UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

University of Pretoria

Faculty of Engineering, Built Environment and Information Technology

Department of Mechanical and Aeronautical Engineering

A Deep Learning Approach Towards Diagnostics of Bearings Operating under Non-stationary Conditions

by

Stephan Baggeröhr

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Mechanical Engineering)

supervised by
Prof. PS Heyns Prof. DN Wilke

2019

Abstract

Keywords: Deep learning, Bearing Fault Detection and Diagnosis, Non-Stationary operating conditions, Unsupervised learning, Information maximisation

Faults in bearings usually manifest as marginal defects that intensify over time, allowing for well-informed preventative actions with early Fault Detection and Diagnosis (FDD) protocols. Detection of the fault begins with capturing, for example, acceleration signals from a machine. Traditionally, handpicked descriptive statistical features (mean, RMS, skewness, kurtosis, etc.) or spectral diagrams obtained from these signals are then used for FDD. However, machine signals are often generated under non-stationary operating conditions of varying loads and speeds, requiring further intervention. More advanced signal processing techniques (spectral kurtosis, or cyclostationary analysis) are hence used to account for the non-stationarity of the signal. This is usually done by separating acceleration signals into deterministic and random components [Abboud et al., 2019]. Fault detection in bearings is possible by observing the random components of the signal.

A wealth of research [Zhao et al., 2019, Cerrada et al., 2018, Khan and Yairi, 2018, Liu et al., 2018] has been invested in machine learning based techniques to circumvent the problems associated with non-stationary signals. Many of these methods require vast amounts of historical data to train. Machines typically spend most of their life operating in a healthy condition, therefore, most historical data is occupied with data that comes from a healthy machine condition, training these methods are difficult, due to the shortage of data from a machine running in an unhealthy condition. Furthermore, well-performing machine learning algorithms still require a domain expert to extract features that are known to be fault sensitive. Deep learning is a recent approach in data analysis whereby feature extraction is incorporated within the training of the algorithm. The algorithm is given the ability to find and extract its own features. The architecture of the algorithm allows for the extraction of complex hierarchical non-linear features. To the author's knowledge, no attempt has been made to make full use of the power of deep learning together with the known structure of bearing acceleration signals to perform FDD.

In this work, a bearing FDD methodology is developed using deep learning approaches. A model based on Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) is used to learn a lower dimensional representation of an acceleration signal. A regularization strategy based on information maximization is used, which allows deterministic and random components of the signals to be learned separately. This representation is subsequently used to perform bearing FDD. The algorithm is trained in a completely unsupervised manner on exclusively healthy data and requires no preprocessing of that data. Furthermore, no auxiliary signals such as a shaft encoder, which contains information about the machine operating condition, is required for the algorithm to work. The methodology was tested on well known benchmark datasets, and it was shown to be robust against non-stationary operating conditions. The algorithm can learn its own fault metric and by observing the trajectory of the signal representation, it is also able to diagnose the type of fault.

Acknowledgements

I would first like to thank my supervisor Prof Stephan Heyns of the Department of Mechanical and Aeronautical Engineering at University of Pretoria. Thank you for the opportunity to expand my mind in a field I would never have dreamt of entering. Thank you for sharing your passion with this subject and showing me that the field of Asset Management is world of ill-defined problems, complex solutions, and massive machines, ie. an engineer's dream. Your unending patience, guidance and advice was paramount to the completion of this work, and for that I will be forever grateful.

I would also like to acknowledge Prof Nico Wilke of the Department of Mechanical and Aeronautical Engineering at University of Pretoria as my co-supervisor. Thank you for all the chapter by chapter advice and feedback you so willingly gave. I will miss our long discussions related to machine learning, statistics, optimization, maths, science, philosophy or whatever topic of interest was on the table that day, the list is endless. You have really influenced the way I think about and approach new problems.

A few more acknowledgements go to Dr Eric Bechhoefer for his generous donation of data and insights, to Dr Dany Abboud for his help with the baseline this work was compared against, and to the Eskom Power Plant Engineering Institute (EPPI) for funding the research.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them. Thank you.

Author
Stephan Baggeröhr

Conference proceedings based on this work

S. Baggeröhr, W. Booyse, P.S. Heyns, and D.N. Wilke. Novel Bearing Fault Detection using Generative Adversarial Networks. *31st Conference on Condition Monitoring and Diagnostic Engineering Management COMADEM*, 243-250, 2018

Co-authored journal articles based on this work

K. Wang, P. Chen, Y. Li, M.J. Zuo, P.S. Heyns, S. Baggeröhr. A new health condition monitoring scheme for wind turbine bearings based on deep convolutional generative adversarial networks. *Journal of Sound and Vibration*, submitted

Contents

Nomenclature

Abbreviations

ALI	Adversarial Learned Inference
ANC	Adaptive Noise Cancellation
ANN	Artificial Neural Network
BCF	Bearing Characteristic Frequency
ELBO	Evidence Lower Bound
FDD	Fault Detection and Diagnosis
FSI	Fault Severity Index
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
PHM	Prognostics and Health Management
RMS	Root Mean Squared
ROC	Receiver operating characteristics
RUL	Remaining Useful Life
SOTA	State of the art
SVM	Support Vector Machine
TSA	Time Synchronous Averaging
VAE	Variational Autoencoder
WPD	Wavelet Packet Decomposition

Greek Symbols

λ	Learning rate
ω, η	Trainable encoder/decoder parameters
ϕ, θ	Trainable generator/discriminator parameters
$\sigma(\cdot)$	Activation function

Roman Symbols

$\mathbf{a}$	Activation matrix
$\mathbf{b}$	Bias matrix
$\mathbf{h}$	Hidden unit matrix
$\mathbf{W}$	Weights matrix
$\mathbf{X}$	Observed variables vector
$\mathbf{Z}$	Hidden variables vector
$H[\cdot]$	Entropy
$I[\cdot]$	Mutual information
L	Loss function

$p(\cdot)$	Real probability distribution
$q(\cdot)$	Generated probability distribution

Chapter 1

Introduction

1.1 Background

Many industries, such as those in the mining or power generation sector, have a large number of high valued legacy assets. The productivity of these companies often relies on these ageing assets. Increasing the reliability and availability of such ageing assets remains an important business goal, if they are to remain competitive. One way of achieving higher reliability is through prognostics and health management (PHM). Some of the ways in which PHM can improve the reliability of an ageing asset are [Roy et al., 2016]:

- Engineering for extending life of high value assets with optimum costs
- Better understanding of the foundations of asset in-service degradation
- Applying new technologies to improve efficiency and effectiveness of the maintenance: large scale data analysis, automation and autonomy

Rotating machinery is one of the main asset classes with industry and, hence, has driven research for PHM. Rotating machinery is generally supported by bearings whose failure can lead to entire system shutdown. As a result, the bearings of a machine are considered one of its most critical components. Bearing related faults can account for as much as 40% of the total number of failures in induction motors [Zhang et al., 2011]. As a result, bearing fault detection and diagnosis (FDD) are important if one is to avoid the more catastrophic failure consequences of large rotating machinery. Faults, however, usually manifest as marginal defects that intensify over time, allowing for well-informed preventative maintenance schedules with early FDD.

PHM consists of three distinct stages, namely: data acquisition, data processing and maintenance decision-making. During data acquisition, signals are captured from the machine in various forms. These include, for example, vibration-, acoustic-, or temperature-signals. Data can also come in the form of oil analyses, where debris in the oil can indicate the ensuing fault. Data, in whatever form, contains information about the condition of the machine to various degrees. During the second stage of PHM, advanced data processing techniques can be used to extract machine condition information from the data. Using a suitable signal processing technique, the location of the fault can be identified. However, estimating the severity of the fault is a bit more challenging and thus, is an active area of research.

The data acquisition method most often used in practice is the vibration signals obtained from accelerometers. This is due to the ease of installation of accelerometers and the high amount of diagnostic and prognostic information contained within the signal. Due to the high sampling rate and large amount of noise in vibration signals, it has always been necessary to improve the quality of the data or extract features before any faults could be detected. Traditionally, these features are generally extracted from either the time domain or frequency domain or a combination of both. A useful feature is one that is only sensitive to faults and performs well in signals that have a low signal-of-interest (fault) to noise ratio. Statistical features in the time domain are a popular choice for bearing diagnostics. For example, kurtosis is used as it is a measure of the signals impulsiveness [Randall and Antoni, 2011].

Alternatively, in the frequency domain, spectral diagrams are used to manually track the excitation of certain system resonances, usually at much higher frequencies, indicating a fault. More commonly, fundamental fault frequencies can be calculated from the bearing's geometry and tracked in the envelop spectrum can provide diagnostic information. However, machines are frequently operated under non-stationary conditions, such as varying speeds or loads. This makes extracting features with these methods more challenging, as they all work under the assumption that the signal is stationary. As a result, more advanced techniques are required to offset the effects of non-stationary conditions.

Once the dataset is clean, and a good set of features have been extracted, the final stage of PHM can commence with the maintenance decision making. This is done by diagnosing the incipient fault and making a prognosis. Fault classification is normally done by comparing the values of the extracted features for healthy machines with those of unhealthy machines. The fault is identified by comparing the characteristics of the features to those of known fault modes using statistics, machine learning or model-based methods. This philosophy of training is known as supervised learning. Clearly labelled historical data of what is healthy and unhealthy is required to build these models.

Once a fault is identified, a degradation metric or fault severity index (FSI) is selected and used to trend the fault over a predetermined monitoring interval. The resulting trend is then used to make a prediction about the remaining useful life (RUL) of the machine. Using the RUL together with the associated risk of the machine, a decision on how to appropriately maintain the machine can be made.

The diagnosis of the fault is depends mainly on the feature types extracted during the second stage. Features that are also sensitive to operating conditions may result in a misdiagnosis. Figure 1.1 shows the typical impact of wear severity adapted from El-Thalji and Jantunen [2014], specifically for a bearing. The wear severity is a typically increasing function with three distinct phases which indicate the progressively worsening condition of the bearing. A healing phenomenon is typically seen at the initiation of the defect and the just before damage growth.

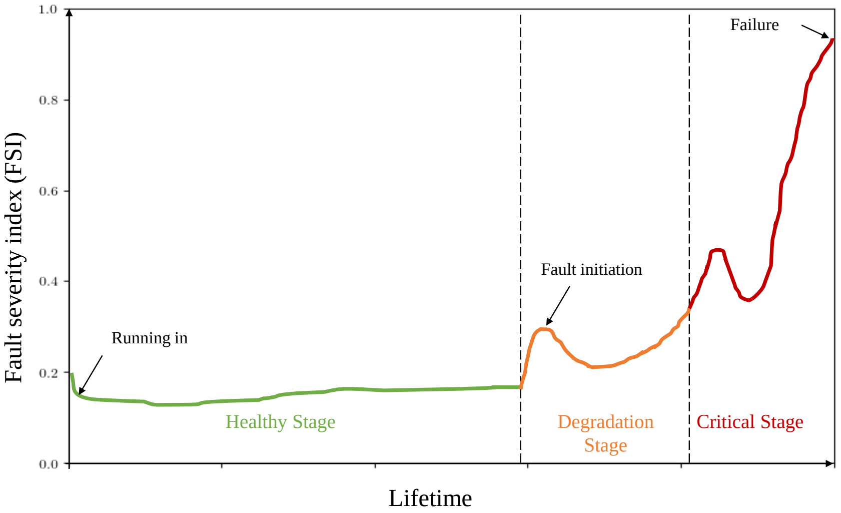


Figure 1.1: Typical dynamic impact of wear severity in a bearing, adapted from El-Thalji and Jantunen [2014].

Knowing what fault features to extract and what fault modes to expect requires expert knowledge or intricate physics-based modelling of the system, both of which can be extremely costly or time-consuming to implement. Furthermore, data driven methods often require large amounts of historical data augmented with event data in order to perform well. Event data, such as breakdowns and overhauls, are usually manually entered by operators, and thus have high variability and sparsity. Above all that, historical process data mainly consists of data from a healthy machine condition because the machine usually spends most of its life operating in a healthy condition. This class imbalance complicates the training of these methods. Hence a supervised approach is neither viable nor reliable.

Furthermore, data driven methods require good discriminative features, and hence careful hand designed features are necessary to create a successful diagnostic algorithm, especially for non-stationary cases. A proposed framework can become very specific to both the asset and the type of faults for which it was developed. A dedicated framework for each individual component is then often needed when a diverse range of assets require monitoring. This adds further costs to the maintenance of machines.

More recently, advancements in deep learning has allowed algorithms more flexibility to automatically find complex hierarchical representative features by exploiting deep network architectures. A comparison of deep learning models against conventional approaches [Zhao et al., 2019] is shown in Fig. 1.2. These algorithms can represent complex functions and as a result, they can extract more discriminative features over a broader range of signals. As industries are preparing to move into Industry 4.0, a new set of industry goals and standards are being formalized. An increase in automation and data capturing is the drive behind achieving these goals. This allows for more data to be captured and new analysis techniques to extract useful information. If industries are to stay competitive, adoption of these techniques is essential. In PHM the ultimate goal is to increase machine availability and reliability, without extensive re-engineering between assets.

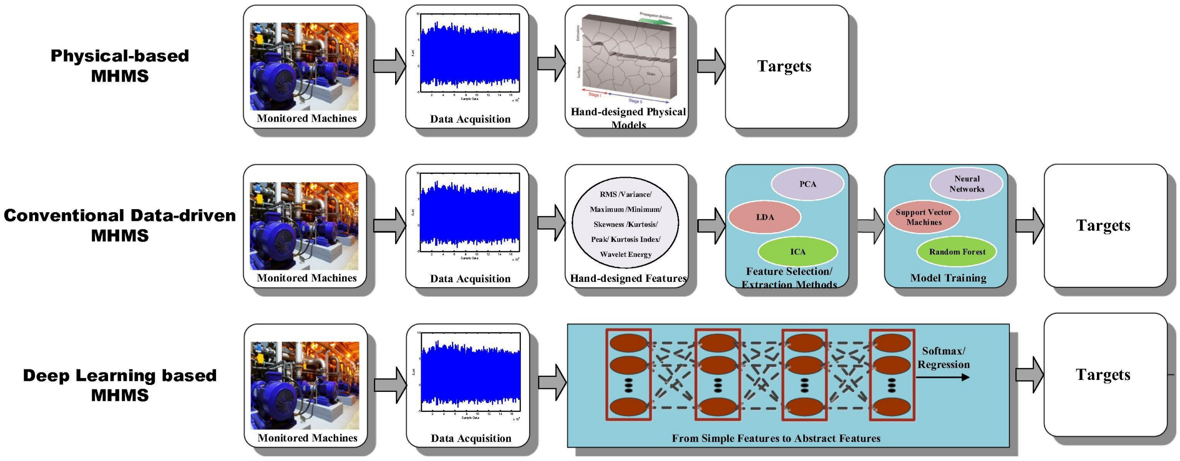


Figure 1.2: Comparison of various fault diagnosis frameworks [Zhao et al., 2019].

1.2 Literature review

It is important to understand how faults in bearings develop before embarking on fault detection and diagnosis. The presence of faults will affect the measured vibration response. Furthermore, non-stationary operating conditions will also influence the vibration waveform. It is crucial to understand all the fault mechanisms before an effective FDD strategy can be developed. This section covers the details behind faults in bearings that are operating in non-stationary conditions. It continues with a review of some of the conventional data driven methods used to diagnose faults, which includes data preprocessing, feature extraction and models for trending.

1.2.1 Bearing faults and detection

Bearing faults

A bearing comprises four main components as shown in Fig. 1.3: (1) the rolling elements, (2) the inner race, (3) the outer race, and (4) the cage. By varying any of these main components, a wide selection of bearings are commercially available, the design of which is based on the function that the bearing has to perform within a rotary machine. Faults can manifest in any number of these components. These faults can be grouped either as a single point defect, a multiple point defect or as a distributed fault.

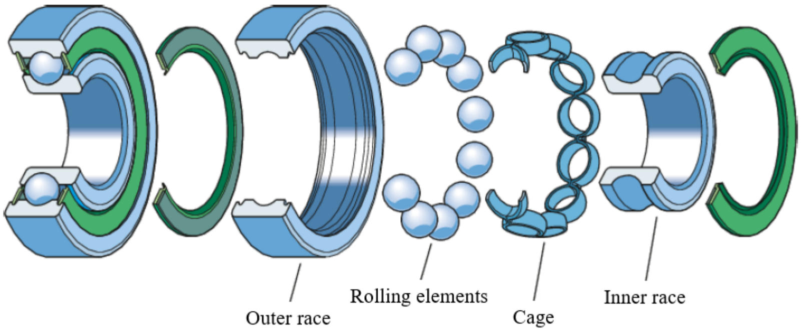


Figure 1.3: Typical ball bearing components.

A single point defect usually occurs during the early stages of fault development. They are characterised by small localised areas of damage appearing on any of the aforementioned components. Examples of such defects are spalls, corrosion, pits, scratches or dents. These types of faults can produce a series of impulses as a result of the sharp discontinuity presented by the fault on the bearing's rolling surface. These impulses excite the system, causing it to resonate which results in a series of broadband bursts. The amplitude of these bursts are modulated by two factors [Randall and Antoni, 2011]:

- The strength of the burst is proportional to the load on the rolling elements, which is modulated at the same rate at which the fault passes through the load zone.
- The transfer function of the path between the fault and the response transducers (accelerometers) varies.

Hence, different faults give rise to predictable bearing characteristic frequencies (BCFs) at which the amplitude of the response is modulated. The BCFs can be calculated from the bearing's geometry, with each main component of the bearing having a unique frequency: F_{IRF} = inner race fault frequency, F_{ORF} = outer race fault frequency, F_{CF} = cage fault frequency, and F_{BF} = ball or rolling element fault frequency. These frequencies can be calculated from the shaft rotation frequency F_S,

$$F_{IRF} = \frac{N_B}{2} F_S \Big(1 + \frac{D_B \cos(\theta)}{D_P}\Big) \tag{1.1}$$

$$F_{ORF} = \frac{N_B}{2} F_S \Big(1 - \frac{D_B \cos(\theta)}{D_P}\Big) \tag{1.2}$$

$$F_{CF} = \frac{1}{2} F_S \Big(1 - \frac{D_B \cos(\theta)}{D_P}\Big) \tag{1.3}$$

$$F_{BF} = \frac{D_P}{2D_B} F_S \Big(1 - \frac{D_B^2 \cos^2(\theta)}{D_P^2}\Big), \tag{1.4}$$

where N_B is the number of rolling elements or balls, D_B is the ball diameter, and D_P is the ball pitch diameter as shown in Fig. 1.4. The ball contact angle θ, is the angle between the centre line of the bearing and the direction of force the rolling elements make with the outer race [Stack et al., 2003]. These equations reveal the dependence of these frequencies on the shaft rotational speed. Thus these frequencies are sensitive to the operating condition of the machine, and therefore do not make ideal fault features in variable operating conditions. This is further exacerbated by the modulation due to loading on the shaft.

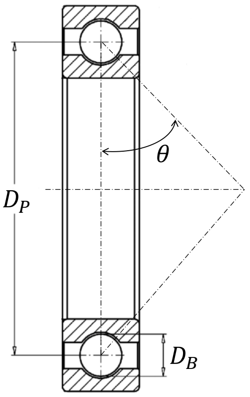


Figure 1.4: Ball bearing geometry used for BCF calculation.

It should be noted that some bearing faults are not continuously impulsive and dynamically change as the wear progresses. This is known as a bearing self-healing phenomenon, whereby for example, faults such as spalls start with sharp edges, which the become rounded off, before a new edge is formed. All the while, the impulsiveness of the signal will increase due to the sharp edge, then decrease when the edge is rounded off, and the subsequently the impulsiveness of the signal will increase again.

Nevertheless, understanding single point defects is important towards understanding how faults produce vibrations in rotating machinery. Single point defects are seldom observed in practice. In reality, in-service defects are often a combination of several, possibly overlapping, single point defects. The vibration signal may produce spectral lines at the expected BCF in the envelope spectrum, however their relative amplitudes may be different from what is expected [McFadden and Smith, 1985].

Another diagnostic approach is physics-based models. These types of models attempt to simulate the vibration caused by faults using lumped mass/spring/damper or finite element models. Early models assumed a deterministic and periodic series of impulses, caused by the fault, with a constant impulse period, T. This was later improved by introducing a stochastic component to the modelling of the impulses. The stochastic component was attributed to the random slippage of the rolling elements that occurs during operation. This slippage results in an uncertainty of the arrival time between the impulses. The nature of the stochastic behaviour of fault bearing signals was then well approximated as a cyclostationary signal [Antoni, 2009]. As a result, cyclostationary based methods worked well for bearing diagnostics.

The vibration response of a bearing operating in non-stationary conditions can also be approximated as the response of a linear system, shown in Fig. 1.5, where the transfer function is itself a function of time, $\mathcal{H}(t, \tau)$ [Antoni, 2009]. The amplitude modulation of a signal is due to two factors. First, the entry and exit events of the rolling elements of the fault into the load zones. And second, the variations in the signal due to changes in the transmission path from the fault to the transducer. The amplitude modulation effects can all be represented by $\mathcal{H}(t, \tau)$. The signal, as measured by the accelerometer is given by $\mathbf{X}(t)$, with the source signal of interest represented as $\mathbf{u}^*(t)$. In bearing FDD the signal of interest $\mathbf{u}^*(t)$, is the impulse signal generated by the fault. In reality, the measured response contains vibration components from various other sources from the machine, such as signals produced by gear meshing. These additional sources are all lumped in $\mathbf{u}_i(t)$. Measurement noise is represented by $\mathbf{n}(t)$. Often at times, an auxiliary signal, represented as $\mathbf{a}_i(t)$, is measured by a tachometer or shaft encoder (represented by $\mathcal{L}_i$) and provides the instantaneous shaft rate of the machine. Many signal processing techniques require this auxiliary signal to remove the effects of the non-stationary operating conditions.

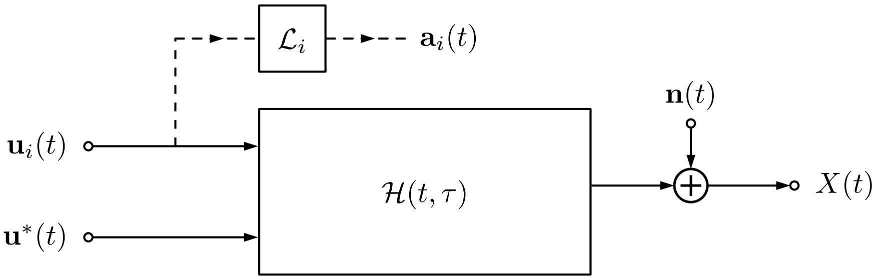


Figure 1.5: Schematic of a linear system $\mathcal{H}(t,\tau)$ that can be used to approximate the response of a machine to a bearing fault.

Throughout its lifetime a bearing fault can start out as a single or multiple localised faults, which gradually evolves and spreads over a larger area producing a distributed fault. Distributed faults, also known as generalized roughness, usually indicate imminent failure. These types of faults are generally further accelerated by a lack of proper lubrication or misalignment. The resulting vibration signal produced by this type of fault has varying degrees of complexity, often with no BCFs present in the signal spectrum, thus making these types of faults harder to predict. Consequently, researchers tend to focus more on diagnostic methodologies based on single or multiple point defects alone [Dolenc et al., 2016]. Antoni and Randall [2002] modelled the acceleration response of a bearing with distributed faults as a combination of periodic components and random components, as Eq. 1.5:

$$X_b(t) = p(t) + B(t) \tag{1.5}$$
$$\mathbb{E}[B(t)] = 0, \tag{1.6}$$

where $p(t)$ accounts for the periodic components and $B(t)$ for the purely random with zero mean, but cyclostationary components of the signal. This assumes that the vibration response of a bearing measured by an accelerometer has two components namely the deterministic component and the random component.

Fault detection and diagnosis

Bearing fault detection and diagnosis is performed in three distinct stages [Cerrada et al., 2018]:

- Fault detection
- Fault diagnosis
- Fault severity

The aim of fault detection is simply to assess whether the bearing is operating as expected or not. More formally, healthy rotating machinery can be expected to vibrate, either due to its inherent operation or due to its manufacturing/installation flaws. After an extended period of observation, the response of the machine due to normal operating conditions forms a unique vibration signature. The presence of a fault will, however, alter this vibration signature. Hence, a fault can be detected by comparing a sample of unknown bearing condition to a vibration sample of a healthy bearing. Note that the vibration signals of machines are often dominated by strong deterministic signals, such as shaft and gear mesh harmonics, and thus changes in the bearing condition leads to only very minor changes in the overall vibration level. Therefore a more targeted approach is required to track the severity of the fault.

In the second stage, fault diagnosis aims to identify what type of fault is present and where it is located. This is especially important if the fault is located in a critical part of the machine that demands immediate action. The final stage, aims to identify the severity of a fault using a fault severity index (FSI). Here the severity refers to the overall health of the bearing and not the exact dimensions of the fault. Fault size estimation is still an underdeveloped research area. As the fault grows, the FSI's value will proportionally increase. Using prior failure data of the machine, an estimation into the condition of the machine can be made. Knowing the type of fault for the third stage, although potentially important for final maintenance decisions, is not necessary for the assessment of the severity of fault, and thus can be performed independent of the second stage. Conversely, knowing the type and location of a fault does not require knowledge of the severity of that fault. Hence, these two stages of bearing FDD can be separated into two independent methodologies.

With this background, some of the methods researchers have proposed to detect and diagnose faults in bearings are next presented.

1.2.2 Signal processing approaches

Signal processing used in bearing diagnostics involves techniques in which deterministic or discrete components of the signal are removed and the remaining residual signal, containing the random components, are used to make a judgement on both the fault severity and fault type. After the separation of the signal, the fault component of the remaining signal is then enhanced with further signal processing techniques or feature extraction methods which are known to be sensitive to the faults. Some of the methods used to remove the deterministic signals are detailed in this subsection.

Deterministic/random separation

Separating a vibration signal into deterministic and random components is a very powerful tool in diagnostics and is often the first step for many diagnostic methods. Some of the most effective methods of performing this separation include Time Synchronous Averaging (TSA), linear prediction models, adaptive noise cancellation and discrete/random separation [Randall et al., 2011].

Time synchronous averaging (TSA) is the oldest technique used to separate deterministic components. In this method, periodic components of a signal are obtained from averaging a number of signal elements, corresponding to one period of interest. Speed fluctuations have to be taken into account by order tracking or sampling the signal in the angular domain, before applying TSA. A downside to this method is that it must be performed for each periodic signal of interest separately. Randall [2011] showed this by removing the periodic vibration signals of a gearbox taken from a mining shovel.

Linear prediction models are used to predict the deterministic component of a signal at a time step based on a certain number of samples from the previous time steps. A residual signal can then be obtained by taking the difference between the predicted and actual signals. The resulting residual signal contains the impulses and random components associated with the bearings. Avendano-Valencia and Fassois [2014] did a review on stationary vs non-stationary modelling methods for the analysis of an in-operation wind-turbine. Among the models that were studied were autoregressive (AR), AR-moving average (ARMA) as stationary models and time-varying-AR (TAR), functional series-TAR (FS-TAR) and adaptive functional series

TAR (AFS-TAR) models as the non-stationary models. They found that non-stationary TAR models are sufficient to model wind turbine signals. The linear predictive models allowed for the separation of deterministic components through residual analysis or similar techniques.

Adaptive noise cancellation (ANC) is a method whereby a filter is obtained from a reference signal containing some relationship to the component of interest within a primary signal. In many cases a self adaptive approach is used, whereby a delayed version of the primary signal is used as the reference signal. The resulting filter, filters out the periodic components of the signal, again leaving the random components associated with bearing faults. Antoni and Randall [2004] showed examples where adaptive noise cancellation was used to separate bearing signals from gear signals. Similarly, Wang et al. [2015] used an ANC algorithm to remove the interfering gearbox signal from the bearing signal tested both numerically and experimentally. Their method used only one accelerometer, without the need for a speed reference signal, showing that the information content of a single accelerometer is sometimes enough to perform diagnostics.

Discrete/random separation is similar to the adaptive noise cancellation approach, with the difference being the filtering process takes place in the frequency domain as opposed to the time domain. Abboud et al. [2016] used generalised TSA to perform discrete and random separation of signals operating in non-stationary domains. A deterministic signal is produced by tracking a specific speed profile using a known reference speed signal such as a tachometer.

Enhancement and feature extraction

Separating the vibration signal is the first step of diagnosis. The next step used by researchers is to enhance the fault carrying signal using more advanced signal processing technique. This is crucial to ensure a robust fault detection methodology.

Envelope analysis is the de-facto standard method used for bearing diagnostics [McFadden and Smith, 1984]. The vibration signal is bandpass filtered in a frequency range corresponding to structural resonance caused by fault impulses. This is followed by amplitude demodulation to obtain an enveloped signal. The frequency range of the bandpass filter is normally high and thus low frequency components normally associated with gear mesh frequencies are filtered out [Randall and Antoni, 2011]. This approach is not exclusive to acceleration signals as Nguyen et al. [2015] used envelope analysis to detect symptoms of defected bearings using acoustic emission (AE) signals. Furthermore, the method has been adopted for non-stationary cases, such as Ming et al. [2016]. They used an iterative approach to perform envelope analysis to extract fault features of a bearing operating in fluctuating load conditions.

Borghesani et al. [2013] explained that the method of using squared envelope analysis spectrum (SES) has gained popularity as a bearing diagnostic tool. Especially, when the SES is paired with computed order tracking (COT) to extend its use to cases with small speed fluctuations. He went on to further highlight the importance of extending this methodology to cases for high speed variance and load transients. Borghesani et al. [2013] then developed a SES and COT based approach for use in highly variable operating conditions.

Wavelet analysis is well suited for application in non-stationary signal analysis. Wavelet analysis is performed by choosing a basis function and expanding a signal in terms of this basis function. Wavelet analysis is similar to Fourier analysis, where the basis function for Fourier analysis is the sinusoid $e^{i2\pi t}$. Wavelet expansions allows for more localised spatial and frequency information through the translation and dilation of the basis function. Ericsson et al. [2005] conducted an investigation on techniques used for the automatic detection of local defects in bearings. They concluded that wavelet based approaches are well suited for the task. Rafiee et al. [2010] proposed a wavelet-based signal processing technique to extract features for gear and bearing diagnosis. Kumar and Singh [2013] provided a methodology in which outer race defect sizes of taper roller bearings can be estimated using a wavelet based approach. The same authors also provided a wavelet based fault localizing technique

[Singh and Kumar, 2013]. Khanam et al. [2014] also provided a technique in which fault size estimation is performed for a bearing using the discrete wavelet transform.

Kurtosis based approaches can be used to obtain frequency bands with the highest levels of impulsivitity. The use of spectral kurtosis as a tool for bearing diagnostics was first introduced by Antoni and Randall [2006]. One of the strongest diagnostics tools developed using kurtosis is the kurtogram. Lei et al. [2011b] improved the feature extraction methods of kurtogram using Wavelet Packet Transform (WPT). Wang et al. [2016] reviewed the use of spectral kurtosis for diagnosing faults in rotating machinery. More recently, Miao et al. [2017], provided an improved method for using kurtosis especially for bearings operating in harsh working conditions.

Cyclostationary analysis based methods were introduced after it was realised that faulty bearing signals can be approximated as cyclostationary signals [Antoni et al., 2004]. Antoni [2009] went on to provide a wealth of examples of cases where cyclostationary models can be useful in the diagnosis of mechanical systems. [Urbanek et al., 2013] proposed a method whereby second order cyclostationary components of a vibration signal were extracted and used for the diagnosis of bearings in wind turbines, which naturally operate with fluctuating speeds and loads.

Recently, Antoni and Borghesani [2019] investigated the evolution of signal characteristics during various phases of machine degradation. The investigation was sparked by the sensitivity of traditional diagnostic indicators being sensitivity to non-stationary conditions in the form of cyclostationarity and non-Gaussianity (or an increase in impulsiveness). Traditionally, these factors have been dealt with separately. For example, testing non-Gaussianity using kurtosis knowing effects of cyclostationary signals will be ignored. They developed a systematic approach whereby these factors of non-stationarity are tracked independently and producing a new FSI based on the most factors that are most dominant in the signal.

All these methods highlight the importance for robust features for a well functioning diagnostic method, especially in non-stationary operating conditions. What is interesting to note is that all these methods are based on hand designed features. Intricate knowledge (sampling rates, shaft speeds, machine loads etc.) of the machine and its signal are often required before performing any one of these methods. Further more, auxiliary signals are almost always needed to ensure the effects of operating conditions can be filtered from the signal. Since learning based approaches offer a chance to learn based purely on the data, it will be discussed next.

1.2.3 Learning based approaches

Learning based approaches are often incorporated with the signal processing techniques discussed previously. The main goal of learning based approaches, is to model the probability distribution of the features extracted and subsequently matching these distributions to known fault cases. In some cases, the features extracted from the model itself are far more discriminative when they are compared to hand designed features.

Artificial Neural Networks

Inspired by the biological model of our brain and its many neural pathways, an artificial neural network (ANN) creates a non-linear mapping between input variables and targets. The ANN is trained in a supervised manner, meaning the targets need to be known. In bearing diagnostics, this means a dataset is required with fault labels already attached to signals. Regardless, many fault detecting algorithms have been constructed using ANNs. By combining neural network with wavelet analysis, Lei et al. [2011a] were able to identify faults and fault severity of locomotive roller bearings. De Moura et al. [2011] performed a comparison between Principal Component Analysis (PCA) and ANN on features extracted from vibration signals of bearings for fault diagnosis. Although supervised learning dominates, it is not the only approach that can be used with ANNs; Heyns et al. [2012] proposed a residual analysis technique in which dominant vibration signals are filtered out using an ANN. Also, an ANN can be used as a feature extractor as is the case with Muruganatham et al. [2013], who extracted features from a vibration signal and used them in an ANN for further fault classification.

Support Vector Machines

Support vector machines (SVMs) aim to model the boundaries between classes represented by the data in the input space. Kernel functions are used to transform data into higher dimensions where hyperplanes can be used to easily separate classes. Again, SVM models are trained in a supervised manner. That is, a dataset complete with the fault classes is required. Diagnosis is performed by comparing a data point to its nearest neighbour, and assigning that class to the unknown data point. Rojas and Nandi [2006] proposed a method in which SVM were used to perform fast classification of bearing faults. The input vector of their method was based on time statistical features. In their approach two cases were analysed; one where the full signal was used and another where only signal segments were used. This is an example where learning is used to improve hand designed features. In a similar approach, both Abbasion et al. [2007] and Wen et al. [2016] proposed using wavelet coefficients as input vectors for a SVM. They were then able to accurately classify faults in signals with unknown class labels. SVMs are not limited by the type of input features that can be used. Alternative feature vectors that have also been used include: features based on ensemble empirical mode decomposition [Zhang and Zhou, 2013], or higher order spectral features [Saidi et al., 2015].

Gaussian Mixture Models

Gaussian mixture models (GMM) is a group of methods that attempt to model the probability distribution of features by approximating it as a linear combination of Gaussian distributions. This approach can model any continuous density by adjusting their means and variances based on model evidence (input data). It is a model that is very easily trained, however it requires a large amount of data to improve accuracy. Nevertheless, Yu [2011] developed an approach in which a feature extraction approach was optimized to find the best features pertaining to bearing faults. A fault degradation metric was then defined as the log-likelihood of a GMM based model of these features. Liu et al. [2015] defined a fault metric as a GMM modelled on a baseline historical factor when presented with new results. This showcases a method in which a probability model can be used as a fault metric, for cases where the data class balance is skewed. Again, GMMs are not limited in their approach or type of data used, such as Aye and Heyns [2017], who predict remaining useful life of bearings using acoustic emissions and GMM based models.

Hidden Markov Models

Hidden Markov Models (HMMs) is a learning framework used especially for sequential data. Markov models can be used to represent the distribution of a sequence of events, or the probability of one event following another. It is assumed that in this sequence, one event is dependent on a finite number of previous events. Hidden Markov models assume that a hidden sequential variable is used to generate a single point, and models the sequential relationship between the hidden variables. The hidden variable is some unseen process that generated the data. Ocak and Loparo [2005] used features from an amplitude modulated vibration signal from both healthy and a faulty bearing to train a HMM. The HMM model was used as a fault detector by calculating the probability of new features, given the features from the training set. Marwala et al. [2006] compared HMM models to GMM models in their ability for fault detection in bearings. Features were extracted from vibration signals and used to train one of either models. For this application, they concluded that HMMs outperformed GMMs. However, it was noted that HMM models are far more computationally expensive than GMM, and the latter should be preferred when time constraints are an issue. There have been many approaches in the way HMM models have been used: Boutros and Liang [2011] developed a methodology in which bearing faults were detected and diagnosed using HMMs, with features obtained from a filter bank. Tobon-Mejia et al. [2012] were able to estimate the RUL of bearings using HMM and features obtained using Wavelet Packet Decomposition (WPD). This allowed them to provide confidence intervals along with their RUL metric further supporting maintenance decision making. Liu et al. [2014] used HMM together with zero crossing features to assess the performance and degradation of bearings. Zhang et al. [2015] used a locality preserving projection for features in a continuous HMM model. The feature extracting method was used to specifically preserve the local structure of the data manifold. Zhou et al. [2016] likewise used a dimensionality reduction technique on a set of traditional signal processing features to feed into a coupled-HMM to diagnose bearings. This shows the versatility that the HMM model has by simply introducing a hidden (latent) variable into the modelling of sequential processes.

1.2.4 Supervised, semi-supervised and unsupervised learning

The majority of machine learning tasks can be considered a special case of determining the joint probability of two random variables, $p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X} \in \mathbb{R}^{n \times k}, \mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ ($\mathbf{x}_n \in \mathbb{R}^k$) is any observed input variable and $\mathbf{Z} \in \mathbb{R}^{n \times m}, \mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ ($\mathbf{z}_n \in \mathbb{R}^m$) is any unobserved/latent or target variable. The form of the variables $\mathbf{X}$ and $\mathbf{Z}$ gives rise to some of the many common tasks that a machine learning algorithm can solve. For example, a discrete variable $\mathbf{Z}$ will give rise to a classification type problem. A continuous $\mathbf{Z}$, however, will give rise to a regression type problem. The data used to train the model will define whether the algorithm is supervised or unsupervised.

Supervised learning approaches can be categorized as approaches that are trained using both input data ($\mathbf{X}$) along with their target vector ($\mathbf{Z}$) [Bishop, 2006]. In bearing FDD, input data can be the features obtained from the raw vibration signal and the target vector can be the machine condition (healthy or unhealthy). The target could also be the RUL directly. Obtaining these target labels requires an extensive collection and storage of historical data with an equal distribution of data over all the classes to train a model. In some cases the data may just not be available, as is the case with new machines.

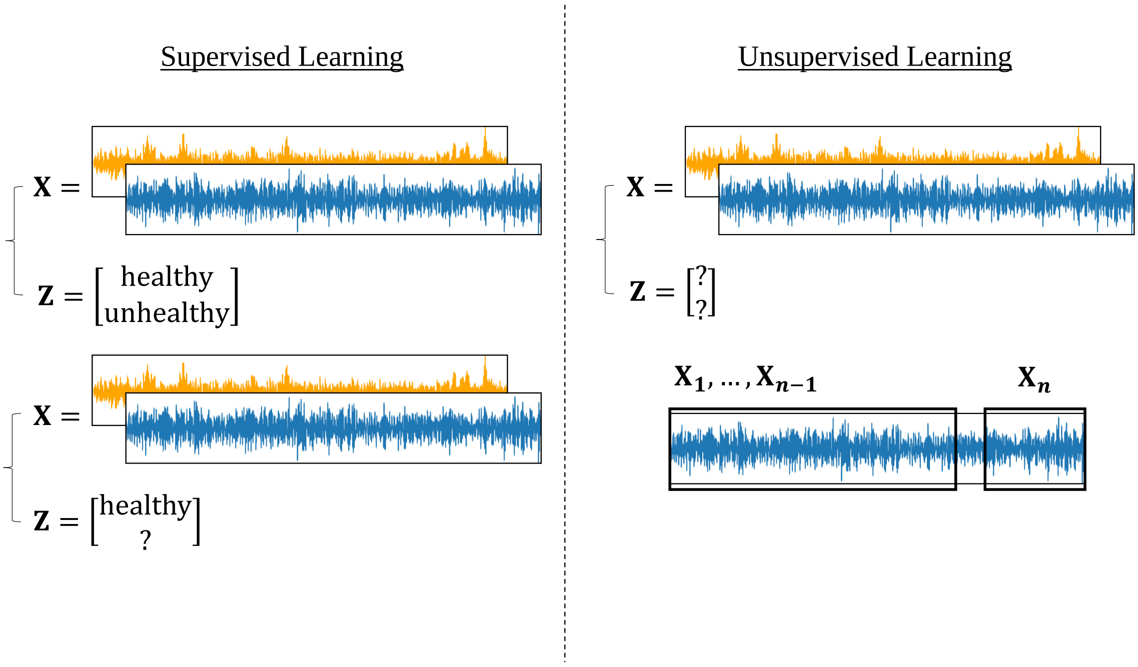
Unsupervised learning algorithms are trained using examples only from the input data ($\mathbf{X}$). In the unsupervised case the input is known as the observed data. It is important to note that the collected data in this instance is available without any labels. These types of algorithms learn using the structure in the data. In bearing FDD the vibration signal is used as the observed data.

Semi-supervised learning occurs when both labelled and unlabelled examples of data are available and used to estimate the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ or to predict $\mathbf{Z}$ from $\mathbf{X}$. The goal of semi-supervised learning is to learn a representation of the data such that examples from the same class are represented in a similar manner [Goodfellow et al., 2016].

It is not always easy to distinguish between unsupervised and supervised learning as there is no formal definition as to what constitutes a target or a label. As a result, at the intersection between a supervised and a unsupervised learning algorithm, the definition can vary between authors. This confusion can be highlighted with a small example. The chain rule of probability states for a random vector $\mathbf{X} \in \mathbb{R}^n$ where $\mathbf{X}$,

$$p(\mathbf{X}) = \prod_{k=1}^{n} p(\mathbf{x}_n | \mathbf{x}_1, ..., \mathbf{x}_{n-1}),$$

therefore the task of estimating $p(\mathbf{X})$ which is inherently unsupervised, may be cast as n supervised learning problems, by individually estimating $p(\mathbf{x}_1), ..., p(\mathbf{x}_{n-1})$.



Supervised Learning
$\mathbf{X} =$
$\mathbf{Z} = \begin{bmatrix} \text{healthy} \\ \text{unhealthy} \end{bmatrix}$
$\mathbf{X} =$
$\mathbf{Z} = \begin{bmatrix} \text{healthy} \\ ? \end{bmatrix}$
Unsupervised Learning
$\mathbf{X} =$
$\mathbf{Z} = \begin{bmatrix} ? \\ ? \end{bmatrix}$
$\mathbf{X}_1, \dots, \mathbf{X}_{n-1}$
$\mathbf{X}_n$

Figure 1.6: Example of cases that illustrate the definition of supervised and unsupervised learning used in this work.

In this work, a supervised algorithm will be defined as an algorithm that explicitly uses labels given by some external expert or through verified fault cases. An unsupervised learning algorithm is an algorithm that is trained purely on examples of the machine in question, regardless if the algorithm is trained using a supervised approach. Note, by this definition, semi-supervised learning (i.e. cases in which only partial labels are available), would be defined as supervised learning, as external labels are used in the training of the algorithms. Figure 1.6 shows cases which are considered supervised and unsupervised as per definition used in this work.

1.3 Scope of research

Performing accurate and reliable fault detection and diagnosis of bearings that are operating in non-stationary conditions remains an important and challenging task. With the fourth industrial revolution (4IR) upon us, it is imperative for industries to keep up to date with the latest trends and techniques in data analysis in order to achieve a competitive edge. Bearing FDD offers a good opportunity in which new deep learning techniques can be utilized to assist, together with current diagnostics techniques, and reach Industry 4.0 goals [Wang et al., 2018].

Currently, there are two dominant approaches towards bearing diagnostics. With the first approach, signal processing techniques are used to initially filter the signal to improve the signal of interest. This is followed by a transformation of the signal from the time domain into the frequency or order domain. Fault specified frequencies are then identified and tracked. The second approach requires a user to manually engineer fault specific features and track their degradation using a statistical model. Both of these approaches rely heavily on user experience and domain knowledge. This makes scaling and implementing the framework more challenging when a diverse range of similar assets needs monitoring.

An improvement in computational power has allowed a closing of the gap between conventional FDD approaches and deep learning based approaches. For a machine diagnostic algorithm to be implemented within the new framework of industry 4.0, the decisions have to be made as autonomously as possible. Deep learning incorporates feature extraction within its training protocol, as shown in Fig. 1.7. The deep learning extracted features can be used to augment subject matter experts' knowledge with new insights.

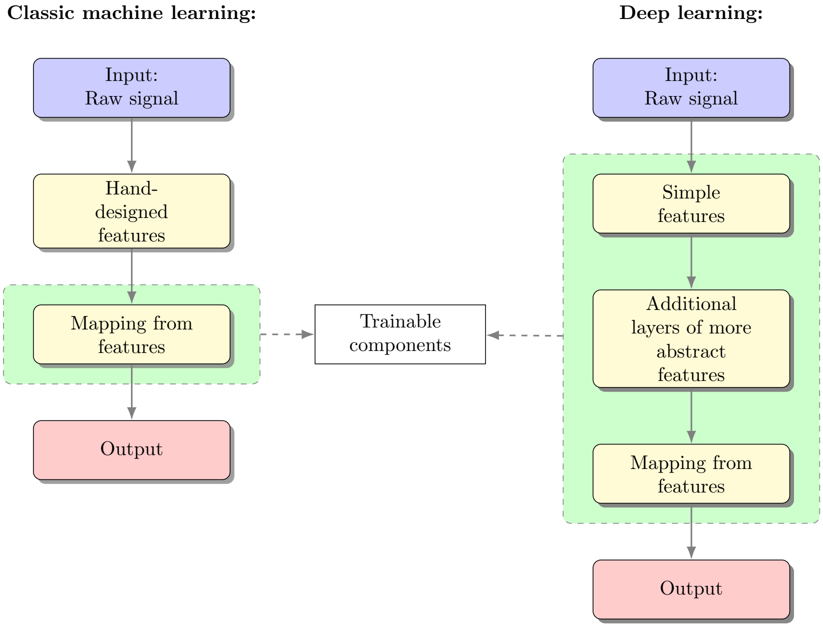


Figure 1.7: Comparison between classic learning and deep learning frameworks [Goodfellow et al., 2016].

With this in mind, the goal of this study is to investigate the use of deep learning algorithms for a bearing fault diagnosis methodology to provide a scalable and robust FSI for machines operating in non-stationary conditions [Gao et al., 2015]. To keep in line with the current diagnostic approaches, the proposed method should:

- Require no signal processing of the data.
- Require no expert intervention when training.
- Require no labelled or historic data to function.
- Require no auxiliary signal (eg. tachometer or shaft encoder).
- Be easily interpreted by machine operators.

The data that will be analysed in this study will be vibration signals from accelerometers. The acceleration of a machine containing a bearing, as measured by an accelerometer, provides a wealth of information about the internal forces experienced by that machine. Therefore, the acceleration signal is able to provide details about the severity and location of a bearing fault. Vibration signals are sufficient and no other data is needed in order to make a diagnosis. Accelerometers are relatively easy to install on many parts of a machine, further justifying their use in many diagnostic methods. Before any deep learning algorithm can be used it is assumed that the data lies on a lower dimensional manifold embedded in a higher dimensional space. Validation of the methodology is done using open source benchmark datasets.

The Current state of the art (SOTA) approaches in bearing diagnostics is to separate the deterministic and random components of the signal. Separating the deterministic components of the signal often requires knowledge of the instantaneous phase of a shaft within the machine. This requires additional measuring components and channels. To include such components in some machines, can be difficult or impractical. Furthermore, installing shaft encoders or tachometers in legacy assets is challenging. The sampling rate of these components needs to be high enough to get accurate resolution of the shaft rate when using time-domain based interpolation methods. This adds to the expense of the methodology. Therefore a methodology that does not require auxiliary signals would be advantageous. Researchers have proposed tacholess order tracking methods that extract phase information from the acceleration signals themselves, indicating that phase information is already contained within the accelerometer signal. The idea of separating random and deterministic components of the signal is a very important diagnostic concept and will be incorporated in this work.

Since all the diagnostic information is within the acceleration signal, the goal of the model is to learn the structure of the acceleration data. With the knowledge that the data has deterministic and random components, the algorithm is set up and regularized in a way that allows a representation of the random and deterministic components to be learned separately. The models are taken from deep learning work aimed at image generation, and hence some adjustments are required for acceleration data. A limitation of this study is the optimization of the many hyper-parameters that are involved in constructing a deep learning model. Where possible, hyper-parameters are taken from literature and adjusted until the algorithm works.

To summarize, this work uses existing knowledge of the data structure of bearing faults [Abboud et al., 2019] and incorporates it together with a deep learning model that is then trained in an unsupervised manner. The contribution of this work is as follows:

- This work shows how a deep learning model can be used as a FSI using an acceleration signal without the need for feature engineering or expert knowledge, providing a scalable FSI in accordance with Gao et al. [2015].
- This work shows how a deep learning model can be used as a FSI in non-stationary operating conditions without the need of auxiliary signals (eg. tachometer or shaft encoder).
- This work shows how a deep learning model can be used to diagnose bearing faults with little to no historical data.

1.4 Document overview

The remainder of this document is structured as follows. In the second chapter, a thorough literature review is performed on both machine learning concepts and their application towards bearing FDD. In the third chapter, the proposed methodology is presented along with a detailed outline of the model structure and parameters that were used for training purposes. All the experimental investigations that were conducted using the proposed method are then presented in Chapter 4. Finally, this study concludes with a summary of all the work and some recommendations for future work in Chapter 5.

Chapter 2

Deep learning: A review

2.1 Introduction

The goal of this chapter is to introduce the reader to some of the important concepts in deep learning. Using these concepts, an in-depth discussion is included on the similarities between a bearing diagnosis problem and an inference type problem. Furthermore, the important concept of dimension reduction and representation learning is presented within the context of bearing diagnostics. This chapter reviews some of the proposed deep learning methods for bearing diagnostics that are covered in literature. Please note that the emphasis of this chapter is on bearing diagnostics. Therefore, the derivation of some of the mathematical proofs used in deep learning is out of the scope of this work. Nevertheless, the absence of these derivations should not impact the understanding of the concepts. Where possible, a clear description of their meaning is given. Furthermore, references to original papers are made where the full derivation of the proofs can be found.

2.2 Inference

The problem of bearing FDD can be recast as a problem of evaluating the posterior distribution $p(\mathbf{Z}|\mathbf{X})$, given measured or observed variables $\mathbf{X} \in \mathbb{R}^{n\times k}$, to infer latent or unobserved variables $\mathbf{Z} \in \mathbb{R}^{n\times m}$. The typical observed variables for a bearing diagnostics problem are the measured acceleration (vibration) response of the machine. The latent variables on the other hand, are those variables that influence the vibration of the machine, yet are not measured directly. In rotating machines some examples of these variables can include the condition of the machine as well as machine operating conditions. The machine condition is the goal of the diagnostic algorithm. The latent variables that contain the machine operating conditions include operating speeds and loads. Both the condition of the machine and the machine operating conditions will change the distribution of the measured signal. Hence, if we can compute the posterior distribution, $p(\mathbf{Z}|\mathbf{X})$, we can use it to estimate the condition of the machine based on the observed accelerometer signal. Estimation of the posterior distribution is known as approximate inference. And to do this we need the joint density between the two variables, $p(\mathbf{Z}, \mathbf{X})$. With the joint density, the posterior can then be calculated using Bayes' theorem (Eq. 2.1), as follows

$$p(\mathbf{Z}|\mathbf{X}) = \frac{p(\mathbf{Z}, \mathbf{X})}{p(\mathbf{X})}. \tag{2.1}$$

The denominator of Eq. 2.1 is known as the evidence, and it is obtained by marginalizing the latent variables from the joint distribution following

$$p(\mathbf{X}) = \int p(\mathbf{Z}, \mathbf{X}) p(\mathbf{Z}) d\mathbf{Z}. \tag{2.2}$$

Unfortunately, performing the calculation of Eq. 2.2 is not tractable. This is due in part because the dimensionality of $\mathbf{X}$ of most practical applications is far too large to calculate these probabilities efficiently or in closed form. Therefore, an approximation method is required to estimate the marginal likelihood of Eq. 2.2. Two approximation schemes exist, namely: sampling methods and variational methods. Sampling methods are often computationally expensive to train on a large dataset and require some form of dimension reduction before making it viable. Thus, carefully hand engineered features are designed together with dimension reduction techniques to make sampling methods viable. An alternative approach to this is variational inference. Variational inference is an approach that is much better suited for probabilistic models that are far too complicated for sampling methods and its scalability makes it well suited for large input dimensions [Bishop, 2006].

2.2.1 Variational inference

Variational inference recasts the estimation of the posterior distribution, $p(\mathbf{Z}|\mathbf{X})$ as an optimization problem. Applying a suitable gradient based optimization method then resolves the posterior distribution. To do this, a family of densities, $q(\mathbf{Z}) \in \mathbb{D}$ is introduced on the latent variables. The optimum density requires the following minimization problem

$$q^*(\mathbf{Z}) = \arg \min_{q(\mathbf{Z}) \in \mathbb{D}} \mathrm{KL}\big(q(\mathbf{Z}) || p(\mathbf{Z}|\mathbf{X})\big), \tag{2.3}$$

to be solved, where $\mathrm{KL}(q||p)$ is the Kullback-Liebler divergence or KL-divergence between two densities q and p. The KL-divergence is a measure, adopted from information theory, of similarity between two densities. Note that the closer the densities, the smaller the divergence, $\mathrm{KL}(q||p) \geq 0$. $\mathrm{KL}(q||p) = 0$ if, and only if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X})$ [Bishop, 2006]. Once optimized, $q^*(\mathbf{Z})$ can then be used as an approximation to the actual $p(\mathbf{Z}|\mathbf{X})$. Figure 2.1 shows the variational inference optimization problem schematically. Starting with an initial parameter guess, the parameters $q(\mathbf{Z})$ are updated until the $\mathrm{KL}(q||p)$ distance between the variational distribution, $q(\mathbf{Z})$ and the actual distribution, $p(\mathbf{Z}|\mathbf{X})$ is at a minimum. Here the optimal parameters refer to the model parameters which are used to define the probability density within the family of densities $\mathbb{D}$.

The choice of the family of distributions $\mathbb{D}$ on $q(\mathbf{Z})$ is based in part on the tractability of the calculations, while any parametric distribution can be used. Highly flexible models are thus encouraged when using this method, since more flexibility equates to a better approximation. However, as the complexity of the model increases, the complexity of the optimization increases as well. Fortunately no over fitting occurs even when highly flexible models are used. Normally, neural networks are used because these models are flexible enough to sufficiently approximate the target distribution, while still being relatively easy to train.

2.2.2 Evidence lower bound

Note that the divergence presented in Eq. 2.3 still requires $p(\mathbf{Z}|\mathbf{X})$, which depends on the joint $p(\mathbf{Z}, \mathbf{X})$ and the marginal $p(\mathbf{X})$. This problem is solved by mathematical manipulation of Eq. 2.3 to recover what is known as the evidence lower bound (ELBO) [Blei et al., 2017]. By expanding the KL-divergence term in Eq. 2.3,

$$\begin{aligned} \mathrm{KL}\big(q(\mathbf{Z}) || p(\mathbf{Z}|\mathbf{X})\big) &= \mathbb{E}_{z\sim q(\mathbf{Z})}[\log q(\mathbf{Z})] - \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{Z}|\mathbf{X})] && (2.4) \\ &= \mathbb{E}_{z\sim q(\mathbf{Z})}[\log q(\mathbf{Z})] - \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{Z}, \mathbf{X})] + \log p(\mathbf{X}) && (2.5) \end{aligned}$$

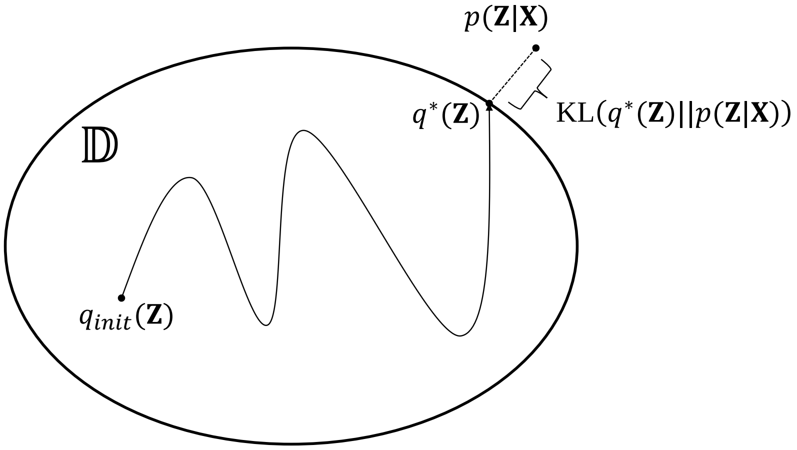

$p(\mathbf{Z}|\mathbf{X})$
$q^*(\mathbf{Z})$
$\mathrm{KL}(q^*(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}))$
$\mathbb{D}$
$q_{init}(\mathbf{Z})$

Figure 2.1: Variational inference schematic, Blei [2019].

the dependence of the optimization problem on the evidence, Eq. 2.2 is revealed. The last term in Eq. 2.5 is constant and does not effect the optimization path. By dropping $\log p(\mathbf{X})$, we can define what is known as the ELBO,

$$\mathrm{ELBO}(q) \quad = \quad \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{Z},\mathbf{X})] - \mathbb{E}_{z\sim q(\mathbf{Z})}[\log q(\mathbf{Z})]. \tag{2.6}$$

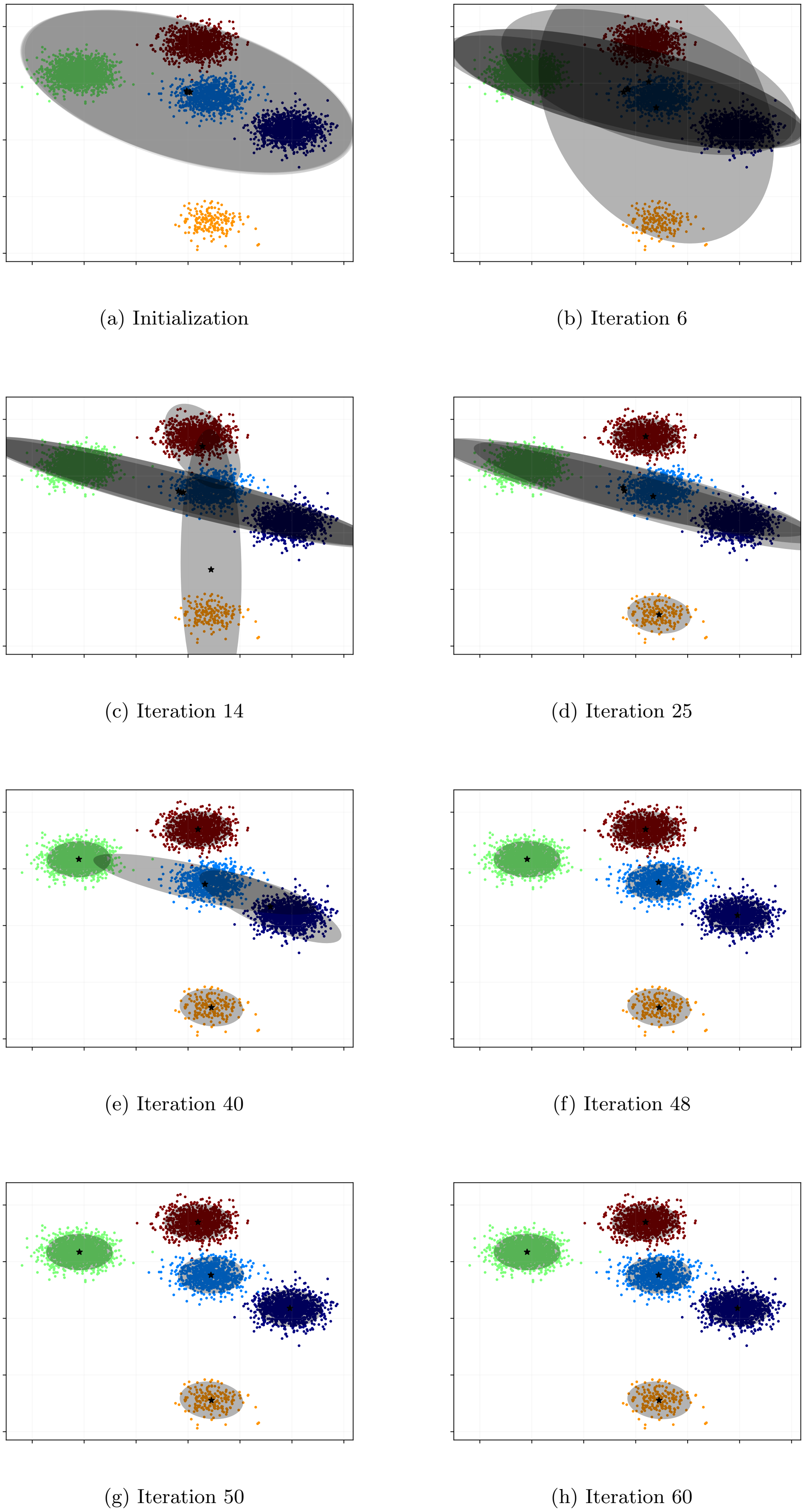
The ELBO is equal to the negative KL-divergence plus the log of the evidence, $\log p(\mathbf{X})$. The log evidence is constant with respect to the latent distribution, $q(\mathbf{Z})$. Therefore maximising the ELBO is equivalent to minimizing the KL-divergence, and can be used as a substitute for the KL-divergence when optimizing.

This concept is well illustrated by the following mixture of Gaussians example. In this example, the observed variables are the samples taken from a Gaussian mixture model (GMM), and the latent variables are the means and the mixture components of the model. The task is to recover these means and mixture components, through only the samples of the observed variables. Maximization of the ELBO a nonlinear problem and therefore requires iteration. Fig. 2.2 depicts the result of variational inference at various iterations. We can see at initialization, the variational factors used to approximate the data distribution overlap and do not cover the entire support of the problem. This however is no problem, as the iterations progress we can see the algorithm explores the data space until it converges correctly to each Gaussian mixture component.

One can rewrite the ELBO as the sum of the log likelihood and the KL-divergence of the prior distributions, $p(\mathbf{Z})$ and $q(\mathbf{Z})$,

$$\begin{aligned}\mathrm{ELBO}(q) &= \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{Z})] + \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{X}|\mathbf{Z})] - \mathbb{E}_{z\sim q(\mathbf{Z})}[\log q(\mathbf{Z})] \quad (2.7)\\ &= \mathbb{E}_{z\sim q(\mathbf{Z})}[\log p(\mathbf{X}|\mathbf{Z})] - \mathrm{KL}\big(q(\mathbf{Z})||p(\mathbf{Z})\big). \quad (2.8)\end{aligned}$$

This reveals that when optimized, the first term in Eq. 2.8 will encourage densities that place their mass on configurations of the latent variable, that explains the observed data. The second term in Eq. 2.8, encourages the densities to stay close to the prior. Thus the optimal solution balances the likelihood and the prior densities. By plotting the ELBO at



(a) Initialization

(b) Iteration 6

(c) Iteration 14

(d) Iteration 25

(e) Iteration 40

(f) Iteration 48

(g) Iteration 50

(h) Iteration 60

Figure 2.2: Progression of variational inference performed on a simple 2-Dimensional Gaussian Mixture Model (GMM). The grey ellipse shows a 2σ standard deviation of the learnt variational approximating factor.

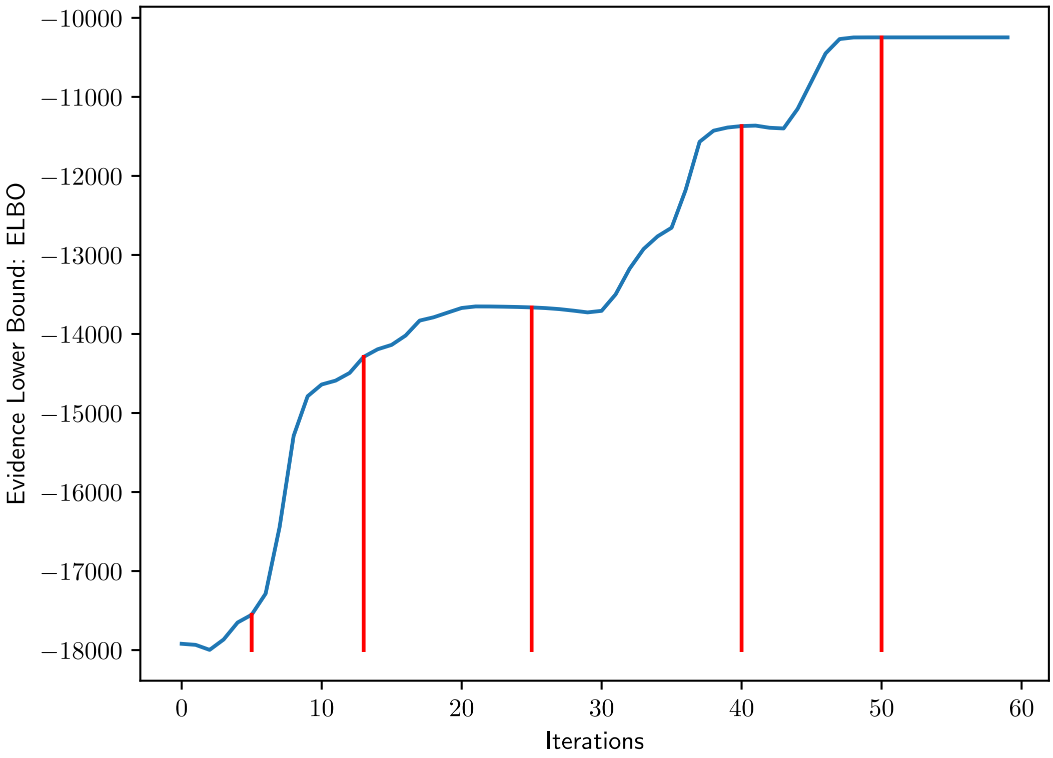


Figure 2.3: Evidence lower bound (ELBO) of the 2D GMM model in Fig. 2.2. Convergence can be seen in Fig. 2.2 after Iteration 48.

each iteration we can get a glimpse into the convergence of this maximization problem. In Fig. 2.3 the red lines correspond to the iterations seen in Fig. 2.2. Here you can see why the name ELBO is given. During each iteration a trade-off is made between the likelihood and the prior which causes these sharp elbow like bends to occur. The first term will encourage the algorithm to explore the input space, while the second will optimize what is already known. Again, this can be seen in Fig. 2.2e, where two of the mixture components have been found whilst the remaining three are still being explored. Although this is a basic example, it clearly demonstrates the power of variational inference to use the observed data to infer the unobserved latent variables.

2.3 Dimension reduction

A very important concept in PHM is dimension reduction. Many proposed algorithms work well but require some form of dimension reduction in order to improve computational efficiency or accuracy of the model. In this section we take a look at some of the technicalities behind dimension reduction as well as some of the important assumptions we need to make about PHM data for any of these techniques to work.

Dimension reduction techniques can be either classified as linear or non-linear. Many proposed methods of bearing diagnostics rely on either technique. One may ask the question, however, why is it that observations in high dimensions can be reduced to a lower dimension representation? An alternative name for dimension reduction is manifold learning, and a very important assumption is made about the structure of the data that allows for the dimension reduction to take place. The assumption is known as the manifold hypothesis.

2.3.1 The manifold hypothesis

Training a bearing diagnostics model is often done under the assumption that the high dimensional feature space sits somewhere on a low dimensional manifold. Data from physical systems tend to reside in high dimensional spaces, but they may have been generated by only a few degrees of freedom in some underlying process. If this is not the case, we would not be able to perform machine learning. Therefore, we make the assumption that the data resides on or near a lower dimensional manifold embedded in the higher dimensional space in this study. We then use a suitable manifold learning technique to extract it from the higher dimensional observations. Put differently, we simply reduce the dimension of the data, whilst retaining the important information about the structure of the data. This assumption is reasonable as the data is generated by machines that only allow for restricted motion within the realm of what is allowed by physics. This significantly constrains the subset of signals that can be observed. Earlier works of vibration condition monitoring refer to this phenomenon as a vibration signature.

The latent variable $\mathbf{Z} \in \mathbb{R}^m$, with $m << n$ is a lower dimensional representation of the observed variables $\mathbf{X} \in \mathbb{R}^n$, since $\mathbf{Z}$ explains all the details and anomalies observed in $\mathbf{X}$ in a structured and compact form. Deep learning relies on this, as the layers get deeper, the algorithm is able to unfold the lower dimensional manifold within the higher dimensional observations to construct decision boundaries in this lower dimensional space.

In unsupervised dimensional reduction, the data observed in isolation is reduced. However, because of the manifold hypothesis, similar data is likely to lie in close proximity on the data manifold and as such can be used to classify unlabelled data.

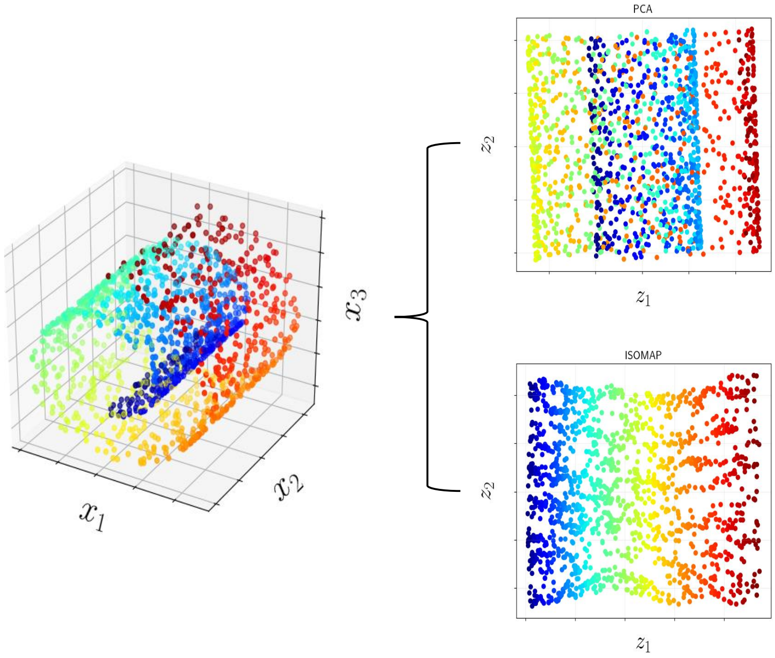


Figure 2.4: The Swissroll dataset is a typical manifold dataset in 3-dimensions. The dataset can easily be reduced to 2-dimensions, however a non-linear method is required to preserve local structure on the manifold. Two methods, PCA (linear) and ISOMAP (non-linear) are shown as an example.

The choice of dimension reduction technique will depend on how the manifold is embedded within the high dimensional space. If the manifold is embedded linearly, a simple PCA works very well. However if the embedding of the manifold is non-linear and more local structure of the manifold must be preserved, a non-linear technique (for example ISOMAP) may be better suited. Figure 2.4 shows a classic Swissroll dataset in 3D. This dataset can easily be represented in 2D. It is important, however to note how the dimensionality reduction technique will affect the representation of the dataset in 2D. Nevertheless, manifold learning is a powerful idea which has been successfully used in condition monitoring. Yuan and Liu [2013] used manifold regularization to take advantage of unlabelled data. This allowed them to create an algorithm that clustered vibration signals measured under the same machine operating conditions, and subsequently perform bearing diagnostics.

2.3.2 Entropy and Mutual Information

In information theory, entropy is a measure of the amount of randomness (or destruction) held by a variable. To put it differently, entropy is a measure of the amount of information in a variable. Given the probability of a variable, $p(\mathbf{X})$, entropy is calculated as [Bishop, 2006]

$$H[\mathbf{X}] = -\int p(\mathbf{X}) \ln p(\mathbf{X}) d\mathbf{X}. \tag{2.9}$$

That is, the entropy is the average amount of information needed to fully define the state of a variable, $\mathbf{X}$. Furthermore, given a joint distribution, $p(\mathbf{X}, \mathbf{Z})$, the entropy of two variables can be calculated in a similar manner. Stemming from this, the conditional entropy, $H[\mathbf{Z}|\mathbf{X}]$ can equally be defined as [Bishop, 2006]

$$H[\mathbf{X}, \mathbf{Z}] = H[\mathbf{X}] + H[\mathbf{Z}|\mathbf{X}]. \tag{2.10}$$

The conditional entropy states that the amount of information needed to define the pair $(\mathbf{X}, \mathbf{Z})$, is equal to the amount of information needed to describe $\mathbf{X}$ plus the amount of information needed to describe $\mathbf{Z}$ given $\mathbf{X}$. Therefore the conditional entropy is the amount of extra information a variable contains, given that a different variable is already known.

In addition, a measure known as mutual information can also be defined using entropy and conditional entropy by the following [Bishop, 2006]

$$\begin{aligned} I[\mathbf{X}, \mathbf{Z}] &= H[\mathbf{X}] - H[\mathbf{X}|\mathbf{Z}] && (2.11) \\ &= H[\mathbf{Z}] - H[\mathbf{Z}|\mathbf{X}]. && (2.12) \end{aligned}$$

Mutual information is, therefore, a measure of the amount of information given about $\mathbf{X}$, by merely observing the variable $\mathbf{Z}$ or *vice versa*. In the context of dimension reduction, we can assess the amount of information lost in our reduction technique, by measuring the mutual information between $\mathbf{X}$ and $\mathbf{Z}$. Thus if $I[\mathbf{X}, \mathbf{Z}] = 0$, it can be said that no information was lost during the dimension reduction process, as knowing $\mathbf{Z}$, is as good as knowing $\mathbf{X}$. (ie. no additional information is needed). The reader does not need to know how these measures are derived in order to understand the remainder of this work. However, it is important to understand the basic idea behind each measure and what it represents as the remainder of the study relies on this understanding.

Representation learning

Representation learning is a philosophy within deep learning in which a semantic, organised structure, is learned automatically by a machine. This is done by utilizing the concepts presented previously when training the model. Usually, we as humans have no problem identifying semantic characteristics in data, which we conduct mostly instinctively. However it is a challenge to define how a machine must learn it. Representation learning based algorithms learn semantics through the structure of unlabelled data, which is well suited for diagnostics of bearings under non-stationary operating conditions. This is even more so, because the model is not centred around the data, but incorporates whatever data is available, making it suitable for cases where only sparsely labelled data is available.

2.4 Machine learning basics

Neural networks are needed to have a model with enough flexibility to approximate the manifold from bearing data. Machine learning is the class of methods by which these neural network models are defined, assembled and subsequently trained. This section covers some of the basic concepts in this regard.

2.4.1 Model building blocks

There are a vast number of ways neural network (NN) based models can be built, however deep learning is interested in a class of models based on a feed forward NN with multiple layers. NNs are considered universal function approximators, and are comprised of several layers of functions feeding into one another. Each layer extracts features from the previous layer. As the layers get deeper, more discriminative features are then able to be extracted. Shallow layers extract local features whilst deeper layers tend to extract more global features. Hence, a hierarchy of features can be learnt by using a NN. A discussion of all the NN layer configurations is outside the scope of this work, however two important layers used in this work are discussed next, namely, fully connected (FC) and convolution (Conv) layers.

Fully connected layers

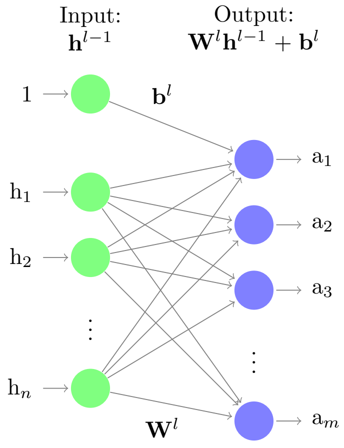


Figure 2.5: Concept of fully connected layers. Every input value or feature is connected to every output value. The size of the input, n can be greater or less than the output, m, resulting in a $n \times m$ weight matrix $\mathbf{W}^l$ and a m bias vector, $\mathbf{b}^l$.

The most basic function for a layer consists of a transformation of an input variable followed by a non-linear activation function $\sigma(\cdot)$,

$$\mathbf{a}^l = \mathbf{W}^l\mathbf{h}^{l-1} + \mathbf{b}^l \tag{2.13}$$

$$\mathbf{h}^l = \sigma(\mathbf{a}^l), \tag{2.14}$$

where $\mathbf{W}^l$ and $\mathbf{b}^l$ are a set of model parameters known as weights and biases respectively that require training. The superscript l indicates the layer number. Many layers can be added together consecutively, where the output of one layer is the input to the next layer. Such a network is known as a fully connected neural network. $\mathbf{a}^l$ are the activations which are passed through some non-linear activation function $\sigma(\cdot)$. The output of the activation function is known as the hidden unit, $\mathbf{h}^l$, which is subsequently fed into the next layer of the network.

The activation function is a vital aspect of the neural network, as it introduces non-linearity into the model. Without these activation functions, the model would essentially be a collection of linear transformations, which in itself is a linear transformation, and learning would be limited. The activation functions are, in essence, what allows the stacking of layers and the learning of hierarchical features to occur.

The activation functions are generally sigmoid functions such as the logistic sigmoid or the tanh function, both of which are examples of the exponential class of functions. More recently, non-exponential functions have gained increasing popularity over functions from the exponential class to improve the convergence of training. There are many activation functions to choose from, and investigating each is beyond the scope of this work. Shao et al. [2018], however, included a full investigation into the use of 15 activation functions in their intelligent rolling bearing fault diagnosis model. They concluded that no one specific activation is better than another, but an ensemble of various activation functions within a network allows for overall improved accuracy.

Convolution layers

A special case of a FC layer, is a 1D convolution layer or a temporal convolution layer. Similar to a FC layer it consists of a transformation followed by a non-linear activation function. However, the function has a smaller set of weights $\mathbf{W}^l$, which are convolved over the input layer,

$$\mathbf{a}^l = \mathbf{W}^l \otimes \mathbf{h}^{l-1} + \mathbf{b}^l \tag{2.15}$$

$$\mathbf{h}^l = \sigma(\mathbf{a}^l). \tag{2.16}$$

Figure 2.6 shows a schematic layout of the function of a convolution layer. Here, the convolution weights, $\mathbf{W}^l$, are convolved over the input, $\mathbf{X}$, to produce as set of new features.

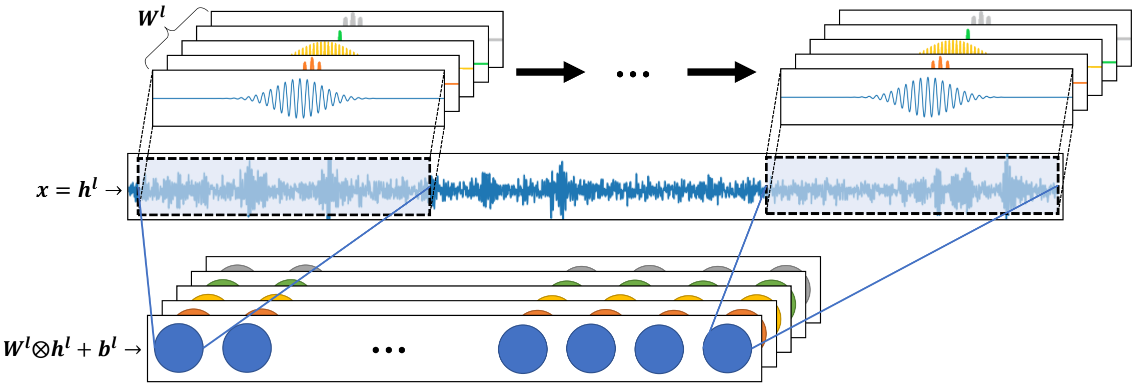


Figure 2.6: Conceptual design of convolution layers. A set of trainable kernels, $\mathbf{W}^l$ are convolved across the input values $\mathbf{X}$, where each kernel represents a feature of importance.

The fact that fewer weights are used in convolution layers means they are far more efficient than FC in terms of training time and memory requirements. Performance is not lost when using convolution layers as they provide a strong and useful inductive bias as to what the convolution neural network (CNN) algorithm can learn [Huszar, 2018].

Convolution layers are equivariant to translation. A function f is equivariant to function g if

$$f(g(x)) = g(f(x)).$$

Suppose we have a time series $\mathbf{x} = \{x_1, ..., x_T\}$ and $\bar{x} = \{x_\alpha, ..., x_{\alpha+l}\}$ is some subset of $\mathbf{x}$ of length l, where $\alpha \in \{\mathbb{Z}|\alpha > 1\}$ and $\alpha + l < T - 1$. Furthermore, suppose there is some shifting function g such that $\bar{x}' = g(\bar{x}) = \{x_{\alpha+1}, ..., x_{\alpha+l+1}\}$. Then

$$W \otimes \bar{x}' = g(W \otimes \bar{x}).$$

This basic property, gives the convolution layer immense power to extract features within a time series signal, regardless of when the event that produced the feature occurred. Thus the features that are extracted are not time or phase dependent, making them useful feature extractors for fault prediction algorithms. Many researchers have investigated the use of CNNs to analyse time series data [Cui et al. [2016], Wang et al. [2016], Wang et al. [2017], Serrà et al. [2018]], and designated them to be simple yet powerful and versatile building blocks.

2.4.2 Model training

A deep learning model's performance is judged by a metric that is defined by the overall goal of the algorithm. Usually, this metric is intractable and thus not optimized directly. Instead, the metric is normally optimized indirectly with a cost function $L(\boldsymbol{\theta})$ (where $\boldsymbol{\theta}$ is the collection of all trainable parameters), chosen by some tractable relation to the metric. The ELBO in Eq. 2.4 is a very good example of a cost function that indirectly optimizes another function (KL-divergence) that approximates the posterior distribution $p(\mathbf{Z}|\mathbf{X})$. The cost function for machine learning algorithms is usually taken as the expectation over the training data or the empirical data distribution $\hat{p}_{data}$ [Bishop, 2006]

$$\begin{aligned} L(\theta) &= \mathbb{E}_{(\mathbf{x})\sim\hat{p}_{data}}\mathcal{L}(f(\mathbf{x}, \theta)) && (2.17) \\ &= \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(f(\bar{x}_i, \theta)) && (2.18) \end{aligned}$$

where $f(\mathbf{x}, \theta)$ represent the neural network layers with model parameters, θ. The cost function can be decomposed as a sum over the training examples, where m is the number of training samples. Gradients of the cost function

$$\nabla_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\nabla_\theta\mathcal{L}(f(\bar{x}_i, \theta)), \quad (2.19)$$

are computed using back propagation [Rezende et al., 2014]. Almost all optimization algorithms used for training of a machine learning algorithm are based on stochastic gradient descent (SGD). The reason is the large size of the training set that is required to get good generalization capacity from these models. Increasing the size of the training set, increases the cost of calculating the gradients. With SGD, smaller batches of the training data are used to estimate noisy expectations of the cost function instead of using the entire dataset at once. The minibatch of size m' is drawn randomly from the training data and used to estimate the gradient from the minibatch, followed by a descent downhill from that estimation, with a learning rate of λ.

$$\begin{aligned} \nabla_\theta L' &= \frac{1}{m'}\nabla_\theta\sum_{i=1}^{m'}\mathcal{L}(f(\bar{x}_i, \theta)) && (2.20) \\ \theta &\leftarrow \theta - \lambda\nabla_\theta L' && (2.21) \end{aligned}$$

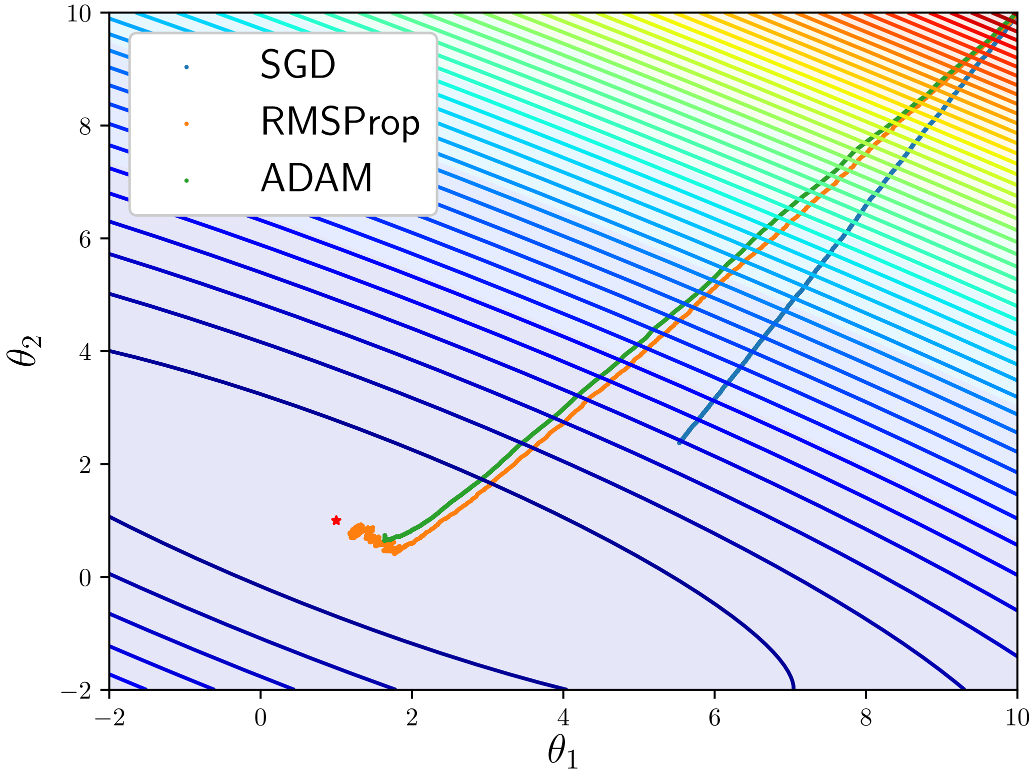


Figure 2.7: Gradient descent paths of 3 stochastic optimization techniques after 25 iterations in the parameter space of a convex function. Optimum is at value (1, 1). Note the slow convergence of SGD compared to ADAM and RMSProp.

Although SGD solves the issue of large training datasets, it is slow to converge to optimum points. Figure 2.7 shows the SGD algorithm on a convex function compared to two popular algorithms, RMSProp and ADAM, which are discussed next. A discussion on all the optimization techniques are out of the scope of this work, however two methods which have been used in this work, RMSProp and ADAM are briefly highlighted.

RMSProp

RMSProp is based on resilient back propagation adapted for mini batch learning [Hinton et al., 2012]. In resilient back-propagation (RProp), the magnitude of the partial derivatives gradients are ignored and only the sign is taken into consideration for each upgrade of parameters [Riedmiller and Braun, 1992]. When used as a minibatch optimizer, RProp may result in unstable behaviour. RMSProp prevents this from happening by averaging the gradients over the minibatches, which keeps the updates bounded.

ADAM

ADAM [Kingma and Ba, 2014], standing for adaptive moments, is the de-facto optimizer for a large number of machine learning and deep learning cases. It is well suited for problems with large data and parameters. ADAM computes an individual adaptive learning rate for each parameter from different estimates of the first and second moments of the gradients.

2.5 Deep learning for bearing FDD

Aided by the enhancement of computational power, improved data acquisition, increased data storage capacity and multi-disciplinary research, deep learning approaches towards bearing FDD are gaining traction within academic research. Researchers are making use of the ability of deep learning algorithms to extract complex features without the need for expert intervention, or signal processing techniques. These algorithms come in many shapes and forms which depend on the data availability and the end goal of the CBM strategy. This section covers a few of these methods, considering deep learning models that are trained end-to-end to perform FDD on rotating machines.

2.5.1 Deep neural nets

Because of their ability to extract hierarchical features, deep neural nets have been used in bearing diagnostics in the following ways. Janssens et al. [2016] were some of the first researchers to apply deep learning towards fault detection of rotating machinery using vibration data, in what they called a feature learning approach. They used the discrete fourier transform (DFT) of a normalized data from two accelerometers as inputs to a network that comprised one convolution layer, one fully connected layer and a classification layer in a supervised setting. Interestingly, they concluded that a deeper architecture would not yield better results.

Nevertheless, Guo et al. [2016] used a deep convolution neural network and proposed a hierarchical based training strategy to classify bearing faults from vibration signals. Their network comprised of three convolution layers followed by two fully connected layers ending in a softmax layer. The whole network was trained end-to-end using a softmax loss function and gradient descent. Once the fault was classified, they further trained another network to evaluate the size of the fault.

Ince et al. [2016] used raw motor signals to perform fault detection by fusing fault feature extraction and classification into one deep neural network. The network comprised of four 1D adaptive convolution layers followed by three fully connected layers. The network was trained end-to-end with labelled data to classify the signals into one of two classes: healthy or faulty.

Jia et al. [2018b] used a deep neural network comprising two sets of normalized convolution layers and pooling layers followed by three fully connected layers to classify raw vibration signals into one of 8 bearing fault classes. Examples of both single point and multi-point faults were used. The authors used a weighted softmax loss function to deal with the class imbalance, where the weights were chosen based on the imbalanced degree of the dataset.

Zhang et al. [2018] proposed a 6 layer CNN network followed by a fully connected layer and a softmax layer respectively, to classify raw vibration signals into one of 10 different fault classes. The authors deal with the issue of varying machine loads and noisy environments by suggesting a training procedure that includes drop-out, mini-batch training and ensemble learning.

An alternative approach was offered by Ding and He [2017], where they used the wavelet packet transform to generate wavelet-images which were fed into a network consisting of 3 convolution layers to learn features. These features were fed into a fully connected layer and used to classify faults in bearings in a full end-to-end algorithm using labelled data.

It is evident from all these approaches that deep learning was used as the feature extractor in the algorithm. Very few methods used hand designed features, however the underlying framework of their approaches are similar. In all the cases presented above, the features which had been extracted using deep learning methods were then subsequently used in a supervised setting, still requiring the need for expensive labelled data.

2.5.2 Variational Autoencoders

An autoencoder (AE) is an unsupervised dimension reduction deep feed forward neural network that attempts to reconstruct the input data from reduced dimensional latent variables. A variational autoencoder (VAE) [Kingma and Welling, 2013], applies this concept to perform inference. A variational autoencoder performs variational inference using a neural network as the family of distributions for the variational model. Training of the VAE is done by optimizing the ELBO presented in Eq. 2.8. A re-parametrization trick [Kingma et al., 2015] is used to allow for gradient based optimization. The network learns to represent the data in lower dimensions by encoding the input data into a latent space, **Z**. **Z** takes the form of a mean and a variance, $\mathbf{Z} = E(\mathbf{X}) = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\eta} \approx \boldsymbol{q_{\phi}(\mathbf{Z}|\mathbf{X})}$, where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\phi = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. The input data is then reconstructed using a decoder network, $\tilde{\mathbf{X}} = D(\mathbf{Z}) \approx p_{\theta}(\mathbf{X}|\mathbf{Z})$.

An example of an encoder-decoder network is shown in Fig. 2.8. The encoder network takes as an input, a portion of the signal, **X** and encodes it down to its latent representation **Z**. This network can comprise of many of the neural network building blocks mentioned previously. Similarly, the decoder network takes the latent representation **Z** and attempts to reconstruct the input to produce $\tilde{\mathbf{X}}$. With the re-parametrization trick, Eq. 2.8 now takes the form of Eq. 2.22. By encoding into a latent dimension that is much lower than the input dimension, the features that are extracted retain important discriminative information. With a VAE, the posterior distribution, $p(\mathbf{Z}|\mathbf{X})$, is approximated by the fully trained encoder and learnt entirely through optimization of the loss function

$$\mathcal{L} = \frac{1}{2}(1 + \log(\boldsymbol{\sigma}^2) - \boldsymbol{\mu}^2 - \boldsymbol{\sigma}^2) + \log(p_{\theta}(\mathbf{X}|\mathbf{Z})). \tag{2.22}$$

A few authors have used autoencoders in their approach towards bearing diagnostics. A two step methodology of training using unlabelled data in a feature extraction step followed fine tuning a classifier utilizing the labelled data is the predominant diagnostic methodology used with autoencoders. This methodology was first introduced in the work of Sun et al. [2016], where they used a sparse autoencoder to learn features represented by the encoded vibration

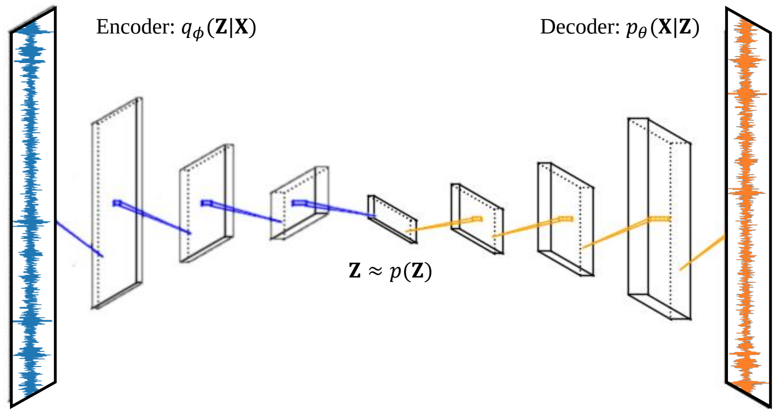

Encoder: $q_\phi(\mathbf{Z}|\mathbf{X})$
Decoder: $p_\theta(\mathbf{X}|\mathbf{Z})$
$\mathbf{Z} \approx p(\mathbf{Z})$

Figure 2.8: Schematic layout of the structure of an autoencoder.

signals of an induction motor. They encouraged sparsity within the weights of the network using a KL-based penalty function. The features that were extracted were entirely machine learned. These features were then used in a neural network classifier trained and fine tuned using labelled data.

Thirukovalluru et al. [2016] used stacked de-noising autoencoders on traditional features extracted from a vibration signal in the time and frequency domains respectively. The AE was used to extract higher level features from these traditional approaches. Layer-wise training was used in the first phase of training in order to initialize the weights. The weights were then fine-tuned to increase classification performance. Although deep learning was used, hand engineered features were still being used as inputs, and the full power of deep learning not yet exploited.

Similarly, Shao et al. [2017] used a deep auto-encoder to learn features from a bearing signal in non-stationary environments. The AE was trained using an entropy based loss function making the features insensitive to complex and non-stationary background noises. The learned features were then fed into a softmax classifier and fine tuned using labelled data. The same authors went on to propose another deep AE where weighted majority voting was used to classify bearing signals into one of twelve fault states [Shao et al., 2018].

Following a same structure, Jia et al. [2018a] trained a normalized sparse auto-encoder with local connection network to get features from raw vibration signals of a gearbox and bearing using data in the frequency domain. The features were then used in a ten class fault classifier. Again, we see labelled data that was used to pre-train and improve the weights of the network.

Lu et al. [2017] used a stacked de-noising autoencoder (SDAE) for signals from machines working in fluctuating operating conditions. The SDAE was trained in an unsupervised manner in order to extract features. Again, these features were then used in a supervised learning protocol to create a classifier for bearing faults.

Even with advanced models such as VAEs, we can see that the deep learning model does all the heavy lifting by extracting the most discriminative features. However, these proposed methodologies often require the user to use these machine learned features in a supervised setting with labelled data.

2.5.3 Generative adversarial networks

Generative adversarial networks (GANs) [Goodfellow et al., 2014] exploits deep learning architecture to build a representation of a target distribution density without explicitly parametrizing a set of density functions. A GAN, like an autoencoder, is comprised of two sub-networks known as a generator network, $G_\theta(\mathbf{Z})$ and a discriminator network, $D_\phi(\mathbf{X})$. Here θ and ϕ are the trainable parameters of the network (such as the weights and biases) and $\mathbf{X}$ is again a portion of the vibration waveform. An example of a GAN network with the generator and the discriminator is shown in Fig. 2.9.

Training of the networks is achieved by setting the generator network against the discriminator network in a two player non-cooperative game expressed as the min-max optimization

$$\begin{aligned}\min_\theta \max_\phi L\big(D_\phi(\mathbf{X}), G_\theta(\mathbf{Z})\big) = \quad & \mathbb{E}_{\mathbf{X}\sim P_{data}}[\log D_\phi(\mathbf{X})] + \\ & \mathbb{E}_{\mathbf{Z}\sim \mathcal{U}/\mathcal{N}[0,1]}[\log(1 - D_\phi(G_\theta(\mathbf{Z})))]. \qquad (2.23)\end{aligned}$$

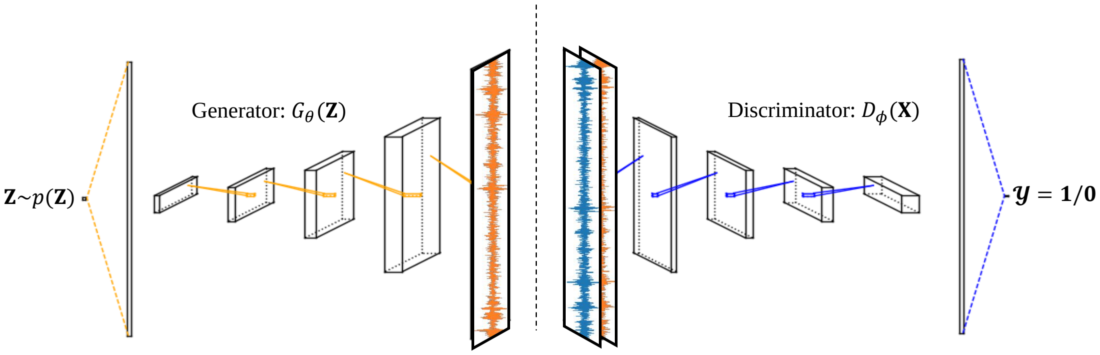
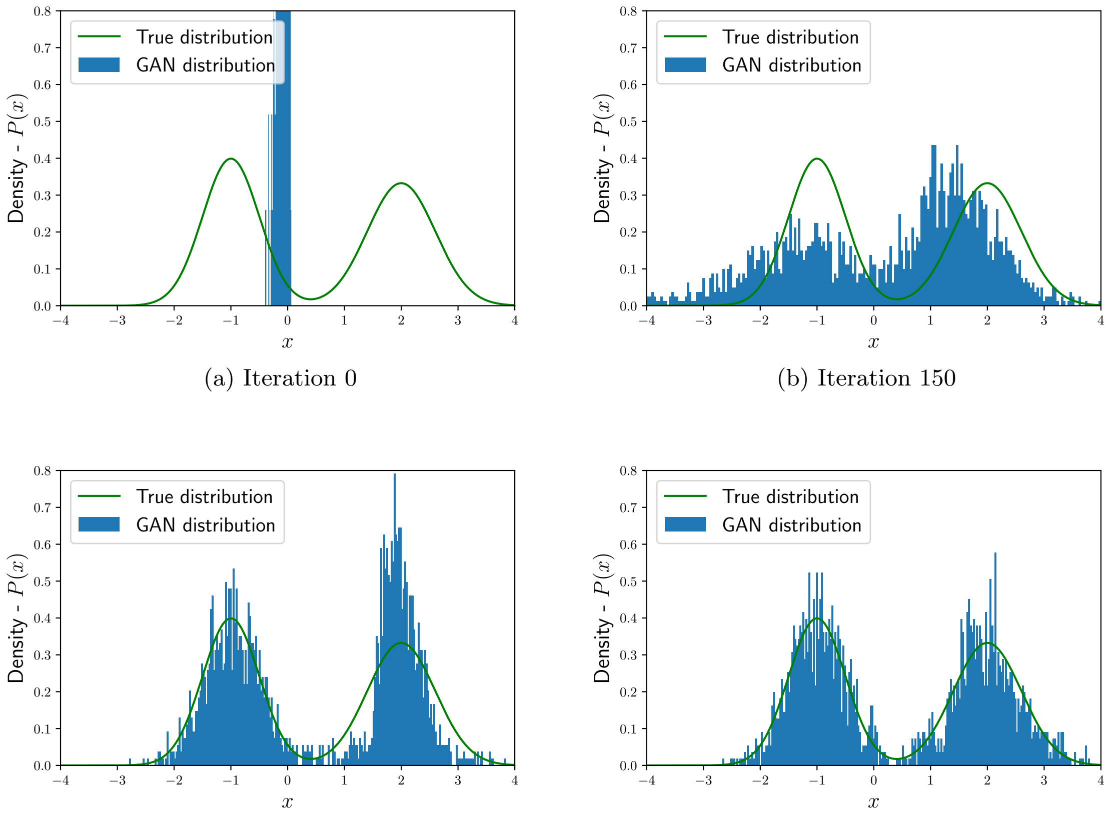

Z~p(Z)
Generator: $G_\theta(\mathbf{Z})$
Discriminator: $D_\phi(\mathbf{X})$
y = 1/0

Figure 2.9: Schematic layout of the structure of a Generative Adversarial Network (GAN).

The generator network passes random noise, $\mathbf{Z} \sim \mathcal{U}[0,1]$ or $\mathbf{Z} \sim \mathcal{N}[0,1]$ through the network and produces a sample from a parametrized distribution, $\tilde{\mathbf{X}} \sim q_\theta$. The discriminator network tries to estimate the probability that the query sample was either produced by the generator or from the training set, $\mathbf{X} \sim p_{data}(\mathbf{X})$. Each sub-network updates its own parameters with gradients derived from the cost function defined in Eq. 2.23 and back-propagation of the error. The generator improves the produced samples based on the feedback (gradients) obtained from the discriminator. Theoretically, training is completed, when Nash-equilibrium is reached. In game theory, Nash-equilibrium occurs when a player has reached a point that no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged. At Nash-equilibrium, the discriminator can no longer distinguish between samples produced by the generator and samples drawn from the target data distribution, and thus $q \approx p_{data}$. The discriminator, now unsure of whether the sample is real or generated, outputs a probability of $D_\phi\big(\mathbf{X}$ or $G_\theta(\mathbf{Z})\big) \approx 0.5$.

We can link the GAN back to inference in the following manner. In variational inference models, learning is achieved by optimizing for the KL-divergence between the real data and the model. Learning in GANs can be achieved by optimizing the density ratio between the



(a) Iteration 0

(b) Iteration 150

(c) Iteration 250

(d) Iteration 400

Figure 2.10: Progression of a GAN as it learns the distribution that generated samples of **X**. In this example, **X** was sampled from a Gaussian with two mixture components.

real data and the model [Mohamed and Lakshminarayanan, 2016]. The discriminator is trained to classify between observed data and generated data. For the problem we have the true observed data density, $p_{data}(\mathbf{X})$, as well as the generator's model of the observed data, $q_\theta(\mathbf{X})$. To build a classifier, we introduce a random variable y. When the sample is from the true data distribution, we assign the label $y = 1$, and similarly for the generated sample we assign the label $y = 0$. We can now represent the samples as follows; for the true data, $p_{data}(\mathbf{X}) = p(\mathbf{X}|y = 1)$ and for the generated data, $q_\theta(\mathbf{X}) = p(\mathbf{X}|y = 0)$. Through some manipulation and applying Bayes' rules, the problem of density estimation is equivalent to that of class probability estimation [Mohamed and Lakshminarayanan, 2016]. Now all that remains is simply to choose an appropriate cost function to learn the parameters. For binary classification the sensible choice is the logarithmic loss function:

$$\mathcal{L} = \mathbb{E}_{p(\mathbf{X}|y)p(y)}\Big[- y \log D_\phi(\mathbf{X}) - (1 - y) \log(1 - D_\phi(\mathbf{X}))\Big]. \tag{2.24}$$

Since we know the process that was used to generate the samples, we can add this to the cost function in Eq. 2.24 to recover the cost function in Eq. 2.23. The power of this training protocol to approximate a bimodal data distribution is shown in Fig. 2.10. Here you can see a GAN learn a simple Gaussian distribution, at various stages of its training.

Booyse [2018] used GANs to propose a completely unsupervised PHM framework for machine diagnostics. In his method a GAN is trained on a set of baseline data. The baseline data could correspond to healthy data, however knowledge of the actual condition of the machine is not a requisite. Upon convergence, the trained network has learnt the distribution of this baseline data. The discriminator output was then successfully used as a FSI. Upon convergence, a GAN trained on the baseline data would result in a fault metric with an output of approximately 0.5. However, when presented with new samples, the fault metric would be 0.5 only when the signal corresponds to the baseline data. If the fault severity increased, the distribution of the input data would be different from the baseline data and the output of the discriminator decreases from 0.5 at the same rate. This was the first case of the application of using the discriminator of a GAN as a FSI.

2.6 Summary

The task of diagnosing a fault in a bearing can be considered as an inference type problem. This inference problem has been solved with classic data driven approaches, predominately based on sampling methods. In this chapter the concept of variational inference is summarised in the context of fault detection. With variational inference, inference can be solved through optimization by the introduction of a variational model. Models with greater complexity are encouraged with variational inference, as it improves approximation of the posterior distribution. Deep learning models offer a good balance between complexity and optimization, making them suitable for use in variational inference type cases. Some suitable neural network building blocks were introduced. To develop an unsupervised algorithm, an assumption of a data manifold was made as well as presenting measures about the dimensionality reduction capacity. Cases where deep learning has already been used in bearing diagnostics are summarised in Table 2.1. Evidence shows that deep learning is capable of extracting very good PHM features, without performing any feature engineering, but authors are still using these features in a supervised manner, with the exception of GANs, which are trained in an unsupervised manner on raw data. Since gathering labelled data for bearing FDD is especially difficult and feature engineering on that data often requires auxiliary signals in non-stationary operating conditions, there is capacity to explore deep learning in providing an unsupervised bearing FDD approach using raw vibration signals.

Table 2.1: Summary of machine learning based approaches towards FDD.

Approach	Paper	Supervised/Unsupervised	Feature Engineering
Neural Networks:	Janssens et al. [2016]	Supervised	Yes
	Guo et al. [2016]	Supervised	No
	Ince et al. [2016]	Supervised	No
	Jia et al. [2018b]	Supervised	No
	Zhang et al. [2018]	Supervised	No
	Ding and He [2017]	Supervised	Yes
VAE:	Sun et al. [2016]	Supervised	No
	Thirukovalluru et al. [2016]	Supervised	Yes
	Shao et al. [2017]	Supervised	No
	[Shao et al., 2018]	Supervised	No
	Jia et al. [2018a]	Supervised	No
	Lu et al. [2017]	Supervised	No
GAN:	Booyse [2018]	Unsupervised	No

Chapter 3

Proposed methodology

3.1 Introduction

Up to now, we have seen how the problem of detecting and diagnosing a fault in a bearing can be solved using an inference model. Again, the goal of the model is to estimate the joint distribution between the observed variables (vibration signals) and the unobserved variables (machine condition, operating conditions), $p(\mathbf{X}, \mathbf{Z})$. We have seen how this problem can be solved by a machine with gradient based optimization schemes. In this chapter we will consider the non-stationary characteristics of the signal from a machine operating with fluctuating speeds and loads, and develop a methodology that is able to take these characteristics into account. Furthermore, we will develop a full end-to-end algorithm that will do all the heavy lifting, producing a fault metric that can be used to diagnose the bearing's condition.

3.2 Latent space regularization

Armed with the knowledge of the structure of faulty bearing signal data, this section will cover how this knowledge can be used with deep learning to achieve the goal of fault detection and diagnosis. With VAEs and GANs, the latent space $\mathbf{Z}$ has been mapped randomly to the observed variable $\mathbf{X}$. There is no regulation on how this mapping takes place. Thus, the information contained in latent space of these models is not very useful for diagnostic purposes. We can use the knowledge of the structure of the vibration signal to regularize the latent space, and use it to our advantage. Doing this will allow the model to learn the deterministic and random components from an acceleration signal. To do so we first need to have an untangled latent space without any random mappings.

To get an estimation model for the posterior distribution from a GAN, we will need a two way mapping between the latent variables and the observed variables ($\mathbf{Z} \leftrightarrow \mathbf{X}$), as opposed to the original GAN's one way mapping, ($\mathbf{Z} \rightarrow \mathbf{X}$). This will allow us to perform inference using a GAN with an encoding network approximating the posterior distribution $p(\mathbf{Z}|\mathbf{X})$. Fortunately, two models have already been developed to allow for the two way mapping; Adversarially Learnt Inference (ALI) by Belghazi et al. [2018] and Bi-directional GANs (BiGANs) by Donahue et al. [2016]. However, these models still require some form of regularization as the mappings they learn are still random. The key here is to regularize with mutual information to improve the structure within the latent space and prevent tangled random mappings.

Adding mutual information as a cost term in the training objective will ensure that when a mapping between the observed space and the latent space is learnt, the mapping will remain uniquely identifiable. A consequence of this procedure is that the structure of the latent space is more consistent with the data manifold in question. An example of the latent space entanglement is given in the subsection 3.2.1.

Maximizing mutual information has proven a key aspect in representation learning. Adversarial Autoencoders (AAE) proposed by Makhzani et al. [2015] and infoGAN by Chen et al. [2016] are two models that incorporate mutual information into their objective function in an attempt to improve the representation learning capacity of the models on which they are based, ie: VAEs and GAN respectively. These two models represent alternative approaches in which mutual information can be used to regularize the structure of the latent space, with both models maximising mutual information in some form. The mutual information is maximized between the latent variable and the observed data distribution. Between the two models, the only difference is the lower bound that is used to approximate the mutual information. As a result an AAE is good at reconstruction, whilst an infoGAN performs better at classification Zhou et al. [2018]. The details of these two models will follow after a small example on latent space entanglement.

3.2.1 Example: Latent space entanglement

This example illustrates the mapping that occurs between the latent space of a model with and without the mutual information regularization. In the case without regularization, the mapping is completely random. In contrast, the regularized mapping has more structure. The training data for this example consists of the following: The observed variables are taken from a GMM with 5 mixture components as shown in Fig 3.1b. These will be mapped using ALI to a latent space consisting of a single Isotropic Gaussian distribution, shown in Fig. 3.1a. The same mapping will then be repeated, this time maximizing the mutual information between the latent space and the observed variables. The regularized model in this case is known as ALICE [Li et al., 2017], which is simply ALI with (C)onditional (E)ntropy (Mutual Information) added to its cost function.

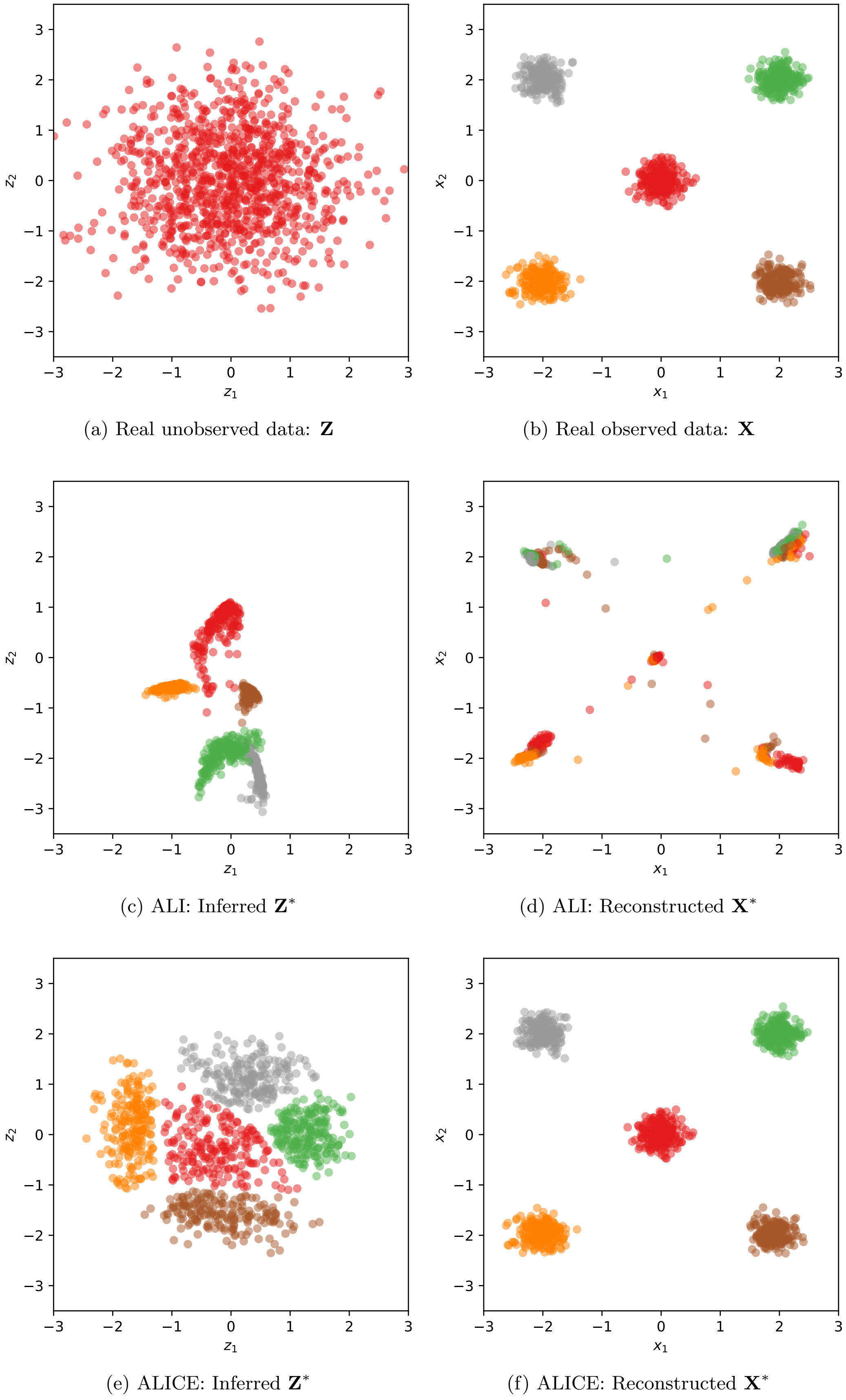
In Fig. 3.1c we have the inferred latent space ($\mathbf{X} \rightarrow \mathbf{Z}^*$) of the model trained without any regularization. We can see that although there is evidence of some clustering of the various mixture components, many of the mapped latent variables overlap. This makes inference using the latent variable impossible as 1 latent variable can be mapped to multiple classes/domains in the observed space. This is exactly what we see in Fig. 3.1d with the reconstructed observed data ($\mathbf{X} \rightarrow \mathbf{Z}^* \rightarrow \mathbf{X}^*$). Here, the reconstructed observed data retains the structure of the original 5 mixture components, however the mapping is completely random, and the data is not mapped back to their correct mixture components or classes.

When we regularize with mutual information, we can see immediately that the inferred latent space in Fig. 3.1e retains far more of the structure of the original latent space, with very few overlaps. Furthermore, in the latent space you can immediately and accurately infer from which mixture component the original observed data point came. When this inferred latent variable is mapped back to the observed space we can see, in Fig. 3.1e, that the mapping retains its original structure. Thus the mapping in the latent space remains unique and identifiable.

By simply maximising the mutual information between the two variable spaces, we can create more structure within the latent space allowing us to perform far more accurate inference, than a model with no structure.

3.2.2 AAE mutual information maximisation

Similar to a VAE, the AAE regularises the latent space by enforcing the latent space structure with an arbitrarily chosen prior distribution, $p(\mathbf{Z})$. This distribution is normally chosen as an Isotropic-Gaussian. The difference between a VAE and an AAE is that the former minimized the KL divergence between the prior latent distribution and the posterior distribution, parametrised by the encoder, whilst the latter does so with an adversarial loss. Here, the marginal density of the latent variable $q(\mathbf{Z})$, calculated as



(a) Real unobserved data: $\mathbf{Z}$

(b) Real observed data: $\mathbf{X}$

(c) ALI: Inferred $\mathbf{Z}^*$

(d) ALI: Reconstructed $\mathbf{X}^*$

(e) ALICE: Inferred $\mathbf{Z}^*$

(f) ALICE: Reconstructed $\mathbf{X}^*$

Figure 3.1: Effect mutual information maximization has on inference. The latent space is effectively regularized to retain more structure, thus improving inference capacity of the latent space. (Generated from code, Li et al. [2017])

$$q_\theta(\mathbf{Z}) = \int q_\theta(\mathbf{Z}|\mathbf{X}) p_{data}(\mathbf{X}) d\mathbf{X}. \tag{3.1}$$

is adversarially trained to match this arbitrary chosen prior distribution $p(\mathbf{Z})$, whilst still minimizing the reconstruction loss objective typical of an autoencoder. The effect is that the latent space is regularized onto the structure of the chosen prior.

The objective function for an AAE is similar to the VAE in Eq. 2.22, however the KL divergence is replaced by an adversarial loss, represented by the the Jensen-Shannon (JS) divergence in Eq. 3.2.

$$\mathcal{L}_{AAE} = JS[q_\theta(\mathbf{Z})||p(\mathbf{Z})] - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z})] \tag{3.2}$$

If we take a look at the conditional entropy between $\mathbf{X}$ and $\mathbf{Z}$,

$$\begin{aligned} H(\mathbf{X}|\mathbf{Z}) &= -\mathbb{E}_{p_{data}(\mathbf{X})}[\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\log q_\phi(\mathbf{X}|\mathbf{Z})]] \\ &= -\mathbb{E}_{p_{data}(\mathbf{X})}[\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z})]] - \mathbb{E}_{p_(\mathbf{Z})}[KL(q_\phi(\mathbf{X}|\mathbf{Z})||p_\theta(\mathbf{X}|\mathbf{Z}))], \\ &\leq -\mathbb{E}_{p_{data}(\mathbf{X})}[\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z})]] \end{aligned} \tag{3.3}$$

and compare the term the second factor to the last term in Eq. 3.2, it can be seen that these are exactly the same terms. The difference between the terms being that the loss function is averaged over the data: $\mathbb{E}_{p_{data}(\mathbf{X})}[\mathcal{L}_{AAE}]$. Thus, by minimizing the reconstruction loss objective of the autoencoder, the mutual information between $\mathbf{X}$ and $\mathbf{Z}$ is maximized.

3.2.3 InfoGAN mutual information maximisation

The infoGAN model is a variation of a GAN, in which a small modification is made to the objective function, L in Eq. 2.23 of the GAN model.

$$\min_\theta \max_\phi L_I\big(D_\phi(\mathbf{X}), G_\theta(\mathbf{Z})\big) = L\big(D_\phi(\mathbf{X}), G_\theta(\mathbf{Z})\big) - I(\mathbf{c}, G(\mathbf{c}, \mathbf{n})) \tag{3.4}$$

Here, mutual information, $I(\mathbf{c}, G(\mathbf{c}$, is optimized together with the original GAN objective function. The small but powerful modification, shown in Eq. 3.4 allows the algorithm to learn meaningful salient representations within the structure of the data, in a completely unsupervised manner. This is achieved by optimizing the mutual information between a small fixed subset of the latent variables, $\mathbf{c}$ and a sample generated from the same $\mathbf{c}$, $G(\mathbf{c}, \mathbf{n})$. Note that $\mathbf{Z} = \mathbf{c} \cup \mathbf{n}$.

The choice of distribution of $\mathbf{c}$ is arbitrary, with some common choices being a continuous or categorical distribution or both. The random and deterministic components of the observed vibration signals can be separated using the latent space. This is done by allowing the variable $\mathbf{n}$ to be mapped without structure, whilst simultaneously allowing the remaining variables, $\mathbf{c}$ to be mapped with structure using mutual information. Note that the mutual information is only maximized on the deterministic variable $\mathbf{c}$. The incompressible noise in the acceleration signal will be mapped in to the random variable $\mathbf{n}$, whilst any deterministic components within the acceleration signal will be mapped deterministically to $\mathbf{c}$. This will allow us to separate the random and deterministic components, which brings us to the proposed methodology.

3.3 Bearing FDD methodology

The methodology for bearing fault detection and diagnosis is highlighted in this section. All the details about the model will follow in the next section. The process of detecting and diagnosing the fault is split into two frameworks. The fault detection phase is used to trend the fault and subsequently make a conclusion about the severity of the fault. No labelled data is required for the fault detection phase. Before training, a baseline or reference subset of the total dataset is preallocated and used to train the model. Consequently, a sample presented for inference is assessed against this arbitrarily chosen baseline dataset. In the case of a new machine, any data that is currently available can be used as the baseline dataset. No additional historical data is required.

During the fault diagnosis phase, the type of fault is classified. In this phase, the clustering capacity of the model can be used to classify the type of fault present in the bearing. In this framework, the model can be adjusted to whatever historical data has been observed, both labelled or unlabelled. The model can leverage any labelled data that may be available. However, labelled data is not a prerequisite for functioning of the model.

3.3.1 Fault detection

The severity of the bearing fault can be tracked by using the discriminator for the random part of the latent variable, $D_n(\mathbf{n})$, where $\mathbf{n} \sim q_\phi(\mathbf{n}|\mathbf{X})$, is the encoded random latent variable of a query sample $\mathbf{X}$. During training, the discriminator, $D_n(\mathbf{n})$, learns a distance metric between samples from the random distribution, $\mathbf{n} \sim \mathcal{N}[0, 1]$ and samples corresponding to the encoded random latent variable $\mathbf{n}$. Hence, when trained on a baseline set of data, the $D_n(\mathbf{n})$ represents a distance metric of the random distribution of baseline dataset. Thus samples drawn from the baseline distribution and presented to the discriminator will have a low distance.

The latent variable $\mathbf{n}$, holds most of the random, incompressible information of the signal. Thus, when a signal which contains a characteristic bearing fault signature is encoded, the random components of the fault will cause the distribution of the random latent variable to shift away from the original baseline distribution with which the network was trained. The discriminator can then be used to quantify this shift.

Furthermore, the metric can be trended in time by presenting sequential examples of the acceleration signal, thus making a trained model suitable for on-line applications. A threshold based on the baseline data that was used to train the network can be used for failure detection. The threshold value can be calculated by obtaining the expected value of the discriminator for all the samples from the baseline set, and an offset added depending on the criticality of the asset. In our study, thresholds as published by other authors are used.

Figure 3.2 shows a schematic of the fault detection framework. Note that no labels of the fault are necessary for this phase of the diagnosis. Because the choice of baseline data is arbitrary, there is no need to use labelled data. Inference can be made with respect to the current condition of the asset.

3.3.2 Fault diagnosis

For the diagnosis of the fault, a slightly different approach is used. The regularization and training of the network ensure samples of similar nature are clustered together within the latent space. This clustering ability can be utilized to diagnose the type of fault in the signal. By comparing the latent representation of a query sample with that of a representation of a known fault mode, a classification can be made. Even in cases where no labels are available, the algorithm still clusters the data based on its similarity in structure. Once a fault has been verified, a label can be associated with the cluster, and consequently used to classify

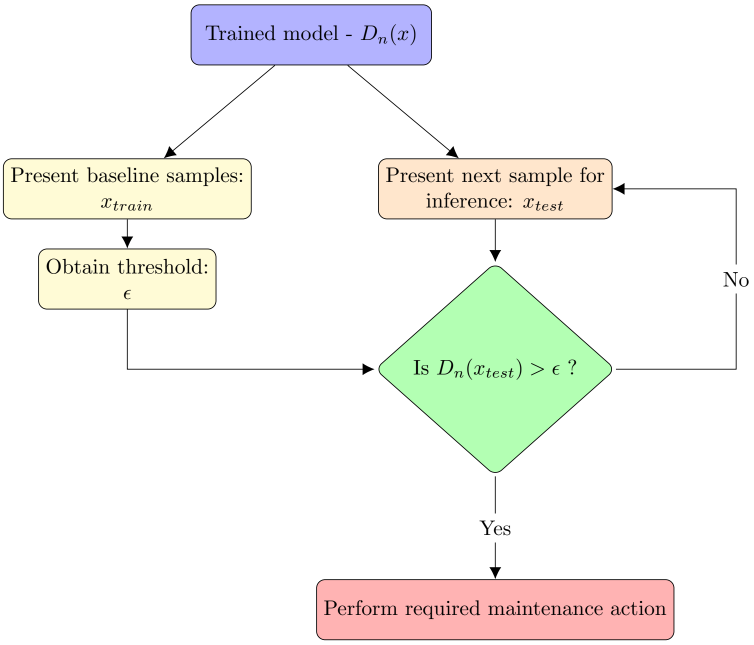

Trained model - $D_n(x)$
Present baseline samples: x_{train}
Obtain threshold: ϵ
Present next sample for inference: x_{test}
Is $D_n(x_{test}) > \epsilon$?
No
Yes
Perform required maintenance action

Figure 3.2: Schematic outline of the fault detection phase of the proposed methodology.

future unlabelled samples.

However, because of the ill-defined nature of the problem, the algorithm will initially struggle to allocate enough space within the latent space to accurately define each of the fault modes and hence, the full data manifold of the asset. This can be overcome if a very small set of labelled samples are available for a semi-supervised setting. These samples can then be used as anchor points in the latent space. The anchor points give the algorithm a good foundation on which to represent the remaining unlabelled set of samples. In cases where the distribution between healthy and unhealthy labels are unbalanced, as is often found in PHM data, the labels that are available may significantly improve the classification accuracy.

A simple numerical problem is used to illustrate this concept. A phenomenological model proposed by D'Elia et al. [2018] was used to generate bearing signals of varying fault severity. A small network is trained with a deterministic dimension of $N_c = 2$ to allow for visualization. In reality this dimension is far too small to encode any discriminative features, nevertheless it is still a good illustrative example. In the first case, no anchor points are used, and a latent representation is learnt. In the second case one sample of a healthy signal, together with one sample from each of the fault modes at maximum severity were used as anchor points. The algorithm automatically aligns unlabelled samples to the anchor points based on their similarity. Furthermore, samples are aligned in increasing order of severity from the baseline (or healthy) anchor point to the anchor point of respective maximum fault severity.

3.4 Bearing FDD model

The proposed model is based on the work done by Zhou et al. [2018] and their proposed Representation GAN (REPGAN). REPGAN is hybrid model of infoGAN and AAE. The full model is comprised of multiple components or sub-networks together with a training protocol that uses the advantages of both the infoGAN and the AAE, to facilitate latent space regularization and increases classification ability. The sub-network components are: an encoder network $Enc(\mathbf{X}) \sim q_\phi(\mathbf{Z}|\mathbf{X})$, a decoder network $Dec(\mathbf{Z}) \sim p_\theta(\mathbf{X}|\mathbf{Z})$, a discriminator network for the signal samples $D_\omega(\mathbf{X})$, and lastly a separate discriminator network for each of the latent variables, $D_{\eta_Z}(\mathbf{Z})$, where $\mathbf{Z} = \big\{\mathbf{c}, \mathbf{s} \text{ or } \mathbf{n}\big\} \in \mathbb{R}$. Here, $\mathbf{c}$ refers to a categorical

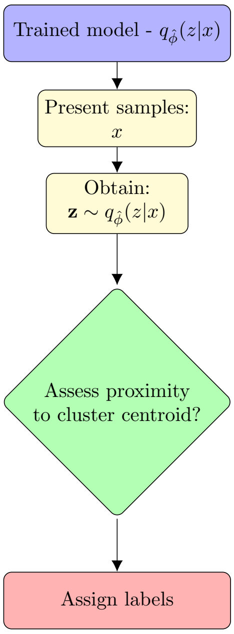

Trained model - $q_{\hat{\phi}}(z|x)$
Present samples: x
Obtain: $\mathbf{z} \sim q_{\hat{\phi}}(z|x)$
Assess proximity to cluster centroid?
Assign labels

Figure 3.3: Schematic outline of the fault diagnosis phase of the proposed methodology.

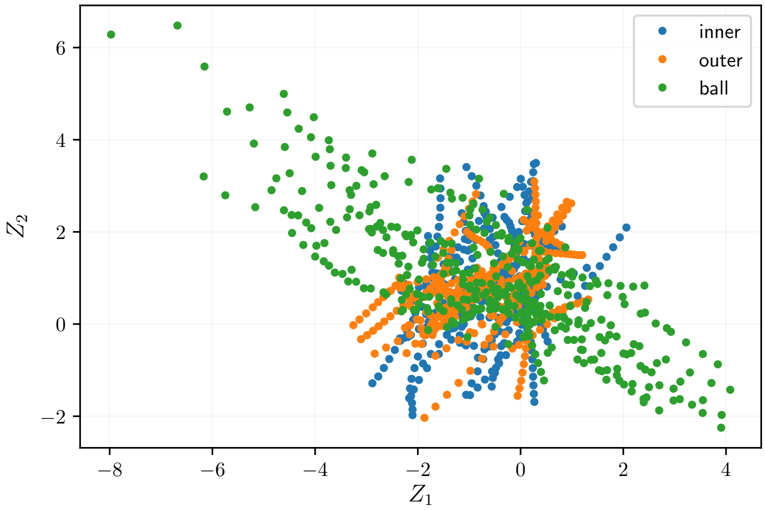


Figure 3.4: Latent variable without anchored points.

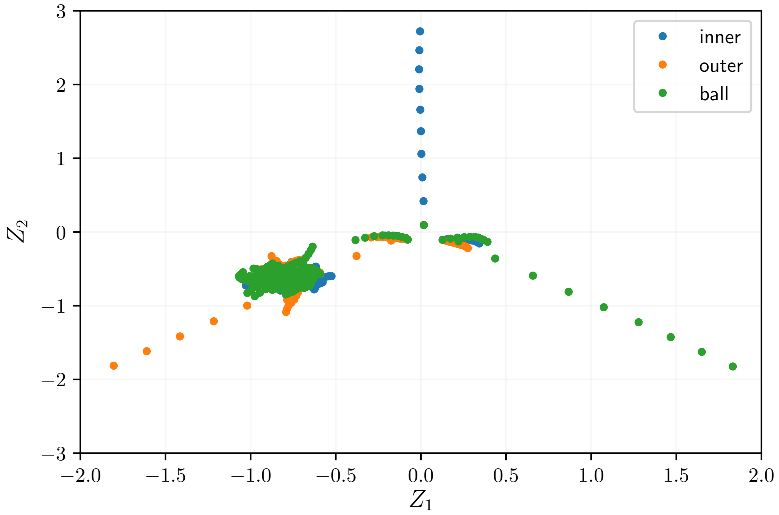


Figure 3.5: Latent variable with anchor points.

distribution, whilst $\mathbf{s}$ refers to a continuous distribution and $\mathbf{n}$ refers to Gaussian noise. It is this discriminator network that will eventually become the fault severity indicator. The model trainable parameters are represented collectively by the variables $\phi, \theta, \omega,$ and η for the encoder, decoder, X-discriminator and Z-discriminator respectively. Training is achieved by alternating between an infoGAN configuration and an AAE configuration.

3.4.1 Model architecture

Figure 3.6 shows how the components form an AAE configuration. Here, the model encodes a sample of the signal down to the latent variable space and then back to the signal space, $\mathbf{X} \rightarrow \mathbf{Z} \rightarrow \tilde{\mathbf{X}}$. With this configuration, the model maps multiple points in the latent space to a single point in the data space. This ensures a good reconstruction ability of the model.

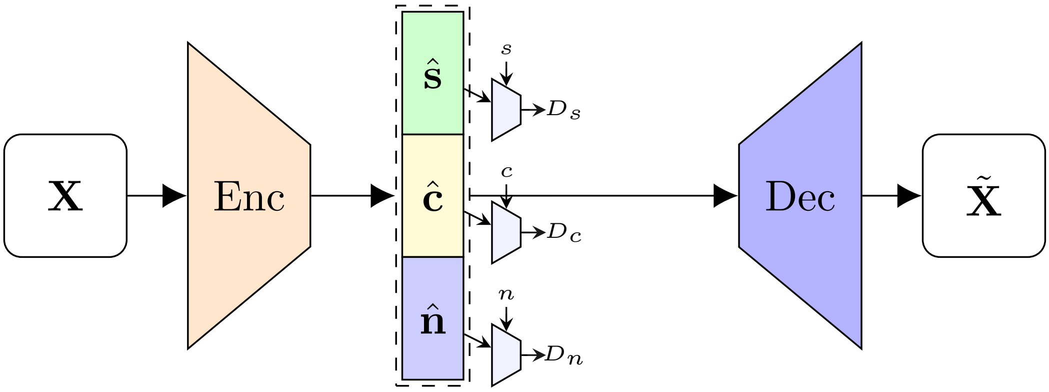


Figure 3.6: AAE configuration of network components.

Similarly, Fig. 3.7 shows how the same components can be re-ordered to make the infoGAN configuration. With the infoGAN configuration, the opposite mapping is learnt, $\mathbf{Z} \rightarrow \mathbf{X} \rightarrow \tilde{\mathbf{Z}}$. Again, mutual information is maximized by minimizing the conditional entropy between the generated signal $\hat{\mathbf{X}}$ and the deterministic latent variables: $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{c}}$. This is a crucial aspect of this model. More precisely, from Fig. 3.7, it can be seen that when the network is trained in an infoGAN configuration, only the deterministic variables are used for the maximization of the mutual information. This ensures that all the deterministic components of the signal are enforced and hence, mapped into these two variables. The encoder and decoder in this configuration, maps multiple points from the data space to a single point in the latent space **Z**, and ensures good classification accuracy of the model. The full details of each of the sub-network components can be found in the Appendix in Section A.1.

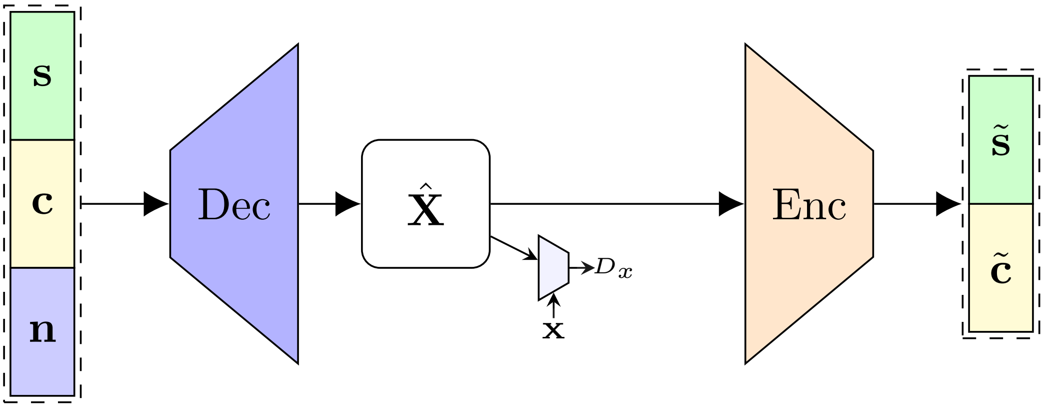


Figure 3.7: InfoGAN configuration of network components.

3.4.2 Training protocol

The objective functions that were used to train the various components are introduced next. It must be noted that, although the architecture can be assembled into the AAE or the infoGAN configurations, the network components (Encoder, Decoder, Discriminators) that make up those configurations are shared between the configurations and have the same trainable parameters. A summary of all the network components and their respective parameters are shown in Table 3.1. The iterations of each training protocol were taken from the original publication [Zhou et al., 2018].

Table 3.1: Summary of model network components and their respective trainable parameters.

Name	Symbol	Parameters
Encoder	$q_\phi(\mathbf{Z}\|\mathbf{X})$	ϕ
Decoder	$p_\theta(\mathbf{X}\|\mathbf{Z})$	θ
Signal discriminator	$D_x(\mathbf{X}, \eta)$	η
Latent variable discriminator	$D_z(\mathbf{Z}, \omega_z)$	ω_z
Signal distribution	$p(\mathbf{X})$	–
Latent variable distribution	$q(\mathbf{Z})$	–

AAE-training objectives

During the AAE configuration the objective functions for the network parameters are as follows:

The first objective is that of the adversarial loss function: Here a Wasserstein GAN (Arjovsky et al. [2017]) is used as the adversarial trainer, which corresponds to one part of the objective function for the AAE. The adversarial loss for the discriminator and the generator (ie: Encoder) is given by

$$\begin{aligned}\mathcal{L}_{Dz} &= \min_{\eta} \mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}[D_z(q_\phi(\mathbf{Z}|\mathbf{X}),\eta)] - \mathbb{E}_{\mathbf{Z}\sim p(\mathbf{Z})}[D_z(\mathbf{Z},\eta)], \quad (3.5)\\ \mathcal{L}_{q_\phi} &= \min_{\phi} -\mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}[D_z(q_\phi(\mathbf{Z}|\mathbf{X}),\eta)] \quad (3.6)\end{aligned}$$

The optimizer used for these loss functions is RMSProp and has a learning rate of 5×10^{-5}. The second objective for the AAE configuration is the typical reconstruction based loss function. Here the loss function is taken simply as a $\ell2$-norm reconstruction loss between the real signal, $\mathbf{X}$, and its reconstruction, $\tilde{\mathbf{X}}$:

$$\mathcal{L}_{Rec_x} = \min_{\phi,\theta} \sum_x \frac{1}{2}||\mathbf{X} - \tilde{\mathbf{X}}||_2^2. \quad (3.7)$$

For the reconstruction the optimizer used is also RMSProp with a learning rate of 1×10^{-3}. Note, the $\ell2$-norm of a vibration signal, may cause blurring when used as a reconstruction based loss function. The work done by Georgiou [2007], provides a thorough breakdown of this issue and proposes a suitable alternative loss function, especially for time based signals such as vibration signals. However, implementing such a loss function was beyond the scope of this work.

InfoGAN-training objectives

The first objective function for the infoGAN configuration is also an adversarial type loss function. Here the objective for the discriminator and the generator (ie: decoder) is given by [Zhou et al., 2018]

$$\begin{aligned}\mathcal{L}_{D_x} &= \min_{\omega} -\mathbb{E}_{\mathbf{X}\sim p(\mathbf{X})}[\log D_x(\mathbf{X},\omega)] - \mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})}[1 - \log D_x(p_\theta(\mathbf{X}|\mathbf{Z}),\omega)], \quad (3.8)\\ \mathcal{L}_{p_\theta} &= \min_{\theta} -\mathbb{E}_{\mathbf{Z}\sim q(\mathbf{Z})} \log \frac{D_x(p_\theta(\mathbf{X}|\mathbf{Z}),\omega)}{1 - D_x(p_\theta(\mathbf{X}|\mathbf{Z}),\omega)}. \quad (3.9)\end{aligned}$$

Note that for the generator loss term, the KL-loss proposed by Salimans et al. [2016] was used to prevent flat gradients from occurring during optimization. The optimizer used for the adversarial loss functions was ADAM with a learning rate of 2×10^{-5}. The objective function for the latent variables however depends on the type of distribution the variable was sampled from. For the categorical variable, $\mathbf{c}$, the loss function is taken simply as the cross-entropy loss [Zhou et al., 2018],

$$\mathcal{L}_{Rec_c} = \min_{\phi,\theta} - \sum_c q(\mathbf{c})[\log q_\phi(\tilde{\mathbf{c}}|(p_\theta(\mathbf{X}|\mathbf{c})))], \quad (3.10)$$

For the continuous latent variable $\mathbf{s}$, the reconstruction loss function is taken as the negative log-likelihood for a Gaussian. Similar to the VAE, a re-parametrization trick is used to allow for gradient back propagation. The loss function is as follows [Zhou et al., 2018]

$$\mathcal{L}_{Rec_s} = \min_{\phi,\theta} - \sum \frac{1}{2}\log(2\pi) - \log(\tilde{\mathbf{s}}_\sigma) - \frac{1}{2}\left(\frac{\tilde{\mathbf{s}}_\mu - \mathbf{s}}{\tilde{\mathbf{s}}_\sigma}\right) \quad (3.11)$$

For both the loss functions, Eqs. 3.10 and 3.11, ADAM was used as the optimizer with a learning rate of 2×10^{-5}.

3.5 Summary

In this chapter the proposed methodology was presented. The algorithm is based on a hybrid model between two deep learning models, namely the adversarial autoencoder (AAE) and infoGAN. Both these models regularize the latent space by maximizing the mutual information. However the outcome of each model emphasizes two different aspects of deep learning respectively, classification and reconstruction. A good balance between classification and reconstruction can be obtained by using a hybrid between the two models. Consequently, the latent space is regularized forming unique and identifiable mappings. This effectively untangles the latent space and allows for more informative inference to be done using the latent space. Specifically, this allows the user to target random and deterministic components of the data, which is very useful for fault detection in bearings. By separating the latent space into random and deterministic variables, we can effectively isolate each component independently. By focusing only on the random latent variables, we are able to infer the severity and the fault types in the bearing.

Chapter 4

Experimental Investigation

4.1 Introduction

In this chapter, the proposed bearing fault detection and diagnosis algorithm was tested using benchmark datasets. Three investigations were performed, each showcasing the capacity of the proposed model. The first dataset, Intelligent Maintenance Systems (IMS) benchmark, is used to test the effect the signal-to-noise ratio has on the fault metric. This dataset represents stationary operating conditions, however the transmission path of the signal to the accelerometer is non-stationary. The second dataset is used to investigate the fault diagnosis capacity of the model. In this experiment, it is shown that using sparsely labelled data in a semi-supervised setting, can significantly improve the untangling of the signal representation space, and consequently fault classification. Finally, the fault methodology is tested on real world wind turbine data. In this experiment the proposed metric is compared to a model in which the non-stationary components are not taken into consideration. The model is able to learn deterministic and random components using the proposed method, thus negating the need for order tracking or time consuming advanced signal processing. The proposed metric is tested against two state of the art bearing diagnostic approaches: a minimum entropy deconvolution (MED) followed by spectral kurtosis (SK) based normalised envelope spectrum approach (MED-SK-NES) and a cyclostationary based improved envelope spectrum (IES) approach. These two baseline models are considered efficient and effective procedures for vibration-based machine diagnostics [Abboud et al., 2019].

4.2 IMS benchmark dataset

This IMS dataset, first introduced by Qiu et al. [2006], contains the acceleration response of four separate accelerometers placed on the housing of four Rexnord ZA-2115 double row bearings that are installed on a single shaft. The shaft was run by an AC motor and kept at a constant speed of 2000 RPM with a constant radial load of $2721,55$ kg. The bearings have a basic dynamic load rating of $13104,28$ kg. A photo and a schematic diagram of the bearing test rig is shown in Fig. 4.1. The dataset contains four run-to-failure tests. This study focuses only on the first two test cases (set no. 1 and set no. 2).

Upon completion of the test-to-failure experiment one (set no. 1), an inner race fault was discovered on bearing 3 and a mixed fault on the roller element and outer race was discovered on bearing 4. The data shows that the surface defect that was found on bearing 4 had self healed [Qiu et al., 2006]. The spalling that was formed in the early stages of testing, was later smoothed over by continuous rolling action. Similarly, an inspection at the end of the test-to-failure experiment two (set no. 2) had revealed that an outer race fault had occurred on bearing 1.

Training was performed with parameters as described in Section 3.4. The results of the proposed methodology are compared against various bearing diagnostic approaches for benchmarking purposes. RMS and kurtosis values are considered as traditional approaches, wile following a similar exposition by Abboud et al. [2019], two methods namely, MED-SK-NES and IES are considered as SOTA. Abboud et al. [2019] use an arbitrary chosen threshold value for fault detection calculated as $\mu + 6\sigma$ of the initial baseline records. The same threshold value is adopted for comparative purposes in this study. Lastly, to evaluate the performance of the fault metric, the receiver operating characteristics (ROC) curve is plotted against the advanced benchmarks.

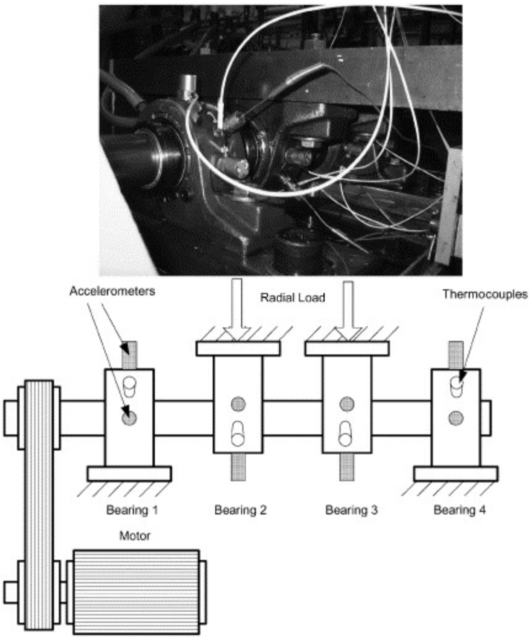


Figure 4.1: Bearing test rig for IMS dataset [Qiu et al., 2006].

4.2.1 Test-to-failure set no. 1

For the first dataset two faults were recorded in the original paper, namely, an inner race fault on bearing 3 and a mixed rolling element fault and outer race fault on bearing 4. The dataset exhibited a self-healing phenomenon of an initial defect.

Approach

The full dataset consists of the acceleration response from 8 channels, two channels for each bearing, corresponding to the x- and y- directions respectively. Only the acceleration response in the y direction for bearing 3 and 4 were used for training purposes. This follows the approach of Qiu et al. [2006], where they only analysed bearings 3 and 4 in their study. Each channel consists of 2156 records of data with each record containing 20480 data points. This equates to a data sampling rate of 20 kHz. A network was trained for each accelerometer on the first 172 records (8% of the full data), which were treated as the baseline level. The network trained on this baseline was used for inference on the remaining set of records as described in section 3.3.

Results

The signals of set no. 1 contain two signatures of interest, specifically the impulse responses due to the inner race fault in bearing 3 and the outer race and rolling element fault in bearing 4. Figure 4.2 shows the results of all the FSI applied to the signal of bearing 3. The RMS in Fig. 4.2a shows a dramatic increase in the power of the signal after record 2000. The kurtosis in Fig. 4.2b shows an increase in the signal's peakedness after record 1800. We can see in Figs. 4.2c and 4.2d that the inner race fault energy increases after record 1800 for the MED-SK-NES and IES based approaches respectively. The proposed fault metric crosses the threshold value at record 2123, as shown in Fig. 4.2e. The variance of the baseline records, with which the threshold was calculated, is high. By ignoring the initial 200 records, we see the fault metric stabilize until record 1500, after which it increases. Only a slight increase is observed for IES in Fig. 4.2d. This suggests that the fault can be detected as early as record 1500. The ROC curve in Fig. 4.2f is based on the fault being detected at record 1500 and indicates that the proposed metric outperforms the two baseline methods employed in this instance.

The FSIs calculated on the signal obtained from bearing 4 are shown in Fig. 4.3. The proposed fault metric, crosses the threshold at record 1474 in Fig. 4.3e. In the same figure, the self-healing phenomenon is seen in the fault metric between records 180 and 950. The self-healing phenomenon is not evident for RMS in Fig. 4.3a, and only marginally detected by the kurtosis in Fig. 4.3b. In Fig. 4.2d it can be seen that the self-healing phenomenon is more apparent using the IES approach than the MED-SK-NES approach in Fig. 4.3c. Both these approaches show an increase in the inner race energy after record 1600. The ROC curve of Fig. 4.3f uses record 1400 as the point of fault detection based on the proposed FSI crossing the threshold. In the figure, the proposed fault metric can be seen outperforming the two baseline methods. In this instance, a metric that is able to detect the fault earlier is favoured as the type of fault that is present in this case leads to very quick catastrophic failure once crack propagation is initiated [Qiu et al., 2006].

Discussion

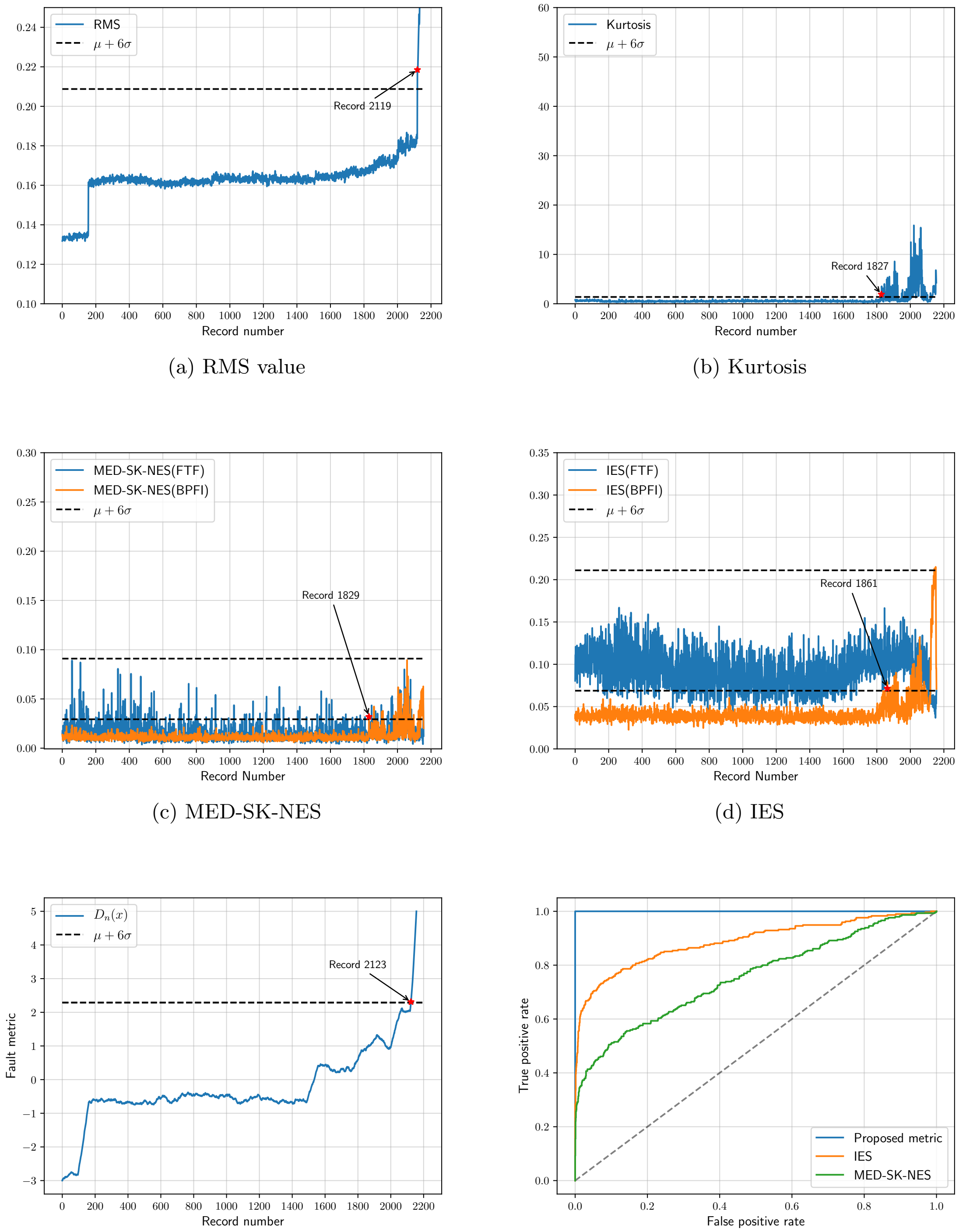
In this first experiment, the proposed fault metric is seen to outperform two SOTA bearing diagnostic techniques, namely IES and MED-SK-NES. With the advanced bearing diagnostic techniques, knowledge of the type of fault, is needed in order to track the correct fault frequency however, this provided information towards fault diagnosis. It is more difficult to do this when two or more faults are present in the signal. The proposed fault metric was most sensitive to the self healing phenomenon that occurred. In this case, the sensitivity may lead to early fault detection which are crucial to prevent catastrophic failures. We see that even with as little as 8% of the available data, the proposed metric is able to outperform the baseline approaches in detecting the fault early.

4.2.2 Test-to-failure set no. 2

In the second test case, a single outer race fault was discovered in bearing 1 upon completion of the test. The training approach and results were as follows.

Approach

The second set comprises the acceleration response for four bearings. This time using only one channel for each bearing. In this set, 984 records of 20480 samples were recorded, again giving a sampling rate of 20 kHz. In a similar manner to the previous set, the first 78 records (8% of the full data) of the dataset are used as a baseline condition and used to calculate the threshold of the FSI. In this instance, the data of the first 3 bearings were analysed following the same approach as Abboud et al. [2019], as these bearings produced the most noteworthy results. Therefore, 3 independent networks were trained as set out in sections 3.3 and 3.4 respectively.



(a) RMS value

(b) Kurtosis

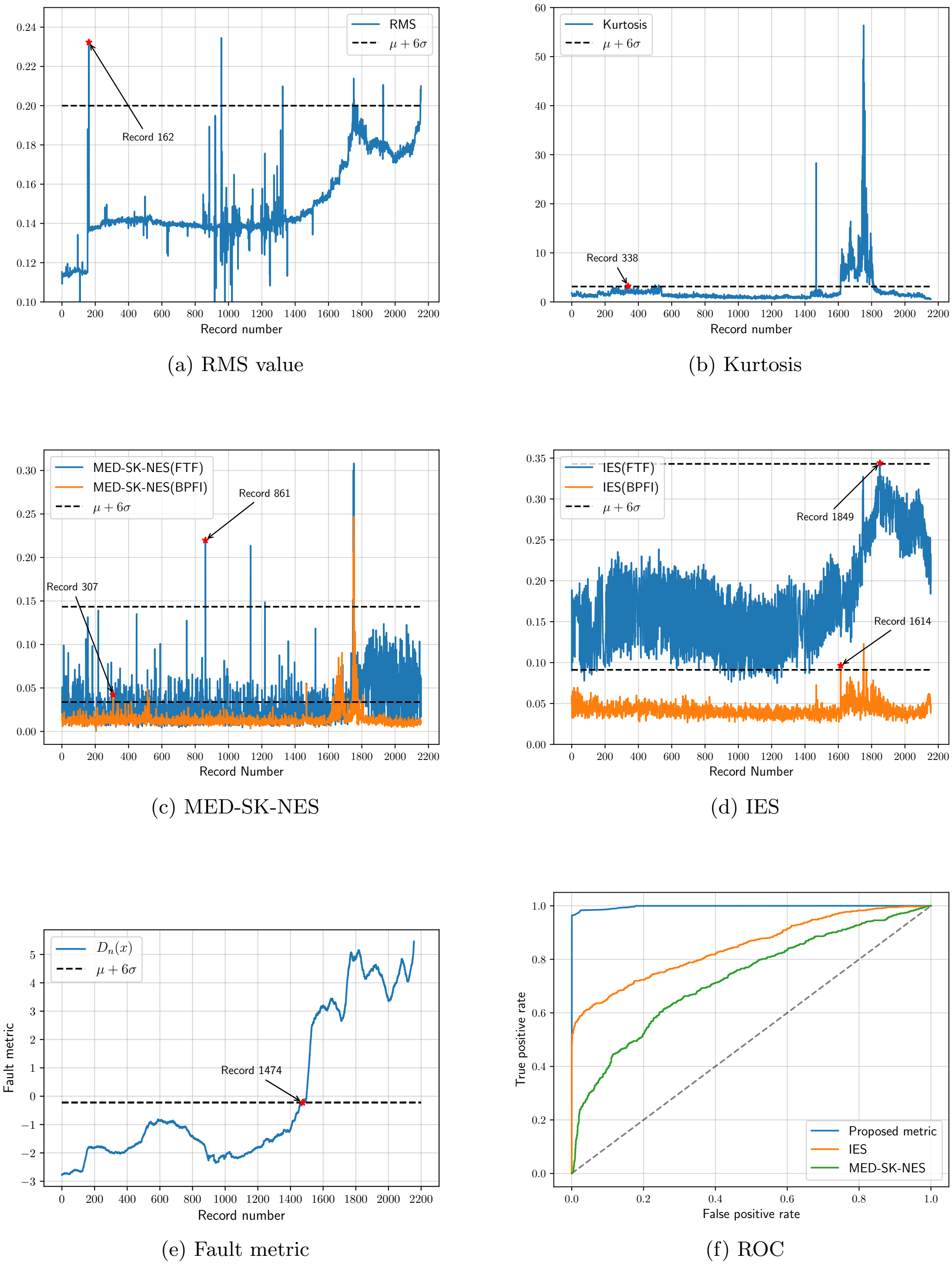
(c) MED-SK-NES

(d) IES

(e) Fault metric

(f) ROC

Figure 4.2: FSI applied on signal of bearing 3 (accelerometer channel 5) of IMS set No. 1.



(a) RMS value

(b) Kurtosis

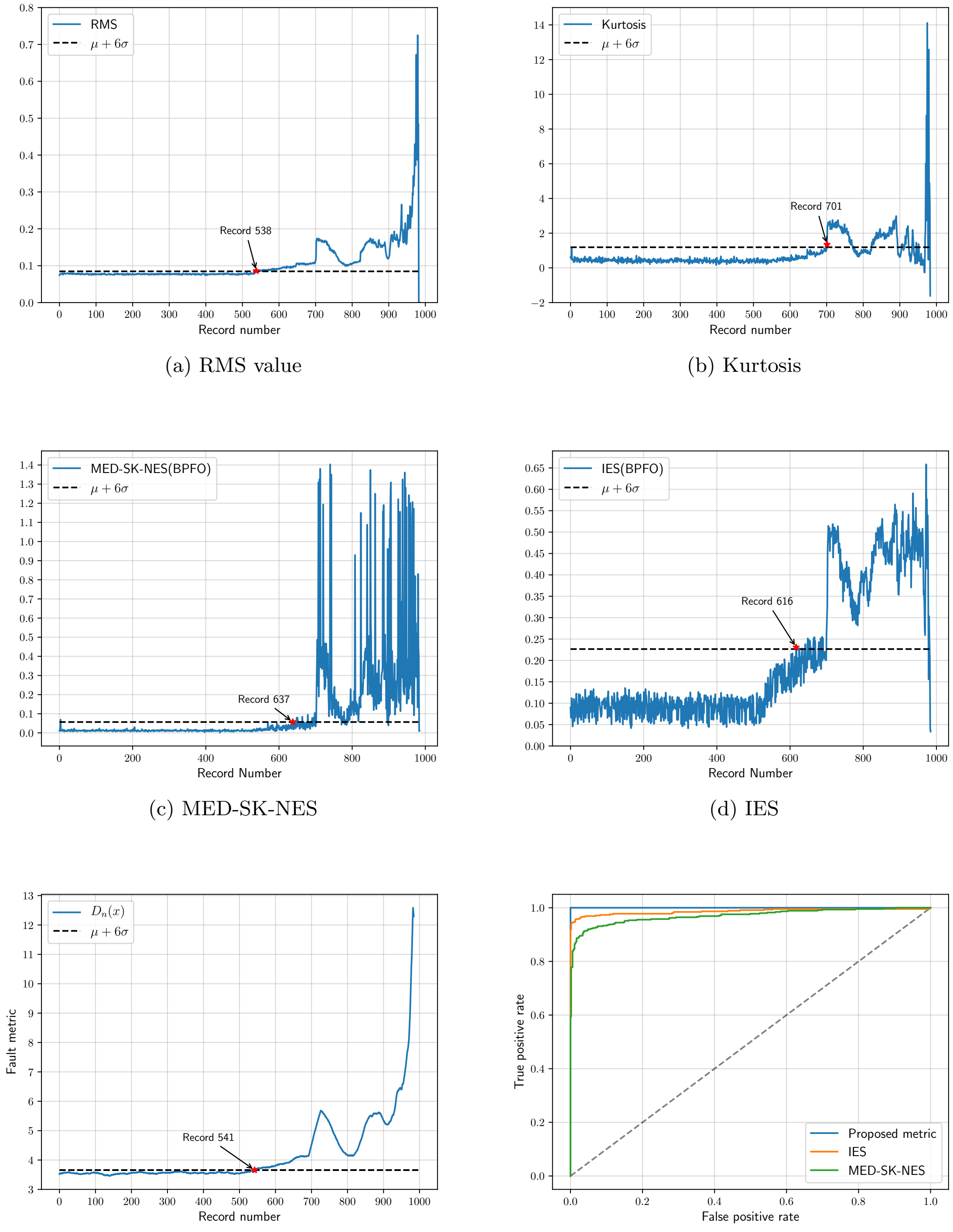
(c) MED-SK-NES

(d) IES

(e) Fault metric

(f) ROC

Figure 4.3: FSI applied on signal of bearing 4 (accelerometer channel 7) of IMS set No. 1.



(a) RMS value

(b) Kurtosis

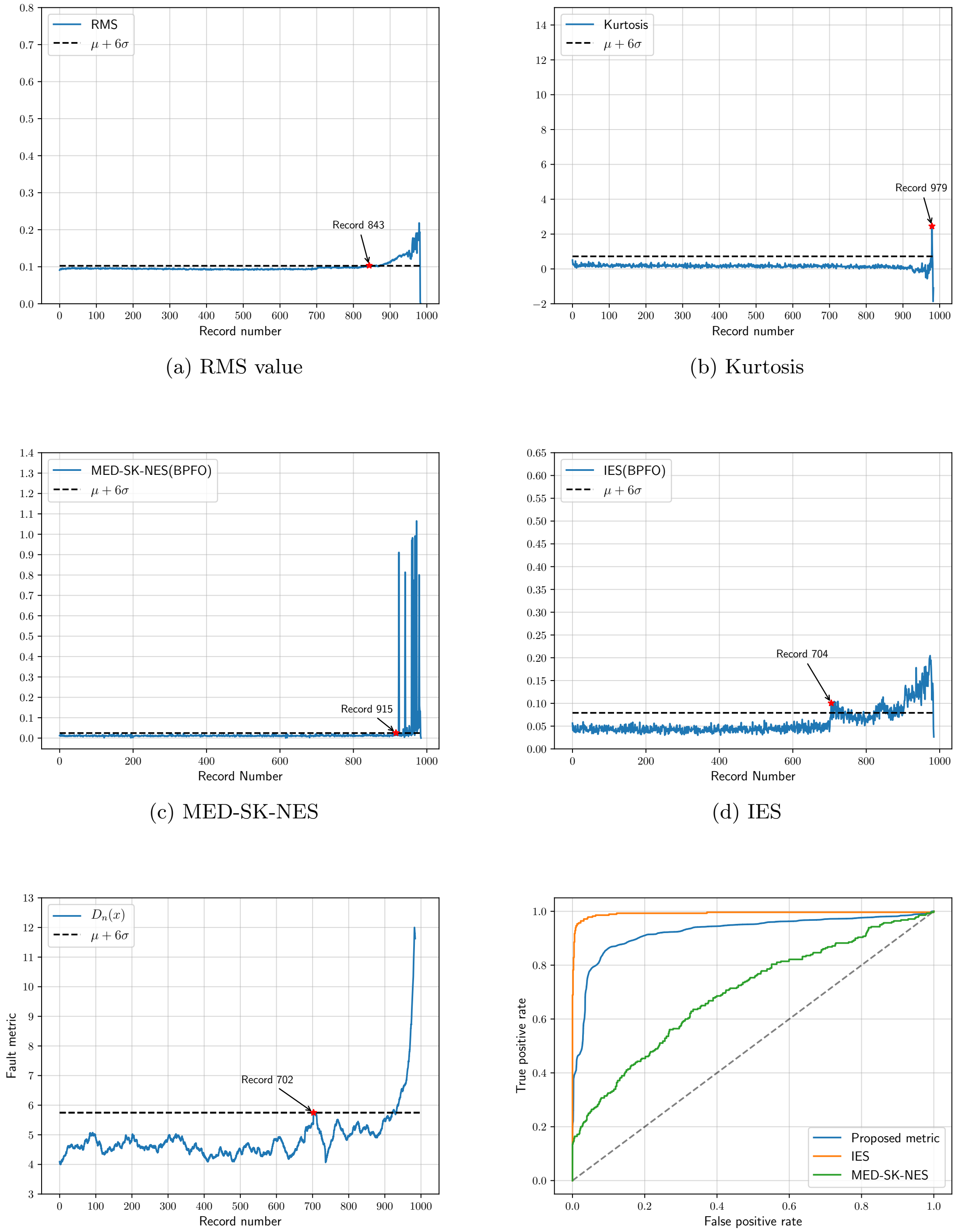
(c) MED-SK-NES

(d) IES

(e) Fault metric

(f) ROC

Figure 4.4: FSI applied on signal of bearing 1 of IMS set No. 2.



(a) RMS value

(b) Kurtosis

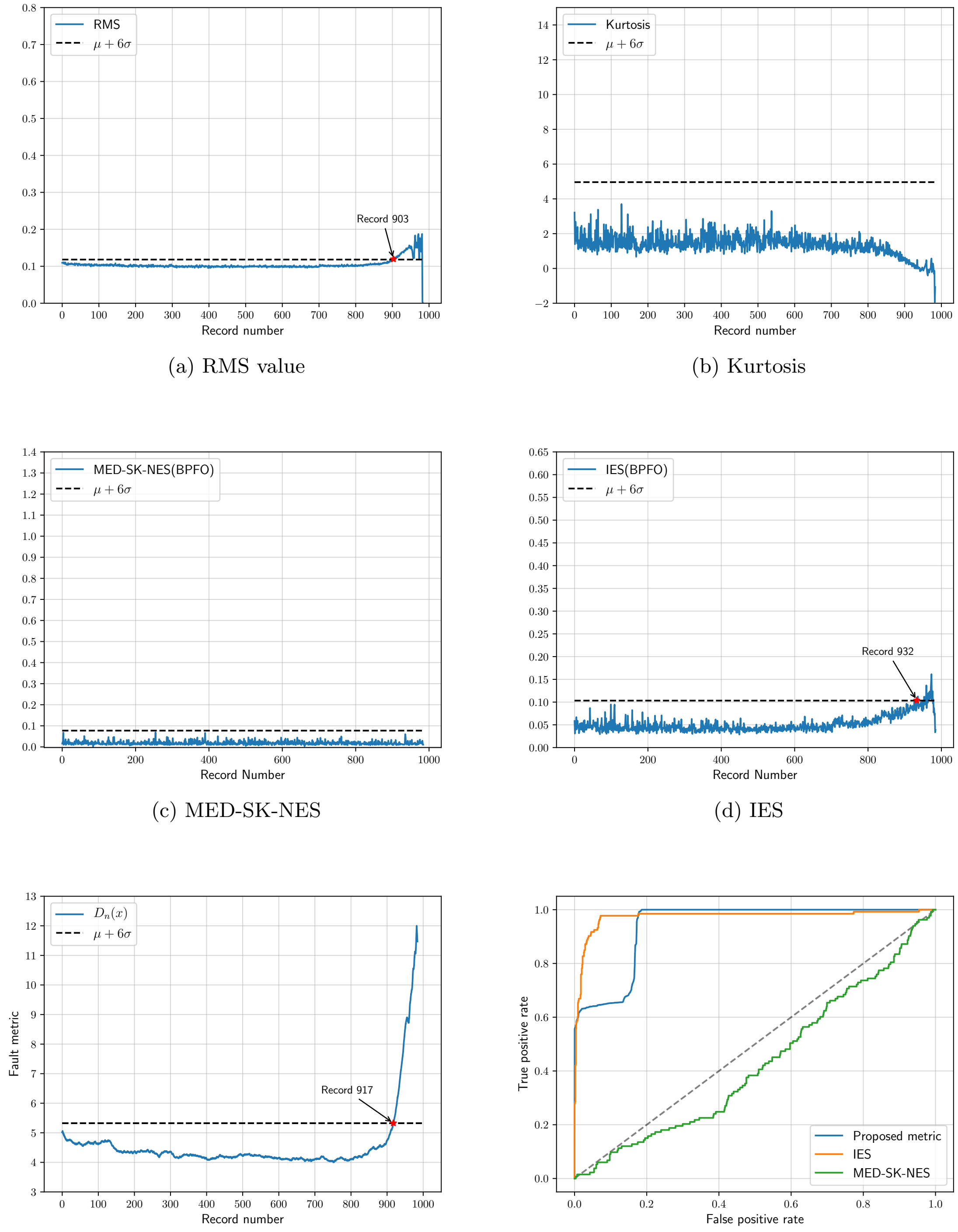
(c) MED-SK-NES

(d) IES

(e) Fault metric

(f) ROC

Figure 4.5: FSI applied on signal of bearing 2 of IMS set No. 2.



(a) RMS value

(b) Kurtosis

(c) MED-SK-NES

(d) IES

(e) Fault metric

(f) ROC

Figure 4.6: FSI applied on signal of bearing 3 of IMS set No. 2.

Results

The signal from the accelerometer placed on bearing 1 corresponds to the case with the highest as the measurements are taken on the bearing with the outer race fault. . Looking at the RMS of bearing number 1, in Fig. 4.4a, we can see that the fault presented itself at record 542. The kurtosis increases around record 600 in Fig. 4.4b. The two traditional approaches have no trouble detecting the faults due to the high SNR. The advanced bearing approaches perform just as well by detecting the fault at record 542 for MED-SK-NES in Fig. 4.4c and 533 for IES approach in Fig. 4.4d. In Fig. 4.4e it can be seen that the fault metric crosses the threshold at record 541 and follows a trend similar to the SOTA bearing diagnostic techniques. Figure 4.4f shows that the fault metric performs just as well as the baseline conditions when record 533 is taken as the fault point.

In Fig. 4.6a we can see that again the RMS approach detects the fault after record 900. The kurtosis of the signal in Fig. 4.5b does not offer any fault detection capacity for this accelerometer signal, corresponding to a remote sensor. We can see similarly, the MED-SK-NES based approach also struggles to detect the fault in Fig. 4.6c. The IES approach performs well for this case by detecting the fault in record 851, as can be seen in Fig. 4.6d. In Fig. 4.6e, the fault metric can be seen crossing the threshold value at record 917. However, looking at the ROC curve of Fig. 4.6f, the IES approach performs slightly better than the proposed approach, and significantly outperforms the MED-SK-NES based approach.

Lastly, in Figs. 4.5a and 4.5b, we see the RMS and kurtosis values of the signals increasing after record 900. These results are expected, as the SNR of this case is lower since the fault signal of interest has to travel a longer transmission path to get to the sensor. The advanced approaches perform significantly better in Figs. 4.5c and 4.5d. The IES approach does the best out of all the benchmark approaches, detecting the fault at record 700. The proposed fault metric in Fig. 4.5e is seen to cross the threshold at record 702. The ROC curve in Fig. 4.5f is based on the fault being detected at record 700 and shows the IES approach performing slightly better than the proposed metric, which in turn performs significantly better than the MED-SK-NES approach.

Discussion

In test set no. 2, we have a similar situation to the first where signals of varying SNR are tested by using sensors which get further away from the only fault, namely an outer race fault at bearing 1. In this case, the SNR is decreasing with each subsequent signal presented. The first signal corresponds to the case with the highest SNR measured at bearing 1. It was shown that the proposed metric is able to trend the fault just as well as the benchmark approaches. As the SNR of the signal decreases, the proposed metric outperforms the MED-SK-NES approach. It was also seen that the proposed metric performs only slightly worse than the IES approach.

4.3 Bearing fault dataset

The second dataset was the fault dataset released by the Society for Machinery Failure Prevention Technology (MFPT) with an accompanying tutorial in bearing envelope analysis [Bechhoefer]. The intended goal of the dataset was to facilitate research into bearing analysis. This dataset is comprised of acceleration signals recorded from a bearing test rig that was equipped with a NICE bearing with 8 rolling elements, with varying operating conditions. The first case is presented as the baseline condition, and corresponds to signals that were obtained from a healthy bearing (no fault) at a shaft rotational speed of 25 Hz and a load of 122.47 kg. The second and third cases correspond to the signals obtained from a bearing with an outer and inner race fault respectively. The signals obtained for the fault datasets were sampled at varying loads between 0 kg and 136.08 kg at a constant shaft rotational speed of 25 Hz. The goal of using this dataset is to test the algorithm with signals of multiple known fault modes, and thus provide insights into the clustering abilities of the proposed model. This dataset is used as a proxy for test-to-failure datasets with multiple fault modes in non-stationary operating conditions of speed and loads.

4.3.1 Approach

The architecture used for this experiment was adjusted slightly for visualization purposes. In this case, only two continuous dimensions were used in the encoder layer, in contrast to the proposed categorical, continuous and noise dimensions. This is to illustrate the clustering power of the proposed method using signal data. Hence, for this experiment the dimension of the output of the encoder was 2. The mutual information between these two latent dimensions and the signal was maximised during training. Two approaches were used for training the network. In the first case, the network was trained using all the data in a completely unsupervised manner. In the second case, anchor points were used to associate areas in the latent space with signals from known fault modes. In this case, only one of each fault mode were used as the anchor point, resulting in a semi-supervised setting.

4.3.2 Results

By setting the latent variables to only 2 continuous variables, the algorithm learns a projection of the data manifold on a 2D continuous sphere. Figure 4.7 shows the two latent dimensions, namely $N1$ and $N2$. This case shows how the algorithm can cluster examples of similar structure together in a completely unsupervised manner.

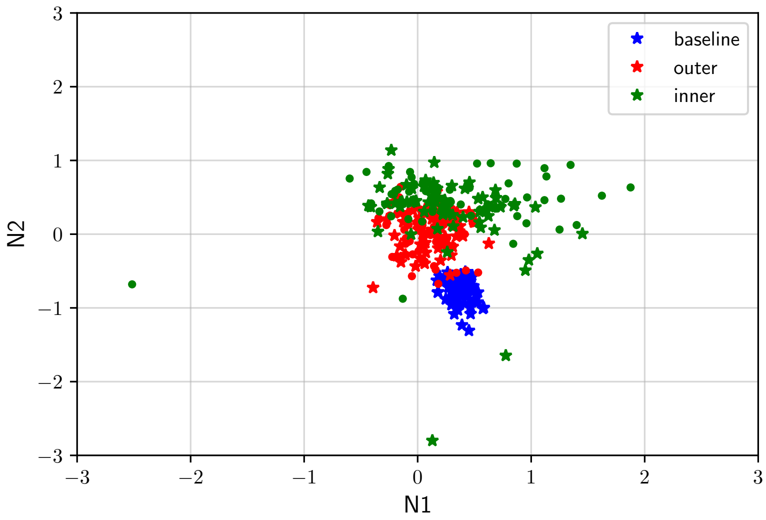


Figure 4.7: Clustering of the latent representation of samples from various fault modes, trained in an unsupervised setting.

In the second case where anchor points were used, as shown in Fig. 4.8, it can be seen that the anchor point is able to further separate the clusters of fault modes, helping to improve the classification accuracy of the proposed method.

4.3.3 Discussion

The aim of this experiment was to investigate the clustering ability of the algorithm, which subsequently can be used as a fault diagnosis tool. Here, two cases of the algorithm were presented. In the first case, the network is trained on all the data in an unsupervised manner.

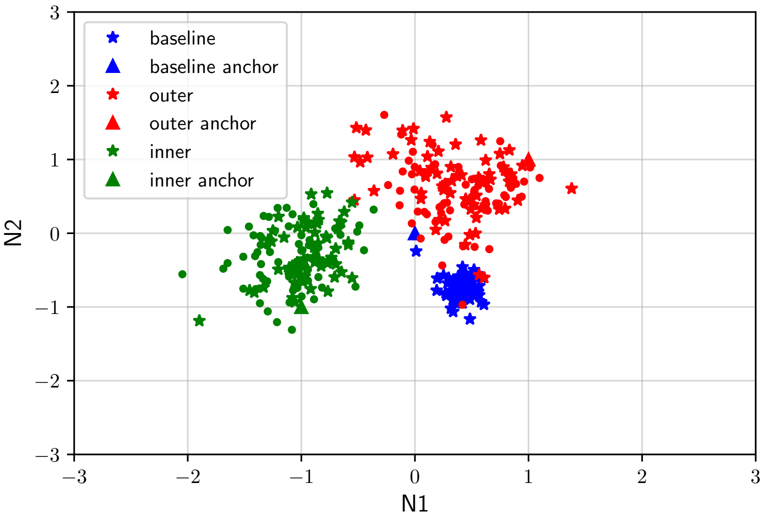


Figure 4.8: Clustering of the latent representation of a sample from various fault modes, trained with a network using anchor points of each fault mode, resulting in a semi-supervised setting.

That is, no labels were given to the network during training. It is assumed that we are working with a limited set of the data, anticipating new data as the machine runs. Regardless, all the data available can be used to diagnose the fault. From Fig. 4.7, it can be seen that the algorithm does provide some unsupervised clustering capacity. In this instance, there would be no diagnosis possible, in the absence of labels, without the help of an established bearing fault diagnostic method. When a diagnosis has been made using the established method, the proposed methodology can then be used to cluster similar samples.

In the second instance anchor points were added to the training protocol. It can be seen that the model clusters similar faults around the anchor points, thus helping to separate the various fault modes. When compared to the first part of this experiment, it was noted that adding only one labelled example of each fault mode increased the ability for the mutual information to be optimized.

4.4 High speed shaft bearing fault dataset

The third and final experiment was conducted on the high-speed bearing dataset (HSBD), which was collected from a 2MW commercial wind turbine [Bechhoefer et al., 2013]. This dataset represent a real-world dataset and is inherently non-stationary due to the varying wind speeds and loads found in nature. This dataset consists of the acceleration measurements captured over a period of 50 consecutive days at an interval of 6s on each recording. The acceleration signal was captured at a sampling rate of 97656 Hz. A tachometer signal was also recorded during the same recording window as the acceleration signal, allowing the exact shaft rate of the wind turbine to be calculated. Doing this shows that the wind turbine had an average shaft rate of 30.9 Hz with the shaft rate varying as high as 15% of the nominal rate at times. Figure 4.9 shows the shaft rate of the wind-turbine over the recording window of 6 seconds for each of the 50 days the experiment was conducted, revealing the varying speeds of each recording window. In the original paper, an inner-race fault was discovered after the data had been collected and the fault severity, obtained using the envelope spectrum, was seen to increase over the recording period.

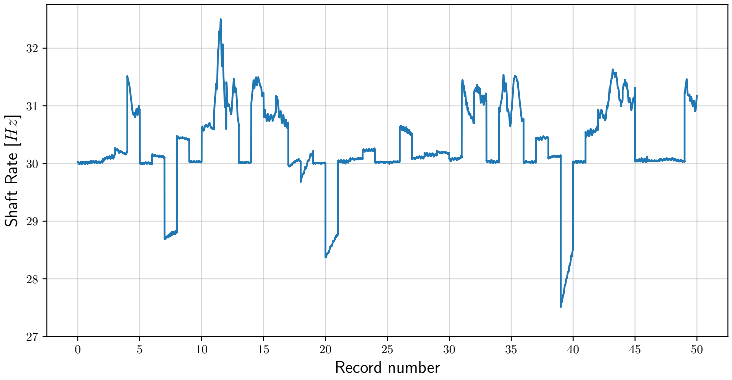


Figure 4.9: Shaft rate of wind turbine recorded over a window of 6 seconds for 50 consecutive days.

4.4.1 Approach

For this experiment, the network model presented in Section 3.4 was used with all the documented parameters. The approach used here was to increase the amount of data available to the network for training, saving a set of weights along the way. This produced three separate instances of trained networks using data from days 1-3. Initially, the network was trained with data from only the first day. Upon convergence, the network's parameters were then saved and data from the next day of recordings was added to the training pool. This was repeated until all the data of the first 3 days were seen by the model. This approach simulates live data coming in batches at a time.

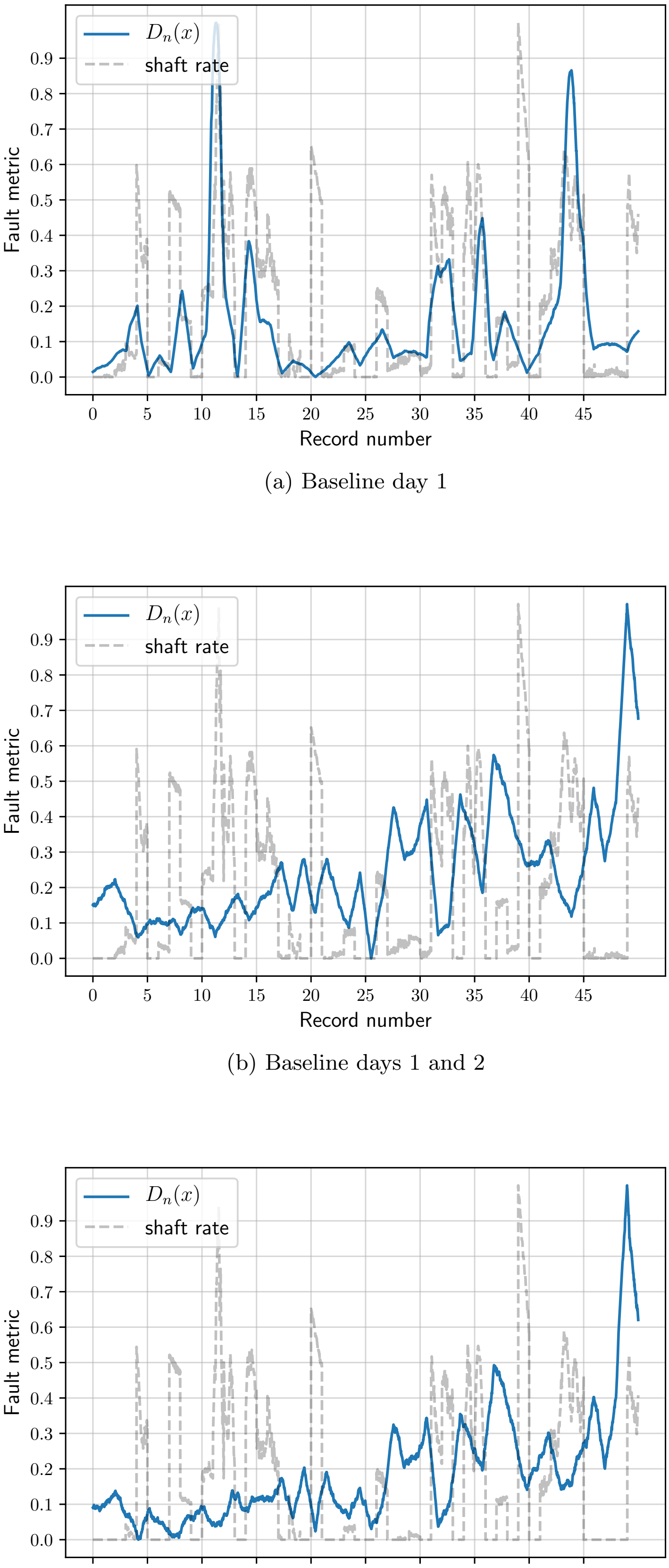
Additionally, a BIGAN [Donahue et al., 2016] based network was trained alongside the proposed model, as a SOTA GAN benchmark for comparison purposes on untangling the latent space. The BIGAN model represents the case of inference with a deep learning model without any regularization. The BIGAN thus creates random, unstructured mappings between the latent space and the observed space. With the BIGAN approach the non-stationary components of the signal are not specifically taken into account and can be used for comparison.

The proposed metric was compared against the advanced bearing approaches of Abboud et al. [2019]. With these approaches, the MED-SK-NES and IES were tracked at the inner-race fault frequency. Both of these two approaches involved order tracking and filtering the vibration signal and represent the latest advanced signal processing approaches for bearing diagnostics.

4.4.2 Results

Figure 4.10 shows the results of the proposed fault metric obtained from a network trained consecutively using the data from day 1 up until day 3. For comparison, the shaft rates that were present in the training data of the model was used to calculate the remaining % of the shaft rate that the network has not yet seen, this % is also included in grey. This allows an opportunity to empirically compare the fault metric with the shaft rates not yet seen by the model, subsequently showing the cross correlation of the metric to unseen operating conditions. It can be seen from Fig. 4.10, that as more data is used for the baseline condition, the network finds it easier to separate the deterministic and random components of the signal, since there are more deterministic components to map.

The results of the BIGAN, where the non-stationary components of the signal are not taken into consideration, are shown in Fig. 4.11. Here it can be seen that the BIGAN based fault metric is sensitive to the operating conditions that the network has not seen. This shows the necessity to regularise the latent space to account for deterministic and random components of the signal.

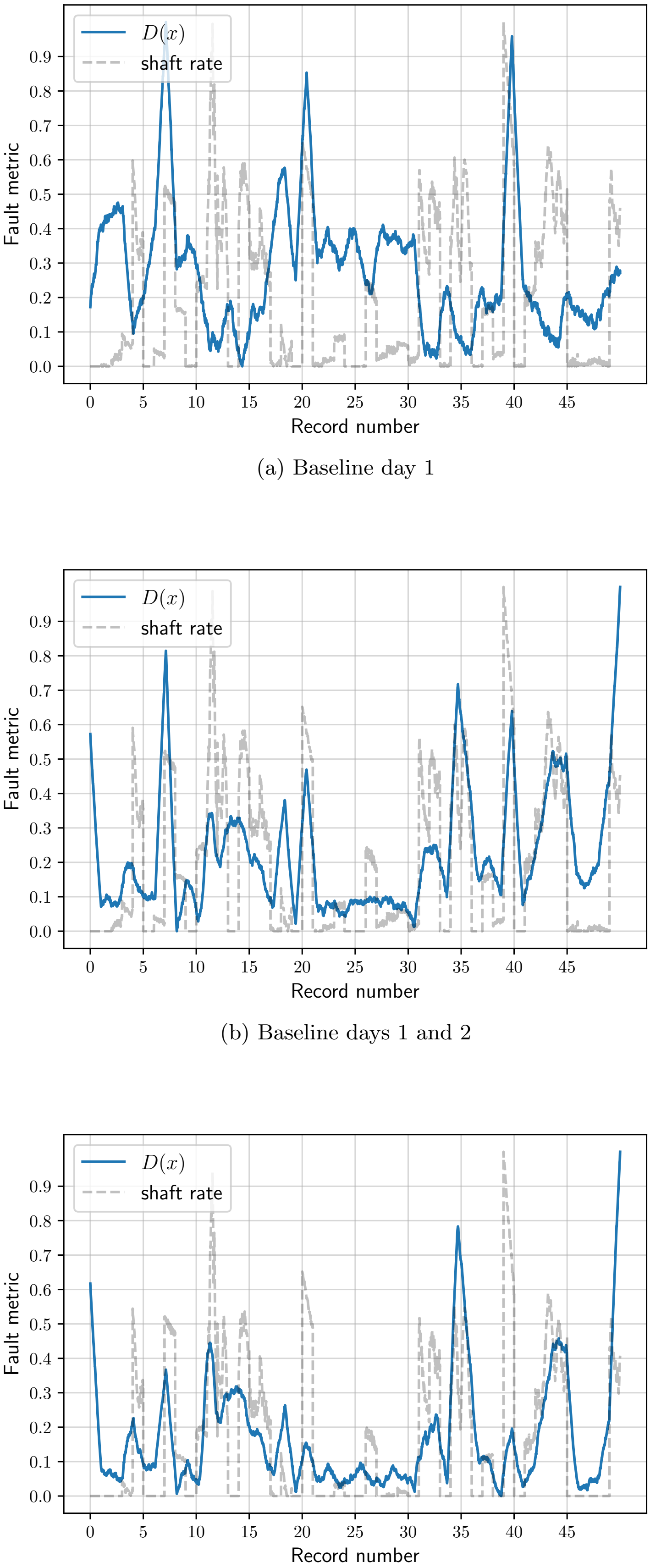


(a) Baseline day 1

(b) Baseline days 1 and 2

(c) Baseline days 1, 2 and 3

Figure 4.10: Proposed fault metric applied to HSBD dataset trended over the 50 day observation window. Figures (a)-(c) represent the cases of increasing the amount of data used as baseline reference. % Unseen shaft rate is shown in grey.



(a) Baseline day 1

(b) Baseline days 1 and 2

(c) Baseline days 1, 2 and 3

Figure 4.11: BIGAN fault metric applied to HSBD dataset trended over the 50 day observation window. Figures (a)-(c) represent the cases of increasing the amount of data used as baseline reference. % Unseen shaft rate is shown in grey.

The proposed methodology was further compared against a BIGAN using an order tracked version of the dataset and presented in Fig. 4.12. This experiment is used to show the proposed method's capacity to count for the modulations caused by speed fluctuations. In this instance, the order tracking will remove the frequency modulation caused by speed fluctuations. Here it can be seen that the BIGAN and the proposed method are both able to track the fault. However, the proposed methodology using the raw signal can be seen to be more sensitive than BIGAN using the order tracked signal. The sensitivity can be explained by the amplitude modulations that are not corrected by order tracking will still influence the signal, especially since it is not accounted for in the unregularized BIGAN based approach.

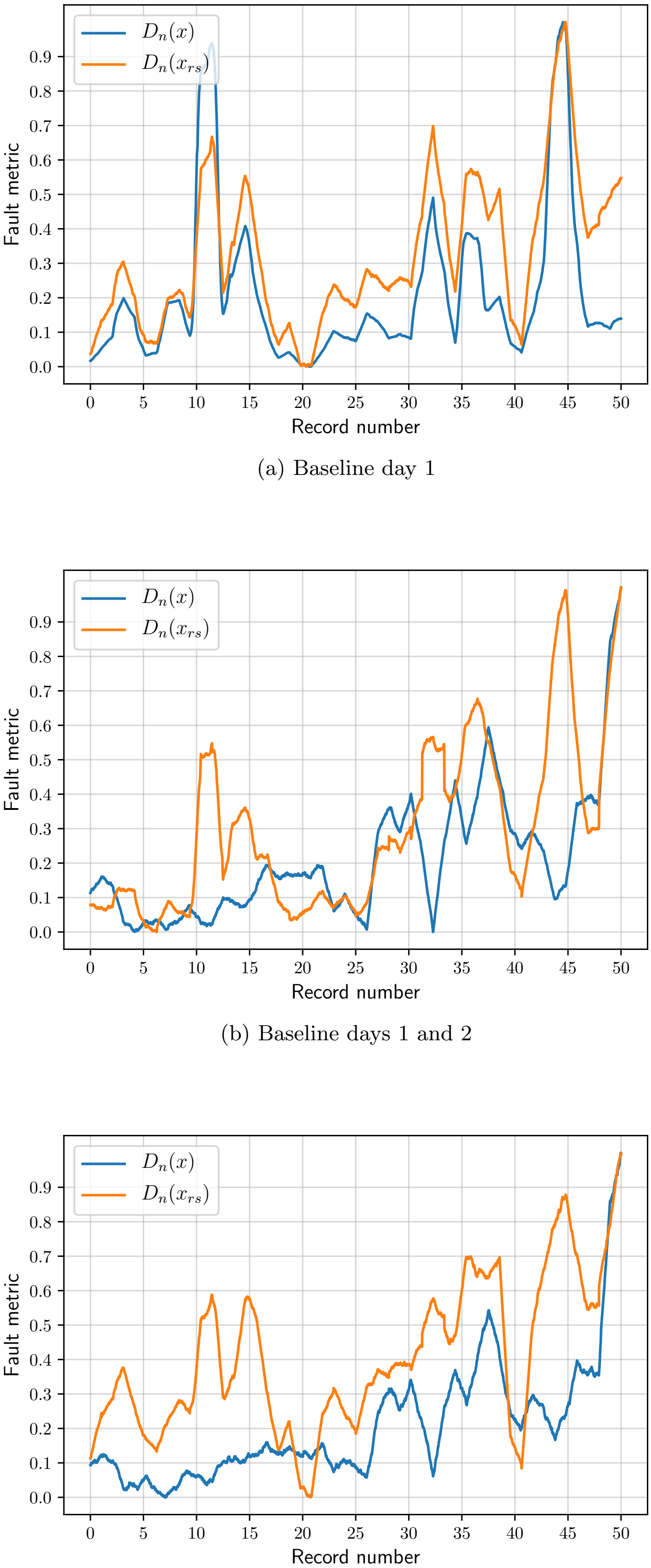
The final comparison is done using the SOTA signal processing techniques on the order tracked vibration signal compared to the proposed method on the raw vibration signal, and is shown in Fig. 4.13. Figures 4.13a and 4.13b show the baseline approaches in which the fault specific frequencies and their respective harmonics are tracked as a FSI. The arithmetic mean of the first three harmonics are used as a FSI [Abboud et al., 2019]. In Fig. 4.13d we see the proposed fault metric following the same trend as the two baselines approaches. The ROC curve of Fig. 4.13e is based on detecting the fault at record 15. It shows that the proposed fault metric performs better than the MED-SK-NES based approach, with the IES approach performing slighly better.

4.4.3 Discussion

The dataset was chosen as a representation of a real world dataset. From these results we can see that the network is able to learn a mapping of the deterministic and random components of the vibration signal. The mapping provides an FSI that is robust against any fluctuating conditions. With less than 8% of the total amount of data being seen (first 3 days) by the network, the model has sufficient capacity to separate the deterministic and random components to perform accurate inference on the remaining set of data, even when, the majority of the operating conditions have yet to be seen. Alternatively, if we look at the BIGAN approach, overall the fault metric increases, however, the BIGAN fault metric is sensitive to the fluctuating operating conditions and may trigger unnecessary early warnings. This is subsequently reduced when the signal is order tracked as expected. This shows that the regularization of the latent space is enough to perform inference without the need of a tachometer signal. The fault metric is shown to be on par with SOTA bearing diagnostic techniques. In the baseline approaches, after the preprocessing of the data, the FSI is developed using fault specific frequencies, thus knowledge of the bearing geometry is required. The proposed approach is able to provide a similar metric using raw unprocessed data.

4.5 Summary

This chapter presented the results of the proposed methodology on three well known datasets. Each dataset was chosen to highlight an important diagnostic feature of the proposed method. The IMS dataset represents cases in which the machine operating condition was kept constant, however the SNR varies from signal to signal. It was seen that the proposed fault metric is able to provide a good trend of the fault, even in cases of low SNR. The fault diagnosis capacity of the proposed method was tested with the second dataset. It was found that the model has good clustering capacity, which can be further improved by adding labelled data. Small amounts of labelled data, go a long way to regularize the representation of the latent space and improve fault classification. Lastly, the proposed method was tested on real world data in the form of signals obtained from a wind turbine. It was seen that the method is able to account for the operating conditions the network had not yet seen, and performed better than a model in which the non-stationary components were not taken into account. It was further seen that the model is able to work just as well as a similar model that has no

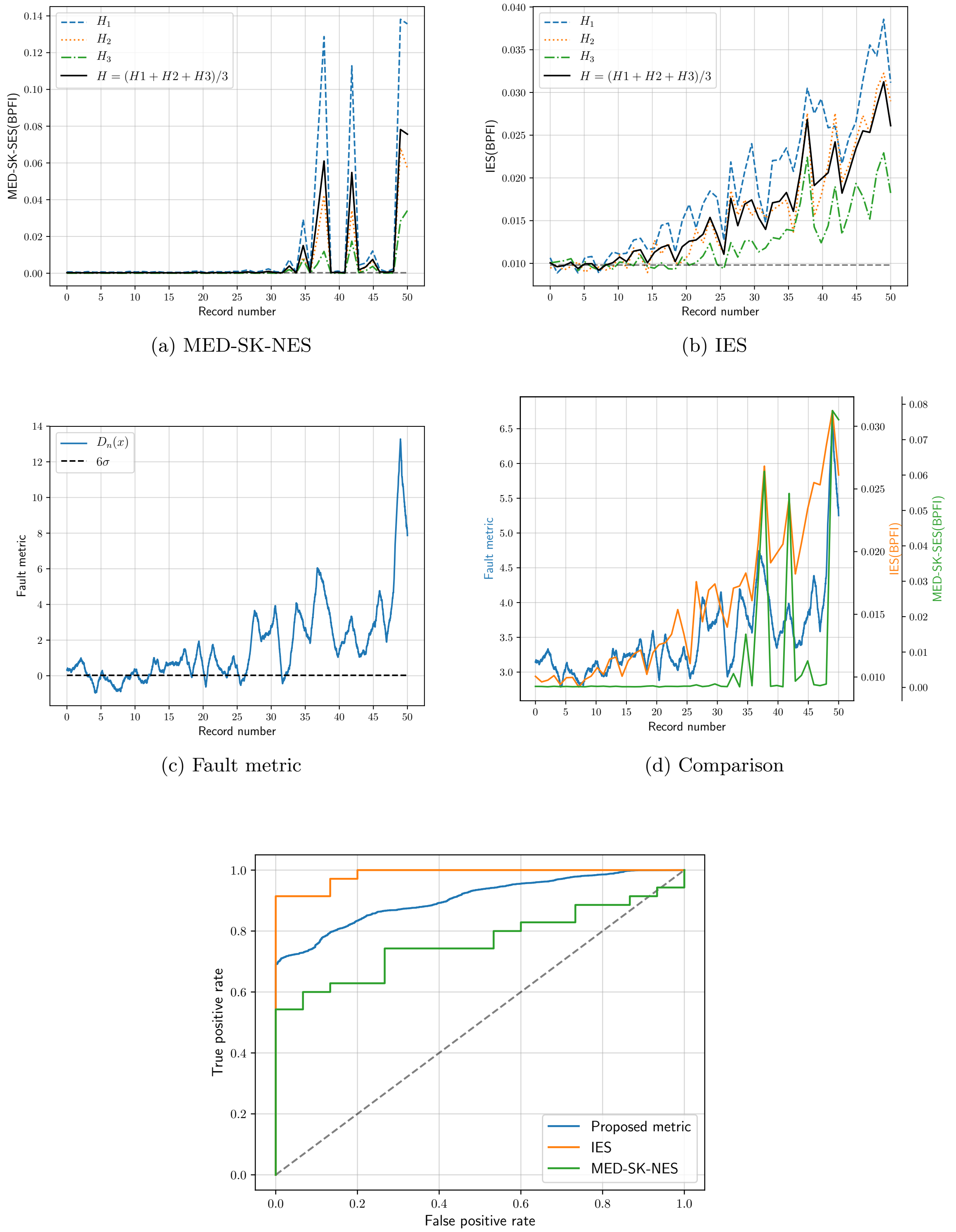


(a) Baseline day 1

(b) Baseline days 1 and 2

(c) Baseline days 1, 2 and 3

Figure 4.12: Comparison between the proposed diagnostic metric (regularized) in blue on the raw vibration signal x, against a BIGAN (un-regularized) in orange using an order tracked re-sampled version of the signal, x_{rs}.



(a) MED-SK-NES

(b) IES

(c) Fault metric

(d) Comparison

(e) ROC

Figure 4.13: Comparison between the proposed diagnostic metric on raw vibration signal against the MED-SK-NES and IES signal processing approaches on the order tracked signal. H_1, H_2, and H_3 refers to the first, second and third harmonics of the fault frequency.

latent space regularization using an order tracked version of the signal. Lastly, the proposed metric was able to provide a FSI as good as the SOTA bearing diagnostic techniques requiring significantly less set-up effort.

Chapter 5

Conclusion

5.1 Conclusions

With Industry 4.0 upon us, there is a need for scalable maintenance decision making. Fault detection and diagnosis has been and remains a crucial step towards increasing the reliability of machines. It also offers the perfect opportunity to allow for new insights provided by deep learning algorithms, towards achieving Industry 4.0 goals. However, a few key challenges need to be overcome before this can happen.

Towards this goal, this work proposes a fault detection metric that moves away from using hand designed features in a supervised learning methodology to an unsupervised learning methodology utilizing machine learnt features. Unprocessed vibration data from accelerometer sensors are taken as the only input into the model, and after training, an easily interpretable fault metric is produced. This has the result of reducing the amount of human hours spent hand crafting features and transfers some of the decision making to machine learning. To do this, the feature extraction step, that is so prevalent in many previously proposed methodologies, is now performed by the deep learning algorithm. In the past, hand crafted features were extracted based on years of experience and intuition into the mechanics of the machine and the fault. These intuitions are then translated into fault metrics using advanced signal processing techniques. This expert knowledge is often time consuming and expensive and can be difficult for machine operators to interpret.

In non-stationary operating conditions, protocol suggests that the effects of the non-stationary operating condition be removed before any fault detection takes place. This often requires an auxiliary signal of the machine operating conditions, such as a tachometer signal. The auxiliary signal is used to remove the effects of the change in shaft rotational speed, due to the fluctuating machine operating conditions. Adding this extra channel can be costly or impractical at times.

Majority of the data sampled from a machine can be classified as healthy data, whilst a small subset of the data is from unhealthy cases. Therefore there is a big class imbalance often found in PHM data. This makes training a model in a supervised manner that much more difficult. Industries either do not have the historical data or the data suffers from this large class imbalance. Therefore, the aim of the proposed model is to provide an unsupervised algorithm that can be trained on data that is available.

These goals were achieved by recasting the problem of bearing diagnostics as an inference type problem. Using variational inference, the problem of diagnostics can now be easily optimized by a machine. Most gradient based optimization agents are able to perform this operation. Most of the heavy lifting is thereby transferred to the deep learning models where both feature extracting and fault metric are constructed. From the literature, it was seen that two deep learning frameworks take centre stage for this task, Autoencoders and Generative Adversarial Networks.

In order to account for the non-stationary components, the same approach as current bearing diagnostic techniques of separating the signal into random and deterministic components is followed. To do this, the latent space of the model was regularized using mutual information, which allowed the model to learn a representation of the signal in the form of structured latent variables. These variables were separated into a deterministic part and a random part. Bearing diagnostics can be performed by measuring the difference that the presence of a bearing fault will have on the distribution of the random components of the measured signal. This difference is easily quantified by the discriminator of the proposed model.

It was demonstrated that the proposed method provides an end-to-end model for a bearing fault metric. The method was tested on a number of benchmark cases, each representing a different non-stationary condition. The fault metric is able to detect a fault in cases with low signal-to-noise-ratios over various non-stationary transmission paths from the fault to the transducer. The methodology can also be used as a fault classifier, and incorporate whatever data may be available at the time. In real world conditions, where non-stationary conditions are inherent, the proposed fault metric was able to detect and trend the fault without the need for a tachometer signal. Furthermore, the model was seen to be robust against the non-stationary operating conditions, even when these operating conditions were not seen by the algorithm at the time of training.

The significant contribution of this model is the amount of effort needed by the user to extract a FSI. This makes implementing the FSI across cloud-based platform, significantly more scalable. No information about the bearings is needed when training and analysing the FSI. The model simply takes raw data as the input and returns an easily interpretable FSI.

5.2 Recommendations

This study serves as a platform for future GAN related work in PHM, in particular, studies that would aim to develop techniques in a fully unsupervised setting. The deep learning literature is dominated by image based analyses. Many of the model architectures and parameters in this work were taken from this literature. Therefore intensive parametric studies should be done in an attempt to improve the model's accuracy and feature extraction performance. Incorporating new neural network building blocks for the deep learning models can also be done in an attempt to extract further discriminative features.

There is a need within the literature to provide a good open source dataset that represents real world operating conditions of PHM data that can be used to benchmark new and existing diagnostic methodologies. A dataset linked more closely to industrial practice will help researchers develop more reliable models, whilst making integration of methods easier.

This work was focused on bearing diagnostics, however there is no reason why it cannot be implemented on other components and their related faults. Gearbox faults produce more deterministic components within the signal and as such, performing the same analysis and focusing on the deterministic components of the latent space, may lead to further breakthroughs in fault diagnostics.

Lastly, on the data science side, work can be done on improving the resolution of the latent space, by incorporating methods that further disentangle the latent space and offer more semantic structure learning. This may eventually lead to the extraction of machine operating condition information, such as shaft speed or loads, directly from the vibration waveform alone in a fully unsupervised setting.

Bibliography

S. Abbasion, A. Rafsanjani, A. Farshidianfar, and N. Irani. Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine. *Mechanical Systems and Signal Processing*, 21:2933–2945, 2007.

D. Abboud, J. Antoni, S. Sieg-zieba, and M. Eltabach. Deterministic-random separation in nonstationary regime. *Journal of Sound and Vibration*, 362:305–326, 2016.

D. Abboud, M. Elbadaoui, W. A. Smith, and R. B. Randall. Advanced bearing diagnostics: A comparative study of two powerful approaches. *Mechanical Systems and Signal Processing*, 114:604–627, 2019.

J. Antoni. Cyclostationarity by examples. *Mechanical Systems and Signal Processing*, 23: 987–1036, 2009.

J. Antoni and P. Borghesani. A statistical methodology for the design of condition indicators. *Mechanical Systems and Signal Processing*, 114:290–327, 2019.

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

J. Antoni and R. B. Randall. Differential diagnosis of gear and bearing faults. *Journal of Vibration and Acoustics*, 124:165, 2002.

J. Antoni and R. B. Randall. Unsupervised noise cancellation for vibration signals: Part II - A novel frequency-domain algorithm. *Mechanical Systems and Signal Processing*, 18: 103–117, 2004.

J. Antoni, F. Bonnardot, A. Raad, and M. El Badaoui. Cyclostationary modelling of rotating machine vibration signals. *Mechanical Systems and Signal Processing*, 18:1285–1314, 2004.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. 2017. URL `http://arxiv.org/abs/1701.07875`.

L. D. Avendano-Valencia and S. D. Fassois. Stationary and non-stationary random vibration modelling and analysis for an operating wind turbine. *Mechanical Systems and Signal Processing*, 47:263–285, 2014.

S. A. Aye and P. S. Heyns. An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. *Mechanical Systems and Signal Processing*, 84:485–498, 2017.

E. Bechhoefer. A quick introduction to bearing envelope analysis. Technical report.

E. Bechhoefer, B. V. Hecke, and D. He. Processing for Improved Spectral Analysis. *Annual Conference of the Prognostics and Health Management Society*, pages 1–6, 2013.

M. I. Belghazi, S. Rajeswar, O. Mastropietro, N. Rostamzadeh, J. Mitrovic, and A. Courville. Hierarchical Adversarially Learned Inference. pages 1–16, 2018. URL `http://arxiv.org/abs/1802.01071`.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. M. Blei. Variational inference: foundations and innovations. URL: `https://mlssafrica.com/programme-schedule/`, 1 2019.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877, 2017.

W. Booyse. Traversing The Manifold , Unsupervised deep learning for critical asset failure prediction. Master's thesis, University of Pretoria, 2018.

P. Borghesani, R. Ricci, S. Chatterton, and P. Pennacchi. A new procedure for using envelope analysis for rolling element bearing diagnostics in variable operating conditions. *Mechanical Systems and Signal Processing*, 38:23–35, 2013.

T. Boutros and M. Liang. Detection and diagnosis of bearing and cutting tool faults using hidden Markov models. *Mechanical Systems and Signal Processing*, 25:2102–2124, 2011.

M. Cerrada, R. V. Sánchez, C. Li, F. Pacheco, D. Cabrera, J. Valente de Oliveira, and R. E. Vásquez. A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing*, 99:169–196, 2018.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. 2016. URL `http://arxiv.org/abs/1606.03657`.

Z. Cui, W. Chen, and Y. Chen. Multi-Scale convolutional neural networks for time series classification. 2016. URL `http://arxiv.org/abs/1603.06995`.

E. De Moura, C. Souto, A. Silva, and M. Irmão. Evaluation of principal component analysis and neural network performance for bearing fault diagnosis from vibration signal processed by RS and DF analyses. *Mechanical Systems and Signal Processing*, 25:1765–1772, 2011.

G. D'Elia, M. Cocconcelli, and E. Mucchi. An algorithm for the simulation of faulted bearings in non-stationary conditions. *Meccanica*, 53:1147–1166, 2018.

X. Ding and Q. He. Energy-fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66:1926–1935, 2017.

B. Dolenc, P. Boškoski, and D. Juričić. Distributed bearing fault diagnosis based on vibration analysis. *Mechanical Systems and Signal Processing*, 66-67:521–532, 2016.

J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. pages 1–18, 2016. URL `http://arxiv.org/abs/1605.09782`.

I. El-Thalji and E. Jantunen. A descriptive model of wear evolution in rolling bearings. *Engineering Failure Analysis*, 45:204–224, 2014.

S. Ericsson, N. Grip, E. Johansson, L. E. Persson, R. Sjöberg, and J. O. Strömberg. Towards automatic detection of local bearing defects in rotating machines. *Mechanical Systems and Signal Processing*, 19:509–535, 2005.

R. Gao, L. Wang, R. Teti, D. Dornfeld, S. Kumara, M. Mori, and M. Helu. Cloud-enabled prognosis for manufacturing. *CIRP Annals - Manufacturing Technology*, 64(2):749–772, 2015.

T. T. Georgiou. Distances between time-series and their autocorrelation statistics. volume 364, 2007. URL `http://arxiv.org/abs/math/0701181`.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

X. Guo, L. Chen, and C. Shen. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement: Journal of the International Measurement Confederation*, 93:490–502, 2016.

T. Heyns, P. S. Heyns, and R. Zimroz. Combining discrepancy analysis with sensorless signal resampling for condition monitoring of rotating machines under uctuating operations. *International Journal of Condition Monitoring*, 2:52–58, 2012.

G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012. URL https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

F. Huszar. inference, 2018. URL https://www.inference.vc/.

T. Ince, S. Kiranyaz, S. Member, L. Eren, M. Askar, and M. Gabbouj. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. *IEEE Transactions on Industrial Electronics*, 63:7067–7075, 2016.

O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, and S. Van Hoecke. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *Journal of Sound and Vibration*, 377:331–345, 2016.

F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing. A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. *Neurocomputing*, 272:619–628, 2018a.

F. Jia, Y. Lei, N. Lu, and S. Xing. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mechanical Systems and Signal Processing*, 110:349–367, 2018b.

S. Khan and T. Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018.

S. Khanam, N. Tandon, and J. K. Dutt. Fault size estimation in the outer race of ball bearing using discrete wavelet transform of the vibration signal. *Procedia Technology*, 14:12–19, 2014.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 2014. URL http://arxiv.org/abs/1412.6980.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. 2013. URL http://arxiv.org/abs/1312.6114.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc., 2015.

R. Kumar and M. Singh. Outer race defect width measurement in taper roller bearing using discrete wavelet transform of vibration signal. *Measurement*, 46:537–545, 2013.

Y. Lei, Z. He, and Y. Zi. EEMD method and WNN for fault diagnosis of locomotive roller bearings. *Expert Systems With Applications*, 38:7334–7341, 2011a.

Y. Lei, J. Lin, Z. He, and Y. Zi. Application of an improved kurtogram method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, 25:1738–1749, 2011b.

C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin. ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching. pages 1–22, 2017. URL http://arxiv.org/abs/1709.01215.

L. Liu, X.-y. Li, W. Zhang, and T.-m. Jiang. Fuzzy reliability prediction of rotating machinery product with accelerated testing data. *Journal of Vibroengineering*, 17:4193–4211, 2015.

R. Liu, B. Yang, E. Zio, and X. Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, 2018.

T. Liu, J. Chen, and G. Dong. Zero crossing and coupled hidden Markov model for a rolling bearing performance degradation assessment. *Journal of Vibration and Control*, 20:2487–2500, 2014.

C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Processing*, 130:377–388, 2017.

A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. 28:6, 2013.

A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial Autoencoders. 2015. URL `http://arxiv.org/abs/1511.05644`.

T. Marwala, U. Mahola, and F. V. Nelwamondo. Hidden Markov Models and Gaussian Mixture Models for Bearing Fault Detection Using Fractals. *International Joint Conference on Neural Networks, Vancouver, Canada*, pages 3237–3242, 2006.

P. D. McFadden and J. D. Smith. Vibration monitoring of rolling element bearings by the high-frequency resonance technique - a review. *Tribology International*, 17:3–10, 1984.

P. D. McFadden and J. D. Smith. The vibration produced by multiple point defects in a rolling element bearing. *Journal of Sound and Vibration*, 98(2):263–273, 1985.

Y. Miao, M. Zhao, J. Lin, and Y. Lei. Application of an improved maximum correlated kurtosis deconvolution method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, 92:173–195, 2017.

A. B. Ming, W. Zhang, Z. Y. Qin, and F. L. Chu. Fault feature extraction and enhancement of rolling element bearing in varying speed condition. *Mechanical Systems and Signal Processing*, 76-77:367–379, 2016.

S. Mohamed and B. Lakshminarayanan. Learning in Implicit Generative Models. 2016. URL `http://arxiv.org/abs/1610.03483`.

B. Muruganatham, M. A. Sanjith, B. Krishnakumar, and S. A. V. S. Murty. Roller element bearing fault diagnosis using singular spectrum analysis. *Mechanical Systems and Signal Processing*, 35:150–166, 2013.

P. Nguyen, M. Kang, J.-m. Kim, B.-h. Ahn, J.-m. Ha, and B.-k. Choi. Robust condition monitoring of rolling element bearings using de-noising and envelope analysis with signal decomposition techniques. *Expert Systems With Applications*, 42:9024–9032, 2015.

H. Ocak and K. A. Loparo. HMM-Based Fault Detection and Diagnosis Scheme for Rolling Element Bearings. *Journal of Vibration and Acoustics*, 127:299, 2005.

H. Qiu, J. Lee, J. Lin, and G. Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289:1066–1090, 2006.

J. Rafiee, M. A. Rafiee, and P. W. Tse. Application of mother wavelet functions for automatic gear and bearing fault diagnosis. *Expert Systems with Applications*, 37:4568–4579, 2010.

R. B. Randall. *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. 2011.

R. B. Randall and J. Antoni. Rolling element bearing diagnostics-A tutorial. *Mechanical Systems and Signal Processing*, 25:485–520, 2011.

R. B. Randall, N. Sawalhi, and M. Coats. A comparison of methods for separation of deterministic and random signals. *International Journal of Condition Monitoring*, 1:11–19, 2011.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning, ICML 2014*, 4:3057–3070, 2014.

M. Riedmiller and H. Braun. RPROP - A Fast Adaptive Learning Algorithm. *Proceedings of the International Symposium on Computer and Information Science VII*, 01:4–10, 1992.

A. Rojas and A. K. Nandi. Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines. *Mechanical Systems and Signal Processing*, 20:1523–1536, 2006.

R. Roy, R. Stark, K. Tracht, S. Takata, and M. Mori. CIRP Annals - Manufacturing Technology Continuous maintenance and the future – Foundations and technological challenges. *CIRP Annals - Manufacturing Technology*, 65:667–688, 2016.

L. Saidi, J. Ben Ali, and F. Fnaiech. Application of higher order spectral features and support vector machines for bearing faults classification. *ISA Transactions*, 54:193–206, 2015.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. 2016. URL `http://arxiv.org/abs/1606.03498`.

J. Serrà, S. Pascual, and A. Karatzoglou. Towards a universal neural network encoder for time series. 2018. URL `http://arxiv.org/abs/1805.03908`.

H. Shao, H. Jiang, H. Zhao, and F. Wang. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 95:187–204, 2017.

H. Shao, H. Jiang, Y. Lin, and X. Li. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mechanical Systems and Signal Processing*, 102:278–297, 2018.

M. Singh and R. Kumar. Thrust bearing groove race defect measurement by wavelet decomposition of pre-processed vibration signal. *Measurement*, 46:3508–3515, 2013.

J. R. Stack, T. G. Habetler, and R. G. Harley. Fault classification and fault signature production for rolling element bearings in electric machines. *IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, SDEMPED 2003 - Proceedings*, 40(3):172–176, 2003.

W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement: Journal of the International Measurement Confederation*, 89:171–178, 2016.

R. Thirukovalluru, S. Dixit, R. K. Sevakula, N. K. Verma, and A. Salour. Generating feature sets for fault diagnosis using denoising stacked auto-encoder. *2016 IEEE International Conference on Prognostics and Health Management, ICPHM 2016*, 2016.

D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot. A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models. *IEEE Transactions on Reliability*, 61:491–503, 2012.

J. Urbanek, T. Barszcz, and J. Antoni. Time – frequency approach to extraction of selected second-order cyclostationary vibration components for varying operational conditions. *Measurement*, 46:1454–1463, 2013.

J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018.

T. Wang, M. Liang, J. Li, W. Cheng, and C. Li. Bearing fault diagnosis under unknown variable speed via gear noise cancellation and rotational order sideband identification. *Mechanical Systems and Signal Praessing*, 62-63:30–53, 2015.

Y. Wang, J. Xiang, R. Markert, and M. Liang. Spectral kurtosis for fault detection , diagnosis and prognostics of rotating machines : A review with applications. *Mechanical Systems and Signal Processing*, 66-67, 2016.

Z. Wang, W. Yan, and T. Oates. Time Series Classification from Scratch with Deep NN - a strong baseline. pages 1578–1585, 2017. URL `http://arxiv.org/abs/1611.06455`.

J. Wen, H. Gao, S. Li, L. Zhang, X. He, and W. Liu. Fault diagnosis of ball bearings using Synchrosqueezed wavelet transforms and SVM. In *Proceedings of 2015 Prognostics and System Health Management Conference, PHM 2015*, 2016.

J. Yu. Bearing performance degradation assessment using locality preserving projections and Gaussian mixture models. *Mechanical Systems and Signal Processing*, 25:2573–2588, 2011.

J. Yuan and X. Liu. Semi-supervised learning and condition fusion for fault diagnosis. *Mechanical Systems and Signal Processing*, 38:615–627, 2013.

P. Zhang, Y. Du, T. G. Habetler, and B. Lu. A Survey of Condition Monitoring and Protection Methods for Medium-Voltage Induction Motors. *IEEE Transactions on Industry*

Applications, 47:34–46, 2011.

S. Zhang, Y. Zhang, L. Li, and J. Zhu. Rolling Elements Bearings Degradation Indicator Based on Continuous Hidden Markov Model. *Journal of Failure Analysis and Prevention*, 15:691–696, 2015.

W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing*, 100:439–453, 2018.

X. Zhang and J. Zhou. Multi-fault diagnosis for rolling element bearings based on ensemble empirical mode decomposition and optimized support vector machines. *Mechanical Systems and Signal Processing*, 41:127–140, 2013.

R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115: 213–237, 2019. URL `http://arxiv.org/abs/1612.07640`.

H. Zhou, J. Chen, G. Dong, H. Wang, and H. Yuan. Bearing fault recognition method based on neighbourhood component analysis and coupled hidden Markov model. *Mechanical Systems and Signal Processing*, 66-67:568–581, 2016.

Y. Zhou, K. Gu, and T. Huang. Unsupervised Representation Adversarial Learning Network: from Reconstruction to Generation. 2018. URL `http://arxiv.org/abs/1804.07353`.

Appendix A

Model architecture

A.1 Network components

The architecture of the individual network components which were used for the majority of the experimental work, and which are different to the original publications, are presented next.

A.1.1 Encoder

The encoder of the network consists of four convolution layers followed by a fully connected layer, whose output is then fed into three separate fully connected layers to give each of the latent variable components. Between each layer, leaky rectified linear unit (leaky-RELU, Maas et al. [2013]) activation function is used with a leak rate of 0.1. Figure A.1 shows a schematic diagram of the network. A summary description is given in Table A.1. The convolution layers all have a kernel size of 25 with a stride of length 5. The number of filters of each layer increases by a factor of 2, for each layer starting at 32 for the first convolution layer and ending with 256 in the last convolution layer. The output of the last convolution layer, is fed into a fully connected layer with a dimension of 1024. The output layer can be considered as the recognition layer, whose output is fed into three separate fully connected layers for each of the latent variable representations (deterministic and random).

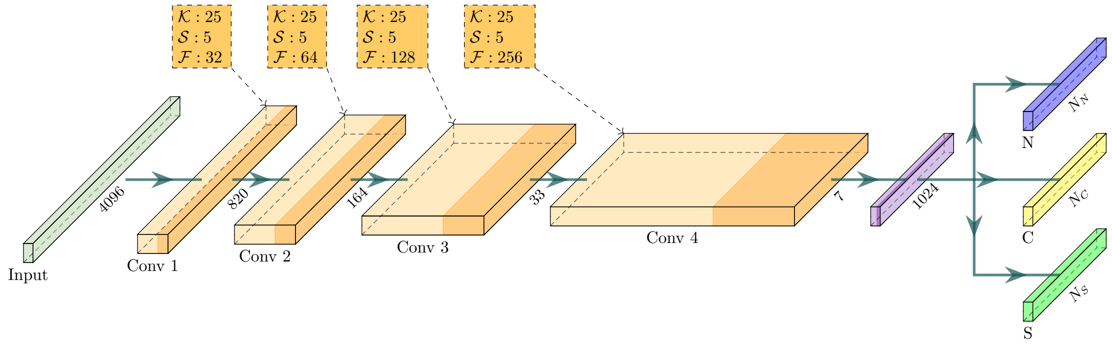

$\mathcal{K}:25$ $\mathcal{S}:5$ $\mathcal{F}:32$
$\mathcal{K}:25$ $\mathcal{S}:5$ $\mathcal{F}:64$
$\mathcal{K}:25$ $\mathcal{S}:5$ $\mathcal{F}:128$
$\mathcal{K}:25$ $\mathcal{S}:5$ $\mathcal{F}:256$
4096
Input
820
Conv 1
164
Conv 2
33
Conv 3
7
Conv 4
1024
N_N
N
N_C
C
N_S
S

Figure A.1: Schematic outline of encoder network. $\mathcal{K}$: Kernel size, $\mathcal{S}$: Stride length, $\mathcal{F}$: No. of Filters.

A.1.2 Decoder

The decoder network is comprised of a fully connected layer followed by four deconvolution layers. The output dimension of 2048×16, of the initial fully connected layer is chosen such that the output decoder has the same dimension as the input to the encoder, namely 4096.

Table A.1: Summary of encoder network layers.

Layer	Structure
Input	$x = [batch, 4096]$
1	$Conv1D_{25\times 32, stride=5}$ $\rightarrow leakyRELU(\cdot)$
2	$Conv1D_{25\times 64, stride=5}$ $\rightarrow leakyRELU(\cdot)$
3	$Conv1D_{25\times 128, stride=5}$ $\rightarrow leakyRELU(\cdot)$
4	$Conv1D_{25\times 256, stride=5}$ $\rightarrow leakyRELU(\cdot)$
5	FC_{1024} $\rightarrow leakyRELU(\cdot)$
6	**c:** $FC_{N_c=3\times 20}$ $\rightarrow softmax(\cdot)$ **s:** $FC_{N_s=2\times 5}$ $\rightarrow s_\mu = 1\times 5$ $\rightarrow s_\sigma = \exp(1\times 5)$ **n:** $FC_{N_n=128}$

In this instance, rectified linear units (RELU) were used as the activation functions between each layer as shown in Fig. A.2. The output of the decoder has no activation function such that pre-processing of the data is not required. A summary of the decoder network is given in Table. A.2. The deconvolution layers also have a kernel size of 25. The stride length, however, is set to 4 and the filters decrease by a factor 2 for each layer, starting at 128 to 32 at the second last layer. The output layer of the decoder is a deconvolution layer with a filter size of 1.

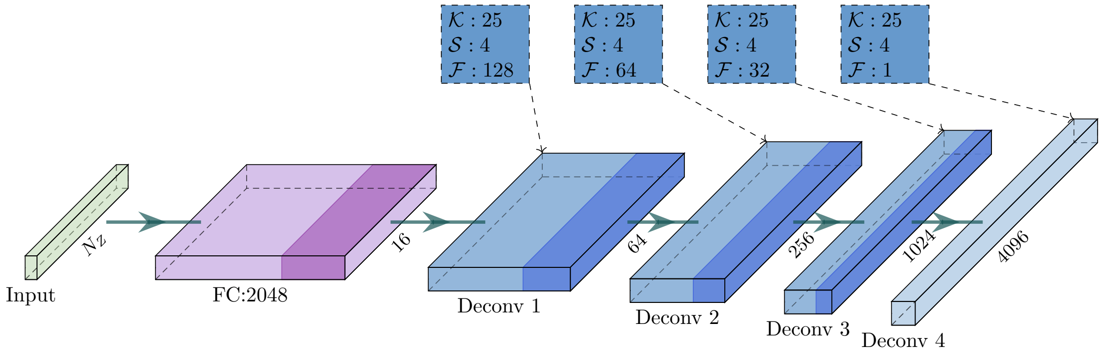


Figure A.2: Schematic outline of decoder network. $\mathcal{K}$: Kernel size, $\mathcal{S}$: Stride length, $\mathcal{F}$: No. of Filters.

A.1.3 Latent variable discriminators

The discriminators used for each of the latent variables, when training with an AAE configuration, each have the same architecture. The only difference between them is that the input dimension for the respective latent variable changes. Note that three separate discriminators are used, each with their own set of parameters. Only the structure of the discriminator remains the same. The discriminators are comprised of two hidden layers, with a hidden

Table A.2: Summary of decoder network layers.

Layer	Structure
Input	$x = [batch, 193]$
1	$FC_{16\times2048}$ $\rightarrow RELU(\cdot)$
2	$Deconv1D_{25\times128,stride=4}$ $\rightarrow RELU(\cdot)$
3	$Deconv1D_{25\times64,stride=4}$ $\rightarrow RELU(\cdot)$
4	$Deconv1D_{25\times32,stride=4}$ $\rightarrow RELU(\cdot)$
5	$Deconv1D_{25\times1,stride=4}$

dimension of 3000 each. Again, leaky RELU is used between each layer as summarized in Table. A.3. The dimension of the final layer is 1 and is used for adversarial training. Note the output of the discriminator is left linear. As shown in Fig. A.3.

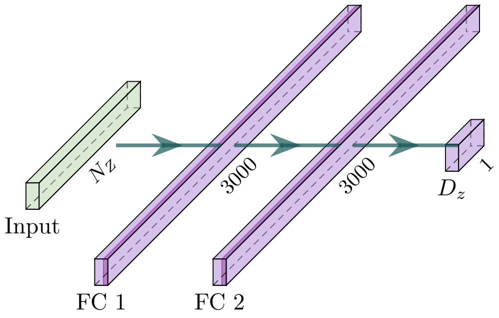


Figure A.3: Schematic outline of z-discriminator network.

Table A.3: Summary of z-discriminator network layers.

Layer	Structure
Input	$x = [batch, N_c$ or N_s or $N_n]$
1	FC_{3000} $\rightarrow leakyRELU(\cdot)$
2	FC_{3000} $\rightarrow leakyRELU(\cdot)$
3	$D_{c/s/n} = FC_1$

A.1.4 Signal discriminator

Lastly, the discriminator that is used in the infoGAN-like structure has an architecture similar to the encoder, with the difference that the output layer is only 1 dimension, to allow for adversarial training. The discriminator architecture is shown in Fig. A.4. In this case, the last layer is passed through a sigmoid activation function as summarized in Table. A.4.

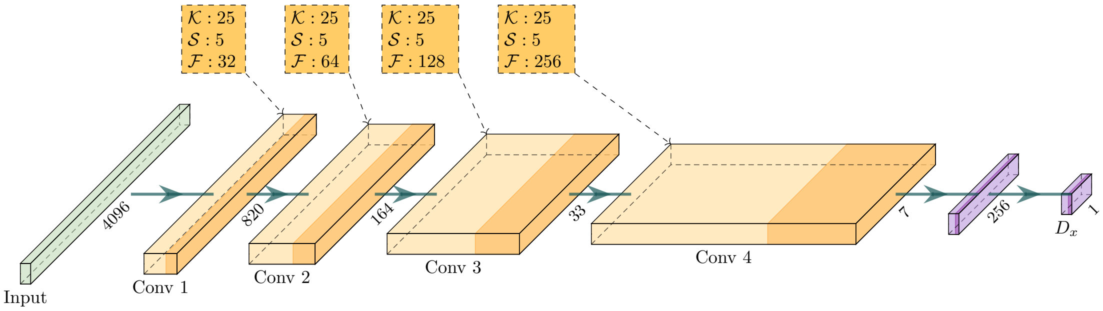


Figure A.4: Schematic outline of x-discriminator network.

Table A.4: Summary of x-discriminator network layers.

Layer	Structure
Input	$x = [batch, 4096]$
1	$Conv1D_{25\times 32, stride=5}$ $\rightarrow leakyRELU(\cdot)$
2	$Conv1D_{25\times 64, stride=5}$ $\rightarrow leakyRELU(\cdot)$
3	$Conv1D_{25\times 128, stride=5}$ $\rightarrow leakyRELU(\cdot)$
4	$Conv1D_{25\times 256, stride=5}$ $\rightarrow leakyRELU(\cdot)$
5	FC_{256} $\rightarrow leakyRELU(\cdot)$
6	$D_x = FC_1$ $\rightarrow sigmoid(\cdot)$