*, *18*, 1–6.

NASA. (2012). *Femto bearing.* `https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository`. (Accessed: 11-12-2020)

NASA. (2020). *Turbofan engine degradation simulation.* `https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository`. (Online; accessed 15 July 2019)

NASA. (2022a). *Bearings.* `https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository`. (Online; accessed 06 June 2020)

NASA. (2022b). *Prognostics Center of Excellence Data Set Repository.* `https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/`. (Online; accessed 17 March 2020)

Navarrete, R. (2018). *Embeddings and prediction of dynamical time series* (Doctoral).

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated life test. In *Proceedings of the ieee international conference on prognostics and health management.*

Ng, S. I. (2018). *Reproducing kernel hilbert spaces & machine learning.* `https://ngilshie.github.io/jekyll/upyear/2018/02/01/RKHS.html`. (Online; accessed 17 January 2020)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 1345–1359.

Paterno, M. (2004). *On random-number distributions for c++.*

Pincus, S. M. (1995). Approximate entropy (ApEn) as a complexity measure . *Chaos Int. J. Nonlinear Sci.*, *5*, 110–117.

Pincus, S. M., Gladstone, I. M., & Ehrenkranz, R. A. (1991). A regularity statistic for medical data analysis. *Journal of Clinical monitoring*, *7*, 335–345.

Pincus, S. M., & Goldberger, A. L. (1994). Physiological time-series analysis: what

does regularity quantify. *Am J Physiol Heart Circ Physiol*, *266*, H1643–H1656.

Prasad, H., Ghosh, M., & Biswas, S. (1985). Diagnostic monitoring of rolling element bearings by high frequency resonance technique. *Journal of Mechanical Design*, *28(4)*, 439–448.

Prognosability [Computer software manual]. (2018). `https://www.mathworks.com/help/predmaint/ref/prognosability.html`. (Accessed: 06-11-2021)

Qi, J., Mauricio, A., & Gryllias, K. (2019). Prognostics of rolling element bearings based on entropy indicators and particle filtering. In *Proceedings of the surveillance, vishno and ave conferences.*

Ragab, M., Chen, Z., Wu, M., Foo, C. S., Kwoh, C. K., Yan, R., & Li, X. (2021). Contrastive adversarial domain adaptation for machine remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, *17*, 5239–5249.

Richman, J., & Norman, R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *Am J Physiol Heart Circ Physiol*, *278*, H2039–H2049.

Sadhu, A., Prakash, G., & Narasimhan, S. (2017). A hybrid hidden markov model towards fault detection of rotating components. *Journal of Vibration and Control*, *23*, 3175–3195.

Segla, M., Wang, S., & Wang, F. (2012). Bearing fault diagnosis with an improved high frequency resonance technique. In *Proceedings of the ieee 10th international conference on industrial informatics.*

Sikora, E. (2016). Detection of bearing damage by statistic vibration analysis. In *Proceedings of the iop conference series on materials science and engineering.*

Small, M. (2005). *Applied nonlinear time series analysis.* World Scientific Series on Nonlinear Science.

Smith, G., & Shibatani, M. (2020). *Fault diagnosis of gearbox with a transferable deep neural network.*

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the

case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing*, *64–65*, 100–131.

Soualhi, A., Medjaher, K., & Zerhouni, N. (2014). Bearing health monitoring based on hilberthuang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation And Measurement*, *465*, 261 – 276.

Sparrow, C. (2012). *The lorenzequation: Bifurcations, chaos and strange attractors* (S. N. York, Ed.).

Tableman, M., & Kim, J. S. (2004). *Survival Analysis Using S - Analysis of Time-to-Event Data*. Chapman & Hall/CRC.

Takagi, T., & Sugeno, M. (1985). An experiment in linguistic synthesis with a fuzzy logic controller. *IEEE Transactions on Systems, Man, and Cybernetics*, *15*, 116–132.

Takens, F. (1981). Detecting strange attractors in turbulence. *Lecture notes in Mathematics*, *898*, 366-381.

Tandon, N., & Choudhury, A. (1999). A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. *Tribology International*, *32(8)*, 469–480.

Taylor, J. (1980). Identification of bearing defects by spectral analysis. *Journal of Mechanical Design*, *102(2)*, 199–204.

Tong, Z., Li, W., Zhang, B., & Zhang, M. (2018). Bearing fault diagnosis based on domain adaptation using transferable features under different working conditions. *Shock and Vibration*, *1*, 1 – 12.

Trendability [Computer software manual]. (2018). `https://www.mathworks.com/help/predmaint/ref/trendability.html`. (Accessed: 12-11-2021)

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9(86)*, 2579 – 2605.

Voronkin, V., Mikhailov, V., & Pavlov, K. (1988). Diagnostics of bearings in electric machinery by using high-frequency vibration signatures. *Journal of Mechanical*

Design, *2(4)*, 407–411.

Wang, L., & Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, *3*, 807–814.

Wang, Q., Michau, G., & Fink, O. (2019). Domain Adaptive Transfer Learning for Fault Diagnosis. *arXiv e-prints:1905.06004* . Retrieved from `http://arxiv.org/abs/1905.06004`

Wang, X., Liu, F., & Zhao, D. (2020). Cross-machine fault diagnosis with semi-supervised discriminative adversarial domain adaptation. *Sensors*, *20*, 1–20.

Wei, X., Jia, L., Guo, K., & Wu, S. (2014). On fault isolation for rail vehicle suspension systems. *Vehicle System Dynamics*, *52*, 847–873.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*, 1–40.

Worden, K., Staszewski, W. J., & Hensman, J. J. (2011). Natural computing for mechanical systems research: A tutorial overview. *Mechanical Systems and Signal Processing*, *25*, 4–111.

Wu, S., Wu, C., Lin, S., Lee, K., & Peng, C. (2014). Analysis of complex time series using refined composite multiscale entropy. *Physics Letters A*, *378*, 1369–1374.

Wu, S.-D., Wu, C.-W., Lin, S.-G., Wang, C.-C., & Lee, K.-Y. (2013). Time series analysis using composite multiscale entropy. *Entropy*, *15*, 1069–1084.

Yan, R., & Gao, R. (2004). Complexity as a measure for machine health evaluation. *IEEE Transactions on Instrumentation and Measurement*, *53*, 1327–1334.

Yan, R., & R.X., G. (2007). Approximate entropy as a diagnostic tool for machine health monitoring. *Mechanical Systems and Signal Processing*, *21*, 824–839.

Yang, R., Kang, J., Zhao, J., Li, J., & Li, H. (1989). Computer aided fault diagnosis of crank systems using engine vibration data. In *Proceedings of the 1st international machinery monitoring and diagnosis conference.*

Yang, R., Kang, J., Zhao, J., Li, J., & Li, H. (2014). A case study of bearing condi-

tion monitoring using spm. In *Proceedings of the prognostics and system health management conference.*

Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *Proceedings of ieee international conference on aircraft utility systems.*

Zadeh, L. A. (1965). Fuzzy set. *Information Control*, *8*, 338–353.

Zao, D., & Liu, F. (2022). Cross-condition and cross-platform remaining useful life estimation via adversarial-based domain adaptation. *Scientific Reports*, *12*(878).

Zhang, B., Li, W., Tong, Z., & Zhang, M. (2017). *Bearing fault diagnosis under varying working condition based on domain adaptation.* `https://doi.org/10.48550/arXiv.1707.09890`. (Online; accessed 17 January 2020)

Zhang, L., Xiong, G., Liu, H., & Guo, W. (2010). Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy inference. *Expert Systems with Applications*, *37*, 6077–6085.

Zhang, R., Tao, h., Wu, L., & Guan, Y. (2017). Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access*, *5*, 14347–14357.

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, *17*, 1 – 21.

Zhao, R., Wang, J., Yan, R., & Mao, K. (2016). Machine health monitoring with lstm networks. In *Proceedings of 10th international conference on sensing technology.*

Zheng, J., Pan, H., & Chen, J. (2017). Rolling bearing fault detection and diagnosis based on composite multiscale fuzzy entropy and ensemble support vector machines. *Mechanical Systems and Signal Processing*, *85*, 746–759.

Zhiqiang, C., Shengcai, D., Xudong, C., Chuan, L., Ren-Vinicio, S., & Huafeng, Q. (2017). Deep neural networks-based rolling bearing fault diagnosis. *Microelectronics Reliability*, *75*, 327–333.

Zhou, J., Zheng, L.-Y., Wang, Y., & Gogu, C. (2020). A multistage deep transfer learning method for machinery fault diagnostics across diverse working conditions and devices. *IEEE Access*, *8*, 80879–80898.

Zhu, J., Nostrand, T., Spiegel, C., & Morton, B. (2014). Survey of condition indicators for condition monitoring systems. In *Proceedings of the annual conference of the prognostics and health management society.*

Zhu, K., & Li, H. (2014). A roller bearing fault diagnosis method based on hierarchical entropy and support vector machine with particle swarm optimization algorithm. *Measurement*, *47*, 669–675.

Zhu, K., & Li, H. (2015). A rolling element bearing fault diagnosis approach based on hierarchical fuzzy entropy and support vector machine. *Journal of Mechanical Engineering Science*, *230*, 2314–2322.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... He, Q. (2020). *A comprehensive survey on transfer learning.*

Zill, D. G., & Cullen, M. R. (2009). *Differential equations with boundary value problems.*

APPENDIX I

Measuring Suitability of Health Indicators

Here, the formulas used to compute how suitable health indicators are for estimating RUL are given.

I.1 Monotonicity

Monotonic trends are matching i.e. all features increasing or decreasing in value together as time progresses. The quantity monotonicity quantifies the monotonic trend in condition indicators as the system evolves toward failure("Monotonicity", 2018). An output of 1 means the condition indicator is perfectly monotonic while 0 indicates a non-monotonic feature.

$$\text{Monotonicity} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\text{sgn}(x_j(k+1)-x_j(k))}{N_j-1}\right|$$

I.2 Prognosability

This is a measure of the variability of a feature at failure based on the trajectories of the feature measured in several run-to-failure experiments("Prognosability", 2018). Prognosable feature has less variation at failure relative to the range between its initial and final values. A highly prognosable feature feature has an output of 1 while 0 means the feature is not prognosable.

$$\text{Prognosability} = exp\left(-\frac{\text{std}_j(x_j(N_j))}{\text{mean}_j|x_j(1)-x_j(N_j)|}\right)$$
$$j = 1, 2, 3, \cdots, M$$

I.3 Trendability

This is a measure of similarity between the trajectories of a feature measured in several run-to-failure experiments. A more trendable feature has trajectories with the same underlying shape("Trendability", 2018). The output is 1 if the input to the trendability function is perfectly trendable and 0 if it is non-trendable.

$$\text{Trendability} = \min_{j,k} |\text{corr}(x_k, x_j)| \;\; j, k = 1, 2, 3, \cdots, M$$

For all the three metrics, x_j represents the vector of measurements of a feature on the j_{th} system, M is the number of systems monitored, and N_j is the number of measurements on the j_{th} system. corr$(\cdot)$ is the correlation function.

For this work the number of systems M were three i.e. B1_1, B2_1, and B3_1. The number of measurements per system N_j were the number of examples for each system.