Latent analysis of unsupervised latent variable models in fault diagnostics of rotating machinery under stationary and time-varying operating conditions

by

Ryan Balshaw
15020658

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

in the

Department of Mechanical and Aeronautical Engineering
Faculty of Engineering, Built Environment and Information Technology

Supervised by
Prof. PS Heyns, Prof. DN Wilke, Dr. S Schmidt

UNIVERSITY OF PRETORIA

2020

Acknowledgements

The author gratefully acknowledges the contribution of the Eskom Power Plant Engineering Institute (EEPEI) in the funding of this research. I would like to acknowledge and thank my supervisors, Prof. Stephan Heyns, Prof. Nico Wilke and Dr. Stephan Schmidt. Their insightful analysis, guidance and support throughout this research process were pivotal to the completion of this work. I would also just like to thank my friends and family for their support.

ABSTRACT

Latent analysis of unsupervised latent variable models in fault diagnostics of rotating machinery under stationary and time-varying operating conditions

Author:	Ryan Balshaw
Supervisors:	Prof. PS Heyns, Prof. DN Wilke, Dr. S Schmidt
Department:	Mechanical and Aeronautical Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Mechanical Engineering)
Keywords:	Deep Learning, Principal Component Analysis, Variational-Auto-Encoders, Generative Adversarial Networks, Anomaly Detection, Asset Integrity Management, Gearbox fault detection, stationary operating conditions, time-varying operating conditions

Vibration-based condition monitoring is a key and crucial element for asset longevity and to avoid unexpected financial compromise. Currently, data-driven methodologies often require significant investments into data acquisition and a large amount of operational data for both healthy and unhealthy cases. The acquisition of unhealthy fault data is often financially infeasible and the result is that most methods detailed in literature are not suitable for critical industrial applications.

In this work, unsupervised latent variable models negate the requirement for asset fault data. These models operate by learning the representation of healthy data and utilise health indicators to track deviance from this representation. A variety of latent variable models are compared, namely: Principal Component Analysis, Variational Auto-Encoders and Generative Adversarial Network-based methods. This research investigated the relationship between time-series data and latent variable model design under the sensible notion of data interpretation, the influence of model complexity on result performance on different datasets and shows that the latent manifold, when untangled and traversed in a sensible manner, is indicative of damage.

Three latent health indicators are proposed in this work and utilised in conjunction with a proposed *temporal preservation* approach. The performance is compared over the different models. It was found that these latent health indicators can augment standard health indicators and benefit model performance. This allows one to compare the performance of different latent variable models, an approach that has not been realised in previous work as the interpretation of the latent manifold and the manifold response to anomalous instances had not been explored. If all aspects of a latent variable model are systematically investigated and compared, different models can be analysed on a consistent platform.

In the model analysis step, a latent variable model is used to evaluate the available data such that the health indicators used to infer the health state of an asset, are available for analysis and comparison. The datasets investigated in this work consist of stationary and time-varying operating conditions. The objective was to determine whether deep learning is comparable or on par with state-of-the-art signal processing techniques. The results showed that damage is detectable in both the input space and the latent space and can be trended to identify clear condition deviance points. This highlights that both spaces are indicative of damage when analysed in a sensible manner. A key take away from this work is that for data that contains impulsive components that manifest naturally and not due to the presence

of a fault, the anomaly detection procedure may be limited by inherent assumptions made in model formulations concerning Gaussianity.

This work illustrates how the latent manifold is useful for the detection of anomalous instances, how one must consider a variety of latent-variable model types and how subtle changes to data processing can benefit model performance analysis substantially. For vibration-based condition monitoring, latent variable models offer significant improvements in fault diagnostics and reduce the requirement for expert knowledge. This can ultimately improve asset longevity and the investment required from businesses in asset maintenance.

Table of Contents

List of Abbreviations

AAE	Adversarial auto-encoder
Adam	Adaptive moment estimation method
AE	Auto-encoder
ANN	Artificial neural network
AR	Auto-regression
AUC	Area-under-the-curve
BCF	Ball cage frequency
BGS	Bayesian geometry compensation
BPFI	Ball pass frequency for the inner race
BPFO	Ball pass frequency for the outer race
BSF	Ball spin frequency
C-AIM	Centre for Asset Integrity and Management
CBM	Condition-based maintenance
CCR	Cumulative contribution rate
CF	Crest factor
CNN	Convolutional neural network
COT	Computed order tracking
CPW	Cepstrum pre-whitening
DBM	Deep Boltzmann machines
DBN	Deep belief network
DFT	Discrete Fourier transform
DLS	Disentangled latent space
ELBO	Evidence lower bound
EM	Expectation maximisation
ES	Expert system
FFNN	Feed-forward neural network
FT	Fourier transform
FTF	Fundamental train frequency
GAN	Generative adversarial network
GMM	Gaussian mixture model
HI	Health indicator
HMM	Hidden Markov model
IES	Improved envelope spectrum
IFT	Inverse Fourier transform
IMS	Intelligent Maintenance Systems
JSD	Jensen-Shannon Divergence
KL	Kullback-Leibler
LDA	Linear discriminant analysis
LHI	Latent health indicator
LL	Log-likelihood
MED	Minimum entropy deconvolution
MI	Mutual information

MLP	Multi-layer perceptron
MMD	Maximum mean discrepancy
MSE	Mean squared error
NES	Normalised squared envelope spectrum
NLL	Negative log-likelihood
OT	Optimal transport
PC	Principal component
PCA	Principal component analysis
PHM	Prognostics and health management
RES	Residual signal analysis
RMS	Root mean square
RNN	Recurrent neural network
RPROP	Resilient propagation
RUL	Remaining useful life
RY-GAN	Representation Yielding GAN
SES	Squared envelope spectrum
SK	Spectral kurtosis
SN	Spectral normalisation
SNR	Signal-to-noise ratio
SOTA	State-of-the-art
SVM	Support vector machine
TC	Total correlation
TSA	Time synchronous averaging
UP	University of Pretoria
VAE	Variational auto-encoder
VI	Variational inference
WAE	Wasserstein auto-encoder
WGAN	Wasserstein GAN
WPT	Wavelet packet transform

List of Symbols

Chapter One

Roman symbols:

d	Roller element diameter
D	Bearing pitch diameter
f_m	Gear mesh frequency
f_s	Shaft speed
F_s	Sampling frequency
L_s	Signal length
L_{sft}	Window shift increment size
L_w	Model window length
N	Roller element count
N_t	Gear teeth number
$p(\cdot)$	A probability distribution
$\mathbf{t}$	Target variable
$x(t)$	Time-series signal
$\mathbf{x}$	Vector of variables

Greek symbols:

θ	model parameters
ϕ	Bearing contact angle

Chapter Two

Roman symbols:

$Bern(\cdot)$	Bernoulli distribution
$\mathbf{c}$	Categorical latent component
$Cat(K, p)$	Categorical distribution
D	Dimensionality
$D_\sigma(q\|p)$	Divergence metric
$D_\phi \| D_\chi$	Data discriminator

$D_\omega \vert D_n$	Latent **n** critic
D_ς	Latent **c** discriminator
$\mathbb{E}$	Expectation operator
E_ϕ	Encoder network
G_θ	Generator or Decoder network
$\mathbf{I}$	Identity matrix
$k(\cdot,\cdot)$	Gaussian kernel
$\mathcal{L}$	Objective function
$\mathbf{n}$	Noise latent component
$\mathcal{N}$	Normal distribution
$q(\cdot)$	Parametric distribution
$r(\mathbf{x})$	Density ratio
$\mathbf{s}$	Continuous latent component
$\mathbf{S}$	Data covariance matrix
$\mathbf{U}$	Eigenvector transformation matrix
$\mathbf{w}$	Weight vector
$\mathbf{x}$	Vector of variables
$\tilde{\mathbf{x}}$	Reconstructed input
$\mathbf{X}$	Matrix of variables
$\mathbf{z}$	Latent variable

Greek symbols:

α, β	Model parameter
ε	Shape parameter
$\boldsymbol{\varepsilon}$	Noise distribution
$\theta, \phi, \chi, \omega, \varsigma$	Network parameters
μ	Mean
σ^2	Variance
λ	Enforcement parameter

Chapter Three

Roman symbols:

$D_\chi(\mathbf{x})$	Data discriminator
$D_n(\mathbf{n})$	Latent critic
f_s	Shaft speed
F_s	Sampling frequency
L_d	Discrepancy signal length
L_s	Signal length
L_{sft}	Window shift increment size
L_w	Model window length
N	Roller element count
$\mathbf{n}$	Noise latent space vector

$p(\cdot)$	A probability distribution
t	Time
$\mathbf{v}$	Latent velocity
x	Signal
$\tilde{x}$	Reconstructed signal
$\mathbf{x}$	Vector of variables
$\mathbf{X}$	Matrix of variables
$\mathbf{z}$	Latent variable
$\mathbf{z}_t$	Latent vector at time instance t

Greek symbols:

σ^2	Variance
σ	Standard deviation

Chapter Four

Roman symbols:

dB	Decibel
F_{dam}	Damage amplitude
$h_i(t)$	Impulse response function
$M_i(t)$	Modulation function
P_i	Signal power
$q(\cdot)$	Instantaneous loading function
$\mathcal{T}_i$	Impulse time
$x_i(t)$	Simulated component
$\otimes$	Convolution operator

Greek symbols:

ε	White noise
ζ_i	Damping ratio
σ^2	Variance
$\theta_{ref}(t)$	Instantaneous angular position
ω_n	Natural frequency
$\omega_{ref}(t)$	Instantaneous shaft speed

Chapter 1 Introduction

1.1 Background

Industrial processes, particularly those applicable to sectors such as mining, manufacturing and power generation, are subject to increased production demand. The productivity of such processes must be high, to satisfy demands placed on the process. The productivity of industrial processes is directly linked to the assets that these processes use, where often the expectation in these processes is that they must maintain operation with minimal intervention or downtime. However, deterioration of physical assets is inevitable, which naturally leads to the requirement for asset maintenance.

Maintenance of industrial assets has evolved, from the original method of unplanned maintenance, where only significant asset damage resulted in an intervention. This method, however, can be costly as the down-time of critical assets can lead to severe financial privation. Time-based preventative maintenance was subsequently introduced and functions on the principle of periodic maintenance to ensure that asset reliability is maintained. This method can be expensive as periodic maintenance does not offer improved cost per unit financial gain, and gave rise to a more efficient approach known as Condition Based Maintenance (CBM) (Jardine et al., 2006). CBM requires that maintenance be performed when there is clear evidence of a problem within a physical asset.

Vibration-based CBM is the most common technique used when performing CBM, and is based on the principle that the health characteristics of an asset are intrinsically contained within a vibration signal. However, this is not the only type of health characteristic detection, with acoustic emission, oil debris, and magnetic chip detection being used for CBM (Večeř et al., 2005). The vibration signal itself is not a sufficient indicator, but rather the signal covariates that relate to the asset's state. Rotating machinery is a common asset group within industry and resulted in the extensive development of CBM techniques to ensure that reliability is maintained. Gearboxes in industrial applications are not only critical components but also a severe expense should a critical failure occur. For example, wind turbine statistics show that 17% of failures are due to gearboxes alone, which also requires the longest downtime to perform gearbox maintenance. Bearings are an even larger contributor, with bearing failure causing 76% of wind turbine failures (Sheng, 2016).

Vibration-based CBM operates on the principle of three distinct operational steps, namely, data acquisition, data processing, and maintenance decision-making (Jardine et al., 2006). Data acquisition is the process of obtaining the necessary vibration signals through the use of sensors. Data processing is the step whereby the covariates of the system are extracted from the signal, and often requires advanced processing techniques. After the necessary covariates are extracted, maintenance decision-making allows for health assessment of the asset. This decision-making is typically split into two categories,

namely, diagnostics and prognostics. Diagnostics is a methodology whereby the covariates are mapped to specific fault cases in the fault space. Prognostics refers to the prediction of the Remaining Useful Life (RUL) of the asset. Prognostics differs in the sense that one aims to estimate the time to critical fault failure, as opposed to classifying a fault directly.

The issue associated with diagnostic CBM is that once a fault is identified, one cannot prevent the associated downtime and thus there is often little time for preparation. Prognostic CBM, however, is a predictive-preventative methodology. A superior, alternate framework, Prognostics and Health Management (PHM), aims to incorporate the facets of diagnostics and prognostics. PHM is a framework that aims to provide early detection and isolation of a fault such that it can be monitored and tracked throughout the asset's life-cycle. In that way, it attempts to isolate fault characteristics and trend the variation in these characteristics to perform maintenance when required. The underlying goal of PHM is to ensure an that the downtime of an asset is minimised through early system fault identification (Lee et al., 2014). Once the fault has been isolated and trended, a failure threshold identifies when maintenance is required (Lei et al., 2018).

There are three methods of PHM that are often utilised, namely, *i)* data-driven, *ii)* physics-driven and *iii)* hybrid-driven PHM. Data-driven PHM refers to an empirical model structure that consists of statistical techniques and features extracted from data. Data-driven models typically try to utilise manually-extracted features from signals (notice here the link to vibrational CBM) and attempt to predict the health of the machine on these covariates alone. In a data-driven framework, one needs access to both healthy and unhealthy data, which, when dealing with critical assets, is uncommon (Ramasso, 2014, Ramasso et al., 2015).

Physics-driven modelling requires a mathematical model of the asset, where this model requires prior knowledge of the failure mechanisms that may occur. The mathematical model parameters are usually unknown and thus need calibration, either through extensive experimental or empirical data (Jardine et al., 2006, Liao and Köttig, 2014). Hybrid-driven modelling approaches attempt to utilise the advantages of both techniques to maximise on the prognostic ability of the model. Recently, however, a drive towards deep-learning-based PHM has arisen, which can be considered an alternative to the data-driven approaches. This technique aims to extract features from data through the use of deep learning techniques, as opposed to hand-crafted covariates (Liao and Köttig, 2014, Lee et al., 2014).

At this stage, the level of prior knowledge that one typically has available has not been addressed. Due to the inherently dangerous nature of failure in critical assets, typically one does not have access to extensive fault data. Usually, a vast amount of healthy asset data is available. With this in mind, it is necessary to distinguish between fault detection, diagnosis, and severity. Fault detection and severity is one form of PHM, and fault diagnosis is another. One can apply both deep learning and signal processing approaches to both, with respective advantages and disadvantages that one can exploit. Deep learning offers benefits such as a reduced requirement for domain expert knowledge, no vibration signal alteration as the raw vibration signals are used as inputs and improved diagnosis result interpretation (Lei et al., 2020b).

The disadvantages of deep learning include the requirement for target fault labels in supervised learning applications and the requirement for computing resources. Furthermore, the type and location of a fault are not always known or identified, with signal processing approaches often focusing heavily on

this investigation. In this work, deep learning methods are investigated to perform fault detection and severity trending by only using healthy asset data.

The aim of this study is to perform anomaly detection in rotating machines using latent variable models by considering sensible metrics and the sensible analysis of the metrics. The unsupervised context of these models is induced through the use of only healthy asset data, whereby sensible metrics are used to measure the deviation from this healthy state. In this process, time-series data from an asset undergo a series of transformations between the initial observation of the signal and the final analysis metric, with the latent variable model facilitating this transformation. The model will provide a fault severity indicator or fault metric to the user that serves as an indication of the state of the asset. This metric can then be trended and interpreted to determine whether asset maintenance is required by analysing the deviation relative to the healthy state. This study will focus on the transformation of the data prior to the model seeing this data and on all aspects of a latent variable model that can provide a fault severity indicator. In Figure 1.1, a high level overview of this process is given, whereby a signal undergoes some transformation to obtain an output metric that can be trended through time to detect deviations from the healthy state of the asset.

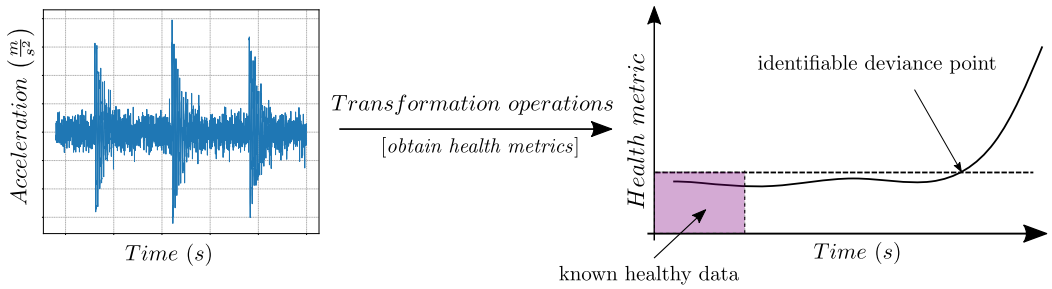


Figure 1.1. An illustration of the process followed in latent variable model-based anomaly detection. A time-series signal undergoes a series of transformations to produce a health metric that is trended and then interpreted to determine if damage has occurred or detectable.

In this study, the techniques used operate by using the temporal coherence in time-series data to capture a state of the system. This state, as highlighted in Figure 1.1, is typically identified as the system in a healthy condition and we use metrics to measure the deviation from this state. This work refers to this as an *unsupervised* approach, to ensure that there is consistency between the literature for latent variable models. However, it is clear that some knowledge of the data is exploited and this exploitation allows for deviation identification to occur.

1.2 The Nature of Gearbox Faults

Vibration data obtained from gearbox applications consists of discrete-time waveform recordings from a measurement device such as an accelerometer. This data may then contain covariates that are indicative of damage. However, it is often non-trivial to extract this information. In this work, the nature of faults in gearbox applications will be discussed as it is important that the reader understand how the presence of different faults manifests in the vibration waveform.

1.2.1 Gear Faults

Gear vibration, when measured, is known to be the prominent source within a vibration signal and has a strong deterministic component. This is due to the interaction of multiple gear teeth within one revolution of the machine. A gearbox set-up, in its most simplified form, consists of two gears that transmit torque and can vary the speed of the output shaft proportional to the input (Shigley and Mischke, 2005). The relationship between the input and output shaft speed, which are sometimes referred to as the gear frequencies, for a parallel shaft gearbox is

$$\frac{f_{s_j}}{f_{s_i}} = \frac{N_{t_i}}{N_{t_j}} \tag{1.1}$$

where N_t and f_s is the number of teeth and the gear shaft frequency on gears i and j respectively, otherwise identified as the shaft speed. It is obvious to note that each gear will have a corresponding shaft frequency (Sharma and Parey, 2016). Due to the dynamics of gear interactions, deterministic phenomena is generated that occurs periodically with the shaft speed due to the interactions between the gear and pinion teeth. The frequency of this interaction present in a gearbox is known as the gear mesh frequency, which is given by

$$f_m = \frac{N_{t_i} f_{s_i}}{60}, \tag{1.2}$$

where this form for the mesh frequency assumes that the gear shaft frequency (f_s) has units of revolutions per minute. One can also note that a gear mesh frequency is shared between interacting gears. In a complicated gearbox, i.e. one that consists of many gears and gear ratios, it is clear that there will be many gear mesh frequencies present. Due to natural eccentricities within a gearbox, modulation occurs, and a gearbox will always have side-bands on the frequency spectrum, around the mesh frequency. Gears also typically have integer harmonics in the frequency spectrum due to the periodic nature in which they operate. Wear inside a gearbox manifests as changes in the meshing frequency energy (Martin, 1987).

Three typical fault types can occur in a gearbox, namely a local fault, a distributed fault and a deflection fault. Local faults, such as a single tooth fault, will typically not manifest around the meshing frequency as this fault acts like an impulse within the system. This response typically manifests in low-level side-bands in the frequency spectrum of a signal. A distributed fault, such as multiple defective teeth, manifests in the form of high amplitude side-bands (Martin, 1987). Gearbox faults such as pitting, scoring and tooth spall are alternative faults that can also develop during the operational life-span of a gearbox (Sharma and Parey, 2016).

1.2.2 Bearing Faults

Bearings, a rotating element often used in gearboxes, is considered a vital component in the effort to produce rotation. Bearings typically assist in load transmission between the input and output of a gearbox, by facilitating shaft rotation (Saruhan et al., 2014). Bearing defects, however, are far more prevalent within gearboxes and thus cannot be ignored when considering gearbox faults. A rolling-element bearing, one of the more common bearing types within a gearbox, typically exhibit faults that manifest as a single point defect, a multiple point defect or a distributed fault. Single point faults give rise to predictable fault frequencies for specific components, namely, the inner race, outer race, rolling element or cage of a bearing. These faults often manifest through a crack or corrosion pits within these bearing elements (Martin, 1987).

These faults all manifest in the form of impulses that are modulated due to the bearing housing. This modulation causes the vibration signal of a bearing fault to initially manifest in a resonance frequency band. At the same time, the impulse periodicity depends on the mode of application of the bearing.

For example, consider the case where the outer race is stationary. Should the fault be a crack in the outer race, periodic impulses can be expected, provided the operation is at a constant speed. However, if the fault is on the inner race, the fault will only be detected when the race moves through the loading zone of the bearing (McInerny and Dai, 2003).

Bearing faults have been identified to occur at different characteristic fault frequencies, assuming a stationary outer race of the bearing which is a reasonable assumption for most applications. There are four frequencies applicable to angular contact bearings. They are *i)* the Ball Pass Frequency on the Outer race (BPFO), *ii)* Ball Pass Frequency on Inner race (BPFI), *iii)* Ball Spin Frequency (BSF) and *iv)* the Bearing Cage Frequency (BCF). The latter is often referred to as the Fundamental Train Frequency (FTF). All of these frequencies are a function of the shaft frequency, which is denoted as f_s and given in Hz, and can be presented as

$$BPFO = f_s \frac{N}{2}\left(1 - \frac{d}{D}\cos(\phi)\right), \tag{1.3}$$

$$BPFI = f_s \frac{N}{2}\left(1 + \frac{d}{D}\cos(\phi)\right), \tag{1.4}$$

$$BSF = f_s \frac{D}{2d}\left(1 - \left[\frac{d}{D}\cos(\phi)\right]^2\right), \tag{1.5}$$

$$BCF = f_s \frac{1}{2}\left(1 - \frac{d}{D}\cos(\phi)\right), \tag{1.6}$$

where D is the pitch diameter, d is the roller element diameter, ϕ is the contact angle and N is the number of roller elements within the bearing.

1.2.3 Operating Condition Problem

It is noticeable that the gear and bearing fault frequencies are all inherently functions of the input shaft speed. It is also known that variations in operating condition can modulate the amplitude of vibration signals (Stander and Heyns, 2006, Schmidt and Heyns, 2020). Thus, when operating under time-varying conditions, it is difficult to identify whether there has been any form of fault development as the fault frequency varies proportionally to the operating conditions. This problem has created a juxtaposition in research, as the operating condition type limits many techniques detailed in the literature. Thus, there has been a shift in the literature to try and address the time-varying speed problem (Abboud et al., 2017). For the remainder of this review, it shall be emphasised, where applicable, to what extent the current research addresses the variational speed problem.

1.2.4 Transmission Path Effects

Due to manufacturing restrictions, it may be difficult for an accelerometer to be placed directly on or near the fault. Hence, accelerometers are often attached to the surface of the set-up housing. This, however, induces a natural transfer function between the excitation source and the measurement device, where this transfer function is a result of the transmission path between the two entities. The transmission path can affect both the amplitude and phase of the vibration waveform, which complicates the fault inference procedure (Stander and Heyns, 2006, Borghesani et al., 2012).

1.2.5 Fault Occurrence

It is clear that for bearing and gear faults, a frequency component is often present in the time waveform that indicates the presence of a fault. In the work of Antoni (2009), an apparent reference is made to the covariates indicative of damage in the waveform, with gear and bearing faults manifesting in different components of the vibration signal. For faults that manifest as impulses in the signal, the presence of a fault frequency indicates that the impulse itself is only detectable in the time waveform with some periodicity. This implies that if one observes segments of a vibration signal, segments

may exist that are representative of a healthy signal as they may contain no-fault information. This is, however, subject to numerous assumptions, and these assumptions are addressed in Section 3.3 as the reader must be aware of the potential effect of the rate of occurrence of fault components in time waveforms.

1.3 Related Work

There are two main objectives that are key to effective PHM, namely diagnostics and prognostics (Lee et al., 2014). In fault diagnostics, the goal is to be capable of performing fault detection, fault isolation and fault severity while for prognostics the goal is to perform asset health assessment to detect emerging failure and predict the asset RUL (Lee et al., 2014, Gao et al., 2015). Fault detection entails that one detects the presence of a fault, regardless of the fault type. The requirement is that this detection is made early into fault incipience, as typically one would prefer to detect the fault as early as possible. Fault detection in PHM is based on the notion that a gearbox, in its healthy state, has a standard condition that is given by characteristic properties, where these properties are contained in the vibration signal (Gao et al., 2015). Thus, one aims to detect a fault when a deviation in behaviour is detectable.

Fault isolation is the process of determining which component is responsible for the detected fault and to determine what type of fault is present. Here the aim is to not only infer the fault type but to ensure that the fault severity is known. Fault severity then refers to the severity of the fault and is often defined relative to a known baseline state. This baseline state is given through a degradation metric or health indicator (HI) and can be trended over the lifetime of an asset to infer the fault severity (Gao et al., 2015). In typical applications of PHM, there are three levels of reasoning for fault diagnosis and prognosis. Fault detection applications are only concerned with detecting and trending the severity of a fault, without identifying the type of fault. Fault isolation only classifies a fault without considering the severity of the fault. The third application is that of RUL prediction, often captured by prognosis techniques (Lee et al., 2014). Gearbox fault analysis is imperative to ensure efficiency in industrial asset reliability. Gearbox fault analysis contains two regions of interest, namely gear analysis and bearing analysis as these elements are the more likely to develop faults (Sheng, 2016).

1.3.1 Signal Processing Approaches

Signal processing techniques are prevalent in literature, with a large body of work directed towards gearbox fault problems. A brief overview of some of these techniques relevant to this study is documented.

1.3.1.1 Traditional Approaches

Time-Statistical Approaches: the Root-Mean-Square (RMS) is a feature that is typical within signal processing environments. The RMS of a signal, otherwise known as the quadratic mean, is thought to be a representation of the energy within a signal $x(t)$, whereby

$$RMS = \sqrt{\frac{1}{T}\int_0^T x(t)^2 dt}. \tag{1.7}$$

An alternative metric is the crest factor (CF) of a signal. The CF measures signal impulsivity. The CF is the ratio of the maximum value of a zero-mean signal and the signal RMS given as

$$CF = \frac{||x(t)||_\infty}{RMS}, \tag{1.8}$$

where $||\cdot||_\infty$ is the $L-infinity$ norm. Sait (2011) noted that normal, stationary operating conditions result in a crest factor between 2 and 6. There are other types of statistical features that one can also use, such as the peak-to-peak value and peak value. Next, since the assumption here is that the signal

is stationary, one can calculate the well known statistical moments of the signal, namely, the mean, standard deviation, skewness and kurtosis (Sait, 2011). Skewness is known to be a measure of the symmetry of the distribution. For example, if a signal has an equivalent count of small and large amplitude components, the skewness should be approximately zero. The kurtosis of a signal is a measure of the variation of the signal from a typical Gaussian signal. Thus, any kurtosis greater than 3 indicates that the signal is not Gaussian-like (Zhu et al., 2014, Večeř et al., 2005). Impulsive signals lack Gaussian-like characteristics, which are then amplified in the kurtosis measure. The kurtosis of a time-series signal is

$$Kurtosis = \frac{1}{\sigma^4 T}\int_0^T x(t)^4 dt, \tag{1.9}$$

where σ^2 is the signal variance. Večeř et al. (2005) applied standard time-domain statistical features to a gearbox under then notion that the distribution for the time series amplitude without gear mesh frequencies was inherently different from that of a Gaussian signal as wear became prevalent. In their work, many standard approaches were compared for detection and severity forecasting in the presence of gear wear.

However, most of the proposed statistical features thus far are highly sensitive to fluctuating operating conditions (Zimroz et al., 2014). It is not to say that one cannot apply traditional approaches to varying operation conditions. Zimroz et al. (2014) attempted to perform bearing detection and severity trending using the regression parameters of a linear regression through standard statistical features versus motor power draw. The statistical features included RMS and peak-to-peak values. Zimroz et al. (2014) found that by using both statistical features and operating condition data, the regression parameters proved fruitful in bearing fault detection and severity forecasting.

1.3.1.2 Advanced Approaches

Frequency Analysis Approaches are techniques that utilise a transformation of the signal from the time domain to the frequency domain. Conventional frequency domain approaches typically used in practice is that of Fourier spectral analysis amplitude threshold detection. However, this technique is outdated and advanced techniques have since replaced it, with significant advances coming from the proposition of cyclostationarity (Antoni, 2009). Cyclostationarity is the notion that a signal has statistical features that are periodic in time and that this periodicity manifests within the energy flow of the signal (Gardner et al., 2006). Cyclostationarity is specifically relevant to gear and bearing applications, as the repetitive nature of their operation introduces periodic energy releases.

More formally, a random signal in the time domain is said to be cyclostationary at the n^{th} order if the n^{th} order statistical cumulant is a periodic function of time. A first-order cyclostationarity signal is one with a periodic mean, which implies that the mean is not ergodic but rather cyclo-ergodic (Capdessus et al., 2000). Numerous techniques have been suggested to estimate the periodicity of the signal means. A common technique is to use Time Synchronous Averaging (TSA) where the TSA can be defined as

$$\bar{x}_{TSA}(n) = \frac{1}{N_r}\sum_{i=0}^{N_r-1} x[n+iN_s], \text{ where } 1 \leq n \leq N_s, \tag{1.10}$$

where the signal $x(n)$ is processed to contain N_r rotations consisting of N_s points per rotation (Capdessus et al., 2000, Abboud et al., 2017, Schmidt et al., 2018). It is interesting to note that Antoni (2009) identified that gear components show strong first-order cyclostationarity, whereas bearing faults are second-order cyclostationary. The Squared Envelope Spectrum (SES) was found to be sensitive to the presence of second-order cyclostationarity within a time-series signal, which indicates that it is a powerful technique to use for bearing fault detection. It is possible to split the first-order and second-order components of a signal, which then allows one to decompose and analyse these components of a

vibration signal separately (Borghesani et al., 2012). This is commonly known as deterministic/random separation in literature, and there are numerous techniques to ensure that sufficient separation has occurred.

For gearbox fault detection, typically the first operation is to separate the gear and bearing information by performing deterministic-random separation, sometimes referred to as discrete-random separation. One can then perform health analyses on the deterministic and on the residual parts, where gear health information is often located in the former and bearing health information in the latter (Borghesani et al., 2012). In some applications, the TSA of a signal is used to determine the deterministic part, whereby the signal is segmented and averaged over one shaft revolution. One can then subtract the TSA from the original signal, which gives a residual signal. However, the inherent assumption in traditional techniques is that the system has constant operating conditions. In the case of varying operating conditions, typically one needs to perform order tracking to change the signal from the time domain to the order domain.

Order tracking is the process of re-sampling a vibration signal at a rate proportional to the speed of the rotating machine. This re-sampling is done at constant angular increments and is implemented with either analogue instruments known as synchronous approaches, or digitally as a post-operation, referred to as asynchronous techniques. The difference between the two is that the former dynamically adjusts the sampling rate based on the speed of the shaft, and the latter uses a tachometer signal to re-sample a signal in an off-line setting. The asynchronous approach is known as Computed Order Tracking (COT) (Munck and Fyfe, 1991, Fyfe and Munck, 1997). After order tracking, one can evaluate the signal using the Fourier Transform, where the spectrum is referred to as an order spectrum as opposed to a frequency spectrum. This classification is because the signal is a function of the shaft speed and its multiples rather than the base sampling frequency. Borghesani et al. (2012) investigated gearbox fault diagnostics under time-varying operation conditions. Order tracking was used as a pre-processing strategy to overcome the operating condition problem. However, other pre-processing approaches were used, such as the phase domain averaging approach proposed by Stander and Heyns (2006) or the improved synchronous average proposed by Coats et al. (2009).

After a signal is decomposed into its deterministic and random components, enhancement techniques are then often used to enhance the source of interest in the decomposed elements. Abboud et al. (2017) stated that this is often achieved by filtering the decomposed signal around a high-energy frequency band. Methods to determine the band of interest are well detailed in the literature. Randall and Antoni (2011) emphasised the usage of the Kurtogram or a wavelet de-noising analysis scheme. The Kurtogram is a powerful technique developed to detect and localise impulse events within a signal that manifest in a specific frequency band. The Kurtogram is an implementation of the spectral kurtosis, which is a function of frequency and frequency resolution (Sawalhi, 2004). Many alternatives to the Kurtogram have been proposed in the literature. Detailed works include Antoni (2016), Tse and Wang (2013a,b), Barszcz and Jabłoński (2011), Lei et al. (2011), Wang et al. (2013) and Niehaus et al. (2020).

A popular signal processing technique is that of the squared envelope spectrum (SES), which is a frequency domain transform of the Squared Envelope of a signal. The SES is given by

$$SES(\alpha) = |DTFT_{n\to\alpha}\left\{|A\left\{x[n]\right\}|^2\right\}|, \tag{1.11}$$

where $DFTF$ is the discrete-time Fourier Transform, $A\left\{\cdot\right\}$ is the complex analytic signal obtained through the Hilbert transform of the original signal and α is the cyclic frequency variable given in Hz

(Randall and Antoni, 2011). The SES is known to be a powerful second-order cyclostationary analysis technique and is well used in literature. However, it is often noted in the literature that it is beneficial to perform some form of signal pre-processing before looking at the SES, with a technique called cepstrum pre-whitening (CPW). CPW often used due to its simplicity in implementation (Borghesani et al., 2013). The implementation procedure of CPW is given as

$$x_{cpw} = IFT\left\{\frac{FT(x)}{|FT(x)|}\right\}, \tag{1.12}$$

where FT and IFT denote the use of the Fourier transform and the inverse Fourier transform. One can then analyse the SES of the CPW signal to see what fault information lies around the fault frequencies of interest.

For signal processing, specific techniques are considered to be the current state-of-the-art (SOTA). The first SOTA technique is the minimum-entropy-deconvolution (MED) spectral kurtosis (SK) normalised squared-magnitude of the squared envelope spectrum (NES), given as a full acronym as MED-SK-NES. The work of Abboud et al. (2019) provides an in-depth analysis of the MED-SK-NES process. The MED-SK-NES technique operates by first using MED and SK filtering to highlight the impulsive fault components in a signal as a form of signal pre-processing and then using the NES to analyse the amplitude of specific frequency components. In the work of Abboud et al. (2019), the performance of MED-SK-NES and the Improved Envelope Spectrum (IES) is compared for three datasets, with the results showing that the methods were able to detect bearing faults in both stationary and non-stationary operating condition cases. For a detailed analysis, implementation guide and discussion of the MED-SK-NES process, please see Appendix D.

A recent focus in the signal processing literature is to develop a statistical analysis framework that can quantify the impulsiveness and cyclostationarity of time-series data. Antoni and Borghesani (2019) proposed a set of condition indicators that track cyclostationary and non-Gaussian components independently. This approach uses a null hypothesis differentiation between healthy and abnormal states to design indicators using the logarithm of the generalised likelihood ratio between the null and alternative hypothesis. A set of indicators is obtained by varying the assumptions given to the null and alternative hypotheses where these assumptions explore cyclostationary signal components under Gaussian and non-Gaussian conditions, generalised Gaussianity against Gaussian cyclostationarity, unknown cyclic periods, non-Gaussianity against Gaussianity and impulses in Bernoulli-Gauss cyclostationary cases. This exploration allows for an interpretation of the anomalous state information under a deviation from stationarity and/or signal Gaussianity. Wang and Tsui (2018) proposed a bearing health indicator that is dimensionless and thus is not affected by variations in the asset operating condition. This health indicator is also formulated with analytical upper and lower bounds which allows for the indicator to be evaluated with respect to known bounds which has benefits in bearing prognosis applications. The upper bound of this indicator corresponds to the healthy bearing condition and the lower bound corresponds to a failure condition.

1.3.1.3 Discrepancy Analysis

Discrepancy analysis is a well researched and used technique in vibration-based condition monitoring, with many applications in gear diagnostics and rolling element bearing diagnostics. Discrepancy analysis is fundamentally akin to residual signal analysis (RES), which is a methodology that seeks to remove any healthy components from data by estimating the first order cyclo-stationary components. RES has strong links to deterministic-random separation, with the works of Abboud et al. (2017), Randall et al. (2011) and Randall and Antoni (2011) offering a concise introduction to the topic. Discrepancy analysis is the generalised technique of comparing a signal from some system with a

model of the healthy condition of the system. In discrepancy analysis, a discrepancy signal transform, otherwise known as a discrepancy measure, is obtained for signal segments or features thereof and this discrepancy measure serves as a measure of the deviation from the model of the healthy data (Schmidt et al., 2019a, Heyns et al., 2012d).

In the work of Heyns et al. (2012d), gearbox fault diagnostics were investigated using discrepancy analysis. The process followed was first to utilise Computed Order Tracking with a windowed re-sampling scheme, the window length was carefully chosen based on the relationship between the shaft speed and the gear mesh frequency. The authors used inter-window interpolation to emphasise the signal edges. From this, a Gaussian Mixture Model (GMM) was fit to the healthy training data partitioned using the windowed re-sampling scheme. The GMM then served as the healthy model, and the Negative Log-Likelihood (NLL) served as the discrepancy measure for any observed windowed sample. By considering all of the discrepancy measures for any given signal, a discrepancy signal was obtained. As multiple shaft rotations are present in the discrepancy signal, the synchronous average was computed and, combined with an order-domain analysis. Their work showed a capacity for gearbox diagnostic analysis under non-stationary load conditions. Heyns et al. (2012c) compared Auto-Regressive (AR) and neural network (NN) based time-series regression to compute a discrepancy signal residual between the predicted and actual time-series values. The discrepancy signal was then re-sampled using the instantaneous shaft angular speed, and the residual signal envelope was computed, deemed the discrepancy transform. The spectrum and cepstrum of the residual signal envelope were analysed for the case of a gear fault under time-varying operating conditions.

Schmidt et al. (2019a) proposed a discrepancy analysis approach that operates by using the Wavelet Packet Transform (WPT) and Order tracking to extract features from the *RMS* of windowed wavelet coefficients from 2^N independent Wavelet Coefficient signals. The healthy data in this case then became an $\mathbb{R}^{16}$ space for each of the wavelet coefficient signals, in which the window *RMS* was considered for variations in one wavelet coefficient signal space. After this point, a multivariate Gaussian distribution was fit to wavelet coefficient *RMS* space with the Mahalanobis distance used as a discrepancy measure. A discrepancy signal then came from measuring the discrepancy metric for the *RMS* of each windowed segment through all wavelet coefficients. To account for the inherent *RMS* dependencies on the shaft speed, the calculation of a discrepancy measure was standardised based on training data and its dependence on shaft speed. It was shown that the methodology could detect faults under non-stationary operating conditions as well as to detect faults in the spectrum of the discrepancy signals.

The main contribution of this work is to show that discrepancy analysis is essential to be applied to unsupervised deep learning approaches to gain additional insight and to draw improved conclusions. In unsupervised learning methods applied to this work, a HI is applied to time-series data fed through a latent-variable model. A HI is synonymous with a discrepancy measure, and one can draw parallels in the analysis of discrepancy signals for a complete time-series signal. Fink et al. (2020) referred to signal-reconstruction based unsupervised learning as *residual-based approaches*, an apparent reference to residual analysis and by extension, discrepancy analysis.

1.3.2 Learning Approaches

The goal of learning-based approaches is to use data to understand the patterns, trends and observations in the data for understanding and prediction. Learning approaches is an overarching term given to supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. A natural progression of complexity is present in learning approaches, that stems from statistical learning

to machine learning and finally to deep learning. To initialise this discussion, the various forms of learning-based approaches will be broadly defined and discussed.

1.3.2.1 Learning-based approaches

Supervised learning is a learning-based approach that assumes access to observed samples $\mathbf{x}$ and target labels $\mathbf{t}$. The form of the target variable dictates the learning task applied to the data, where discrete variables $\mathbf{t} \in \mathbb{Z}$ are used for classification and continuous variables $\mathbf{t} \in \mathbb{R}$ are used for regression tasks. The goal of a supervised learning task is learn a discriminative function to capture the conditional distribution $p(\mathbf{t}|\mathbf{x})$ by maximising the likelihood. This discriminative function is also required to generalise to the entire input space $\mathbb{R}^n$ such that predictions can be made for both observed and new data.

Unsupervised learning is a class of machine learning that does not use target labels $\mathbf{t}$ and only the observed samples $\mathbf{x}$. With these samples, unsupervised learning focuses on three main aspects, namely clustering, density estimation or data visualisation. Unsupervised clustering is where the goal is to discover groups of similar examples in data, density estimation refers to methods that aim to determine or model the input space data distribution and unsupervised visualisation refers to techniques that project high dimensional data in a $\mathbb{R}^2$ or $\mathbb{R}^3$ space (Bishop, 2006).

Semi-supervised learning is an approach that lies between supervised and unsupervised learning. This approach assumes that there are observed samples $\mathbf{x}$ and some of these samples have observed target variables $\mathbf{t}$. Two forms of semi-supervised learning are investigated in the literature, referred to as inductive or transductive semi-supervised learning (Kingma et al., 2014). The former is another term given to supervised learning techniques while the latter seeks to learn from the observed data to predict only on test data (Chapelle et al., 2006). The difference seems ambiguous but is better understood by relating inductive learning to learning for the entire input space while transductive learning is only concerned with unlabelled data only. In a semi-supervised setting, it is common to allow for the generative function applied to $p(\mathbf{x})$ or $p(\mathbf{t}, \mathbf{x})$ to share parameters with the discriminative model used to capture $p(\mathbf{t}|\mathbf{x})$. In this process, the supervised learning scheme is used in conjunction with the unsupervised learning scheme and a nature trade-off develops between maximising the supervised conditional log-likelihood $\log p(\mathbf{t}|\mathbf{x})$ and the unsupervised density estimation log-likelihood $\log p(\mathbf{x})$ or $\log p(\mathbf{t}, \mathbf{x})$ (Goodfellow et al., 2017). If this trade-off is then controlled to allow for sufficient flexibility from both frameworks, a model can be obtained that may hinge off the benefits of both frameworks.

The importance of learning-based approaches is the ability to access sources of information that can be used to infer the state of a system. These representative metrics are common to learning-based approaches. Many learning approaches used in PHM utilise classification or regression information to determine the type of fault or predict the RUL of an asset, while other methods use reconstruction loss information as a damage measure. These methods are viable techniques that can be applied, but the important factor reduces to the cost required to produce the relevant data and data labels. For time-series data, vibration samples are easy to obtain but the issue of data labelling may be problematic. In this work, the problem of data labels is negated by only using data from an asset in a healthy state. From this, we use metrics available from the models considered to infer damage. This is an important difference as a large portion of learning-based literature is devoted to classification or regression performance. Lei et al. (2020b) identifies two issues with current approaches, the first is that assets often operate for long periods of time in a healthy state and the time spent in an unhealthy state is often low. The second is the data labelling problem. It this work, healthy system data is used with

techniques from learning-based approaches to produce metrics that arise from an unsupervised learning framework, thereby negating the need to labelled fault data.

1.3.2.2 Statistical Learning

Statistical learning is an overarching term given to the objective of determining a function that can relate some inputs to some outputs. The initial applications of statistical learning are considered the building blocks for more complex techniques used in machine learning such as neural networks (Bishop, 2006, Hastie et al., 2009). Common statistical learning tasks are supervised regression and classification and in a statistical learning framework, linear models are used to predict outputs to data given inputs. Statistical learning is the cornerstone of learning-based approaches and the basic elements of regression and classification are seen as the foundational work for machine learning and deep learning. The difference is that statistical learning is concerned with how to handle to data to predict outputs while the other learning types are concerned with maximising performance.

Statistical learning techniques have been used for vibration data and have been integrated into many facets of signal processing techniques. For example, Zimroz et al. (2014) used a regression analysis framework to detect deviations in wind turbine data by monitoring the linear regression model coefficients. Jiang et al. (2009) compared the classification performance of linear discriminant analysis with a proposed supervised manifold learning algorithm on a variety of datasets. Schmidt and Heyns (2019) used a Gaussian mixture model to perform probabilistic condition inference by using Bayes rule to infer the class of a fault. This approach, albeit not explicitly a statistical learning approach as the expectation maximisation (EM) algorithm is used to fit the GMM model to the available class data, uses techniques from probabilistic generative models (Bishop, 2006).

1.3.2.3 Machine Learning

Machine learning is a numerical approach to statistical learning that uses neural networks to improve model performance through increased parameter flexibility and model non-linearity. Machine learning has risen in popularity over recent years due to the increase in computational power availability as well as data dimensionality and quantity (Zhao et al., 2019). The rise in data-driven health monitoring using machine learning as opposed to signal processing approaches is due to the reduced dependency on expert techniques to extract useful information from a vibration signal. There are three popular approaches to machine learning, namely: Artificial Neural Networks (ANNs), Expert Systems (ES) and Hidden Markov Models (HMMs) (Jardine et al., 2006, Liu et al., 2018b). ANNs were conceptually formulated on the notion of how the human relates an input to output (Bishop, 2006). Concerning machine health monitoring, the typical application of supervised regression and classification is for RUL and fault classification. An inherent requirement for machine learning approaches is to have input features that relate to the output. This requirement is a serious drawback of conventional machine learning approaches, as the quality of manually designed input features can be a deciding factor in the performance of an ANN and said features often require expert knowledge (Khan and Yairi, 2018).

Many machine learning techniques have been applied to PHM, such as Jiang et al. (2019a) who used a variant of typical ANNs, namely Convolutional Neural Networks (CNNs) for fault diagnosis for a wind turbine gearbox. The operation principle was to use vibration signals as input with multiple individual levels that all perform *1-dimensional* convolution on the signal. A supervised learning strategy was used to classify known wind turbine fault conditions.

Support Vector Machines (SVMs) are decision machines developed with the intent to ensure that the

objective function is convex during model optimisation, thereby ensuring that any solution found is a global optimum. This implies that the SVM weight Hessian is positive-definite for all values within the model parameter space. SVMs operate on the principle that a margin exists that is the smallest distance between the input samples to the machine and the boundary that it uses for decision support. By maximising this margin, which can be shown to be a function of support vectors (data-points from the input training set), one can perform classification and even regression (Bishop, 2006). Jedliński and Jonak (2015) used wavelet coefficients obtained from the Continuous Wavelet Transform, a method detailed in Sadowsky (1994), as input features. An SVM and a simple Multi-Layer Perceptron (MLP) were used to classify the gearbox condition in a binary case with healthy or damaged classes. The authors also compared the wavelet coefficient to vibration signal features as inputs, but this worsened the classification performance.

1.3.2.4 Deep Learning

The main problem associated with the standard machine learning approach is the requirement for manually defined features. This process can result in information loss as the time-series data is compressed into user-defined features. To circumvent this, a more recent field of interest has developed, namely that of deep learning. In a deep learning framework, the aim is to extract representations of a raw input through the use of a deep neural network. This deep neural network will typically consist of many non-linear layers, with the non-linearity coming from the activation functions. Thus, deep learning does not require manually extracted features but attempts to learn features without any guidance. In that way, deep learning seeks to replace the extensive expert knowledge of typical health monitoring methodologies with a network that learns methods for fault detection, isolation and severity trending (Zhao et al., 2019, Hoang and Kang, 2019).

Techniques utilising deep learning are on the rise, with multiple approaches applied to fault detection and isolation. Typical deep learning methods include Auto-Encoders (AEs) which were originally proposed by Rumelhart et al. (1986), Deep Belief Networks (DBN) proposed by Hinton et al. (2006), Deep Boltzmann Machines (DBM) proposed in Salakhutdinov and Hinton (2009), Recurrent neural networks (RNNs) for which Hochreiter and Schmidhuber (1997) introduced foundational work, deep convolutional neural networks (CNNs) proposed by LeCun et al. (1998), Generative Adversarial Networks (GANs) proposed by Goodfellow et al. (2014) and Variational Auto-Encoders (VAEs) introduced by Kingma and Welling (2013). Zhao et al. (2019) provide a detailed investigation for specific examples of these models applied to time-series data.

Zhang et al. (2018) proposed and implemented a Training Interference CNN that used a raw time-series signal as input with multiple convolutional layers for bearing fault classification. Due to low data samples, the author chose to perform data augmentation of a time-series signal by segmenting the signal with a certain amount of overlap between segments. The validation as to why this is consistent is that vibration measurement is typically conducted at very high frequency, and thus the dimensionality of a single sample is very high. However, in rotating machinery, typically a single record has multiple revolutions and thus by performing a form of manual convolution, one can increase the number of training samples. The authors used the Case Western Reserve Bearing Dataset in a supervised classification setting, as this dataset contains many bearing faults. It was found that the Training Interference CNN outperformed typical machine learning and deep learning techniques.

San Martin et al. (2019) proposed the usage of *VAE*s as a form of unsupervised dimensionality reduction for bearing fault diagnosis in a supervised classification setting. In their work, a comparison of raw signal input, spectrogram input, and manually extracted feature inputs was done, which aimed

at highlighting the potential benefits and pitfalls of supervised fault classification using different input data formats. Multiple variations of the *VAE*s were tested and compared with the well-known dimensionality reduction technique, that of Principal Component Analysis (*PCA*). For bearing fault classification, the authors found that *PCA* and the *VAE*s had similar performance, even with extensive studies into the *VAE* form. The authors also noted that a spectrogram or the manual features as inputs improved performance while raw signal inputs appeared to hinder performance.

Booyse et al. (2020) used a GAN to perform fault detection and trending in an unsupervised setting. The data discriminator was used as a HI, and its fault detection performance was analysed for many datasets. It is important to note that Booyse et al. (2020) trained their models on only healthy data, which keeps to the definition of an unsupervised approach used in this work. The work of Booyse et al. (2020) was one of the first applications of unsupervised deep learning for PHM using raw vibration data. The *GAN* HI was compared to a *VAE* for datasets that varied in fault manifestation. The response to damage from the output of the data discriminator was clear, which indicates that implicit generative models such as a GAN offer significant benefits and that their use in PHM cannot be ignored. Using deep learning to capture healthy data distributions allows for machine learning to be readily implemented and applied in industrial applications.

1.4 Latent Variable Models

Latent variable models and, by extension, semi-supervised and unsupervised learning are two key topics for this work as they form the foundation of the techniques used in this work. Thus, is necessary that the reader understand its application in PHM. However, to understand the application to PHM requires that the concept of a latent variable model be understood. For the purposes of this work, the focus begins on density estimation techniques using latent variable models. This work will focus on the latent manifold of latent variable models and thus the implications of any transition functions or latent manifold information must be clear.

The objective of density estimation techniques is to model the probability distribution $p(\mathbf{x})$ using a set of samples from this distribution. However, a severe limitation exists where the data $\mathbf{x}$ may exist in a high dimensional space which makes the process of modelling this distribution complex and infeasible. To overcome this high dimensionality constraint, the assumption is made that there exists a lower-dimensional subspace on which all data points from $p(\mathbf{x})$ lie, where this assumption is often referred to as the manifold hypothesis (Fefferman et al., 2013, Goodfellow et al., 2017). The manifold hypothesis exists as a strong argument can be made that physical laws often constrain the low dimensional manifold of data and that the observed data are simply a manifestation of these laws.

The assumption made from the manifold hypothesis is that there exists an unobserved latent variable $\mathbf{z}$ that explains the relationship between the observed variables in $\mathbf{x}$. A generative view of a latent variable model is given through the conditional distribution $p(\mathbf{x}|\mathbf{z})$ with some prior $p(\mathbf{z})$ over the latent variable space. This then allows for the generation of new samples through the directed graph of sampling $\mathbf{z} \sim p(\mathbf{z})$ and then sampling a new observation $\mathbf{x}$ from the conditional distribution $p(\mathbf{x}|\mathbf{z})$. To perform density estimation, it is now required that the joint distribution $p(\mathbf{x},\mathbf{z})$ is marginalised with respect to the latent variable $\mathbf{z}$ through

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}, \tag{1.13}$$

which is intractable as an analytical solution can rarely be developed for most real-world data. Furthermore, if we attempt to use Bayes theorem to compute the posterior distribution $p(\mathbf{z}|\mathbf{x})$ to perform model inference, with the posterior distribution given through

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}. \tag{1.14}$$

The end result is an intractable solution due to the presence of $p(\mathbf{x})$ in the denominator (Bishop, 2006, Goodfellow et al., 2017). However, techniques such as Variational Inference or GANs that allow one to capture and model the generative distribution and the posterior distribution. Figure 1.2 visualises the typical process followed by latent variable models. In Figure 1.2, there are two elements, namely the input space shown in $\mathbb{R}^3$ and the latent space shown in $\mathbb{R}^2$. It is important to emphasise the presence of two functions f_ϕ and g_θ, where these functions are used to transition between the input space and the latent space through a mapping between the two spaces. The link to probabilistic approaches is also shown, as the functions are used to define the generative distribution and the posterior distribution used for model inference.

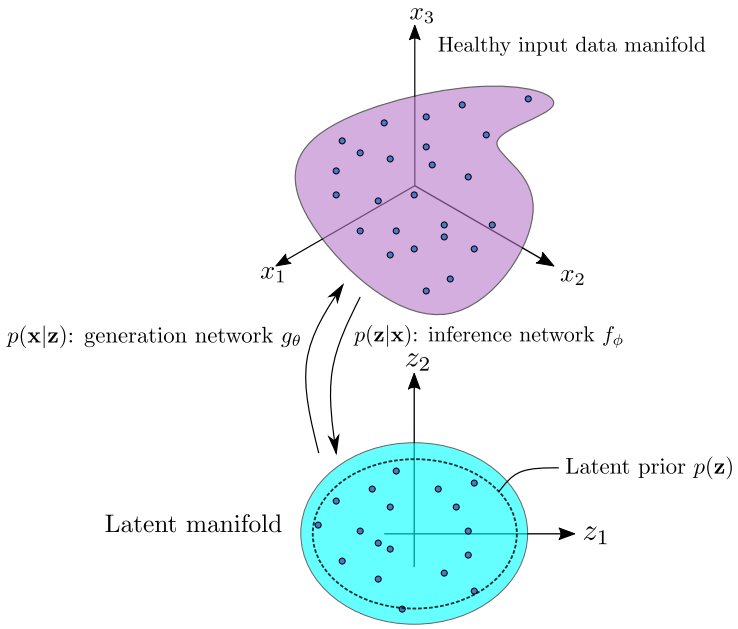


Figure 1.2. An illustration of the process of latent variable models. Notice the location of the latent prior $p(\mathbf{z})$ and the learnt latent manifold through the inference network f_ϕ.

In Figure 1.2, reference is made to the practical application of latent variable models, whereby the parametric functions f_ϕ and g_θ are used to emphasise the learnt transition between the input space and the latent space. The placement of the latent prior with respect to the latent manifold is also emphasised, to highlight how it is critical to ensure that f_ϕ transforms input feature samples in a region similar to $p(\mathbf{z})$. This ensures that the data samples can be generated through sampling $p(\mathbf{x}|\mathbf{z})$ from a sample $\mathbf{z} \sim p(\mathbf{z})$. Not all latent variable model approaches learn both parametric functions, for example, GANs typically only focus on learning a powerful parametric function g_θ. It is crucial to note that *VAEs* and *GANs* differ through their approach to density estimation, where the former is explicit while the latter is implicit. The benefit of implicit density estimation approaches is that they allow for increased transition function flexibility as they are not constrained to be a specific distribution type. For an in-depth discussion in this regard, please refer to Section 2.6.1.

1.4.1 Application to Vibration Data

For the application of PHM, the objective is to develop a framework that uses healthy time-series data to capture and model the data distribution of the asset in a healthy condition. Vibration data has to be correctly processed and applied to the models, where in this work the processing step is key to obtain useful information from the latent manifold. To process vibration data, an assumption is made that any given signal of length L_s, which is often of very high dimension $L_s \gg 1$, can be segmented and broken down into segments of length L_w, where L_w is referred to as the model window length. The mathematical operation of data processing can be regarded as a non-symmetric Hankel matrix with the addition of a shift term L_{sft} that controls the overlap between adjacent windows. This is given as

$$X_{signal} = \begin{bmatrix} x(0) & x(1) & x(2) & \cdots & x(L_w) \\ x(L_{sft}) & x(L_{sft}+1) & x(L_{sft}+2) & \cdots & x(L_w+L_{sft}) \\ x(2L_{sft}) & & \cdots & & x(L_w+2L_{sft}) \\ \vdots & & \ddots & & \vdots \end{bmatrix}, \tag{1.15}$$

where X_{signal} is a matrix of size $\mathbb{R}^{\lfloor T \rfloor \times L_w}$ with $T = \frac{L_s - L_w}{L_{sft}}$ and $\lfloor \cdot \rfloor$ denotes the use of the floor operator. To obtain a Hankel matrix, as per the definition of a Hankel matrix, the shift term L_{sft} must be set to one. All signals available in a given dataset are then processed using the process followed in Equation (1.15). This then gives rise to the *temporal preservation* approach that is crucial to this work. The *continous-time* approach is a technique used in this work that aims to preserve the element of time in the model analysis metrics. By processing vibration data using Equation (1.15) with $L_{sft} = 1$, the element of time is preserved data observed by the model and thereby allows for metric responses to evolve over time as opposed to previous work, which produced metric responses which were independent of time. Figure 1.3 emphasises the difference between previous approaches and the approach proposed in this work. In Figure 1.3, the classic deep learning processing approach and the *temporal preservation* approach followed in this work are given, where the difference is slight but of utmost importance to this work.

From Figure 1.3, it can be seen that there are the two processes begin in the same way, whereby the available signals are split into a training and test set, and the signals are processed using Equation (1.15) with L_{sft} set to $0.5L_w$. Following this, the training data is further partitioned into training and validation data, and the entire dataset is normalised based on the features from the training data. It is emphasised here that the training data only consists of signals that are considered to be representative of an asset in its healthy state. Once the processing stage is completed, the parameters of the different transition functions, defined broadly with the variables θ and ϕ, are optimised. Once the optimisation procedure is complete, the difference between the standard approach and the proposed *temporal preservation* approach becomes evident, with the latter processing all of the data using $L_{sft} = 1$ for Equation (1.15) and then bypassing the model training stage. The reason for this decision is to ensure that the element of time in the vibration data is preserved, a decision that is further discussed in Section 1.5. The final step followed by both approaches is to evaluate then all data with the health indicators available to the model, where the health indicator is a degradation metric, is model specific and finally, the model health indicator performance is then assessed.

After we model or learn the distribution of an asset in a healthy condition, we can use the model to analyse whether newly observed data is healthy or unhealthy. This analysis can occur in two regions, firstly the likelihood of a new sample can be assessed given the healthy historical data and secondly the latent representation of a sample can be evaluated based on where it lies on the latent manifold. The former analysis region is well detailed in the literature, with Booyse et al. (2020) providing an

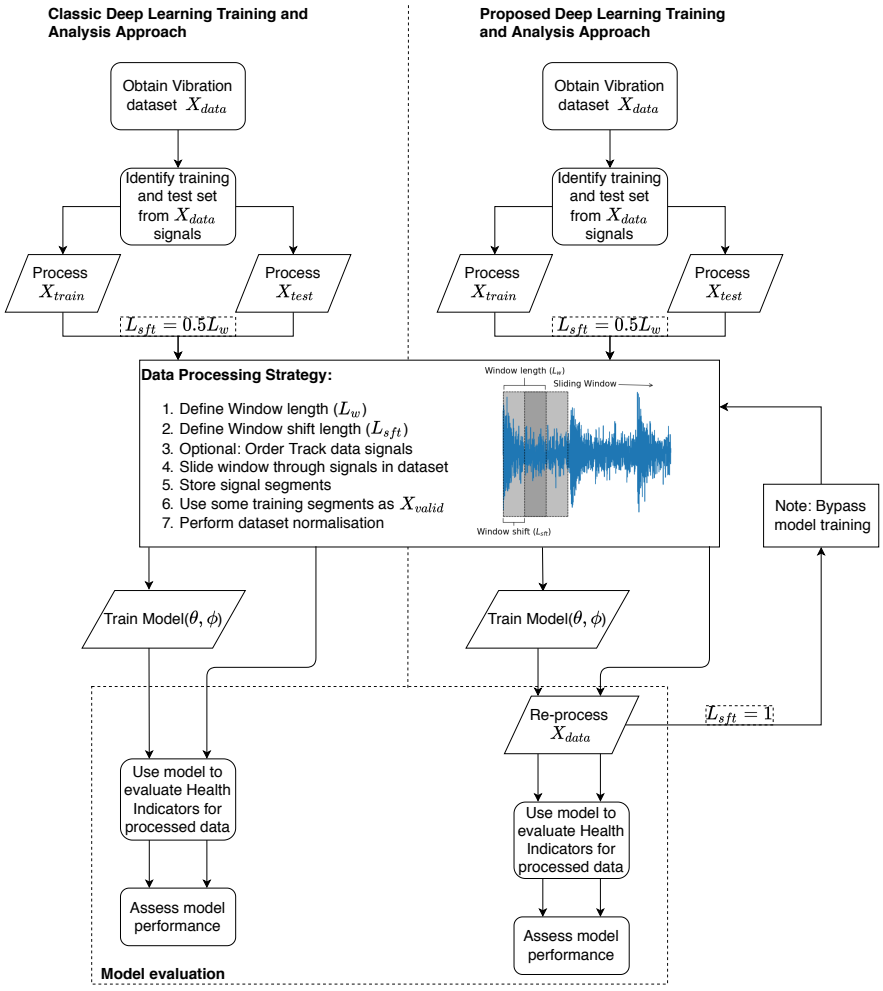


Figure 1.3. A detailed schematic of the typical deep learning model evaluation procedure versus the methodology proposed in this report. Notice the slight but key difference after model training, whereby L_{sft} is set to one, resulting in one index window shifts when processing the data for model evaluation. The effect of this change induces metric responses which evolve over time, while the classic approach produces metric responses that are independent of time.

excellent introduction and formulation of latent variable models to PHM. The latter analysis region is a key idea to this work, and thus the concept of the latent manifold will be clearly introduced to the reader.

Certain health indicators arise from the choice of latent variable model formulation. The type of health indicator is dictated by the choice of distribution parametrisation, which is an important aspect when analysing the meaning and interpretation of the health indicator. Models that assume a Gaussian distribution use the reconstruction loss, which in turn can be used as a health indicator to measure whether the model can reconstruct an input. For models that wish to use implicit density estimation techniques, the discriminator function used in the adversarial training scheme can be used to measure whether data is anomalous. However, the latent manifold is an aspect of latent variable models that can also be analysed, which requires that sensible latent metrics be formulated as degradation metrics.

In order to produce sensible latent metrics, it is imperative that the latent manifold be untangled. The implication of an entangled manifold is that damage may then be hidden in the manifold, thereby ensuring that the detection of anomalous instances is non-trivial. However, a disentangled manifold will allow for the intrinsic structure in the data to be represented in the manifold and will ensure that anomalous instances will be detectable in the manifold.

1.4.2 The Latent Manifold

The latent manifold of a latent-variable model can be seen as an embedding of the data distribution in a lower-dimensional space, where the natural constraints of physical laws govern this space. The use of the latent manifold for vibration data is evident in the dimensionality of vibration signals, which will be given in this work by L_w as any vibration signal is processed with Equation (1.15). However, this input feature space is still of high dimensionality, and the use of a lower-dimensional manifold is applicable. In the latent manifold, it is expected that the vital information that is used to capture the input data be represented and thus the latent space can be seen as a compressed representation of the input data (Jiang et al., 2009, Fefferman et al., 2013).

In the latent manifold, it is expected that input data that possess similar features be closer together, where the concept of closeness refers to the representation of the distinguishable features for similar input data be placed in close proximity on the latent manifold. This process is often referred to as Representation Learning, which entails that the latent manifold must represent the important features of the input data. In this representation, the manifold can either exist in a linear or non-linear subspace, where the difference refers to how information is compressed and makes reference to the fact that the linear manifold is represented by a linear hyperplane. The assumption of the subspace of the manifold has significant implications on the model used, as typically a parametric function is used to represent the transition from the input space to the latent space. This is also important in a probabilistic sense, whereby to perform model inference a parametric function is used to represent the parameters of the distribution $p(\mathbf{z}|\mathbf{x})$ and this function can either be formulated as a linear or non-linear function. The decision is also influenced by the data space, where a non-linear transition function is capable of capturing and transitioning between a non-linear input and latent space. In contrast, a linear transition function can only perform linear operations. If the data lies near a non-linear manifold embedded in a high-dimensional space, linear transition functions will be problematic and incapable of transitioning between $\mathbf{x}$ and $\mathbf{z}$.

To emphasise the choice of linear or non-linear parametric functions, consider the S-curve data in an $\mathbb{R}^3$ space shown in Figure 1.4. For this example, let the colour used in Figure 1.4 be a hypothetical indication of the damage present in the data. Here, the input space is non-linear, and the purpose of this example is to examine how a linear vs non-linear transition function maps this data to an $\mathbb{R}^2$ manifold. For the linear case, PCA was used as a transition function while for the non-linear case, a technique known as Isomap was used, a method detailed in Tenenbaum (2000).

In Figure 1.5, the linear and non-linear transition function results are shown, where the non-linearity of the data space can clearly be seen to affect where data lies in the latent manifold. In Figure 1.5(a), it is clear that the linear transition function cannot capture the non-linearity in the dataset and thus data that lies close to one another in the input space are not correctly placed in the latent manifold. The implication of this is that the presence of damage has not been clearly uncovered in the data. However, for the non-linear transition function, data that have similar features or spatial characteristics are represented in similar locations in the latent manifold. This is beneficial as it clearly indicates that additional non-linearity can assist in clearly uncovering the presence of the damage in the data.

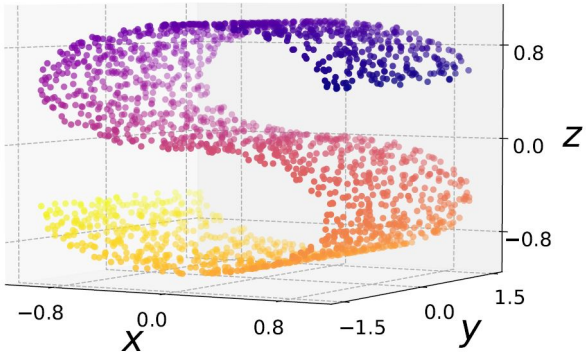
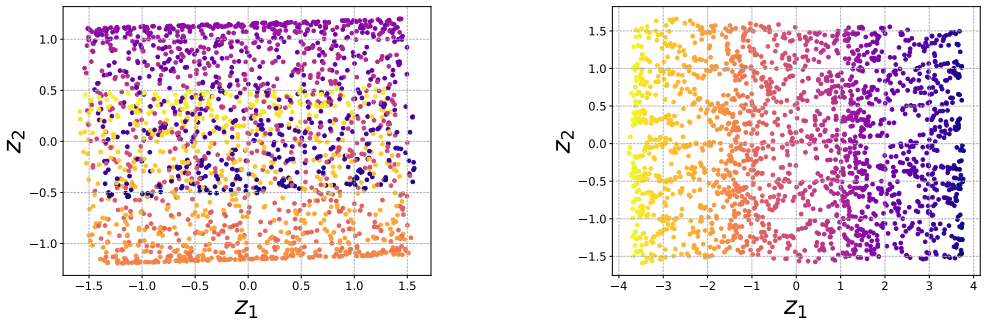


Figure 1.4. The S-curve dataset in its $\mathbb{R}^3$ data space.



(a) Linear transition function manifold.

(b) Non-linear transition function manifold.

Figure 1.5. The learnt manifold for the S-curve dataset shown in Figure 1.4 using a linear (PCA) and a non-linear (Isomap) dimensionality reduction technique for (a) and (b) respectively.

To obtain a latent representation using latent variable models, it is crucial that model inference be performed otherwise no access to the latent manifold is possible. The perspective that separates dimensionality reduction techniques and model inference are the assumptions that a probability distribution constrains the latent manifold and that samples can be drawn from this distribution. This then gives the generative ability to latent variable models, as one has access to a simple distribution to sample from when obtaining generated data samples and if one can perform model inference, the representation of input data is all constrained to lie in the same place. Techniques such as AEs do not have such an ability and the latent space from the encoder is unconstrained, and thus the user has no way of knowing where the data lies in the manifold.

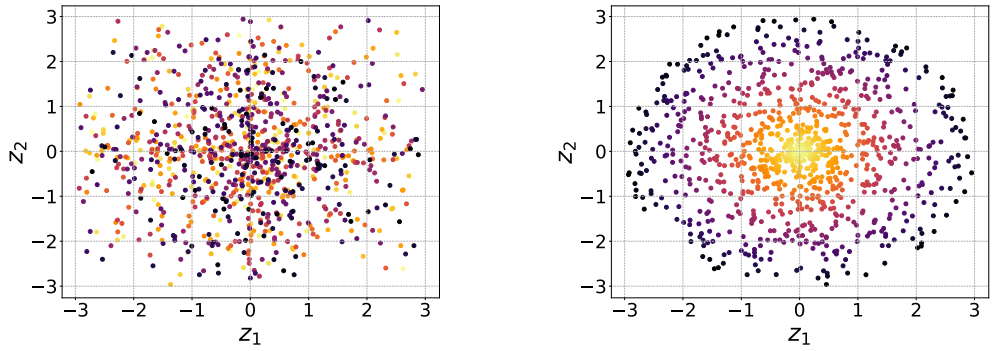
1.4.3 Latent Manifold Entanglement

Latent variable models attempt to approximate the intrinsic geometry of high dimensional data manifolds by learning low-dimensional latent-space variables and a transition function that embeds data manifolds into a latent manifold if the goal is to perform model inference. The recent focus of latent variable models is into uncovering the semantic meaning of the latent representations with an analysis of whether disentangled latent representations can be obtained (Bengio et al., 2013). A disentangled representation is one where each latent dimension captures the underlying factors of variation in the date distribution. To uncover and disentangle the intrinsic structure of the data in the latent distribution implies that the poignant information has been effectively captured during data compression and that

the latent distribution captures the explanatory sources in the data (Bengio et al., 2013).

However, to learn a disentangled latent representation is non-trivial as many alternative representations are equally viable. The representation types that can be achieved are a tangled or disentangled representation. A random or tangled latent representation implies that the latent space contains the highest entropy possible, meaning it captures the least structure and hence the least information in the latent space. The implication being that a tangled latent space cannot be effectively used as a metric for anomaly detection. It is essential that the point be made here that to have a prior $p(\mathbf{z})$ as a Gaussian does not imply that the latent space will be random, a random representation implies one where the embedding or compressing function captures no structure in the data. A tangled representation may recognise and uncover some structure in the form of factors of variation in the data, but these factors are entangled in the latent space. In a tangled latent space, it is expected that perturbing a single latent dimension will result in changes in a variety of generative data factors. A disentangled representation is one where each latent dimension captures a factor of variation in the data, and the resulting latent manifold captures the factors of variation in the data. If this is achieved, it allows for changes in these factors to be detected which is a necessity for this work. The implication of a disentangled manifold is aligned with the preservation of information, whereby the goal is to disentangle the factors of variation to ensure that the amount of information present in the data that is disregarded is as little as possible.

For this work, it is important to find untangled latent manifolds that disentangle these factors of interest. When damaged time-series data is introduced, it is undesirable that the response to damage be hidden or tangled with other potential factors of variation which highlights the importance of disentanglement. In a practical context, if there are time-varying operating conditions present alongside damage, we do not want fluctuations in the operating condition to affect the manifold response to damage. A visual illustration of manifold entanglement is given in Figure 1.6 where this figure is used to illustrate the differences between an entangled and disentangled manifold. Figure 1.6 differs from Figure 1.5, whereby the former aims to show how the data generative factors can be entangled and the latter focuses on how linear or non-linear techniques can affect the learnt manifold depending on the non-linearity present in the input space. Figure 1.6 illustrates how generative data factors, with colour given to the shaft speed of the machine, can be hidden in the manifold or can be correctly disentangled to give a latent space where the traversal along a single dimension changes a single generative factor. It is clear that in a disentangled manifold the shaft speed has been correctly disentangled and the manifold is able to capture the intrinsic factors in the data.



(a) Entangled manifold

(b) Disentangled manifold

Figure 1.6. An example of latent entanglement, with (a) showing an entangled manifold and (b) showing a disentangled manifold. Note that the colours here are representative of generative data factors such as the shaft speed.

1.5 Scope of Research

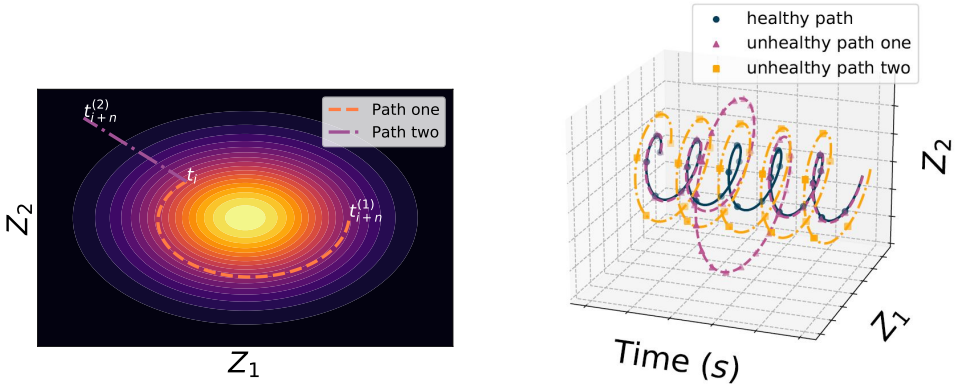
This research explores three major ideas:

1. Previous data-driven approaches for vibration-based condition monitoring overlook implicit assumptions in the model design. The nature and implications of these assumptions on the model performance are explored in detail in this work and it is shown that the effect of model window length has significant implications on result interpretation and performance. This highlights the need to sensible result analysis procedures when dealing with time-series data that extend through all aspects of a model and the results obtained.
2. One will expect different unsupervised latent variable models to perform differently with an increase in model and formulation complexity. This has however not been explored systematically. In this work such a systematic investigation is conducted. The latent variable models considered are *PCA*, *VAE*s and *GAN* models that can be used for model inference, such as the $DLS-GAN$ proposed in Ding and Luo (2019) and the model proposed in this work, the $RY-GAN$ model. The performance of the models much be analysed against signal processing for performance quantification purposes, thus the results obtained will be benchmarked against state-of-the-art signal processing techniques.
3. It is shown in this work that the latent manifold of latent-variable models is equally responsive to damage. This is a natural expectation for a manifold representative of healthy data but has this idea has not been sensibly investigated. A trivial reformulation of the model analysis procedure can introduce additional latent metrics. This observation is the crux of the present work and focuses on the traversal through the latent space and the detection of anomalous instances in the manifold. This reformulation manifests through the *temporal preservation* approach which differs from standard approaches in the preservation of time available in time-series data applications. This subtle change is shown to offer substantial improvements over the standard approach.

To understand the response of the latent manifold to damage, it is useful to consider a visual thought

experiment. Consider two parametrisations of a latent manifold learnt by a model trained on healthy data, one with and one without a dependent time variable. Figure 1.7 details this idea and can be used to assist the thought experiment. We assume that a model is trained only on healthy signals and $\mathbf{z}$ defines the latent variable space.

For the case without a dependent time variable, there are two ways in which the latent space can respond to damage, one where the model projects anomalous data far from the original healthy region and another where there are more significant changes in the distance travelled through the manifold. In Figure 1.7(a), the contour colours are that from an isotropic Gaussian distribution and represent the distribution governing healthy data in the latent space and path one and two represent two potential manifold traversal cases. A traversal through the manifold along either path from time instance t_i to t_{i+n} could either force anomalous data off the manifold or grow in path distance travelled. Both cases are equally viable, but the distinction must be made. If the vector norm of a latent instance is a damage metric, the point at $t_{i+n}^{(1)}$ will not be deemed anomalous.



(a) manifold response: type one

(b) manifold response: type two

Figure 1.7. The potential manifold response that can be obtained from latent variable models with and without the presence of time visualised explicitly. Note that in (a) the healthy data distribution in $\mathbb{R}^2$ space is indicated by colour and the contour lines present.

An even better intuitive understanding arises through Figure 1.7(b), which details the inclusion of the time component present in vibration data. It is clear that from the original healthy latent traversal, the two anomalous cases manifest differently. For growth in radius, the inter-point distance may be preserved, but the model pushes some instances far off the manifold before returning to its original path. For growth in the inter-point distance, the radius change is not significant, but the model is still indicating that this instance is unlike previously seen healthy data. These two responses may be equally indicative of damage but cannot readily be detected under the same latent HI. The purpose of this work is to show that the latent manifold is equally responsive under the right analysis formulation.

For deep learning applications to time-series data, it is common for the data to be treated in a practical machine learning context. However, it is argued in this work that this decision is unwise and can detract from the ability of the user to understand the response of the model to anomalous data. By simply adding in a second processing step as shown in Figure 1.3, one can uncover model results

by merely preserving the time element. This benefit is predominantly noticeable when analysing the latent manifold, as we can track the position and trajectory of latent instances through time, as was highlighted in Figure 1.7. This manifold tracking allows one to investigate the response of a latent variable model to anomalous instances, if the model is representative of a healthy asset. The analysis of the latent manifold augments model intuition and can introduce physical interpretation into the latent manifold and the response therein to anomalous data.

The decision to add in a processing step after a model has been trained seems trivial, however, it is an analysis step that is not considered in current deep learning practices. The benefit is realised in the analysis of the latent manifold as it introduces a sensible manner of interpreting data and the interaction between the chosen processing strategy and the model. As this work will show that the latent manifold is interpretable under the *temporal preservation* approach, it is argued that it also provides the ability to quantify the addition of model complexity, as it establishes a consistent baseline for performance. Suppose one considers all aspects of any given model, where these aspects are both the latent and input space. In that case, significant improvements can be made in quantifying whether certain assumptions or decisions increase the return on investment to the user.

Three datasets are considered and investigated in this work. Each of these datasets were selected with a specific goal in mind. These datasets were carefully chosen with a critical theme: to show the performance of latent variable models under stationary and non-stationary operating conditions for gearbox condition-based monitoring applications. The first dataset is a phenomenological model that is a synthetic dataset with constant operating conditions. This dataset was chosen as it allows for explicit control over the dataset parameters, and it allows for the simulation of both inner and outer race bearing faults. In the phenomenological model dataset, the aim is to two-fold, it aims to show the difference in model performance for the models considered, and it also illustrates the effect of the assumed model window length L_w on model result interpretation. For a further explanation in the impact of the latter, please refer to Section 3.3.

The second dataset considered is the Intelligent Maintenance Systems (IMS) dataset which was introduced in the work of Qiu et al. (2006). This dataset was investigated as it offered a wide variety of sensory data, fault cases and is a run-to-failure dataset. This dataset has bearing faults that manifest naturally under stationary operating conditions. The aim of the analysis is to show the performance ability of the proposed latent metrics on a real dataset for constant operating conditions.

The final dataset considered in this work is a gear-tooth fault dataset under time-varying operating conditions that have been extensively analysed in the works of Schmidt et al. (2018) and Schmidt and Heyns (2020). This dataset consists of two experimental datasets that are joined, where the first contains vibration data for a healthy experimental set-up and the second contains data from the set-up where a tooth fault was manually seeded and left to run until complete tooth failure occurred. The aim of this analysis is the investigate the performance of unsupervised latent variable models under time-varying operating conditions.

For the various models considered in this work, the objective is to show the effects of model complexity on the response results obtained from the model. The decision to include a model complexity investigation is to highlight the strengths and weaknesses of different models, with a focus on how the model formulation affects the results obtained. This work does not attempt to improve or formulate a

novel signal processing approach. The focus was on the application of unsupervised data-driven models to gearbox PHM in the presence of both stationary and time-varying operating conditions.

1.6 Document Overview

Chapter 2 introduces latent variable model literature and summarises the important aspects and principles thereof. The required techniques associated with model formulation, derivation and implementation are discussed. The focus is on the fundamental formulation of latent variable models in an unsupervised learning framework. The latent variable models which were considered, namely, *PCA*, *VAE* and *GAN*-based methods, are described and derived in detail. The $RY-GAN$ model is also proposed and explained to the reader. The implications and applications of these models are discussed for vibration data.

Chapter 3 presents an analysis of latent variable models in their ability to learn a latent manifold, with a focus on the flexibility of the transition functions between the input and latent space and on how the various models approach latent disentanglement. The multiple health indicators and the proposed latent health indicators available from the models used in this work are then introduced and formulated for the reader.

Chapter 4 introduces the reader to the phenomenological model dataset investigated in this work, and the properties of this model are discussed in some detail. The responses from the various health and latent health indicators are shown for the dataset and compared in detail.

Chapter 5 presents the IMS dataset and performs an analysis on three of the available bearing datasets. The ability of the various health and latent health indicators is critically discussed and compared across the various models considered. A comparison to various state-of-the-art signal processing techniques is also conducted. This was done to allow for the health indicator performance can be quantified and analysed against signal processing.

Chapter 6 presents the gear-tooth fault dataset and performs an analysis of two versions of this dataset. The difference in the datasets was introduced through a low-pass filtered version of the dataset, which was done to quantify model performance in terms of dataset complexity. The health indicator response performance is compared to various signal processing techniques. This was done to ensure that a suitable baseline is formed to ensure that the metric performance is fully quantified.

Chapter 7 presents the report conclusions and recommendations for future work. The performance of the various models, health indicators and latent health indicators is discussed and work for future investigations are proposed.

Appendix A details important machine learning and supervised learning literature that is important but not fundamental to this work. This chapter was included as supervised learning techniques provide a solid background into the probabilistic treatment of statistical learning.

Appendix B details the optimisation schemes, further details the implementation of the $\beta-TC-VAE$ method, introduces the training schemes for the $DLS-GAN$ and $RY-GAN$ models and provides important information related to the networks used in this work. The network architecture and

motivations for network design are given and the important hyper-parameters are detailed for the datasets considered.

Appendix C presents the important parameters for the phenomenological model. These parameters allow for the mathematical model to be defined such that the model is representative of a real application with reasonable characteristics.

Appendix D provided a detailed derivation and analysis of MED-SK-NES. This was done to allow for a concise analysis of the implementation aspects for the MED-SK-NES technique to provide a level of insight into using MED on time-series data.

Appendix E presents results that were deemed interesting for the IMS dataset result analysis section. These results were included to allow for a full result analysis reflection to occur.

Chapter 2 Unsupervised Learning

2.1 Chapter Abstract

The purpose of this section is to provide a literature background into unsupervised latent variable techniques and to detail the $DLS-GAN$ and $RY-GAN$ methods used in this work.

2.2 Introduction

Unsupervised learning is a variant of machine learning whereby one assumes that one does not have training target labels and thus the aim is to learn the representation of the distribution $p(\mathbf{x})$ as opposed to $p(\mathbf{t}|\mathbf{x})$. Often, in unsupervised learning, a model for this distribution utilises latent variables, which are a lower-dimensional representation of the data $\mathbf{x}$. In this way, the latent variables $\mathbf{z}$ act as a method to learn the relationships between data, with the relationship given as the model evidence $p(\mathbf{x}) = \mathbb{E}_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})$. In this section, different unsupervised learning techniques shall be presented. The discussion will consist of Principal Component Analysis (PCA), Variational Auto-Encoders (VAEs), the $\beta-TC-$VAE and important literature for Generative Adversarial Networks (*GAN*s) shall be analysed. It is important to note that the proposed *temporal preservation* approach is used to process any data that the models see. For those interested in a background to supervised learning, please refer to Appendix A.

There is a important difference between the various domains of literature that must be clarified prior to the development of the important literature for this work. The basis of vibration-based condition monitoring has strong roots in signal processing techniques. Signal processing is a powerful field directly focused on obtaining interpretable results from the covariates of a damaged signal that is indicative of damage. However, to obtain these results requires extensive domain knowledge and is built on hand-crafted features and insights into vibration data. The term hand-crafted features is used broadly here but the meaning is simple, a feature or set of features is obtained from a vibration signal where these features require domain knowledge as a prior to understand and interpret either the implementation required to obtain results or meaning of the results.

If we now venture away from signal processing, statistical learning is concerned with using data to uncover observations in the data that can be used for output prediction given an input. A statistical learning application formulation consists of a linear model and often significant data pre-processing is required for small datasets to obtain manually extracted features. The choice of model linearity arises from finding a suitable function to discover the relationship between input and output using statistical methods. In a statistical learning model, often non-linearity is included through the use of basis functions on the features, however the model itself is still a linear function of the unknown parameters used to fit the model (Bishop, 2006, Hastie et al., 2009).

Machine learning is the numerical technique that approaches some of the problems from statistical learning with increased model flexibility and non-linearity and is sometimes referred to as non-linear statistical modelling (Hastie et al., 2009). In machine learning approaches, there is still a requirement for a significant investment into data pre-processing and a need to obtain manually extracted features which requires some domain knowledge for feature significance. However, the use of a neural network allows the model to increase its flexibility as there is significantly more unknown parameters and activation functions are often used to introduce a level of non-linearity into the model. This increased flexibility is what makes neural networks attractive to supervised and unsupervised learning applications.

The caveat of machine learning is the requirement for manually extracted features that can improve the model performance. This is where deep learning has an impact as the goal is to use no data pre-processing and only increase the model flexibility to allow for an increased level of abstraction through each layer of the neural network. The dependency of increased model flexibility implies that highly flexible non-linear models must be used.

In this chapter, the important unsupervised learning techniques to this work will be discussed, where the first goal will to introduce the technique from a machine learning background. However, where necessary, a discussion of how a method is used for condition monitoring is used to highlight how the method is applicable to time-series data.

2.3 Principal Component Analysis

PCA is a linear dimensionality reduction technique that is often used as a pre-processing step in supervised learning. PCA is defined as the orthogonal projection of any input $\mathbf{x}$ into a linear latent subspace, where PCA aims to maximise the variance along the principal components (PCs) of the latent space from the projected inputs. In another manner of thinking, PCA attempts to ensure that in the transformation from the input space to the latent space, data variation is preserved. For this report, the deterministic form of PCA shall be considered, whereby one only performs deterministic movements from the latent space to the input space and vice-versa.

The formulation of PCA shall now be investigated. Consider the case where one aims to project data to a M dimensional space, where one can obtain a latent representation of $\mathbf{x} \in \mathbb{R}^D$, where D is the dimensionality of the input feature space, through

$$\mathbf{z} = \mathbf{U}^T \mathbf{x}, \tag{2.1}$$

which is a linear transformation through the transformation matrix $\mathbf{U}$. The dimensionality of $\mathbf{U}$ provides control of the size of the latent space. If one then wishes to reconstruct $\mathbf{x}$ from the latent space, this can be done using

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{z}, \tag{2.2}$$

where $\tilde{\mathbf{x}}$ is the input reconstruction. Consider now the projection to one latent dimension, a scalar projection, of a given input vector $\mathbf{x}_n$ from a training set through a unit vector $\mathbf{u}_1$, given by $z = \mathbf{u}_1^T \mathbf{x}_n$ (Bishop, 2006). As the aim is to maximise the variance of the transformed data along the principal component $\mathbf{u}_1$, the variance for the projected data can be given as

$$\sigma^2_{latent} = \frac{1}{N} \left(\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}} \right)^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1, \tag{2.3}$$

where $\overline{\mathbf{x}}$ is the mean of the training set and $\mathbf{S}$ is the data covariance matrix given as

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^T. \tag{2.4}$$

To ensure that when maximising the variance the principal component vector does not tend to infinity, a constraint is enforced that the vector must be a unit vector. The constraint is thus: $\mathbf{u}_1^T\mathbf{u}_1 = 1$ (Bishop, 2006). To re-define the objective function with this constraint into an unconstrained objective function, a Lagrange multiplier is used on the constraint leading to the objective function

$$\mathcal{L} = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 + \lambda_1(1-\mathbf{u}_1^T\mathbf{u}_1). \tag{2.5}$$

By taking the closed-form derivative of this function with respect to $\mathbf{u}_1$ and setting it equal to zero, it can be shown that

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1, \tag{2.6}$$

where if one then left multiplies by $\mathbf{u}_1^T$ we can see that

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1. \tag{2.7}$$

This indicates is that the principal component $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ and the variance will be maximum when $\mathbf{u}_1$ is the eigenvector with the largest eigenvalue. If one then wishes to find the other principal components, a constraint is given such that the new principal component is orthogonal to the former which naturally leads to the next eigenvector. The result is that the principal components are ordered and selected based on the M largest eigenvalues of $\mathbf{S}$, where this then gives the form of $\mathbf{U}$

$$\mathbf{U} = \begin{bmatrix} \uparrow & \cdots & \uparrow \\ \mathbf{u}_1 & \ddots & \mathbf{u}_M \\ \downarrow & \cdots & \downarrow \end{bmatrix},\ \mathbf{U} \in \mathbb{R}^{N\times M}. \tag{2.8}$$

For the selection of the number of principal components to use, it is common to assign a variance proportion that is kept in the dataset, which is given by the eigenvalues as these were shown to be equal to the variance of the latent space. By using the cumulative contribution rate (CCR), sometimes referred to as the cumulative percentage,

$$\frac{\sum_{i=1}^{M}\lambda_i}{\sum_{i=1}^{N}\lambda_i}, \tag{2.9}$$

where $M \leq N$ and N is the dimensionality of the latent space, $N \leq D$.One can use the first M principal components that ensure that a high percentage of the variance is preserved (Bishop, 2006). An interesting, alternative formulation of PCA is that of finding $\mathbf{U}$ using

$$\begin{aligned} &\min_{C} \|\mathbf{X}-\mathbf{X}\mathbf{U}\mathbf{U}^T\|_2^2 \\ &\text{subject to } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{aligned} \tag{2.10}$$

which is a objective function designed to minimise the least-squares projection of the dataset $\mathbf{X}$ to a linear subspace under the constraint that each direction be unit vectors and orthogonal to one another (Udell et al., 2016).

PCA is often utilised as a deterministic dimensionality reduction technique but can also be used as a generative model that uses linear transformations to the latent space as opposed to non-linear transformations as are used in the sections that follow. This approach is often referred to in literature as Probalistic PCA, but it is simply a framework that uses a linear latent variable model formulation with Gaussian distributions for the generative distribution $p(\mathbf{x}|\mathbf{z})$ and the latent prior $p(\mathbf{z})$ (Tipping and Bishop, 1999, Bishop, 2006).

The key idea for PCA as a latent variable model is that it is a model that uses linear transformations to and from the latent space. However, the use of a linear transformations may be limiting as it assumes that the data distribution is a linear Gaussian model, with the same assumption made for the generative distribution $p(\mathbf{x}|\mathbf{z})$ and the posterior distribution $p(\mathbf{z}|\mathbf{x})$. This limitation leads to the requirement for more complex latent variable model formulations, which induces complexity in formulating an objective function from the marginal distribution $p(\mathbf{x})$ as this is intractable. This intractability stems from the addition of a neural network with a non-linear hidden layer (Kingma and Welling, 2013).

The application of PCA to time-series data is not unheard of, as detailed in Zhang et al. (2020), however the application has often been used as a dimensionality reduction technique to improve classification or regression performance as opposed to viewing *PCA* as a linear latent variable model. This difference is a crucial component in this work as *PCA* can be seen as a computationally robust and efficient version of methods that use neural networks, which provides a coherent performance baseline and also introduces a model flexibility component in the analysis of vibration data. This is because PCA is a linear model but other techniques offer more flexibility.

2.4 Variational Auto-Encoders

As noted previously, the aim of unsupervised learning is to approximate the distribution $p(\mathbf{x})$ with the use of latent variables $\mathbf{z}$ with the relationship given as $p(\mathbf{x}) = \mathbb{E}_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})$. More formally, assuming that latent variables are drawn from a latent distribution, the model evidence can be given as

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \tag{2.11}$$

which is the marginalisation of the joint distribution $p(\mathbf{x}, \mathbf{z})$. However, often in deep learning, this marginalisation is intractable as this is in integration over an often high dimensional latent space. Furthermore, if one aims to perform model inference in a probabilistic setting, one would use Bayes' theorem to obtain the posterior distribution, which is given as

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}. \tag{2.12}$$

However, this is also intractable due to the intractable marginal distribution. In the work of Kingma and Welling (2013), these issues were addressed using Variational Inference (VI) techniques to formulate an objective function that can be optimised. Thus, the VI concept will be introduced after which Variational Auto-Encoders (VAEs) shall be shown. VI is a technique used to determine the posterior distribution using the approximation from a family of distributions $q(\mathbf{z}) \in \mathcal{K}$, where each distribution is considered to be a candidate for the optimal solution. To evaluate the distributions such that the optimal distribution from the family $\mathcal{K}$ can be found, the KL divergence can be used (Blei et al., 2017). Thus, the optimisation problem is

$$q^*(\mathbf{z}) = \min_{q(\mathbf{z}) \in \mathcal{K})} KL(q(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x})), \tag{2.13}$$

where the complexity of this optimisation problem is now a function of the complexity of the family of distributions chosen by the user. This then induces a trade-off as one needs to select a distribution that best suits the posterior distribution with sufficient complexity while still allowing for computational efficiency to perform the optimisation. This objective function is still, however, intractable due to the dependency on the posterior distribution. Consider for a moment the following expansion of the

natural logarithm of the marginal distribution $p(\mathbf{x})$

$$\begin{aligned}\log p(\mathbf{x}) &= \log\left(\int_{\mathbf{z}} p(\mathbf{x},\mathbf{z})d\mathbf{z}\right)\\ &= \log\left(\int_{\mathbf{z}} p(\mathbf{x},\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}\right)\\ &= \log\mathbb{E}_{q(\mathbf{z})}\left[\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right].\end{aligned} \tag{2.14}$$

Now, using an equality known as Jensen's equality (Jensen, 1906), which is formally given for concave functions as

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)], \tag{2.15}$$

the expansion can be continued as

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x},\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})], \tag{2.16}$$

where the terms on the right hand side is known as the Evidence Lower Bound (ELBO) (Blei et al., 2017). Bearing this term in mind, if one expands the term $KL(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x}))$

$$\begin{aligned}KL(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x})) &= -\mathbb{E}_{q(\mathbf{z})}\left[\log\frac{p(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}\right]\\ &= -\left(\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})]\right)\\ &= -\left(\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x},\mathbf{z})] - \log p(\mathbf{x}) - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})]\right)\\ &= -(\text{ELBO} - \log p(\mathbf{x}))\\ &= -\text{ELBO} + \log p(\mathbf{x}).\end{aligned} \tag{2.17}$$

Thus, one can see that by taking the difference between the log marginal likelihood term and the KL divergence, one can maximise the log-likelihood and minimise the KL divergence by maximising the ELBO. This allows one the opportunity to avoid the use the KL divergence minimisation for the posterior distribution and rather perform maximisation of the ELBO and in doing so, the two intractable issues previously identified can be resolved. However, the derivation of the ELBO is incomplete, as the joint distribution term still needs to be expanded. Through expansion,

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x},\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})]\\ &= \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})] + \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})]\\ &= \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL(q(\mathbf{z})\|p(\mathbf{z})).\end{aligned} \tag{2.18}$$

In the work of Kingma and Welling (2013), the decision was made to use a parametric model on the approximate distribution over the intractable posterior distribution, given as $q(\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x})$. This distribution is often referred to in the literature as the recognition or inference model. In the same manner, it is assumed that the latent variables $\mathbf{z}$ are generated from a prior distribution $p_\theta(\mathbf{z})$ and the parameters θ are considered to be the generative model parameters that are required for the latent variable prior and likelihood distributions $p_\theta(\mathbf{z})$ and $p_\theta(\mathbf{x}|\mathbf{z})$. The term probabilistic encoder refers to the approximate posterior distribution and the term probabilistic decoder refers to the generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$. Both the encoder and decoder are assumed to be parametrised by neural networks, where the encoder aims to learn the parameters of the approximate distribution by taking a given input $\mathbf{x}$ and mapping it to distribution parameters and the decoder aims to learn the mapping from the latent variable space to the original feature space $\mathbf{x}$.

The form of the probabilistic encoder is often chosen to be a multivariate isotropic Gaussian, as it then allows for a given input $\mathbf{x}$ to produce a known distribution form over the latent variables. The form

of the probabilistic decoder is often parametrised either as a Gaussian or as a Bernoulli distribution depending on the application. The probabilistic decoder then allows for a given input $\mathbf{z}$ to be mapped to a distribution for the input features $\mathbf{x}$ from which the latent variable could have been drawn. A Gaussian encoder is often given in the form

$$q(\mathbf{z}|\mathbf{x}_i) \sim \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I}), \tag{2.19}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are given by the parametric non-linear neural networks. The probabilistic decoder is given as

$$p(\mathbf{x}_i|\mathbf{z}_j) \sim \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2\mathbf{I}), \tag{2.20}$$

where again, $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are given by the parametric non-linear neural networks. The objective function minimised for a VAE is

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{x}_i) = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z}_j)\right] + KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i))\|p_\theta(\mathbf{z})\right), \tag{2.21}$$

where ϕ are the recognition model parameters and θ are the generative model parameters. The two terms in the loss function have significantly different implications in how the optimisation occurs. The first term is one that is not uncommon in Machine Learning literature but typically does not compute the integral directly and rather performs empirical sampling of the encoded latent distribution to evaluate the expectation. The second term, fortunately, has a closed-form solution that can be computed under the assumption of the form of the prior and the approximate posterior distribution. Thus, the objective function can be given as

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathbf{x}_i) = \frac{1}{L}\sum_{l=1}^{L}\left[-\log p_\theta(\mathbf{x}_i|\mathbf{z}_{j,l})\right] + KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i)\|p_\theta(\mathbf{z})\right). \tag{2.22}$$

It was noted by Kingma and Welling (2013) that often, the number of samples L required to evaluate the expectation can be set to one if the batch size is sufficiently large. From this point, the loss function dependency on ϕ and θ is dropped for brevity.

2.4.1 VAE Discussion

The approach presented thus far is somewhat complex and is better understood when relating previous literature to the VAE framework. The probabilistic encoder is a network that maps a given feature vector to an assumed distribution that covers the latent space. Typically, the latent space is incomplete by design, meaning its dimensionality is less than the input. Thus, the encoder performs probabilistic dimensional reduction. This implies is that the encoder facilitates dimensionality reduction with the added benefit of a known form of distribution that governs the latent space, which is powerful as this can allow for latent sampling. The probabilistic decoder is then a network that learns to take a given latent space vector and map it back to the original input space, where this input space is now also governed by some explicit distribution. The important element here is that often this mapping is chosen to be deterministic, as is often the case with supervised learning, but this does not have to be the case. One could train a network that can learn a stochastic distribution on the input space which then allows one to perform sampling in the reconstructed space. This is powerful for anomaly detection as one can use this space to justify how unexpected a given reconstruction is.

Figure 2.1 serves as a visual explanation of how a VAE network structure is formulated. In this figure, the networks served to parametrise Gaussian isotropic distributions. One can immediately note is how both the encoder and decoder networks learn a mean and variance vector that corresponds to the latent space and the reconstructed output. However, it is clear that if one is to sample in the latent space, one cannot easily perform gradient descent. This lead to the proposed *re-parametrisation trick* which introduces an external multivariate zero mean, unit covariance noise distribution from which one can sample and then use this sample to develop a latent variable vector (Kingma and Welling, 2013). Given

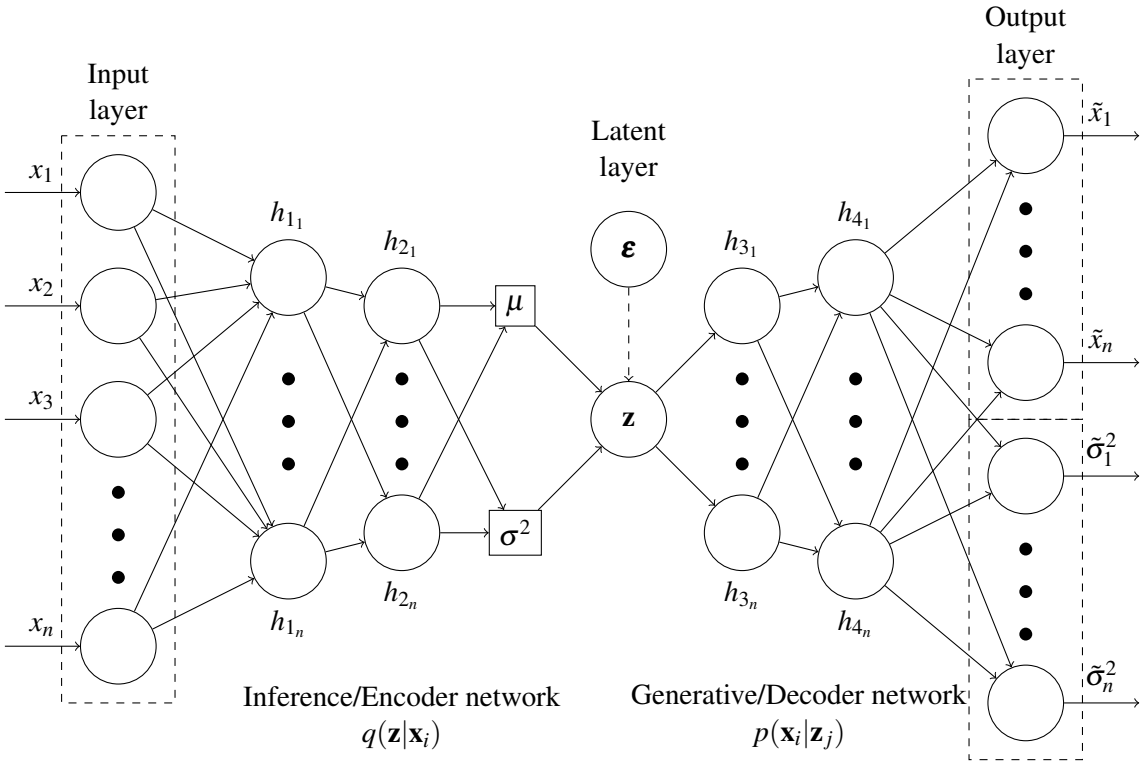


Figure 2.1. Illustration of a simplistic Variational Auto-Encoder. The notation, in this case, is circular elements denote nodes, square elements denote the latent vector distribution parameters and the dotted rectangular elements indicate the inputs and outputs of the network system.

in mathematical notation, this is

$$\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\varepsilon}_i, \tag{2.23}$$

where $\odot$ is the element-wise product operator and $\boldsymbol{\varepsilon}_i$ is the stochastic component sampled from a distribution, given as $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This process allows one to conduct back-propagation through the network structure. The aim now is to develop the necessary terms that can allow for an evaluation of the loss function. The first term in the loss function, under the assumption of a multivariate isotropic Gaussian decoder, can be expanded to be

$$\begin{aligned} \log p(\mathbf{x}_i|\mathbf{z}_j) &= -\frac{1}{2}\left(D_{input}\log(2\pi) + \log(|\boldsymbol{\sigma}_j^2\mathbf{I}|)\right) - \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^T(\boldsymbol{\sigma}_j^2\mathbf{I})^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_j) \\ &= -\frac{1}{2}\left(D_{input}\log(2\pi) + \sum_{i=1}^{D_{input}}\log(\sigma_j^2)\right) - \sum_{k=1}^{D_{input}}\frac{(x_{i,k} - \mu_{jk})^2}{2\sigma_{j,k}^2}, \end{aligned} \tag{2.24}$$

where D_{input} is the dimensionality of the input space and j refers to the elements obtained by passing an input $\mathbf{x}_i$ through the encoder and decoder $\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2 = D_\theta(E_\phi(\mathbf{x}_i))$. One can then drop the term $\frac{1}{2}D\log(2\pi)$ to give the reconstruction objective function. In this work, two forms of a VAE are considered where the main difference between the two is the decision to either assume an identity output covariance matrix and rather learn an output variance. These two forms are referred to as the deterministic (VAE_1) or stochastic (VAE_2) VAEs respectively. If the VAE_1 is used, Equation (2.24) reduces to the MSE loss function. For vibration data, the use of an output variance can aid in quantifying the extent of an anomalous instance and help emphasise anomalous instances in the data.

The second term in the loss function is the KL divergence between the approximate distribution and the

latent variable prior. This term acts as regularisation in the VAE loss function as it penalises deviance from the assumed form from the prior distribution. This then enforces that the VAE learns to map input vectors to a controlled latent space which is useful as this allows the loss function to ensure that the mapping to the latent space can be one that is informative as to what features represent a certain input when compared to the other. Under the assumption that both of these distributions are Gaussian, the KL divergence between two general multivariate Gaussian distributions, $\mathcal{N}_0 \sim (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \mathcal{N}_1 \sim (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, can be expanded to be

$$KL(\mathcal{N}_0, \mathcal{N}_1) = \frac{1}{2}\left[tr(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - D_{latent} + \log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right], \tag{2.25}$$

where $tr(\cdot)$ is the trace operator and $|\cdot|$ indicates a matrix determinant (Kingma and Welling, 2013). In this manner, it is often assumed that the latent space prior is a zero mean, isotropic unit Gaussian, which then leads to the following expansion

$$\begin{aligned} KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i))\|p_\theta(\mathbf{z})\right) &= \frac{1}{2}\left[tr(\boldsymbol{\sigma}_i^2\mathbf{I}) + \boldsymbol{\mu}_i^T\boldsymbol{\mu}_i - D_{latent} + \log\frac{1}{\prod_{k=1}^{D_{latent}}\sigma_{i,k}^2}\right] \\ &= \frac{1}{2}\sum_{k=1}^{D_{latent}}\left[\mu_{i,k}^2 + \sigma_{i,k}^2 - \log(\sigma_{i,k}^2) - 1\right], \end{aligned} \tag{2.26}$$

where i now refers to the latent sample obtained by using the re-parametrisation trick on the outputs of the encoder $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 = E_\phi(\mathbf{x_i})$. Therefore, the loss function for VAEs can be given in full as

$$\mathcal{L}(\mathbf{x}_i) = \frac{1}{2}\left(\sum_{i=1}^{D_{input}}\log(\sigma_{j,i}^2)\right) + \sum_{k=1}^{D_{input}}\frac{(x_{i,k} - \mu_{j,k})^2}{2\sigma_{j,k}^2} + \frac{1}{2}\sum_{k=1}^{D_{latent}}\left[\mu_{i,k}^2 + \sigma_{i,k}^2 - \log(\sigma_{i,k}^2) - 1\right]. \tag{2.27}$$

In practical applications, there are two conventional approaches to the modelling of the variance terms in the probabilistic encoders and decoders. These approaches try to navigate the potential issue of negative variances learnt by the network, as this is an impossible occurrence. The first approach to circumnavigate this issue is to use a linear activation function on the variance output and treat this term as a logarithmic variance, whereby one then takes the exponent of this term to treat it then as a variance. Alternatively, an approach often used in literature is to use a Softplus activation function, as this is a variant of the approach which models it as a logarithmic term. In doing this, the variance can then be treated as normal and thus requires no manipulation to transform it into the correct form. These two approaches are equivalent in how they aim to ensure positive variances in a network, however, the effect it has on the values that the variance might take is highly different. If one uses an exponential form on a linear unit the variance will change exponentially whereas the Softplus activation function ensures a linear variation for an activated neuron. The Softplus activation method can be treated as a dampened form of the exponential approach as it ensures that the variances do not grow exponentially but rather linearly. The effect of this in a VAE context has not been investigated, but it is certainly an interesting contrasting assumption that has clear implications on the performance of the model. For the interested reader, conditional VAEs are detailed in the work of Sohn et al. (2015). A conditional VAE maximises the ELBO of the conditional distribution $p(\mathbf{x}|\mathbf{y})$ where $\mathbf{y}$ are conditional variables.

In the work of An and Cho (2015), a *VAE* based anomaly detection algorithm was proposed that consisted of determining a Monte Carlo estimate for a given sample $\mathbf{x}_i$ through the reconstruction log-likelihood $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\log(p_\theta(\mathbf{x}|\mathbf{z}))$. The ability of this reconstruction probability was then compared to reconstruction errors obtained from an AE, a linear PCA model and a kernel PCA model on two datasets and it was found that the reconstruction probability was an objective and clear anomaly score in comparison to those found from the other models considered.

In the work of Matsubara et al. (2018), an unregularised score was proposed which is the same as that used by An and Cho (2015) but differs in the once-off estimation of the expectation using only the predicted mean of the latent variable from the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$. This metric was compared on a toy and manufacturing anomaly dataset, where the latter consisted of identifying anomalous crack in screw holes. Matsubara et al. (2018) found that a VAE with the unregularised score was found to work well for anomaly detection on problems of varying difficulty.

In the work of Booyse et al. (2020), *VAE*s are presented in the context of anomaly detection on vibration data. A *VAE* was trained on only healthy data and the reconstruction log-likelihood response was compared to the discriminator output from a *GAN* trained on the same data. It was found that a *VAE* could be used for anomaly detection, however the *GAN* consistently outperformed the *VAE*. Booyse et al. (2020) investigated a variety of synthetic and experimental datasets and found that a *VAE* and a *GAN* was able to capture the manifold of healthy asset data and detect anomalous instances in vibration data.

In this work, the latent manifold of *VAE*s is investigated for time-series data to determine whether anomalous instances are detectable only in the input space or if the latent manifold is also able to detect the presence of anomalous instances. The benefit of the *VAE* model is the addition of encoder and decoder network non-linearity and flexibility, which can aid in producing better latent manifolds. To do this, the *temporal preservation* approach is imperative as it allows for the element of time to be preserved in the discrepancy signals, where a discrepancy signal in this work is a signal that contains all the discrepancy metric responses for the partitioned segments obtained using Equation (1.15) with $L_{sft} = 1$. Most approaches to anomaly detection using VAEs only focus on the reconstruction log-likelihood of a model for an instance $\mathbf{x}_i$ but the input space is only one element of the model that is indicative of damage. If the latent manifold captures the information of a healthy asset, the presence of anomalous instances should also manifest in this manifold, however sensible methods of measuring this change must be used.

2.5 β-TC-VAE

Another form of a VAE that is interesting is the β-Total Correlation (TC)-VAE, which is considered to be one of the current state-of-the-art VAE methods. The β-TC-VAE formulation is a variation of the original VAE whereby the KL divergence from the ELBO is decomposed into separate elements. The aim here is twofold, not only does this decomposition yield an alternative formulation to the original VAE, but it also allows for a deeper understanding of the original VAE formulation and what the KL divergence aims to achieve. Due to issues with initially implementing this method, the decision was made to present the β-TC-VAE thoroughly, so that future work can be done more efficiently if required. Consider now the KL divergence from the VAE loss in Equation (2.22), which was given in its mini-batch form. In Chen et al. (2018), they initialise the proof using

$$\frac{1}{N}\sum_{i=1}^{N} KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i)\|p_\theta(\mathbf{z})\right) = \mathbb{E}_{p(\mathbf{n})}[KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i)\|p_\theta(\mathbf{z})\right)], \tag{2.28}$$

where $\mathbf{n}$ now makes reference to the entire dataset of training data $\mathbf{x}$. The decomposed form of the KL divergence was shown to be

$$\mathbb{E}_{p(\mathbf{n})}[KL\left(q_\phi(\mathbf{z}|\mathbf{x}_i)\|p_\theta(\mathbf{z})\right)], = \underbrace{KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n}))}_{\text{Index-Code MI}} + \underbrace{KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j))}_{\text{Total Correlation}} + \underbrace{\sum_j KL(q(\mathbf{z}_j)||p(\mathbf{z}_j))}_{\text{Dimension-wise KL}}, \tag{2.29}$$

where $\mathbf{z}_j$ is used to refer to the j^{th} latent variable (Chen et al., 2018). As noted in Equation (2.29), there are three elements which can be referred to as the index code Mutual Information (MI), the TC and the dimension-wise KL divergence. The intuition between these three elements is: the index code MI can aid in enabling compact and disentangled latent space representations, the TC term can aid in finding independent latent factors in the data distribution and the dimension-wise KL divergence ensures that the latent dimensions do not deviate from the prior distribution. Chen et al. (2018) then argue that the existence of the TC term in the KL divergence is why VAEs can learn disentangled latent representations and give the overall VAE objective function as

$$\mathcal{L}_{\beta\text{-}TC} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{n})p(\mathbf{n})}[\log p(\mathbf{n}|\mathbf{z})] + \alpha KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n})) + \beta KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j)) + \gamma \sum_j KL(q(\mathbf{z}_j)||p(\mathbf{z}_j)), \tag{2.30}$$

where α, β and γ are weighting parameters for the expanded KL divergance, with Chen et al. (2018) stating that one use $\alpha = \gamma = 1$ and modifying β. This is the final objective of the β-TC-VAE. However, there is a clear dependency here on the entire dataset $\mathbf{n}$, which can be problematic when datasets become large in size. One can note that this objective function cannot be easily implemented, thus the author has chosen to present a detailed decomposition of the $\beta - TC - VAE$ objective function in Appendix B.3.

Kim and Mnih (2018) derived a Factor-VAE, which is similar to the β-TC-VAE but an adversarial training approach was used to estimate the TC term in the expanded KL divergence. Esmaeili et al. (2018) provide a unifying investigation into the ELBO and the KL divergence in VAEs, with an intuitive explanation of the expanded KL divergence. In this research, the β-TC-VAE with mini-batch stratified sampling shall be used to compare how a standard VAE performs to a state-of-the-art technique.

2.6 Generative Adversarial Networks

Generative Adversarial Networks (GANs), developed by Goodfellow et al. (2014), are considered to be the forefront of unsupervised learning techniques due to their impressive successes in image generation and to a larger degree, generative modelling as a research endeavour. A GAN, in an unsupervised machine learning framework, is an algorithm whereby two networks are pitted against one another and play a two-player game that terminates when an equilibrium point is reached. Here, the two players are the generator network and the discriminator network, denoted as G and D respectively, and the two-player game methodology is known as an *adversarial network* framework. In this framework, the adversary is the discriminator D. The roles of the players in this game are simple, the objective of the generator is to generate samples similar to those of the ground truth data samples and the role of the discriminator is to determine whether these samples are real or fake. The discriminator is trained to try and improve its ability to label data as real or fake, while the generator is trained to try and improve its ability to make the discriminator label the generated data incorrectly. The equilibrium point required for game termination is the point when the Nash-equilibrium is reached. Nash equilibrium is a point where all players in a non-cooperative game cannot perform any profitable action, based on the state of all other players (Muthoo et al., 1996).

As training progresses in this framework, the hope is that G will become better at producing samples that are similar to those of the data distribution, while D will become better at determining whether samples are from the real distribution. Thus, as training progresses, the players G and D shall update and improve themselves, based on the improvement of the other, to a point where D has equivalent uncertainty on whether real and fake samples are distinguishable. The hope is that when Nash-

equilibrium is reached, G can fully mimic the data distribution and generated samples should be indistinguishable from those in the real distribution. If this point is reached, then G can be seen as a proxy to the real distribution (Goodfellow et al., 2014).

Jiang et al. (2019b) utilised GANs in an AE based framework, whereby vibration data was encoded and decoded using an AE, and the decoded data was sent to a discriminator. However, the raw vibration data was not used but rather a feature extractor that converted a signal into feature representations. Their implementation is akin to an $\alpha-$GAN approach with a pre-defined feature extractor (Rosca et al., 2017). The implementation used by Jiang et al. (2019b), like Booyse et al. (2020), makes use of only healthy vibration data as training data but rather than using the discriminator as a health indicator, an anomaly score based on the reconstruction loss in the input space and the latent space was used. When their approach was implemented on the Case Western Reserve University bearing dataset their technique proved to be comparable to other inference-based techniques such as a Bidirectional GAN (BiGAN) and, by extension, Adversarially Learned Inference (ALI), which are techniques proposed by Dumoulin et al. (2016) and Donahue et al. (2016) respectively.

Liu et al. (2018a) utilised the adversarial auto-encoder (AAE) approach, detailed in Makhzani et al. (2015), alongside their proposed Categorical GAN (CatGAN) approach to produce a technique they call Categorical Adversarial Auto-Encoder. Here, the idea was to allow the adversarial AE to learn to encode data to some prior $p(\mathbf{z})$ and at the same time, the encoded data was used to train a discriminative classifier to classify latent codes to one of K classes. Thus, the assumption is inherently made that the latent space is some clustered distribution, such as K Gaussians, so that labels can be generated for prior samples. In doing so, the hope is to not only encode data to some prior but also perform clustering based on different data. This technique was implemented on vibration data for the Case Western Reserve Bearing dataset, where fault data was also trained on, with the hope that different faults could be clustered to different places in the latent space. Vibration data was pre-processed using a multitude of features and a simple 2D latent space was constructed. Their technique was compared to other latent space clustering techniques such as K-means clustering and was found to be superior for vibration data, even in the presence of non-stationary operating conditions.

GANs have also been specifically designed for anomaly detection in cases where the data is not vibration data, with various techniques such as the Anomaly GAN (AnoGAN) (Schlegl et al., 2017), f-AnoGAN (Schlegl et al., 2019), Efficient GAN-Based Anomaly Detection (EGBAD) (Zenati et al., 2018), GANomaly (Akcay et al., 2019) and Adversarial Dual Auto-Encoder (ADAE) (Vu et al., 2019). In the works of Di Mattia et al. (2019), these techniques are all compared and their performance is evaluated on a variety on datasets, including MNIST and Fashion-MNIST (LeCun et al., 1998, Xiao et al., 2017).

2.6.1 GAN Training

To train a GAN, a framework is used whereby the two players are both parametrised by neural networks, which are then initialised and shall bear the notation $G_\theta(\mathbf{z})$ and $D_\phi(\mathbf{x})$ for the generator and discriminator respectively. As is the case with unsupervised learning, the assumption is made that there are latent variables $\mathbf{z}$ that describe the data, inducing a latent variable model, and a prior for this latent variable distribution is assumed (Rosca et al., 2017). The generator then represents the distribution $p_\theta(\mathbf{x}|\mathbf{z})$, where the prior $p(\mathbf{z})$ is often assumed to be a multivariate, isotropic unit Gaussian. G can also be seen as a parametric function from $\mathbb{R}^z \to \mathbb{R}^D$, where z is the latent space and D refers to the input space, with continuous parameters θ. The discriminator is a network with a single output node and a sigmoid activation function applied to this node, thus a function $\mathbb{R}^D \to \mathbb{R}^1$ with parameters ϕ in

the classic GAN framework. The objective of D is to predict 1 for samples $\mathbf{x} \sim p(\mathbf{x})$ and 0 for samples $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. From a probabilistic perspective, the discriminator gives the distribution $p(t=1|\mathbf{x})$, which assigns a probability to a sample it sees with $t=1$ signifying real data and $t=0$ signifying fake data. One can immediately notice here that there are now some target labels assigned to data, indicating that this training framework, albeit being purely unsupervised concerning data labels, is supervised in its training of the discriminator.

Using the binary cross-entropy loss function from Equation (A.8), the objective function for GAN training can be given as

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p(\mathbf{x})}\left[\log D_\phi(\mathbf{x})\right] + \mathbb{E}_{z\sim p(\mathbf{z})}\left[\log(1-D_\phi(G_\theta(\mathbf{z}))\right], \tag{2.31}$$

where one aims to optimise the discriminator parameters ϕ by minimising the loss function

$$\mathcal{L}_D(\theta,\phi) = -\mathbb{E}_{x\sim p(\mathbf{x})}\left[\log D_\phi(\mathbf{x})\right] - \mathbb{E}_{z\sim p(\mathbf{z})}\left[\log(1-D_\phi(G_\theta(\mathbf{z})))\right], \tag{2.32}$$

and one aims to optimise the generator parameters θ by minimising the loss function

$$\mathcal{L}_G(\theta,\phi) = \mathbb{E}_{z\sim p(\mathbf{z})}\left[\log(1-D_\phi(G_\theta(\mathbf{z})))\right]. \tag{2.33}$$

The training procedure for GANs is implemented as an alternating gradient descent procedure, where it is important to note that it is referred to as descent as the objective functions are minimised rather than maximised. Equations (2.32) and (2.33) are the ones shown in Goodfellow et al. (2014), with the target label elements dropped for brevity due to the trivial nature of their application. One interesting derivation that is key to understanding how GANs learn to approximate $p(\mathbf{x})$ with $p_\theta(\mathbf{x}|\mathbf{z})$ comes about in the derivation of what the optimal values that should be obtained GAN training reaches Nash equilibrium. The entire derivation is beyond the scope of this work and only the key elements from Goodfellow et al. (2014) shall be presented. If one attempts to find an optimal discriminator using $\max_D V(G,D)$, the result becomes

$$D^*_G(\mathbf{x}) = \frac{p(\mathbf{x})}{p(\mathbf{x}) + p_g(\mathbf{x})}. \tag{2.34}$$

where $p(\mathbf{x})$ is the true data distribution and $p_g(\mathbf{x})$ is the generated distribution learnt by the generator. This result shows that in the discriminator update step, density ratio estimation is performed by the discriminator, which often leads to some literature calling the discriminator update the ratio estimation step and the adversarial or generative step (Mohamed and Lakshminarayanan, 2016, Uehara et al., 2016, Goodfellow, 2015). Using this result, the original objective function in Equation (2.31) was reformulated in Goodfellow et al. (2014) and reduced to

$$V(D^*_G, G) = -\log 4 + 2JSD(p(\mathbf{x})\|p_g(\mathbf{x})), \tag{2.35}$$

where JSD is the Jensen-Shannon Divergence between two distributions. This is an important consideration for the GAN framework, as it is now clear that some divergence metric is used to compare the true and generated distributions. This divergence is also slightly better behaved, theoretically, when two distributions are non-overlapping as the KL divergence tends to infinity in the case of non-overlap.

In the application to time-series data, GANs offer significant advantages due to the implicit density estimation performed. This then removes the assumption that the input data is Gaussian, which adds a benefit of model flexibility to capture more complex data distributions. The work of Booyse et al. (2020) was significant in introducing and formulating *GAN*s in a time-series data application, with it shown that *GAN*s offer significant advantages and can clearly detect damage through the use of the learnt discriminator. The implicit density estimation along with G non-linearity offers an advantage over *VAE*s in the type of data the manifold can capture. However, the focus of this work is around

the latent manifold of latent variable models and the current *GAN* formulation only allows for data generation and no model inference. In the sections that follow, the aim is to investigate *GAN*s further and also develop latent variable models that can benefit from the implicit density estimation and recover a latent manifold for the input data.

In the implementation of GANs, GAN training proved to be a tricky procedure, with clear sources of difficulty identified in the literature. These problems have been thoroughly investigated, however, there is no clear direction that has shown to be the correct focal direction. There are three main fields of research focus, namely, GAN loss function improvement, GAN training framework improvement and GAN parametrisation improvement. These fields will now be elaborated on and discussed based on literature.

2.6.2 Loss Function Improvement

The loss functions associated with GANs are often scrutinised due to problematic results noted from the original formulation. Here, the author is referring to how the gradients might back-propagate during training initialisation. It was noted by Goodfellow et al. (2014) that D, due to the simple nature of its operation, is near-optimal performance as there is little to no overlap between $p(\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$. As a result, the original formulation in Equation (2.33) produces insufficient gradients and results in slow or even negligible G training improvement. Two popular improvements from literature are the Non-Saturating loss and the KL Loss (Goodfellow et al., 2014, Sønderby et al., 2016). These losses are implemented to directly improve Generator training, where the non-saturating loss is

$$\mathcal{L}_{G_{NS}} = -\mathbb{E}_{z\sim p(\mathbf{z})}[\log D_\phi(G_\theta(\mathbf{z}))], \tag{2.36}$$

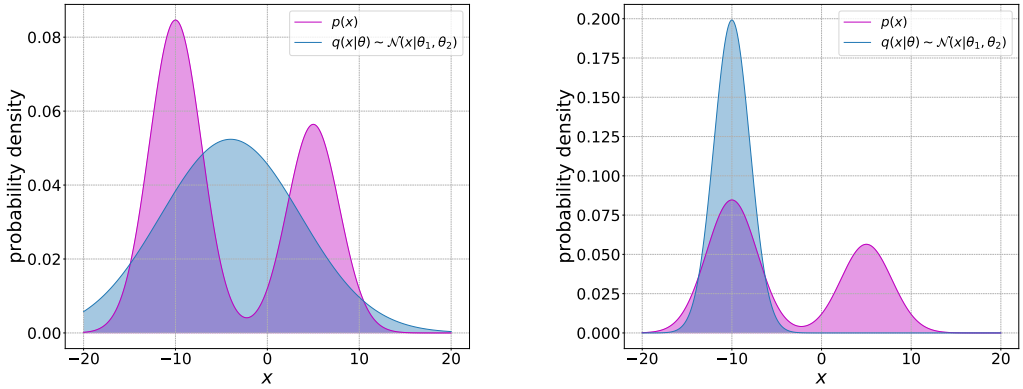
which in non-saturating as the original GAN loss may saturate around 0 during initial training. The fundamental principle of the loss has also changed, where for the original loss the generator aimed to minimise the prediction that the generated data was fake while for the non-saturating loss it aims to maximise the prediction of generated data as real. The non-saturating loss allows for strong gradients during initial training and weaker gradients near the end of training, provided that G improves. However, this weaker gradient result is not always an ideal occurrence, as it may result in premature saturation in the training of G. As an alternative, the KL Loss was derived by Sønderby et al. (2016) based on the principle of mean-seeking versus mode-seeking divergence metrics between distributions. Consider for a moment, the asymmetric KL divergence where the mean-seeking divergence is the forward KL divergence $KL(p||q)$ whereas the mode-seeking divergence is the reverse KL divergence $KL(q\|p)$, where p is the true distribution and q is the approximate distribution. By expanding the KL divergence, these forms are

$$KL(p\|q) = \int p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}, \tag{2.37}$$

$$KL(q\|p) = \int q(\mathbf{x}) \log\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}, \tag{2.38}$$

where the performance of these two types is easily explained through a simple example. Consider the case where one aims to approximate some bi-modal univariate-Gaussian mixture distribution $p(x)$ with a univariate Gaussian distribution $q(x)$ of which one can control the mean and variance, as shown in Figure 2.2. When minimising the forward KL divergence in Equation (2.37), the approach will strongly penalise points where $q(x)$ is very low while $p(x)$ is larger. As a result, the optimal case will result in a $q(x)$ that covers all $p(x) > 0$, thereby tending to satisfy a mean-result of $p(x)$, as shown in Figure 2.2(a). When minimising the reverse KL divergence in Equation (2.38), the approach will strongly penalise points where $p(x)$ is low while $q(x)$ is larger. This result will induce a mode-seeking approach, as shown in the simple example shown in Figure 2.2(b), as the univariate distribution will rather focus on one mode and not placing $q(x)$ where there is no $p(x)$. As noted by Bishop (2006), in many practical applications it is not unreasonable to assume that $p(x)$ may be multi-modal, which then

highlights that the forward KL divergence may be ill-suited, as it may lead to the modal average and thus the parametric distribution $q(x)$ will be a poor approximation. This simple idea extends to other



(a) Optimal forward KL divergence.

(b) Optimal reverse KL divergence.

Figure 2.2. Illustration of the KL divergence for a simple problem of a parametrised univariate Gaussian distribution used to minimise the forward and reverse KL divergence. Notice the mean seeking behaviour of the forward KL divergence and the mode seeking behaviour of the reverse KL divergence.

distributions and for the assumption of the distribution modelled through G, which takes an implicit distribution form, the parametric function is hopefully sufficiently flexible to capture all modes. This assumption of sufficient flexibility is why the reverse KL divergence is favoured, as G will then tend to neglect points where $p(\mathbf{x})$ is not likely to be and may be able to capture all modes.

The derivation of the KL divergence loss is beyond the scope of this work but if one minimises the reverse KL divergence, as detailed in Sønderby et al. (2016), the KL divergence loss that one can use for the optimisation of G can be given as

$$\mathcal{L}_{G_{KL}} = -\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log \frac{D(G_\theta(\mathbf{z}))}{1 - D_\phi(G_\theta(\mathbf{z}))}. \tag{2.39}$$

2.7 GAN Training Framework Improvement

The second problem often addressed and discussed in the literature is the idea of addressing GANs through changing the optimisation scheme or even the methodology of how a GAN is developed. Thus, in this section, the author will discuss different approaches to GAN training as well as different approaches to the adversarial framework on which GANs operate. Due to the inherent difficulties associated with training GANs, there has been a substantial driving mechanism within the literature to improve the optimisation scheme implemented to find the Nash-equilibria of the GAN-game. Here, the author generalises the optimisation scheme to cover both the optimisation technique as well as the GAN framework formulation. For the first point of discussion, GAN optimisation scheme improvement from the optimisation technique and the objective function will be discussed. Note that in the previous section we did cover G objective function improvement, here however the focus will be on both D and G.

2.7.1 Optimisation Scheme Improvement

For the classical optimisation technique used in training GANs, one takes alternating D steps and then G steps until some convergence criteria are met. This technique is referred to as alternating gradient descent or ascent depending on how one defines their loss functions, but as pointed out in the works of Mescheder et al. (2017) this technique does not always guarantee convergence due to inherent issues with the optimisation framework.

In the zero-sum game, if the Hessian is positive definite and the learning rate is sufficiently small, then the alternating gradient descent approach shall converge to the Nash equilibrium. However, due to complexity induced by neural network parametrisation, the Hessian is no longer guaranteed to be positive definite, due to the non-convex nature of the objective functions used in GANs. Furthermore, Mescheder et al. (2017) show that the existence of eigenvalues with large imaginary components in the Hessian can be detrimental to training, with this existence attributed to the non-conservative vector field induced in the original zero-sum game. To alleviate this issue, the authors choose to enforce convexity into the loss functions directly, by adding the norm of the objective functions to the original cost functions. This convexity enforcement led to the development of the Consensus Optimisation algorithm, which can also be seen as a technique that tries to enforce that the non-conservative vector field is conservative locally. The convex enforcement term is developed for GAN training as

$$L_{norm}(\theta,\phi) = \frac{1}{2}\left[||\nabla_\phi \mathcal{L}_D(\theta,\phi)||_2^2 + ||\nabla_\theta \mathcal{L}_G(\theta,\phi)||_2^2\right], \tag{2.40}$$

where one can see that this term enforces that the gradients of the two objective functions must tend to zero during optimisation. Thus, for the case of GAN training, the new objective functions can be shown to be

$$\tilde{\mathcal{L}}_D(\theta,\phi) = \mathcal{L}_D(\theta,\phi) + \lambda L_{norm}(\theta,\phi), \tag{2.41}$$
$$\tilde{\mathcal{L}}_G(\theta,\phi) = \mathcal{L}_G(\theta,\phi) + \lambda L_{norm}(\theta,\phi), \tag{2.42}$$

where λ is some enforcement parameter, that guides how harshly the objective function is penalised for large gradients. The training scheme for Consensus Optimisation is also one that is no longer alternating gradient descent but simultaneous gradient descent. Therefore, G and D are updated simultaneously as opposed to in an alternating pattern (Mescheder et al., 2017, 2018).

Consensus Optimisation, however, is not a silver bullet for GAN training, with other techniques such as unrolled-GANs (Metz et al., 2016) also showing promising results. Unrolled-GANs operate under the principle of an unrolled update procedure for G, whereby G is updated for N updates of D. However, the gradient is back-propagated or 'unrolled' through each of the N updates for D and not for just the future state of D. This unrolled future state of D is then forgotten and D is updated based on the current state of G. A technique proposed by Nagarajan and Kolter (2017), similar to Consensus Optimisation, operates by adding the norm of the gradient of D to the objective function of G. This technique also focuses on the non-conservative nature of the vector field, however, it addresses it only for G and not for both D and G. The loss function in this case is

$$\tilde{\mathcal{L}}_G(\theta,\phi) = \mathcal{L}_G(\theta,\phi) + \lambda||\nabla_\phi \mathcal{L}_D(\theta,\phi)||_2^2. \tag{2.43}$$

These convex gradient approaches are proposed to combat mode collapse in GANs, which they attribute to the optimisation scheme. Mode collapse is a phenomenon whereby, if the data distribution consists of non-overlapping modes, G tends to hop between modes rather than learning to sample from each mode. This occurs due to the nature of G, as it is not enforced that it captures all modes but rather its only objective is to fool D (Goodfellow et al., 2014). Other techniques that try to circumvent mode collapse are to induce gradient penalisation, such as the technique proposed by Roth et al. (2017) which uses a gradient penalty on the discriminator. An alternative to the technique proposed by Roth

et al. (2017) is the R_1 or R_2 gradient penalty proposed by Mescheder et al. (2018), which penalises the discriminator based on either real data or fake data. The form of this penalty is

$$R_n = \frac{\lambda}{2}\mathbb{E}_{\mathbf{x}\sim P}[\|\nabla_{\mathbf{x}} D_\phi(\mathbf{x})\|_2^2], \tag{2.44}$$

where R_1 is for cases where $P = p(\mathbf{x})$ and R_2 is for cases where $P = p_\theta(\mathbf{x})$. Another alternative, as proposed by Booyse et al. (2020), is that of a discriminator penalty that is based on samples from the latent distribution, where the objective here is to produce a discriminator that is invariant to mild perturbations in the latent space. The penalty form is similar to the R_2 penalty but instead of taking the derivative to $\mathbf{x}$ the derivative is taken to $\mathbf{z}$, given as

$$R_z = \frac{\lambda}{2}\mathbb{E}_{\mathbf{x}\sim p_\theta(\mathbf{x})}[\|\nabla_{\mathbf{z}} D_\phi(\mathbf{x})\|_2^2], \tag{2.45}$$

2.7.2 GAN Formulation Improvement

In this section, a discussion about GAN formulations that aim to introduce disentangled latent representations explicitly as well as GAN formulations that aim to improve training by changing the divergence metric that is used to quantify the divergence between $p(\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ will occur. The first method introduced is that of the InfoGAN, then the Adversarial Auto-Encoder (AAE) and finally Wasserstein-GANs (WGANs) and by extension, Wasserstein Auto-Encoders (WAEs).

2.7.2.1 InfoGAN

The Information Maximising GAN (InfoGAN) framework is a mild modification of the original GAN framework. This modification attempts to obtain explicit disentangled representations in an unsupervised manner. The benefit of a disentangled latent representation is the capturing of the factors of variation in the data and thereby ensuring that the important information in the data is captured by the latent manifold. Here the author introduces the term explicit disentanglement, which is not often seen in the literature. The explanation for this term is that for disentanglement in a GAN framework, the form of the disentangled representation is *explicitly* chosen, such as a discrete categorical representation, whereas VAEs attempt to uncover disentangled representations *implicitly*. InfoGAN's originally received criticism for poor performance in comparison to state of the art VAEs, such as FactorVAE (Kim and Mnih, 2018). However, Lin et al. (2019) were able to show that by using techniques to improve GAN training from literature, the performance of InfoGANs could be vastly improved.

InfoGANs make use of mutual information, which is a measure of the amount of information between two random variables. Mutual information can be expressed as the sum of the entropy of one of the random variables minus the conditional entropy, where the variable chosen for just the entropy term is conditioned by the other. The author chose this description as mutual information can be expressed as

$$\begin{aligned} MI(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X), \end{aligned} \tag{2.46}$$

where $H(\cdot)$ is the Shannon entropy of a random variable and it is clear to see that the mutual information between two random variables is interchangeable (Bishop, 2006). Chen et al. (2016) then chose to assume that the latent vector $\mathbf{z}$ comprises three components: a noise component $\mathbf{n}$, a continuous component $\mathbf{s}$ and a categorical component $\mathbf{c}$. These three components then had three objectives in terms of what they aim to disentangle. The noise component is assumed to contain the incompressible noise element of the data distribution, the continuous component is assumed to capture the structured semantic features of the data distribution and the categorical element is assumed to capture any categorical representations of the data distribution. Then, by adding an information-theoretic mutual information measure as a form of regularisation, the objective is to maximise the mutual information

between latent codes $\mathbf{c}$ and $\mathbf{s}$ and the generated distribution $p_\theta(\mathbf{x}|\mathbf{z})$. By maximising this mutual information, the hope is to obtain latent codes that are disentangled and contain features that define the underlying structure of the data. The proposed form of the zero-sum game is given as

$$\min_G \max_D V_I(G,D) = V(G,D) - \lambda [MI(\mathbf{c}, G(\mathbf{n},\mathbf{c},\mathbf{s})) + MI(\mathbf{s}, G(\mathbf{n},\mathbf{c},\mathbf{s}))]. \tag{2.47}$$

One of the issues associated with the conditional entropy term in the mutual information is that one needs the posterior distribution $p(\mathbf{c}|\mathbf{x})$, which is intractable. However, using Variational Information Maximisation (Barber and Agakov, 2004), a lower bound of mutual information can be obtained. In this work the approach for $\mathbf{c}$ is shown, assuming that one has access an approximate distribution $q(\mathbf{c}|\mathbf{x})$, which can be done using a neural network as a parametric function from $\mathbb{R}^D \to \mathbb{R}^c$. The lower bound is shown to be

$$MI(\mathbf{c}, G(\mathbf{n},\mathbf{c})) \geq H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{n},\mathbf{c})} \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}|\mathbf{x})} \log(q(\mathbf{c}|\mathbf{x})). \tag{2.48}$$

However, one can see that there is still a dependency of the intractable distribution $p(\mathbf{c}|\mathbf{x})$, fortunately Chen et al. (2016) show that this can be removed, where the final element of the proof is the consideration of a parametric distribution $q_\varphi(\mathbf{c}|\mathbf{x})$, where we now wish to maximise the mutual information by optimising the parameters φ in $q_\varphi(\mathbf{c}|\mathbf{x})$ through

$$\begin{aligned}
\mathcal{L}_I(G,Q) = q(\mathbf{c}|\mathbf{x})) &= \max_{q(\mathbf{c}|\mathbf{x})} (H(\mathbf{c}) + \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{x}\sim G(\mathbf{n},\mathbf{c})} \log(q(\mathbf{c}|\mathbf{x})) \\
&= H(\mathbf{c}) + \max_{\varphi} \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{x}\sim G(\mathbf{n},\mathbf{c})} \log(q_\varphi(\mathbf{c}|\mathbf{x})) \\
&= H(\mathbf{c}) + \max_{\varphi} \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{n}\sim p(\mathbf{n})} \log(q_\varphi(\mathbf{c}|G(\mathbf{n},\mathbf{c}))).
\end{aligned} \tag{2.49}$$

Therefore, the InfoGAN objective function can be written as

$$\min_{G,Q} \max_D V_{InfoGAN}(G,D,Q) = V(G,D) - \lambda \mathcal{L}_I(G,Q), \tag{2.50}$$

where $H(\mathbf{c})$ is dropped as it is a constant. To implement the InfoGAN approach, we parametrise Q with a neural network, with the decision made by Chen et al. (2016) to share most of the weights of D. The form of the distribution $q_\varphi(\mathbf{c}|\mathbf{x})$ is made to be that of a categorical distribution of $\mathbf{c}$, or a Gaussian distribution for $\mathbf{s}$. Lin et al. (2019) showed that it is more effective to rather choose a factored Gaussian distribution for $\mathbf{s}$ as it is easier to optimise. As a result, the form for categorical latent variables are as follows

$$q(\mathbf{c}|G_i(\mathbf{n},\mathbf{c}_i)) \sim Bern(\mathbf{c}_i|\boldsymbol{\mu}_{c_i}), \tag{2.51}$$

where $\boldsymbol{\mu}_{c_i}$ is a parametric function in $\mathbb{R}^D \to \mathbb{R}^k$ where k is the number of categories. For the continuous latent variable case, the assumption is that the distribution is a Gaussian given as

$$q(\mathbf{s}|G_i(\mathbf{n},\mathbf{s}_i)) \sim \mathcal{N}(\mathbf{s}_i|\boldsymbol{\mu}_{s_i}, \boldsymbol{\sigma}^2_{s_i}\mathbf{I}), \tag{2.52}$$

where $\boldsymbol{\mu}_{s_i}$ and $\boldsymbol{\sigma}^2_{s_i}$ are parametric functions in $\mathbb{R}^D \to \mathbb{R}^s$. One thing to note is that the variance can either be a neural network or assumed to be unity. So, ultimately, to implement an InfoGAN you need an neural network that takes in a generated sample $G(\mathbf{n},\mathbf{c},\mathbf{s})$ and then outputs a predicted form of $\mathbf{c}$ and $\mathbf{s}$. It acts as a form of encoder that is focused on recovering the latent samples in a generated sample. The objective is to then minimise the difference between the original and recovered forms of the disentangled latent elements. In practice, it is common for the categorical variable chosen to be a categorical distribution $\mathbf{c} \sim \text{Cat}(K,u)$ whose form is given as

$$p(\mathbf{t}_i|u(\mathbf{x}_i)) = \prod_{k=1}^{K} u_k(\mathbf{x}_i)^{t_{ik}}, \tag{2.53}$$

where K is the number of classes and u is the parametric function used to predict the class of an input. It is also common practice to let the continuous variable be of the form $\mathbf{s} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.

In the work of Lin et al. (2019), the InfoGAN framework was improved with a technique that introduced a regulariser on the original InfoGAN objective function. This regulariser was referred to as a contrastive regulariser. The regulariser operates by introducing a discriminator that takes two samples generated by setting one latent feature to be the same while randomly sampling the other latent features. The objective of this discriminator is then to predict the latent feature that was shared between the images. The notion behind this proposed method is the enforcement of disentangled latent features which are distinct as possible.

2.7.2.2 Adversarial Auto-Encoder

Adversarial Auto-Encoders (AAEs) are a technique developed by Makhzani et al. (2015) that introduces the adversarial framework into a setting where one aims to perform variational inference. The operational principle of AAEs is to utilise the auto-encoder framework, which allows for a latent space to be recovered. However, the latent space has no form of regularisation which then does not guide the network into how $\mathbf{z}$ should be, allowing it to be distributed freely. Thus, to regularise it to some form, an adversarial framework is placed on the latent space. The objective of this adversarial framework is to produce a latent posterior distribution $q(\mathbf{z}|\mathbf{x})$ that matches an arbitrary prior distribution. This framework, which shall be called the AAE framework, can be seen as an alternative to a VAE, where a VAE uses the KL divergence to regularise the latent space while the AAE framework uses the *JSD* from GAN training to regularise the latent space. The objective function for AAEs can thus be defined as

$$\begin{aligned}\min_{\theta,\phi}\max_{\omega}\mathcal{L}_{AAE} &= \mathcal{L}_{AE}+\min_{G_z}\max_{D_z}V(D_z,E_z)\\ &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}||\mathbf{x}_i - G_\theta(E_\phi(\mathbf{x}_i))||_2^2+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log D_\omega(\mathbf{z})\right]+\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log(1-D_\omega(E_\phi(\mathbf{x}))\right],\end{aligned} \tag{2.54}$$

where D_ω refers to the latent discriminator network, E_ϕ refers to the encoding network and G_θ refers to the decoding network from the AE framework. Samples are drawn from any prior $\mathbf{z}\sim p(\mathbf{z})$, which are then considered to be the real latent samples and the encoding network is updated to ensure that the features it encodes are similar to those latent samples. Due to the requirement that all one needs is samples from the latent distribution, the latent space can be enforced to be any distribution that can be efficiently sampled from. This means that rather than developing an analytical expression, as was the case for the KL divergence, the latent space can be structured into anything just by having access to samples from the distribution you wish to use. This is quite a fundamental difference between VAEs and GANs and shall be discussed in section 2.7.3.

One interesting development in the AAE framework is the ability to use the adversarial framework to develop discrete latent variables. Like InfoGAN, the assumption can be made that the latent space consists of noise and some discrete categorical element. However, an encoding network can be trained to cluster inputs into categories, in an unsupervised fashion. It is interesting that if one uses the adversarial framework on $\mathbf{n}$ and $\mathbf{c}$ separately, where $\mathbf{z}=[\mathbf{n},\mathbf{c}]$ is the latent space, class structure can be enforced and uncovered. Here, one just needs a prior for $p(\mathbf{c})$ which can easily be chosen to be a categorical distribution $\text{Cat}(K,p)$. This then introduces a second discriminator network into the original AAE objective function that then aims to enforce that $q(\mathbf{c}|\mathbf{x})$ matches the prior for $\mathbf{c}$. The encoding network is therefore represented as $q(\mathbf{c},\mathbf{n}|\mathbf{x})$. To enforce that the output of the encoder $q(\mathbf{c}|\mathbf{x})$ is representative of the prior a softmax activation function can be used. In the implementation of categorical AAEs, the author found that it was easier to add a small amount of white noise to any discriminator input on the categorical code D_ζ. This is because samples from the prior $p(\mathbf{c})$ are one-hot vectors, therefore allowing the encoder some time to sort out its production of categorical elements before no useful gradients can back-propagate. One can also use this discrete latent variable approach

in a semi-supervised setting, where one uses minimal labelled samples to aid in how the encoder categorises inputs if the data has categorical labels associated with it.

One important point of discussion in the AAE framework is whether or not it is necessary to use the adversarial framework for simple latent distributions, such as the Gaussian distribution. If one has access to simpler divergence metrics, it might make sense to use a simpler metric to enforce structure into the latent space, where here the author is specifically referring to noise elements (**n**) and even continuous elements **s**. A popular metric in literature is that of the Maximum Mean Discrepancy (MMD) metric, which was introduced in Zhao et al. (2017) in a VAE context and Li et al. (2015) for a GAN context. In the works of Zhao et al. (2017), issues related to how VAEs with the KL divergence metric can lead to undesirable results were highlighted, such as inference failures due to properties of the ELBO objective, modelling bias due to dimensionality differences or a problem identified as the information preference problem. To overcome these issues, the authors proposed replacing the KL divergence with any divergence that satisfies $D(q||p) = 0$, if and only if $q == p$. They also propose adding a MI maximisation term, which could be reduced to the VAE objective function with a secondary divergence. For purposes of this discussion, the *InfoVAE* objective function from Zhao et al. (2017) can be given as

$$\begin{aligned}\mathcal{L}_{InfoVAE} = \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] + (1-\alpha)\mathbb{E}_{p(\mathbf{x})}KL\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right) \\ + (\alpha+\lambda-1)KL(q_\phi(\mathbf{z})||p(\mathbf{z})),\end{aligned} \tag{2.55}$$

where $q_\phi(\mathbf{z})$ cannot be evaluated, thus the authors propose approximating it by sampling $\mathbf{x} \sim p(\mathbf{x})$ and then sampling $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, which is the exact process in which latent samples are obtained in the original VAE framework to allow for gradients to back-propagate through distributions. The authors then recommend setting $\alpha = 1$ for instances when complex data distributions are to be approximated, which then leaves the second KL divergence term in Equation (2.55). This divergence was then replaced, with the authors recommending and AAE approach through the *JSD*, or recommending that one use MMD. Due to the authors' recommendations, the overall principle of the work of Chen et al. (2016) is the analysis of replacing the KL divergence with the MMD divergence. MMD is a technique that was developed by Gretton et al. (2008) that tests whether two distributions are different, based on samples drawn from each distribution. Specifically, MMD investigates the statistical moments of the two distributions and based on the similarity of these moments indicates whether or not the samples are similar. MMD is formulated as

$$MMD(\mathcal{F}, p, q) = \sup_{f\in\mathcal{F}}(\mathbb{E}_{x\sim p}[f(x)] + \mathbb{E}_{y\sim q}[f(y)]), \tag{2.56}$$

where $\mathcal{F}$ is a class of functions $f : \mathcal{X} \to \mathbb{R}$ with p and q defined in the domain $\mathcal{X}$. Gretton et al. (2008) produced a closed form solution that seeks to evaluate the difference in the first moments of the transformed space $\mathbb{R}$. This can be expressed by using a kernel $k(\mathbf{x}, \mathbf{x}^{'}) = \langle\phi(\mathbf{x}), \phi(\mathbf{x})\rangle$, where $\phi(\mathbf{x})$ is a basis function. Gretton et al. (2008) showed that MMD can be expressed as

$$MMD(\mathcal{F}, p, q) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i^{'}=1}^{N}k(\mathbf{x}_i, \mathbf{x}_i^{'}) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j^{'}=1}^{M}k(\mathbf{y}_j, \mathbf{y}_j^{'}) - \frac{2}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}k(\mathbf{x}_i, \mathbf{y}_j), \tag{2.57}$$

where N and M are the number of samples $\mathbf{x} \sim p(\mathbf{x})$ and $\mathbf{y} \sim q(\mathbf{x})$ respectively (Li et al., 2015, Dziugaite et al., 2015). A common choice for the kernel is the Gaussian kernel of the form

$$k(\mathbf{x}, \mathbf{x}^{'}) = e^{-\frac{\|\mathbf{x}-\mathbf{x}^{'}\|_2^2}{\varepsilon}}, \tag{2.58}$$

where ε is a shape parameter that controls the spread of the kernel (Snyman and Wilke, 2018, Bishop, 2006). For MMD in machine learning applications, a common choice is $\varepsilon = 2 * D$, where D is the dimensionality of the vector space. Interestingly, MMD has also been used to replace the GAN

objective as shown in the work of Li et al. (2015). To conclude this section, one now has the option to either use MMD for latent space regularisation or one can use the GAN objective, where the former is regarded to be stable and less susceptible to GAN training problems.

2.7.2.3 Wasserstein GANs

The next section of discussion is that of Wasserstein GANs (WGANs), which is a reformulation of the approach used for GANs. WGANs attempt to use another technique to measure the divergence between the generated parametric distribution $p_g(\mathbf{x})$ and the true distribution $p(\mathbf{x})$. WGANs introduce the concept of the earth-mover distance, which has its roots in optimal transport (OT) theory (Peyré and Cuturi, 2019). OT studies the problem of economically transforming one distribution into another, where these distributions can be discrete point masses or continuous distributions. Let μ and ϑ be probability measures defined on $\mathcal{X}$ and $\mathcal{Y}$, with density functions $\mu(x) = f(x)dx$ and $\vartheta(y) = g(y)dy$. To facilitate this transformation, one needs to define a transportation map $T : \mathbb{R}^d \to \mathbb{R}^d$ that transports discrete points or alters the spatial modification of a distribution. This transformation map must satisfy the condition

$$\int_A g(y)d(y) = \int_{T^{-1}(A)} f(x)dx, \tag{2.59}$$

where $\forall A \subset \mathbb{R}^d$. Given a cost function $c(x,y)$, where this cost function is the cost of transporting a unit mass from x to y, the total transportation cost can be given as

$$C_t = \int_x c(x, T(x))d\mu(x). \tag{2.60}$$

The Monge problem is then the attempt to find the transport map that minimises the total transportation cost under the transformation density constraint. The solution to this problem is known as the OT map and the total transportation cost of an OT map is called the Wasserstein distance $\mathcal{W}_c(\mu, \vartheta)$ which can be given as

$$\mathcal{W}_c(\mu, \vartheta) = \min_{T_{\#\mu}} \int_x c(x, T(x))d\mu(x), \tag{2.61}$$

where $T_\#$ is called the push forward function given by T that produces $\vartheta = T_{\#\mu}$ (Peyré and Cuturi, 2019, Lei et al., 2020a). A formulation was then developed called the Kantrovich formulation, which relaxed transportation maps to transportation plans. The idea is to rather than map points directly, a point x_i can be spread to several locations in y (Peyré and Cuturi, 2019). A joint distribution was defined as $p(x,y)$ where, when marginalised with respect to y or x, equals $\mu(x)$ or $\vartheta(y)$. This gives rise to a coupling which can be defined as

$$\mathcal{U}(\mu, \vartheta) = \left\{ p(x,y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \int_y p(x,y)dy = \mu(x), \int_x p(x,y)dx = \vartheta(y) \right\}. \tag{2.62}$$

The OT map is then found by

$$\mathcal{W}_c(\mu, \vartheta) = \min_{p \in \prod(\mu, \vartheta)} \int_x c(x, T(x))dp(x,y). \tag{2.63}$$

If the cost is chosen to be the distance between points x and y, $c(x,y) = \|x - y\|$, the optimal cost, often referred to as the Earth Mover distance or Wasserstein$_1$ distance, can be shown to be (using the Kantrovich-Rubenstein duality)

$$\begin{aligned} W_1(\mu, \vartheta) &= \inf_{p \in \prod(\mu, \vartheta)} \mathbb{E}_{(x,y)\sim p}[\|x - y\|] \\ &= \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x\sim\mu}[f(x)] - \mathbb{E}_{y\sim\vartheta}[f(y)], \end{aligned} \tag{2.64}$$

where f is a function that is K-Lipschitz on its entire domain $\mathbb{R}$ (Arjovsky et al., 2017, Gulrajani et al., 2017, Tolstikhin et al., 2018). K-Lipschitz functions are those that satisfy $|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$, or rather, the approximate gradient of a function for any set of points x_1 and x_2 is restricted by a constant K on the entire domain. By assuming that f is parametrised by a neural network and is to

be 1-Lipschitz, where f shall be referred to as the *WGAN critic* (D_{WGAN}), this formulation can be re-organised to be

$$\min_{G} \max_{D_{WGAN}} V_{WGAN}(D,G) = \mathbb{E}_{x\sim p(\mathbf{x})}\left[D_\phi(\mathbf{x})\right] - \mathbb{E}_{z\sim p(\mathbf{z})}\left[D_\phi(G_\theta(\mathbf{z}))\right], \tag{2.65}$$

which leads to the following objective functions, for gradient descent purposes, for D_{WGAN} and G respectively

$$\mathcal{L}_D(\theta,\phi) = -\mathbb{E}_{x\sim p(\mathbf{x})}\left[D_\phi(\mathbf{x})\right] + \mathbb{E}_{z\sim p(\mathbf{z})}\left[D_\phi(G_\theta(\mathbf{z}))\right], \tag{2.66}$$

$$\mathcal{L}_G(\theta,\phi) = -\mathbb{E}_{z\sim p(\mathbf{z})}\left[D_\phi(G_\theta(\mathbf{z}))\right]. \tag{2.67}$$

One important element is how the Lipschitz constraint is satisfied. In the work of Arjovsky et al. (2017), gradient clipping was used. However, the authors noted that this is a poor method of enforcing the constraint. In Gulrajani et al. (2017), a gradient penalty of the form

$$GP = \lambda \mathbb{E}_{\tilde{\mathbf{x}}\sim p(\tilde{\mathbf{x}})}[\|\nabla_{\tilde{\mathbf{x}}} D_\phi(\tilde{\mathbf{x}})\|_2 - 1]^2, \tag{2.68}$$

where $\tilde{\mathbf{x}}$ are samples drawn by uniformly varying between points from $\mathbf{x}\sim p(\mathbf{x})$ and $\mathbf{x}\sim p_g(\mathbf{x})$. The enforcement parameter λ is recommended to be set ≥ 1, with Gulrajani et al. (2017) reporting that $\lambda = 10$ worked well in their experiments. This gradient penalty is only applied to the critic, as the 1-Lipschitz enforcement is only for the WGAN critic. One key point is that the *WGAN critic* is no longer a parametric distribution as is the case for discriminators for GANs but rather a function that allows one to calculate the Wasserstein distance and track its deviation as the generator improves its expressibility to capture the data distribution.

Tolstikhin et al. (2018) developed the WAE, which can be viewed in juxtaposition to a VAE, where a WAE uses a WGAN divergence metric to replace the log-likelihood or reconstruction term from the VAE objective and any choice of divergence metric for the latent space regularisation. Two choices were investigated by Tolstikhin et al. (2018), namely, a GAN-based approach or an MMD-based approach. One key assumption made by Tolstikhin et al. (2018) was to not use the Wasserstein-1 distance but rather the Wasserstein-2 distance coupled with a convex penalty term (the Lagrangian of the constrained optimisation problem), which lead to the formulation of the Penalised Optimal Transport (POT) approach (Bousquet et al., 2017). As a result, a WAE approach leads to nothing more than a generalisation of AAEs and offers an alternative latent space regulariser.

A Wasserstein-Wasserstein Auto-Encoder was proposed by Zhang et al. (2019), which uses the Wasserstein-2 distance as a latent space regularisation strategy. Under the assumption that both the prior $p(\mathbf{z})$ and the approximate posterior $q(\mathbf{z}|\mathbf{x})$ are to be Gaussian, the Wasserstein-2 approach reduces to the Fréchet distance (Dowson and Landau, 1982, Heusel et al., 2017). This latent space objective function can be given as

$$W_2(p(\mathbf{z})||q(\mathbf{z}|\mathbf{x})) = \|\mu_p - \mu_x\|_2^2 + \text{tr}(\Sigma_p) + \text{tr}(\Sigma_q) - 2\text{tr}(\Sigma_p^{\frac{1}{2}}\Sigma_q^{\frac{1}{2}}), \tag{2.69}$$

where $\text{tr}(\cdot)$ is the trace operator and μ and Σ refer to the mean and covariance of the distributions. This regularisation term can be seen as a direct alternative to that used in a VAE.

2.7.3 GANs and VAEs

From a probabilistic framework, GANs can be described as an implicit density model where the model, at no point, makes any assumptions about the data distribution that it wishes to approximate. This is a powerful approach as it may allow for more flexible distributions to be approximated, by using density ratio estimation (Goodfellow, 2015, Mohamed and Lakshminarayanan, 2016). VAEs, on the other hand, can be seen as an explicit density model as one assumes choices for the parametric likelihood distributions that may ultimately affect the type of data one wishes to approximate. For this discussion, reference will be made between the ELBO which is to be maximised, as shown in Equation (2.18).

The objective is to derive an alternative loss that can be drawn into a GAN framework. Firstly, the density ratio $r(\mathbf{x})$ can be defined as

$$r(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|t=1)}{p(\mathbf{x}|t=0)} = \frac{p(t=1|\mathbf{x})}{p(t=0|\mathbf{x})} = \frac{D(\mathbf{x})}{1-D(\mathbf{x})}, \tag{2.70}$$

which the reader may recognise in its inverse form if they are aware of the generator loss derivation shown in Sønderby et al. (2016). Mohamed and Lakshminarayanan (2016) showed that this ratio can be related to the discriminator objective function through $D = \frac{r}{r+1}$ and the use of the Bernoulli loss function. There is also a need to define a synthetic likelihood, which can be used as an alternative to the first half of Equation (2.18) (Rosca et al., 2017). This synthetic likelihood can be given as

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z}_j)\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}_i|\mathbf{z}_j)}{p(\mathbf{x})}\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}). \tag{2.71}$$

Consider now the approach of explicit or implicit *posterior distributions* $q_\phi(\mathbf{z}|\mathbf{x})$ (Rosca et al., 2017). For the explicit case, the KL divergence in Equation (2.18) can be reduced to an analytical solution if both the prior $p(\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ are assumed to be factorised Gaussians. In the implicit case, one can use the density ratio trick to replace the KL divergence

$$-KL[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{x})] = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{x})}\right] \approx \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{D_{\mathbf{z}_\psi}(\mathbf{z})}{1-D_{\mathbf{z}_\psi}(\mathbf{z})}\right], \tag{2.72}$$

which introduces a discriminator to enforce latent structure as opposed the KL divergence, which Makhzani et al. (2015) achieved with AAEs. Here, $D_{\mathbf{z}_\psi}$ refers to a discriminator over $\mathbf{z}$ parametrised by ψ. This form of the divergence is using the KL divergence form of the generator update, where here the generator is regarded to be the encoding network (or the posterior distribution that is a parametric function) (Rosca et al., 2017).

Consider now the approach of explicit or implicit likelihood distributions $p_\theta(\mathbf{x}|\mathbf{z})$. If we parametrise this distribution as a Gaussian the classical MSE or L_2 loss function can be used, or if a zero-mean Laplace distribution is used the L_1 loss function can be used. If we assume that this distribution is implicit, we can again use synthetic likelihood with the density ratio trick, while dropping the second term in the synthetic likelihood as it is a constant. This usage case can be shown to result in

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z}_j)\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{D_\phi(G_\theta(\mathbf{z}))}{1-D_\phi(G_\theta(\mathbf{z}))}\right], \tag{2.73}$$

which introduces a discriminator on $\mathbf{x}$, as is the case in the classic GAN framework (Rosca et al., 2017). Here it is again clear that the KL divergence form of the generator objective function is used. Please note that in these derivations it was assumed that one still aims to maximise the ELBO, hence the sign difference in the derivations. One can note that two approaches can be taken to posterior distributions and likelihood distributions, which either leads to the VAE, GAN or AAE framework.

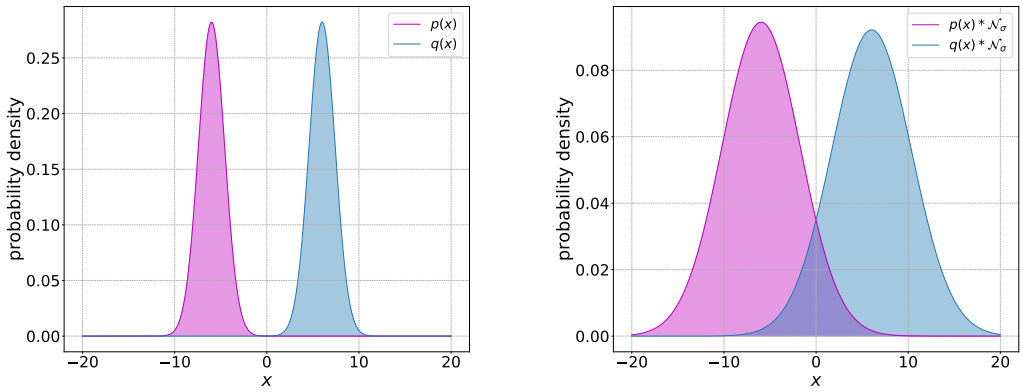
2.7.4 GAN Parametrisation Improvement

GAN parametrisation improvement is broadly described as techniques that attempt to ensure that the adversarial zero-sum game is well defined. These techniques aim to improve GAN training through the introduction of auxiliary elements that aim to smooth the training process. The main techniques that shall be discussed here are instance noise and spectral normalisation. Mescheder et al. (2018) detail a good introduction to parametrisation improvement. The works of Radford et al. (2015) and Heusel et al. (2017) are equally important, thus any interested reader should review these papers. Conditional GANs are an interesting research sphere of GANs, however, they are not paramount to this work. The author suggests that readers interested in this topic review the papers of Mirza and Osindero (2014), Reed et al. (2016), Odena et al. (2016), Perarnau et al. (2016) and Miyato and Koyama (2018) for more information in this regard.

2.7.4.1 Instance Noise

Instance noise is a technique that is considered to be commonplace in recent GAN training techniques. Instance noise aims to assist in the training of the generator, which may be ill-posed during initial training. As noted in Sønderby et al. (2016), often during training initialisation the generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ and the real distribution $p(\mathbf{x})$ are non-overlapping. The result of this is that the classic KL divergence in infinite and the JSD, albeit finite, is near maximum. The clear and obvious side-effect of this result is that discriminator updates will often be encroaching on optimality before the generator is even able to shift its distribution to match the real distribution (Sønderby et al., 2016).

To solve the problem of non-overlapping supports, Sønderby et al. (2016) proposed that one adds some Gaussian noise to any data seen by the discriminator, where this noise is gradually annealed out over the course of training. This allows for a better behaved divergence response and allows the initial state of the zero-sum game to focus on proxy distributions as opposed to the true distributions. Sønderby et al. (2016) show that any divergence metric $D_\sigma(q|p)$, when q and p are subjected to instance noise, becomes equivalent to $D_\sigma(q * \mathcal{N}_\sigma | p * \mathcal{N}_\sigma)$ where $*$ is the convolution operator and $\mathcal{N}_\sigma$ is the noise distribution. Figure 2.3 contains a visualisation of instance noise applied to two non-overlapping Gaussian distributions. One can note how instance noise creates overlapping supports between the two (Sønderby et al., 2016, Roth et al., 2017).



(a) Distributions with no instance noise.

(b) Distributions with instance noise.

Figure 2.3. Instance noise applied to two simple distributions. Notice the clear non-support in the case where there is no instance noise while if one uses instance noise there is now suddenly an overlapping support. Adapted from Sønderby et al. (2016) and best viewed in colour.

2.7.4.2 Spectral Normalisation

Another technique that has received attention is Spectral Normalisation (SN). Spectral normalisation is a discriminator weight normalisation technique that is based on the Lipschitz continuous function approach from WGANs. The overall idea is to determine a discriminator that satisfies

$$\max_{\|f\|_L \leq K} V(G,D), \tag{2.74}$$

where the linear approximation of the gradient $\frac{f(x_2)-f(x_1)}{x_2-x_1}$ is bounded by the Lipschitz constant K. This is achieved by taking each weight matrix $\mathbf{w}$ in the discriminator network and normalising it by the spectral norm (or largest eigenvalue) $\sigma(\mathbf{w})$ of said matrix. The result of this ensures that the Lipschitz

constant of a given layer is equal to 1. Miyato et al. (2018) show that due to the linear manner in which weights are multiplied in neural networks, this normalisation process can be done for each layer in the network independently. Thus, the Lipschitz constant for a neural network function is the product of the spectral norm of the layers in the network. In the implementation of SN, a power iteration method is used to determine the spectral norm of each of the weight matrices and are then iteratively updated through

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla E \left(\frac{\mathbf{w}_t}{\sigma(\mathbf{w}_t)} \right). \tag{2.75}$$

The result of using SN on discriminator networks has shown to improve generator performance on more complex datasets and allows for improved training performance. SN is also computationally cheaper to implement than other methods such as gradient norm regularisation (Miyato et al., 2018). Qin et al. (2018) investigated how SN affects discriminators and it was found that, as opposed to restricting weights to be small, SN restricts the range of loss function values attainable during training and in doing so, prevents vanishing and exploding gradients. This result shows it not the restriction of the discriminator network that matters, but rather objective function restrictions. it was also shown that for strong Lipschitz constraint enforcement, the objective function appears to tend towards a linear function.

2.8 Latent Disentanglement

A topic that needs to be elaborated on is that of the idea of implicit versus explicit disentanglement. In this work, implicit disentanglement is the term given to techniques such as VAEs where the latent space is required to be disentangled but there are no explicit terms in the training scheme that prompt disentanglement of specific feature types. This is a highly debated and discussed idea, with the works of Locatello et al. (2019) providing an informative and sobering investigation into whether it is intrinsically possible to learn disentangled features using the ELBO formulation on which VAEs are built.

Explicit disentanglement, in this work, refers to techniques that aim to obtain latent space elements that are explicitly prescribed to be some structure, such as a discrete latent space element from the InfoGAN framework. For vibration-based condition monitoring, Baggeröhr (2019) used an approach that consisted of a latent space that comprised of three elements, namely, a continuous element $\mathbf{s}$, a categorical or discrete element $\mathbf{c}$ and a noise element $\mathbf{n}$. This explicit latent space $\mathbf{z}$ can be given as $\mathbf{z} = [\mathbf{s}, \mathbf{c}, \mathbf{n}]$. The idea that stems from the notion of a signal decomposition, whereby a signal comprises of two main elements: a deterministic component, the first order cyclo-stationary component, and a residual component, where this residual may consist of noise and any second or higher-order cyclostationary components (Antoni, 2009). The objective of this explicit latent space composition is to let $\mathbf{c}$ and $\mathbf{s}$ capture the deterministic component while $\mathbf{n}$ is left to capture the residual component of a signal.

In the GAN-based frameworks in this work, the premise is to utilise two alternating training approaches in one, where the two approaches are the AAE approach and the InfoGAN approach. The AAE approach is used to ensure that data can be encoded and reconstructed into the prescribed latent space components, while the InfoGAN approach enforces that the mutual information terms $MI(\mathbf{s}|\mathbf{x})$ and $MI(\mathbf{c}|\mathbf{x})$ are maximised. In implementation, the mutual information maximisation can be obtained through the use of the encoder network, which is now treated as the parametric distribution Q from the original InfoGAN framework.

In the presence of only healthy data, the notion is that **n** will be a representation of a healthy signal residual component and will have some prescribed structure, such as a unit Gaussian distribution. In the presence of data that deviates from the training data representation, with this data potentially containing some machine damage information, **n** will exhibit off-manifold responses to data, with the hope that **n** will be the only latent representation to deviate from its normal state. One crucial note to make here is that these techniques are tailored to bearing failure cases, as it is a known property of bearing faults to manifest in the residual component of a signal. This intuition was used and driven in the works of Baggeröhr (2019). However, it is this author's inclination that these techniques can be thought of under a more general framework, that of using disentangled latent components to capture the predictable and known components of a signal, while the noise component is used to capture any additional information about the data. The noise component will also respond to any deviance from the healthy data state.

There is, however, one potential discrepancy that was not discussed or presented in the work of Baggeröhr (2019), that of whether the model behaviour under anomalous data responds according to the model assumptions. The model assumption referred to here is that a GAN-based technique with a latent space $\mathbf{z} = [\mathbf{s}, \mathbf{c}, \mathbf{n}]$ actually presents responses to damage in only the **n** component. In the formulation presented in Baggeröhr (2019) and, by extension, Zhou et al. (2019), there is no model incentive to only respond in **n** but only an incentive to disentangle **s** and **c** to capture the salient data attributes and restrict **n** to be the incompressible data attributes. As shown in Zhou et al. (2019), **n** still had some cluster-dependency, which indicates that the latent noise component still contains some structural information, which is undesirable. To investigate this, this author shall present two alternative formulations where these formulations are derived under two different ideologies: one focused on complete latent separation and the other focused on improved decoder information capture. As with most literature on GANs, the application is heavily biased to image data. Thus, the techniques used in this work will first be presented in their raw form and then adapted to vibration data applications. This adaptation accounts for both categorical and continuous latent variables.

2.9 Disentangled Latent Space Clustering

The Disentangled Latent Space (DLS) Clustering methodology is a technique proposed by Ding and Luo (2019) that aims to produce a latent representation of data that is not only disentangled but also separated into independent parts. This technique can be seen as an alternative to the REPGAN approach from Zhou et al. (2019), a foundational method used in the work conducted by Baggeröhr (2019). The main difference between the DLS methodology and the REPGAN methodology is that DLS aims to separable latent space elements. The benefit of this approach is that it forces the network to learn deterministic components that contain information in the deterministic component of the data while **n** as unstructured noise. To do this, a GAN and a deterministic auto-encoder are integrated, referred to in this work as a *GAN-based model*, to allow for bi-directional mappings between the latent space and the data space alongside the addition of non-Gaussian density estimation using the GAN framework for the data space. For a discussion to take place, the authors present four key model elements, namely, the posterior (encoder) distribution $q_\phi(\mathbf{z}|\mathbf{x})$, the generative (decoder) distribution $p_\theta(\mathbf{x}|\mathbf{z})$, a data discriminative distribution $p_\chi(t = 1|\mathbf{z})$ and a latent Wasserstein metric function $f_\omega(\mathbf{z_n})$. As this is a deep learning application, these distributions are represented by parametric functions $E_\phi(\mathbf{x}), G_\theta(\mathbf{z}), D_\chi(\mathbf{x})$ and $D_\omega(\mathbf{z_n})$ respectively. To aid with explain-ability, Figure 2.4 shall be often referenced to in three sections and these sections are highlighted in the figure.

Ding and Luo (2019) made the assumption that the latent space is split into two representations, based on the prior joint distribution $p(\mathbf{z}) = p(\mathbf{z_c}, \mathbf{z_n})$ where $\mathbf{z_c}$ is the discrete prior categorical latent

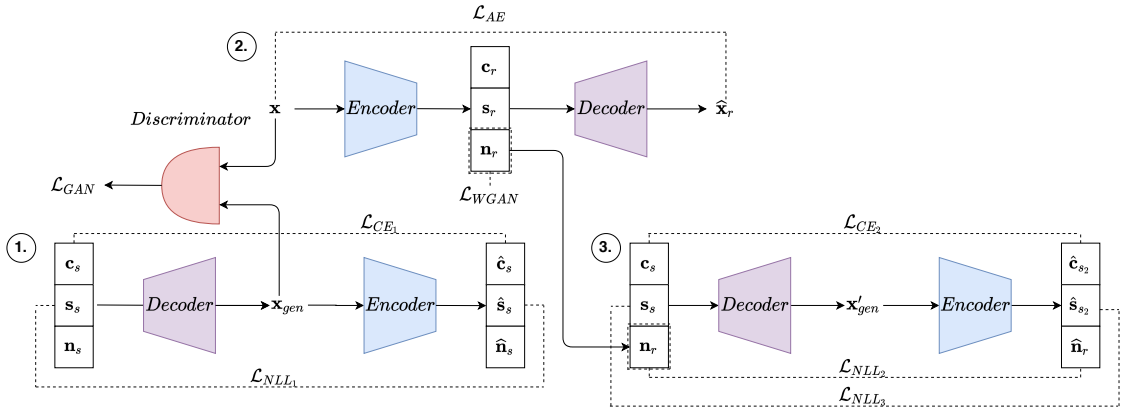


Figure 2.4. A complete overview of the DLS GAN model architecture with three main components, denoted through numbered circles. Part one refers to the InfoGAN framework, part two refers to the AAE framework and part three refers to the latent separation approach that the DLS-GAN method uses. Note that this author chose to use subscripts *s* to refer to elements that are influenced by samples from the latent prior distribution and *r* to refer to elements that are influenced by samples from the data distribution.

representation and $\mathbf{z_n}$ is the prior noise representation. For this discussion, this author will also introduce a third latent component, $\mathbf{z_s}$ which shall be referred to as the prior continuous representation. The objective is now to apply two sets of constraints to the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$, firstly such that the prior $p(\mathbf{z}) = p(\mathbf{z_c}, \mathbf{z_s}, \mathbf{z_n})$ becomes $p(\mathbf{z_c}, \mathbf{z_s})p(\mathbf{z_n})$ and secondly, constraints should also be applied such that $\mathbf{z_c}$ and $\mathbf{z_s}$ are disentangled latent components (capture the generative factors of the data). The former constraint is achieved by penalising the discrete and continuous latent components separately to the noise latent component while the latter constraint is achieved using mutual information maximisation. In this work, two topics shall be referred to, that of latent disentanglement and latent separation whereby the former aims to obtain disentangled latent components while the latter aims to obtain latent code independence. For brevity, it shall be assumed from this point that the posterior distribution generates components $\mathbf{c}$ as a categorical distribution and $\mathbf{s}$ and $\mathbf{n}$ components as a factorised unit-variance Gaussian distribution.

The process in which latent disentanglement, part one of Figure 2.4, is achieved is in lieu with the InfoGAN, however the parametric distribution Q that facilitates mutual information maximisation is simply the encoding distribution $q_\phi(\mathbf{z}|\mathbf{x})$. Under the InfoGAN framework, the first step is to generate samples $\mathbf{z}_g = [\mathbf{c}, \mathbf{s}, \mathbf{n}]$ from which data samples are generated using $\mathbf{x}_g = G_\theta(\mathbf{z}_g)$. These generated samples then are fed through the encoder to obtain the reconstructed latent variable $\widehat{\mathbf{z}}_g = E_\phi(G_\theta(\mathbf{z}_g))$. This then allows for the calculation of the mutual information $MI(\mathbf{c}, \mathbf{s}|G_\phi(\mathbf{z}_g))$ through

$$\mathcal{L}_{MI} = \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{s}\sim p(\mathbf{s}),\mathbf{n}\sim p(\mathbf{n})} \log(q_\phi(\mathbf{c}|G_\theta(\mathbf{c},\mathbf{s},\mathbf{n}))) + \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{s}\sim p(\mathbf{s}),\mathbf{n}\sim p(\mathbf{n})} \log(q_\phi(\mathbf{s}|G_\phi(\mathbf{c},\mathbf{s},\mathbf{n}))), \tag{2.76}$$

which is to be maximised under the InfoGAN framework and it is trivial to see that these two terms reduce to the cross-entropy loss and the Gaussian negative log-likelihood under the assumed latent element prior distributions. These can be given as

$$\mathcal{L}_{CE_1}(\theta, \phi) = \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{s}\sim p(\mathbf{s}),\mathbf{n}\sim p(\mathbf{n})} \left[-\sum_{k=1}^{K} \mathbf{c}_k \log \left[E^{\mathbf{c}}_{\phi,k}(G_\theta(\mathbf{z}_g)) \right] \right], \tag{2.77}$$

$$\mathcal{L}_{NLL_1}(\theta,\phi) = \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{s}\sim p(\mathbf{s}),\mathbf{n}\sim p(\mathbf{n})}\left[\frac{1}{2}\|\mathbf{s} - E^{\mathbf{s}}_{\phi}(G_\theta(\mathbf{z}_g))\|_2^2\right], \tag{2.78}$$

where the author chose to represent the components from the encoder used in each loss in the superscript of E_ϕ and $\|\cdot\|$ is the L_2 norm. As the InfoGAN framework builds on the GAN framework, generated samples $\mathbf{x}_g$ can also be evaluated through a discriminator to improve the generative capacity of the decoder network $G_\theta(\mathbf{z}_g)$. In this work, the original GAN objective function is used to train $D_\chi(\mathbf{x})$ while the KL divergence loss shall be used for $G_\theta(\mathbf{z})$. These losses are given independently as

$$\mathcal{L}_D(\chi) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log D_\chi(\mathbf{x})\right] - \mathbb{E}_{\mathbf{z}_g\sim p(\mathbf{z})}\left[\log(1 - D_\chi(G_\theta(\mathbf{z}_g)))\right], \tag{2.79}$$

$$\mathcal{L}_{G_{KL}}(\theta) = -\mathbb{E}_{\mathbf{z}_g\sim p(\mathbf{z})}\log\left[\frac{D_\chi(G_\theta(\mathbf{z}_g))}{1 - D_\chi(G_\theta(\mathbf{z}_g))}\right]. \tag{2.80}$$

A clear issue with DLS-GAN, up to this point, is that it appears to be only a purely generative model and cannot use any sampled data $\mathbf{x}\sim p(\mathbf{x})$ from the true data distribution to perform inference. However, the fix here is trivial as one has already made explicit assumptions of an encoder and decoder network. In this manner, the standard auto-encoder framework, part two of Figure 2.4, can be applied where the explicit assumption is made that the generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$ is Gaussian. If one assumes a deterministic decoder, the objective function is just the negative log-likelihood under a Gaussian distribution with unit variance given as

$$\mathcal{L}_{AE}(\theta,\phi) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\frac{1}{2}\|\mathbf{x} - G_\theta(E_\phi(\mathbf{x}))\|_2^2\right]. \tag{2.81}$$

For the observant reader, one may note that at this point the latent distributions appear to be unconstrained, while the author mentioned previously that an AAE framework is used. Interestingly, Ding and Luo (2019) only regularise the latent noise component, $\mathbf{n}$, through the use of the MMD objective function. This is interesting in two ways, they imply that to regularise $\mathbf{c}$ and $\mathbf{s}$ obtained from E_ϕ all that is required is to use the InfoGAN framework. In this way, it is implied that the cross-entropy and negative log-likelihood terms are sufficient to guide the encoder to encode data into the required prior forms. It is also interesting as it is directly contrastive to Zhou et al. (2019), who used adversarial latent regularisation techniques for all three latent components alongside the InfoGAN framework. In this work, it is required that one obtain a health indicator from the latent noise component and as such an adversarial regularisation technique shall be used. This method shall be that WGAN latent critic with gradient penalty, (Arjovsky et al., 2017, Gulrajani et al., 2017), which gives an objective function of the form

$$\mathcal{L}_D(\omega) = -\mathbb{E}_{\mathbf{n}\sim p(\mathbf{n})}\left[D_\omega(\mathbf{n})\right] + \mathbb{E}_{x\sim p(\mathbf{x})}\left[D_\omega(E^{\mathbf{n}}_{\phi}(\mathbf{x}))\right] + \lambda\mathbb{E}_{\tilde{\mathbf{n}}\sim p(\tilde{\mathbf{n}})}[\|\nabla_{\tilde{\mathbf{n}}}D_\omega(\tilde{\mathbf{n}})\|_2 - 1]^2, \tag{2.82}$$

$$\mathcal{L}_E(\phi) = -\mathbb{E}_{x\sim p(\mathbf{x})}\left[D_\omega(E^{\mathbf{n}}_{\phi}(\mathbf{x}))\right], \tag{2.83}$$

where care should be taken to see that the generator, in this case, is the encoder network E_ϕ. Please refer to Figure 2.4, part two for clarity in the entire AAE process. It is clear to note that up to this point that there is clear network consistency between the InfoGAN and AAE frameworks through the use of shared encoder and decoder networks.

The final element of the DLS-GAN is that of latent separation, which Ding and Luo (2019) approach in an interesting manner. Part three of Figure 2.4 should be referred to in this regard. In the use of this model, one would typically sample from the data and latent prior distributions to evaluate the expectations in each of the objective functions. From this point, the notation $\mathbf{c}_s, \mathbf{c}_s, \mathbf{c}_s \sim p(\mathbf{z})$ and $\mathbf{x}_r \sim p(\mathbf{x})$ shall be used for the samples. To enforce latent separation, a combined latent representation $\mathbf{z}'$ is used with elements $\mathbf{z}' = [\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_r]$ where $\mathbf{n}_r = E^{\mathbf{n}}_{\phi}(\mathbf{x}_r)$. From this combined latent representation, one needs to simply feed it through the encoder and decoder networks respectively, such that reconstructed latent representation is obtained $\tilde{\mathbf{z}}' = E_\phi(G_\theta(\mathbf{z}'))$. From this point, one can simply use the cross entropy and negative log likelihood loss functions on $(\mathbf{c}_s, \tilde{\mathbf{c}}_s)$ and $(\mathbf{s}_s, \tilde{\mathbf{s}}_s)$ to ensure that they are

recovered correctly and to ensure clear latent separation, one can use the negative log likelihood loss on $(\mathbf{n}_r, \tilde{\mathbf{n}}_r)$. The motivation here is straightforward, one can ensure that $\mathbf{n}_r$ is distinct and separable if, given a random class and continuous representation, it can still be recovered from the generated sample. This may be accomplished if the information contained in $\mathbf{n}_r$ does not contain any structure that is linked to $\mathbf{c}$ and $\mathbf{s}$ which would ultimately change when fed through the decoder. The objective functions required here are

$$\mathcal{L}_{CE_2}(\theta,\phi) = \mathbb{E}_{\mathbf{x}_r\sim p(\mathbf{x})}\mathbb{E}_{\mathbf{c}_s\sim p(\mathbf{c}),\mathbf{s}_s\sim p(\mathbf{s})}\left[-\sum_{k=1}^{K}\mathbf{c}_k\log\left[E^{\mathbf{c}}_{\phi,k}(G_\theta(\mathbf{z}'))\right]\right], \tag{2.84}$$

$$\mathcal{L}_{NLL_2}(\theta,\phi) = \mathbb{E}_{\mathbf{x}_r\sim p(\mathbf{x})}\mathbb{E}_{\mathbf{c}_s\sim p(\mathbf{c}),\mathbf{s}_s\sim p(\mathbf{s})}\left[\frac{1}{2}\|\mathbf{n}_r - E^{\mathbf{n}}_{\phi}(G_\theta(\mathbf{z}'))\|_2^2\right], \tag{2.85}$$

$$\mathcal{L}_{NLL_3}(\theta,\phi) = \mathbb{E}_{\mathbf{x}_r\sim p(\mathbf{x})}\mathbb{E}_{\mathbf{c}_s\sim p(\mathbf{c}),\mathbf{s}_s\sim p(\mathbf{s})}\left[\frac{1}{2}\|\mathbf{s}_s - E^{\mathbf{s}}_{\phi}(G_\theta(\mathbf{z}'))\|_2^2\right]. \tag{2.86}$$

The final objective function used in this model can be given as

$$\mathcal{L}(\theta,\phi,\chi,\omega) = \mathcal{L}_{GAN} + \lambda_{AE}\mathcal{L}_{AE} + \beta_1\mathcal{L}_{WGAN} + \beta_2\mathcal{L}_{NLL_2} + \beta_3\mathcal{L}_{CE_1} + \beta_4\mathcal{L}_{CE2} + \beta_5\mathcal{L}_{NLL_1} + \beta_6\mathcal{L}_{NLL_3}, \tag{2.87}$$

where this objective function depends on one λ_{AE} parameter and six β parameters. Note that λ_{AE} is different from the λ used in the gradient penalty applied to $\mathcal{L}_{WGAN}$. In the work of Ding and Luo (2019), there were fewer parameters due to the lack of a disentangled continuous component. The λ parameter can be seen as a method to increase the reconstruction term in the objective function, while the β parameters control the enforcement of the latent disentanglement and separation terms. It is also recommended that $\beta_1 = \beta_2, \beta_3 = \beta_4$ and $\beta_5 = \beta_6$ to allow for reasonable control of the strength of latent components in the objective function. For a clear training procedure to apply to the DLS-GAN, please see Appendix B.4.

2.10 Representation Yielding GAN

A Representation Yielding GAN (RY-GAN) is proposed in this work as an alternative to the DLS-GAN to provide some method of compatibility between the generative models used in this work. The purpose of this technique is to try and allow for deeper decoder network information capture to allow for a latent representation that captures the crucial information about the data. RY-GAN is a variant of the work of Rosca et al. (2017) that takes into consideration the presence of continuous and categorical latent components. Ultimately, this technique attempts to unify improvements found in literature in the case of models that aim to incorporate GANs into an auto-encoder framework. The main component used from Rosca et al. (2017) was the treatment of the decoder network as a generative distribution but built into an auto-encoder, which aims to improve the expressibility of the decoder network to discourage generator mode collapse. This is a subtle but fundamental difference to the DLS-GAN as the DLS-GAN discriminator only sees samples that are purely generative.

RY-GAN also aims to integrate latent disentanglement through an InfoGAN-like framework with an explicit focus on only the decoder network. The motivation here is akin to the generator in the InfoGAN framework, with the Q distribution serving as a means to guide G into disentanglement. The decoder network is driven towards a point where it can utilise latent information to generate signals that contain similar semantic meaning to the latent samples from which they were obtained and captures the misalignment that the encoder would produce. This misalignment is obvious if one does not update the encoder from the InfoGAN framework, however, it is proposed that this step is not required if latent regularisation and auto-encoder based updates are used in the training framework as the encoder is already guided to produce a suitable latent representation.

In this way, the encoder network serves primarily to capture the constraint of prior regularisation for the three latent elements while the decoder serves to utilise the latent representation for disentanglement and improved generative performance. Figure 2.5 shall be referred to often, with two sections that stem from the AAE-based framework and the InfoGAN framework. For this discussion to take place, the author presents five key model elements, namely, the posterior (encoder) distribution $q_\phi(\mathbf{z}|\mathbf{x})$, the generative (decoder) distribution $p_\theta(\mathbf{x}|\mathbf{z})$, a data discriminative distribution $p_\chi(t=1|\mathbf{z})$, a noise latent Wasserstein metric function $f_\omega(\mathbf{z_n})$ and a discrete latent discriminative distribution $p_\varsigma(t=1|\mathbf{z_c})$. As this is a deep learning application, these distributions are represented by parametric functions $E_\phi(\mathbf{x}), G_\theta(\mathbf{z}), D_\chi(\mathbf{x})$, $D_\omega(\mathbf{z_n})$ and $D_\varsigma(\mathbf{z_c})$ respectively. This notation is preserved between RY-GAN and the DLS-GAN as many equations can be re-used in the RY-GAN case.

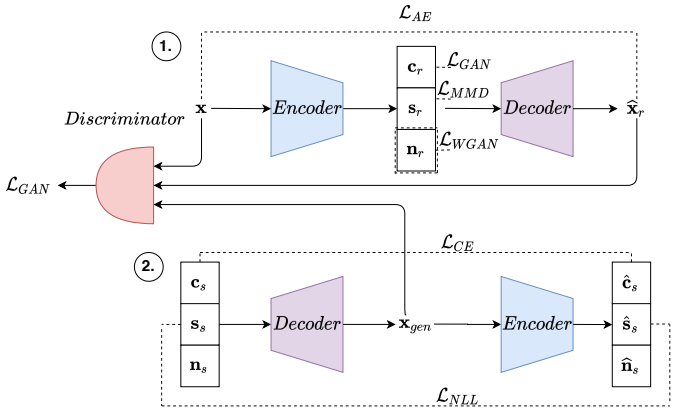


Figure 2.5. A complete overview of the RY-GAN model architecture with two main components, denoted through numbered circles. Part one refers to the AAE framework with latent regularisation while part two refers to the InfoGAN-based framework. Notice the subtle inclusion of the $\mathbf{x}_r$ in the discriminator.

The first component that shall be elaborated on is part one of Figure 2.5 as it is primarily an auto-encoder framework with additional latent regularisation techniques. In the standard auto-encoder framework a Gaussian distribution is assumed for the generative distribution $p_\theta(\mathbf{x}|\mathbf{z})$. For a the deterministic Gaussian distribution, Equation (2.81) can be used. For the case of latent regularisation, the author chose to deviate from the DLS-GAN approach and rather penalise the encoder directly as it is not required to create latent misalignment during training in part two of Figure 2.5. For the latent noise component, a WGAN latent critic with gradient penalty is to be used, with the critic objective functions given in Equation (2.82) and the encoder objective function given in Equation (2.83). The part that deviates from the DLS-GAN approach is that one now penalises the posterior distribution components $q(\mathbf{c}|\mathbf{x})$ and $q(\mathbf{s}|\mathbf{x})$ to match a discrete categorical distribution and a isotropic Gaussian distribution, which are the assumed prior forms of $p(\mathbf{c})$ and $p(\mathbf{s})$ respectively. For the categorical prior constraint, a discriminator is used with an assumed constant amount of instance noise $\sigma_n \sim \mathcal{N}(0,\sigma^2)$ added to each instance that D_ς sees. This was done as the author found that this constant noise aided in training as during the initial stages of training this discriminator over-fits to the fact that one element is perfectly one while the rest are exactly zero, which is unlikely under a softmax activation function which one typically uses when using networks for multi-class classification. The objective functions associated with D_ς and E_ϕ are given as

$$\mathcal{L}_D(\varsigma) = -\mathbb{E}_{c\sim p(\mathbf{c}),\varepsilon\sim\sigma_n}\left[\log D_\varsigma(\mathbf{c}+\varepsilon)\right] - \mathbb{E}_{x\sim p(\mathbf{x}),\varepsilon\sim\sigma_n}\left[\log(1-D_\varsigma(E^{\mathbf{c}}_\phi(\mathbf{x})+\varepsilon))\right], \tag{2.88}$$

$$\mathcal{L}_{E_{KL}}(\phi) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\varepsilon\sim\sigma_n} \log \frac{D_\varsigma(E^{\mathbf{c}}_\phi(\mathbf{x})+\varepsilon)}{1-D_\varsigma(E^{\mathbf{c}}_\phi(\mathbf{x})+\varepsilon)}. \tag{2.89}$$

The next element that requires latent regularisation is $\mathbf{s}$ for which the author has two options, the first is to use a discriminator as was the case in the work of Zhou et al. (2019) and the second option is to use an MMD divergence approach. In this work, the latter shall be used as MMD offers optimisation stability and implementation simplicity. The MMD objective function can be given as

$$\mathcal{L}_{MMD}(\phi) = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}),\mathbf{s}'\sim p(\mathbf{s})}\left[k(\mathbf{s},\mathbf{s}')\right] + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\mathbf{x}'\sim p(\mathbf{x})}\left[k(E^{\mathbf{s}}_\phi(\mathbf{x}),E^{\mathbf{s}}_\phi(\mathbf{x}'))\right] - 2\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}),\mathbf{x}\sim p(\mathbf{x})}\left[k(\mathbf{s},E^{\mathbf{s}}_\phi(\mathbf{x}))\right], \tag{2.90}$$

where $k(\cdot,\cdot)$ is a Gaussian kernel. For part two of Figure 2.5, a variant of the InfoGAN framework is used, whereby only the decoder is updated to ensure that MI maximisation occurs. Under the InfoGAN framework, the first step is to generate samples $\mathbf{z}_g = [\mathbf{c},\mathbf{s},\mathbf{n}]$ from which data samples are generated using $\mathbf{x}_g = G_\theta(\mathbf{z}_g)$. These generated samples then are fed through the encoder to obtain the reconstructed latent variable $\widehat{\mathbf{z}}_g = E_\phi(G_\theta(\mathbf{z}_g))$. Using the MI representation in Equation (2.76) the CE and NLL objective function in the case of RY-GAN are almost identical to those given in Equation (2.77) and Equation (2.78) respectively, with a deviance in the optimised elements in the losses $\mathcal{L}_{CE}(\theta)$ and $\mathcal{L}_{NLL}(\theta)$. This notation is used to indicate that only the decoder is updated when these objective functions are used. The final component of RY-GAN is that of the data discriminator, which follows an intuition used in the $\alpha - GAN$ approach proposed by Rosca et al. (2017). The intuition here is that one uses generated samples, $\mathbf{x}_g = G_\theta(\mathbf{z}_g)$, and the reconstruction of samples $\mathbf{x}\sim p(\mathbf{x})$ passed through the auto-encoder, $\mathbf{x}_r = G_\theta(E_\phi(\mathbf{x}))$, to update D_χ and G_θ respectively. The former case is to improve the training performance and the latter is to improve the generative capacity of the decoder network. The motivation for this choice is also detailed in Rosca et al. (2017). The power in this approach lies in its ability to rather use one discriminator. The two objective functions obtained from this approach, using the KL divergence GAN loss formulation for generator updates, can be given as

$$\begin{aligned}\mathcal{L}_D(\chi) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log D_\chi(\mathbf{x})\right] - \frac{1}{2}\Big(&\mathbb{E}_{\mathbf{z}_g\sim p(\mathbf{z})}\left[\log(1-D_\chi(G_\theta(\mathbf{z}_g)))\right] \\ &+\mathbb{E}_{x\sim p(\mathbf{x})}\left[\log(1-D_\chi(G_\theta(E_\phi(\mathbf{x}))))\right]\Big),\end{aligned} \tag{2.91}$$

$$\mathcal{L}_{G_{KL}}(\theta) = -\frac{1}{2}\left(\mathbb{E}_{\mathbf{z}_g\sim p(\mathbf{z})}\left[\log\frac{D_\chi(G_\theta(\mathbf{z}_g))}{1-D_\chi(G_\theta(\mathbf{z}_g))}\right] + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log\frac{D_\chi(G_\theta(E_\phi(\mathbf{x}))}{1-D_\chi(G_\theta(E_\phi(\mathbf{x}))}\right]\right), \tag{2.92}$$

where there is a clear usage of a factor of $\frac{1}{2}$, which was used in this work to allow for the GAN equilibrium position to be kept to 0.5. The overall objective function that is to be applied to a RY-GAN approach can be given as

$$\mathcal{L}(\theta,\phi,\chi,\omega,\varsigma) = \alpha\left(\lambda_{AE}\mathcal{L}_{AE}\right) + (1-\alpha)\mathcal{L}_{MMD} + \mathcal{L}_{WGAN} + \mathcal{L}_{GAN_{\mathbf{c}}} + \mathcal{L}_{CE} + \mathcal{L}_{NLL}, \tag{2.93}$$

where there are two factors used to balance the training, namely α and λ_{AE}. The former aims to control the influence of the GAN and the reconstruction terms on the decoder network while the latter is used to increase the emphasis of the reconstruction term in the objective function. For a clear training procedure for the RY-GAN, please refer to Appendix B.4.

The use of the DLS-GAN and RY-GAN approaches is to try and obtain a model that has an improved generative capacity through the adversarial training scheme as well as allowing the model to perform model inference. To ensure that model inference and latent disentanglement is possible, techniques from the InfoGAN framework and the Variational inference framework are used in combination to create a model that not only offers model inference but also improved disentanglement and generative capacity. For vibration data, the addition of an adversarial latent critic for the $\mathbf{n}$ latent component offers a metric that can be used to track the latent representation of signal segments with a scalar value. This metric can be analysed in conjunction with the latent health metrics that are detailed in the next chapter.

Chapter 3 Data-Driven Condition Monitoring

3.1 Chapter Abstract

In this section, a succinct model evaluation analysis shall be given and detailed in the context of vibration-based condition monitoring. To initialise this analysis, a clear context is given to the various latent variable models considered in this work to detail the relationship between the models. After this analysis, a discussion is required regarding the application of latent variable models for condition monitoring. The purpose of this discussion is twofold, it is to clarify what health indicators may be available from the various latent variable models and to clarify how current model evaluation practices are performed and how they can be improved.

3.2 Latent Manifolds in Latent Variable Models

In this work, the considered latent variable models are, namely *PCA*, *VAE*s and $\beta - TC - VAE$s with a deterministic and stochastic parametrisation which is a term given to the decision to assume an identity output covariance or a learnt output covariance, the $DLS - GAN$ and the proposed $RY - GAN$ methodology. In these latent variable models methodologies, there is a clear progression in the linearity of the latent manifold and the approach taken to introduce latent disentanglement. *PCA* is a linear latent variable model that uses a linear transformation to perform the transition to and from the latent manifold. This transition is facilitated through the eigenvectors of the training data covariance matrix and there is a natural structuring of the eigenvectors from the largest eigenvalue to the smallest. In this discussion, the term transition function is used to describe any parametric function that is used to transition between the input space and the data space. For the latent variable models considered in this work, a transition function is non-invertible and therefore two transition functions are required for models used for both data generation and model inference.

The assumption of transition function linearity assumes that the latent manifold exists on a linear hyperplane in the latent manifold. This linearity, by design, can be problematic for two reasons. The first is that if there exists any non-linearity in the data, the transition functions will be incapable of handling this non-linearity and the effect will be a complex and entangled latent manifold. The second reason is that the generative and posterior distributions for *PCA* are linear Gaussian distributions, thus the model is designed for data that only consists of Gaussian data. The result of this distribution choice assumes that the data seen by the model is Gaussian, which can be problematic if the data contains of non-Gaussian components.

The purpose of a *VAE* is to move away from linear transition functions as *VAE*s use neural networks to introduce transition functions that are non-linear. The power of a neural network lies in its ability to introduce a parametric function that is flexible and non-linear through the use of non-linear activation

functions. This addition of non-linearity offers flexibility in the latent manifold through a non-linear embedding of the data in the latent space. *VAE*s do keep the assumption of Gaussian generative and posterior distributions, which shares the same limitation identified for *PCA*. The issue often associated with *VAE*s is their ability to capture the factors of variation in the data, with the work of Burgess et al. (2018) indicating that penalisation on the *KL* divergence term in the *VAE* objective function is a capable method of enforcing latent disentanglement. However, this is an implicit disentanglement technique and in the work of Locatello et al. (2019), the idea of latent disentanglement in models that attempt to obtain a factorised posterior distribution is challenged. This decision leads to models that use alternative techniques to explicitly enforce latent disentanglement though the use of MI.

The MI term used in the $InfoGAN$ approach proposed in Chen et al. (2016) provides an explicit method of ensuring that specific latent codes are used by the generator of a *GAN* and that these codes capture important information in the data. This is done by using a network to recover specific latent codes from any generated data. The use of MI can be seen as an explicit disentanglement technique as the network is penalised for producing samples that do not use the latent codes effectively. By incorporating this approach into a framework that considers improving the generative capacity of a latent variable model, the goal for the $DLS-GAN$ and $RY-GAN$ is to obtain a generative distribution that is more flexible and a latent manifold that better captures the factors of variation in the data. This is done by segmenting the latent space into three components, where two components are trained to maximise the MI between the latent variables and the generated data and the third is used to capture any additional information in the data. The $DLS-GAN$ and $RY-GAN$ models use non-linear transition functions, MI and the adversarial *GAN* framework to try and capture the complexity of the data distribution and improve the quality of the latent manifold.

One fundamental detail that must be made clear is that *PCA* is a computationally robust and efficient technique, while *VAE*s and *GAN*s are computationally expensive and less robust during training. This difference allows for *PCA* to be seen as a baseline method and can rationalise the performance of the metrics from the *VAE* and *GAN*-based methods. This can also allow for the quantification of model linearity versus non-linearity and the effect of latent manifold disentanglement.

The purpose of obtaining latent manifolds of good quality is that this work is focused on the latent manifold response to anomalous instances in data. If the latent manifold is highly entangled or equivalent to random noise, the ability to detect the presence of anomalous instances becomes non-trivial and may be infeasible. if the manifold is able to capture the factors of variation in the data through flexible, non-linear transition functions then the process of detecting subtle changes is simplified significantly. As the learnt latent manifold has no knowledge of anomalous data, it is expected that this data will affect the traversal through this manifold in time, where the aim of this work is to detect and identify these changes. In the next section, the proposed Pseudo-Time analysis framework and the various detection metrics are proposed and quantified for the reader.

3.3 Pseudo Time Analysis

The potential that this work aims to highlight is how current unsupervised deep learning approaches tend to make somewhat naive choices when analysing the performance of the models, with the final objective often cast in a format of a binary fault detection scheme, albeit often being an implicit rather than explicit choice. The key element of interest up to this point has been providing evidence that unsupervised deep learning techniques can detect faults in vibration data. This is a promising investigative approach, but it is limiting in that one may never know the fault type and that it never

exploits a key element in any vibratory signal: time. This work will show how this implicit choice comes about and what its relationship is with common data partitioning practices and post-processing analyses often performed in literature.

3.3.1 Vibration Data Preparation

In deep learning, there is a strong driving force to reduce the dependency on feature engineering and allow the model to perform automatic feature extraction. The processing methodology often driven for in CBM-based deep learning is the use of the raw-vibration signal. The problem, however, is that one does typically not have large amounts of raw vibration data for a system's unhealthy state and this data is often obtained with a high sampling frequency. The high sampling frequency is problematic as this results in a model input dimensionality that is often computationally infeasible. A solution to this often used in literature is to use a *direct partitioning* scheme, as was the case in the works of Booyse et al. (2020), Baggeröhr (2019) and San Martin et al. (2019), whereby a prescribed model window length L_w is effectively used to reduce model input dimensionality and increase the size of the training dataset. This window is then randomly moved in a signal or is treated as a moving window with a certain overlap percentage between windows (Booyse et al., 2020, San Martin et al., 2019). The next steps are then straightforward, partition all vibration data using the *direct partitioning* scheme, train a model and then evaluate the model on all data with the objective to determine whether the model can detect damage. In this analysis, often one utilises a HI, which is a broad term given to metrics used to detect damage. In this work, there are three potential HI estimates, namely,

$$HI^{(1)}(\mathbf{x}, \tilde{\mathbf{x}}, \boldsymbol{\sigma}^2) = -\frac{1}{D}\sum_{k=1}^{D}\frac{(\tilde{x}_k - x_k)^2}{\sigma_k^2}, \tag{3.1}$$

which is the reconstruction log-likelihood for an input $\mathbf{x}$ and its reconstructed mean $\tilde{\mathbf{x}}$ and variance $\boldsymbol{\sigma}^2$ obtained from the explicit assumption that the generative distribution $p(\mathbf{x}|\mathbf{z})$ is Gaussian. Note that the terminology and notation used here is broad and not crucial to understand at this time. The literature study will enhance the readers understanding of the HIs. This term can also be considered to be the negative squared Mahalanobis distance under a factored Gaussian distribution. The second is the data discriminator likelihood estimate

$$HI^{(2)}(\mathbf{x}) = D_x(\mathbf{x}), \tag{3.2}$$

which is based on the input feature space $\mathbf{x}$. The rationale of $HI^{(1)}$ is a estimate of the likelihood that the data $\mathbf{x}$ is from the true data distribution. For this work, the data discriminator provides an estimate of how likely any observed data is from the healthy asset data distribution. The third health indicator is that of the Wasserstein metric estimate

$$HI^{(3)}(\mathbf{n}) = D_n(\mathbf{n}), \tag{3.3}$$

which is based on the latent feature space $\mathbf{n}$. The rationale behind this metric is a scalar measure of the deviation of a given latent variable from the learnt latent manifold in the $\mathbf{n}$ space. Typically, one must use a HI to determine the machine condition given any signal, which leads to the use of statistical features such as the mean or RMS of the HI values for a signal. Another treatment perspective is that the partitioning scheme coupled with a healthy indicator gives rise to a discrepancy signal, which one obtains by preserving the sequential order in which signal segments are obtained. One can then calculate the statistical features of this discrepancy signal to arrive at an estimate of the machine condition.

The objective of classic vibration-based anomaly detection has now become apparent, for a signal to be *classified* as anomalous from the training data all of its segments need to be different to those in the reference healthy data. This can be problematic, as it is now crucial that each signal segment contain some indication of damage. One can now link properties of rotating machinery such as fault frequency, fault dynamics and model window length. This observation is best realised through the use

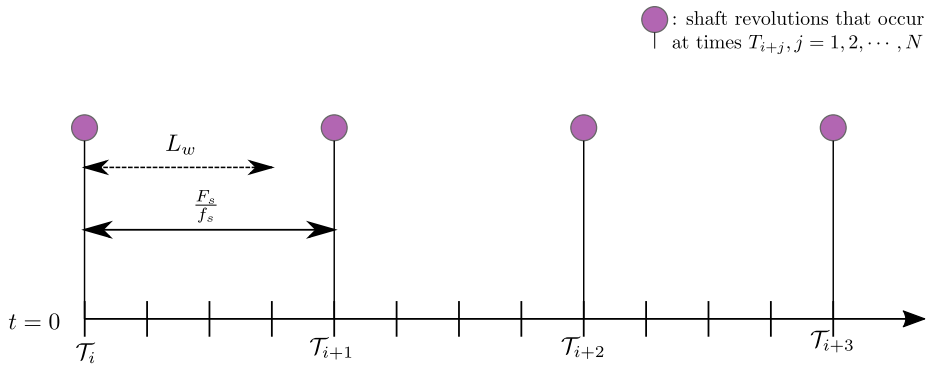


Figure 3.1. The interaction between model window length L_w, signal sampling frequency F_s and the shaft speed f_s. Note here that the shaft speed is used as a proxy for the presence of faults, as faults occur proportional to the shaft speed.

of a simple thought experiment that uses the idea that for any rotating machinery, fault frequencies are proportional to shaft speed. A relation between sampling frequency F_s and shaft speed f_s can give an approximation of a window length that will capture one rotation of the shaft, given as

$$L_w \geq \frac{F_s}{f_s}, \tag{3.4}$$

where F_s is the sampling frequency. One can also make the reasonable assumption that the faults in gearbox condition monitoring will occur in proportion ≥ 1 to the shaft frequency. Therefore, if the objective is to capture fault responses in the data, the window length should, as current literature stands, capture at least one shaft revolution within the window. An obvious problem now lies in slow operating condition cases or faults subjected to fault dynamics such as moving in and out of a loading zone, in the case for inner race faults. In this case, if one neglects to allow for interpretation of how model window length is linked to the fault information they wish to detect, they may end up with results skewed based on implicit assumptions they made. For the case of non-stationary or time-varying operating conditions, an alternative explanation based on the lower bound of Equation 3.4 follows this intuition, given as

$$f_{s_{min}} = \frac{F_s}{L_w}, \tag{3.5}$$

which states that to detect, at a minimum, once per revolution faults, one must ensure that shaft speed is bound by the ratio of the sampling rate and the chosen window length. It is also clearly evident that if one uses a very high sampling frequency, this implicit assumption is also affected as one may have to use large window lengths. In Figure 3.1, a visual representation of this idea is given. It is simple to note that unless the sampling rate, which discretises the measured signal, is low and the shaft speed is high, the window length will never be large enough to ensure that each segment contains damage.

In this work, the author will show that to interpret deep learning model results correctly, one must not only be aware of their implicit assumptions but also exploit the time domain of the vibration data to give a better understanding of what the model is potentially responding to and what is gained by using more complex deep learning methods. It shall also show how current practices limit the exploratory power available to the user and may, in some applications, give results that one may struggle to interpret. Before this statement is clarified, let us introduce a simple re-casting of the evaluation analysis step that most authors tend to neglect. It is often common that one takes the entire dataset and follows the direct partitioning scheme for every signal available. It is proposed here that a simple *temporal*

preservation analysis approach, whereby one incrementally shifts the window over a signal and then evaluate and store the HI values for each increment. This approach is related to discrepancy analysis techniques however this has now been extended to data-driven approaches (Heyns et al., 2012b,a, Schmidt et al., 2019a). This approach then gives a discrepancy signal of length $L_d = L_s - L_w$, where L_d is the discrepancy signal length and L_s is the discrete time-domain signal length. Figure 1.3 contains a visual example of how one would typically process vibration data under the *temporal preservation* analysis approach.

The ramifications of the implicit assumption made through L_w can, however, reduce with fault frequency and shaft speed. For the former, if the fault frequency is proportional at a sufficiently large rate, where the ambiguity here is due to application dependency, one may be guaranteed to obtain faults in every signal segment. For the latter, in the case of sufficiently high shaft speed, anomalies may be present in every signal segment. Another limitation is that as fault frequency and shaft speed increase, one may be tempted to reduce the window length accordingly, but this may result in unreasonably inflexible network architectures. However, the benefits of the *temporal preservation* analysis shall always be present and it applies to other applications, such as interpreting the latent space.

3.3.2 Latent Space Analysis and Metrics

Another missing component in deep learning approaches is an analysis of the latent space and its response to unhealthy data. A large number of latent variable models are detailed and proposed in literature, with a few mentioned in Chapter 2 in this work. However, in the case of vibration-based condition monitoring, there has been no manner of comparing the benefits of different techniques and no unified method of comparing models critically against one another. Here the author is referring to alternative formulations of the same methodology, common in *VAE* literature. The initial motivation for the use of complex latent-variable models is that they may provide some bounds on where healthy data is in the latent space, through latent-variable model regularisation. However, to the best of the author's knowledge, there has been no case in the literature of an in-depth latent space analysis of models trained on vibration data. Note that it may be beneficial for the reader to keep Figure 1.7(b) in mind when reading the following section.

In this work, the author will show that for proper latent space interpretation, one does not only have access to *HI*s but also Latent Health Indicators (*LHI*s), which provides a platform for deeper and more directed research into latent-variable models and their application in vibration data. The *LHI*s are simple to compute and implement in the *temporal preservation* analysis approach detailed in this work and allow for an in-depth and deeper understanding of the role of the latent space in PHM. For one to understand the latent space and its response to damage, one must first incrementally shift the signal observation window through time, to develop a state-space representation of the latent space. The term state-space is used here to refer to the time component included in the latent space, as shown in Figure 1.7(b). Three latent metrics are proposed in this work, where these metrics are trivial to compute and are given as

$$LHI_t^{(1)} = \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2, \tag{3.6}$$

$$LHI_t^{(2)} = \|\mathbf{z}_t\|_2, \tag{3.7}$$

$$LHI_t^{(3)} = \cos^{-1}\left(\frac{\mathbf{z}_{t+1}^T \mathbf{z}_t}{\|\mathbf{z}_{t+1}\|_2 \|\mathbf{z}_t\|_2}\right), \tag{3.8}$$

where $\mathbf{z}_t$ indicates the latent space representation at any point $t \in [0, L_d - 1]$ from the *temporal preservation* analysis procedure and $\|\cdot\|_2$ is the L_2 norm. $LHI^{(1)}$ is the trivial calculation of the latent representation Euclidean norm. $LHI^{(1)}$ can be interpreted as the latent distance norm between two time-continuous interval points and allows one to interpret the inter-time distance characteristics of

the latent space. $LHI^{(2)}$ can also be interpreted, for latent space representations enforced to be an isotropic Gaussian, as the Euclidean norm. This metric directly measures the projection of data from the origin. $LHI^{(3)}$ can be interpreted as the angle between two points in the latent space and allows one to interpret the directional characteristics of the latent space. $LHI^{(1)}$ can analyse the latent velocity or latent distance, which reduces to looking at either the average or total of the $LHI^{(1)}$. This phenomenon exists by considering the discrete velocity

$$\mathbf{v}_t = \frac{\mathbf{z}_{t+1} - \mathbf{z}_t}{\Delta t}, \tag{3.9}$$

where Δt is a constant under the *temporal preservation* analysis procedure if the signal sampling rate is constant. One can then look at the average velocity norm

$$v_{avg} = \frac{1}{L_d - 1} \sum_t \|\mathbf{v_t}\|_\mathbf{2}, \tag{3.10}$$

however if Δt is treated as constant, one can rather use

$$v_{avg} = \frac{1}{L_d - 1} \sum_t \|\mathbf{z_{t+1}} - \mathbf{z_t}\|_\mathbf{2}, \tag{3.11}$$

as Δt is simply a scalar on the norm. One can also analyse the *total path distance* by considering a discretisation of the latent distance integral

$$\begin{aligned} x_{path} &= \int_{t=0}^{t=L_d-1} \|\mathbf{v_t}\|_2 dt \\ &= \sum_t \|\mathbf{v_t}\|_2 \Delta t \\ &= \sum_t \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2. \end{aligned} \tag{3.12}$$

It shall be shown in this work how the combination of the *temporal preservation* analysis and the proposed *LHI*s can lead to an interpretable latent space and introduce insight into how latent variable models respond to anomalous data. The term interpretation is used to describe how these metrics quantify the latent manifold response to anomalous instances. The significance here is that these metrics provide a means of understanding the dynamics of latent manifolds for time-series data and understanding the traversal through the latent manifold. The *LHI*s are trivial to compute but the fundamental principle on which they operate is linked to the manifold hypothesis and the basis of anomaly detection. It is assumed that a model trained on reference vibration data, presumed to be healthy, should be incapable of understanding vibratory data that contain anomalous instances. If so, there should be some measurable model response to damage, whether it be in the latent space or the data space. In Table 3.1, the model analysed in this work and their available *HI*s and *LHI*s are succinctly noted for the reader.

Table 3.1. The available health and latent health indicators for the different latent variable models considered in this work.

Model Type	$\mathbf{HI}^{(1)}$	$\mathbf{HI}^{(2)}$	$\mathbf{HI}^{(3)}$	$\mathbf{LHI}^{(1)}$	$\mathbf{LHI}^{(2)}$	$\mathbf{LHI}^{(3)}$
PCA	✓	✗	✗	✓	✓	✓
$VAE_{1\|2}$	✓	✗	✗	✓	✓	✓
$\beta - TC - VAE_{1\|2}$	✓	✗	✗	✓	✓	✓
$RY - GAN$	✓	✓	✓	✓	✓	✓
$DLS - GAN$	✓	✓	✓	✓	✓	✓

Chapter 4 Phenomenological Model Dataset Analysis

4.1 Chapter Abstract

In this chapter, the author presents the phenomenological model and the performance investigation of the different models considered in this work. There are four key concepts that are key to this investigation:

1. The model window length L_w affects model diagnostic performance
2. The latent manifold is interpretable under the *temporal preservation* approach
3. Model complexity and the progression thereof needs to be highlighted and understood for applicability
4. Model performance must be compared to fully highlight the benefits of complex methods

The reader is asked to keep these concepts in mind when going through the various results, as each dataset offers insights into each of these points. For a detailed collection of the model architectures, learning rates, stopping conditions and hyper-parameters please refer to Appendix B.5. The models used on this dataset are: *PCA*, the VAE_1 and VAE_2 models, a $\beta - TC - VAE$ model with both the unit and learnt output variance denoted as $\beta - TC - VAE_1$ and $\beta - TC - VAE_2$, the $RY - GAN$ model and the $DLS - GAN$ model. This dataset was chosen as it allows for clear user control of the parameters of the data, which allows for an in-depth analysis of the metric response that is not possible with other datasets where certain properties may not be known.

4.2 Dataset Introduction

The phenomenological model was proposed in Abboud et al. (2017) where the model was designed to reproduce the response of a gearbox system with a local gear fault and a distributed wear bearing fault. This model was developed to be a function of the rotational speed and aims to capture the physical phenomena that occur as the rotational speed changes. Here the author is referring specifically to the amplitude modulation that is present in time-varying operating conditions (Urbanek et al., 2017, Schmidt et al., 2019a).

The vibration signal proposed by Abboud et al. (2017) consists of four separate additive elements, given as

$$x(t) = x_{g_d}(t) + x_{g_r}(t) + x_n(t) + x_b(t), \tag{4.1}$$

where the four elements are: the deterministic gear component $x_{g_d}(t)$, the random gear component $x_{g_r}(t)$, the noise component $x_n(t)$ and the bearing component $x_b(t)$. The latter can be composed of an outer race or inner race fault given as $x_{b_o}(t)$ or $x_{b_i}(t)$ respectively. These components are then used

to account for transmission path modulation, which is typical for vibration measurement cases. The transmission path that Abboud et al. (2017) account for is that from the source of excitation to the accelerometer. For all components other than the noise component, this can be expressed as

$$x_{g_d}(t) = h_{g_d}(t) \otimes z_{g_d}(t), \tag{4.2}$$

$$x_{g_r}(t) = h_{g_r}(t) \otimes z_{g_r}(t), \tag{4.3}$$

$$x_b(t) = h_b(t) \otimes z_b(t), \tag{4.4}$$

where h_i is the impulse response function that is convolved, where $\otimes$ denotes the convolution operator, with the true excitation signal z_i. Abboud et al. (2017) state that these impulse responses are those typically found in single degree of freedom sources, which is given in Schmidt et al. (2019a) as a viscously under-damped response function

$$h_i(t) = \exp^{-\xi_i \omega_{n,i} t} \sin\left(\sqrt{1-\xi_i^2}\omega_{n,i} t\right), \tag{4.5}$$

where ξ_i is the damping ratio of component i with an assumed natural frequency $\omega_{n,i}$ in $\frac{rad}{s}$. Next, one can document the source excitations for each component, where the source excitation for the deterministic gear component is given by Abboud et al. (2017) as

$$z_{g_d}(t) = M_{g_d}(\omega_{ref}(t))(1+\mathcal{I}(\theta_{ref}(t)))\sum_{j}^{N_{g_d}} a_{g_d}^{(j)} \cos(jN_{t_g}\theta_{ref}(t) + \varphi_{g_d}^{(j)}), \tag{4.6}$$

where $M_{g_d}(\omega_{ref}(t))$ is a modulation function that is based on the rotational speed, to account for the signal modulation present in varying speed conditions which is detailed and elaborated on in Urbanek et al. (2017). The term $\mathcal{I}(\theta_{ref}(t))$ is used by Abboud et al. (2017) to model gear fault impacts as a function of the shaft position, however, this term shall be excluded in this work as the aim of this dataset is to only analyse bearing fault cases. Inside the summation of Equation (4.6), N_{g_d} is the number of gear mesh components, with $a_{g_d}^{(j)}$ and $\varphi_{g_d}^{(j)}$ referring to the amplitude and phase of the j^{th} mesh component. Finally, N_{t_g} is the number of teeth on the gear considered and $\theta_{ref}(t)$ is the angular position of the shaft, given by the integration of the speed of the shaft

$$\theta_{ref}(t) = \int_0^t \omega_{ref}(\tau) d\tau. \tag{4.7}$$

Consider, briefly, that the speed of the shaft is modelled at a constant speed, the integration of the speed of the shaft then results in $\theta_{ref}(t) = \omega t$. The frequency of the deterministic component is $\omega_d = jN_{t_g}\omega$, which is exactly the form for the gear mesh frequency given in Equation (1.2), with the summation indicator j used for the harmonics of the mesh frequency which should typically be modelled at a lower amplitude. The next component is that of the random gear component whose purpose is to manifest as modulated white noise that can be a simulation of distributed gear damage which, as documented by Schmidt et al. (2019a), aims to complicate the inference procedure for bearing damage detection. This term is given as

$$z_{g_r}(t) = M_{g_r}(\omega_{ref}(t))\varepsilon_{g_r}(t)\sum_{j}^{N_{gr}} a_{g_r}^{(j)} \cos(j\omega_{ref}(t) + \varphi_{g_r}^{(j)}), \tag{4.8}$$

where $M_{g_r}(\omega_{ref}(t))$ is a modulation function, $a_{g_r}^{j}$ and $\varphi_{g_d}^{(j)}$ are the amplitude and phase of the random components and $\varepsilon_{g_r}(t)$ is the white noise component of the signal. Abboud et al. (2017) use the gear ratio to alter the phase of the signal, however this term is just set to one by Schmidt et al. (2019a). The white noise component is modelled as a zero-mean univariate Gaussian

$$\varepsilon_{g_r}(t) \sim \mathcal{N}(0, \sigma_{g_r}^2), \tag{4.9}$$

that has a variance $\sigma_{g_r}^2$ and at each time instant a random sample is drawn from this noise distribution. The noise term is modelled as amplitude modulated white noise

$$x_n(t) = \varepsilon_n(t) M_n(\omega_{ref}(t)), \tag{4.10}$$

where $M_n(\omega_{ref}(t))$ is an amplitude modulating function and the noise component $\varepsilon_n(t)$ is given as

$$\varepsilon_n(t) \sim \mathcal{N}(0, \sigma_n^2), \tag{4.11}$$

which is a zero-mean univariate Gaussian with a set variance σ_n^2. The final term that is modelled is the bearing component, which is given by Schmidt et al. (2019b) for an outer race fault as

$$z_{b_o}(t) = M_b(t) \sum_{i}^{N_\mathcal{T}} F_{dam_o}^{(i)} \delta(t - \mathcal{T}_i), \tag{4.12}$$

which is considered to be a train of Dirac delta functions that are all centred at different $\mathcal{T}_i$ terms. These centres are based on the type of bearing, the slip of the bearing and the shaft speed. To determine these terms one typically uses the expected angle of impacts in the angle domain and then converts to the time domain. Gryllias and Antoniadis (2012) and Schmidt et al. (2019a) introduced slip by making slight adjustments to the impact angle in each rotation, which was accomplished by sampling from a zero-mean univariate Gaussian distribution with a variance of 0.1. As indicated in Abboud et al. (2015) and Abboud et al. (2017), the term $\mathcal{T}_i$ can be modelled as $\mathcal{T}_i = t(i\theta_f - \mu_i)$, where θ_f is given as $\theta_f = \frac{2\pi}{\text{BPFO}}$ which is the angular period of the bearing fault. However, one can also use the instantaneous speed of the shaft and the expected angular impulse, based on the shaft orders of the fault, to determine when an impulse occurs in the angle domain and then convert this back to the time domain.

The bearing damage component is modelled as a term that one can sample from a univariate Gaussian distribution in the form $F_{dam_o} \sim \mathcal{N}(\overline{F}_{dam_o}, \sigma_{dam_o}^2)$, where one can then change the mean of this distribution to characterise the growth in the bearing magnitude. This growth change was made to be monotonically increasing over the different levels, where the signal-to-noise ratio (SNR) was then tuned for the various damage components. If one were to simulate healthy gearbox data, which is necessary for this work, one just needs to set $z_b(t) = 0$.

Finally, one can also model inner race bearing faults, whereby the methodology of implementation is primarily the same as that applied to an outer race fault. A key difference, however, is the presence of fault dynamics with the fault moving in an out of the bearing loading zone, synchronous with the shaft speed. It is reasonable to expect that bearing faults on the inner race, with a stationary outer race, exhibit periodic fault amplitude modulation related to the loading zone of the bearing. An inner race defect can be modelled as

$$z_{b_i}(t) = q(\theta_{ref}(t)) M_b(t) \sum_{i}^{N_\mathcal{T}} F_{dam_i}^{(i)} \delta(t - \mathcal{T}_i), \tag{4.13}$$

where again, one can use the BFPI to determine the impact locations in the angular domain and use the instantaneous speed of the shaft to track when impulses periodically occur based on the BFPI normalised by shaft speed. In this model, it is also expected that one includes bearing slip as a zero-mean univariate Gaussian with the same variance as the case for the outer race fault. As before, one cannot assume that the amplitudes are constant per rotation in the bearing signal and as such the amplitude is randomly sampled from a distribution $F_{dam_i} \sim \mathcal{N}(\overline{F}_{dam_i}, \sigma_{dam_i}^2)$. The modulation function $M_b(t)$ can be assumed to be the same form as the outer race fault, however now the function $q(t)$ is used to simulate the instantaneous loading of the bearing and if one assumes it to be a radial load, its form is

$$q(\theta_{ref}(t)) = \begin{cases} q_0[1 - \frac{1}{2\varepsilon}(1 - \cos\theta_{ref}(t))]^n & : |\theta_{ref}(t)| < \theta_{max}, \\ 0 & : \text{otherwise}, \end{cases} \tag{4.14}$$

where q_0 is the maximum load intensity, θ_{max} is the maximum angular range of the loading zone, ε is the load distribution factor and n is often set to $\frac{3}{2}$ for ball bearings (McFadden and Smith, 1984). In the

work of Gryllias and Antoniadis (2012), the following limits are given for the load distribution factor and the maximum angular range: $\varepsilon < 0.5$ and $\theta_{max} < \frac{\pi}{2}$. Finally, one needs to carefully define the angular ranges of the load distribution, with Schmidt et al. (2019b) suggesting that one shifts the angle to a range of $[-\pi, \pi]$, while it is also possible to use the relation $-\theta_{max} < \theta_{ref}(t) < \theta_{max}$ to develop the distribution. Ultimately, the choice is dictated with respect to first angular reading and to where this is in relation to the loading zone. Figure 4.1 contains a visual explanation of the load distribution for a initial condition of $\theta_{ref}(0) = 0$, which results in a loading factor of $q(\theta_{ref}(t)) = q_0$.

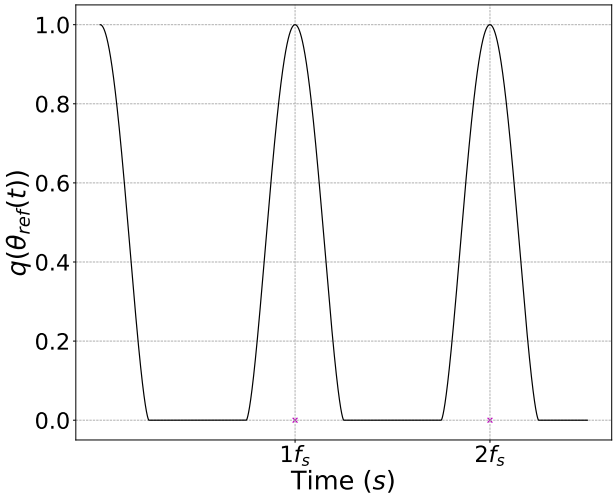


Figure 4.1. The load distribution function $q(\theta_{ref}(t))$ for an initial condition of $\theta_{ref}(0) = 0$, $\varepsilon = 0.499$, $n = \frac{3}{2}$, $\theta_{max} = \frac{\pi}{2}$ and $q_0 = 1$. Notice how the loading and unloading process is synchronous with the shaft rotation f_s, adapted from McFadden and Smith (1984).

The final element of this model is to define signal contribution levels under constant operating conditions. One must take careful note here of the reference to the operating condition level, as this approach will not hold under non-stationary operating conditions. As one has access to the discrete gearbox signal components, one can perform simple component scaling manipulations such that a reasonable model can be obtained. Here, the suggestion is that one uses his direct access to the noise signal to perform Signal-to-Noise Ratio (SNR) based component scaling. The SNR can be given as

$$SNR = \frac{P_{signal}}{P_{noise}}, \tag{4.15}$$

which is the ratio of the average power of any signal with respect to the average power of the noise. For discrete signals, it is common to use the SNR on a decibel scale and the RMS of a signal, which can be given as

$$SNR_{dB} = 10\log_{10}\left(\frac{A_{signal}}{A_{noise}}\right)^2, \tag{4.16}$$

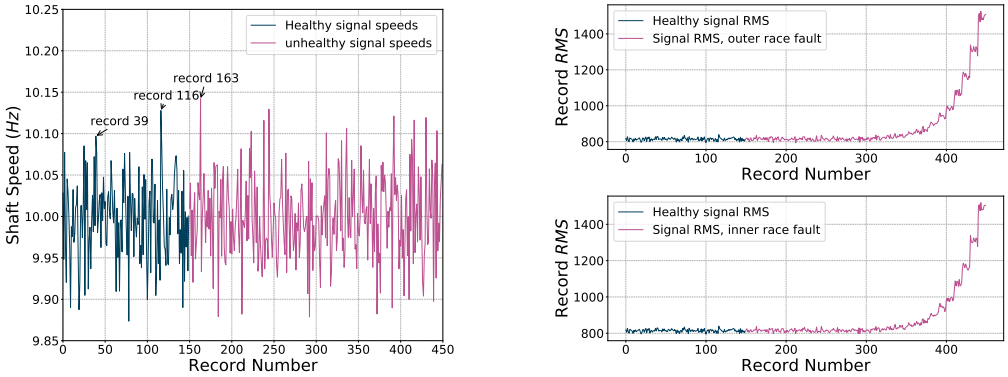
where A refers to the discrete signal RMS. Under the objective of finding a SNR between the components in Equation (4.1) and $x_n(t)$, such that each component can be scaled to a specific SNR, the following relationship can be found

$$C_i = \frac{10^{\frac{SNR_i + 20\log_{10}(A_{x_n})}{20}}}{A_{x_i}}. \tag{4.17}$$

This relationship then allows for model components to be defined in a reasonable relationship for the noise component of the signal, such that any gearbox signals generated by the model are representative of a real application with characteristics that are within reasonable bounds.

4.2.1 Dataset Properties

In this work, a set of prescribed properties shall be used to give a phenomenological model that is representative of a gearbox under constant operating conditions. To introduce model stochasticity, the decision was made to use constant but per-signal varying shaft speeds sampled from a normal distribution. This means that although the speed for a given signal will be stationary, the shaft speed from one signal to the next shall be different. This allows one to control the model complexity as a larger speed variance will control the data distribution complexity that any latent variable model aims to capture. The speed profile distribution that shall be used is $f_{shaft_1} \sim \mathbb{N}(10, 0.05^2)Hz$, with Figure 4.2(a) showing the samples for the generated signal records. In Figure 4.2(b) the signal *RMS* was calculated and one can note that there is a clear progression of damage with levels corresponding to the ten records for each of the thirty increments between $-40dB$ and $10dB$. As the phenomenological model in this work shall be used to simulate bearing faults, bearing properties were required to be chosen to develop fault frequencies. For a concise summary of the model parameters for the transmission paths, mesh coefficients, variance components and noise scaling *SNR*s please refer to Appendix C.



(a) Record shaft speed

(b) Record RMS

Figure 4.2. The signal record shaft speed and *RMS* for the signals generated using the phenomenological model. Note that the shaft speed for the inner race and outer race fault datasets were shared to ensure that any conclusions made were not biased by discrepancies in shaft speed.

To generate a healthy signal distribution, the decision was made to densely sample the healthy signals speed profile to develop a total of one hundred and fifty healthy signals. For the faulty dataset, the author chose to use thirty batches of ten signals per batch weighted using Equation (4.17) for a $SNR_{fault} \in [-40, 10]dB$ with each of the thirty points linearly spaced along the domain. This then gives three hundred signals in the unhealthy dataset, from which both inner and outer race faults shall be explored. To reduce fault analysis complexity, the decision was made to share the inner race and outer race unhealthy dataset speed profile, such that one can explore the effects of different faults without complicating the analysis with different shaft speeds per fault. It is important to emphasise here that the faults are still kept distinct and only the shaft speed is shared, thus giving two unhealthy test datasets under one shaft speed profile.

4.3 Dataset Result Analysis

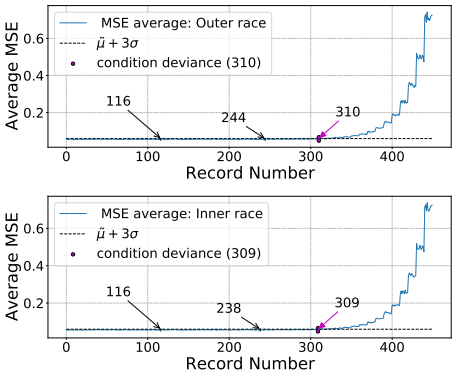
The phenomenological model was investigated due to parameter availability and the control the author had on the dataset. The mathematical model provides access to both inner race and outer race faults, which allowed for model comparisons to occur for different fault cases. The objective here is also not to compare the methods to signal processing, but rather to analyse deep learning models in performance comparisons to one another. The author will show results from the *temporal preservation* data processing approach with the discrepancy signal mean used as the detection metric. A discrepancy signal is obtained by feeding signal segments through the model and preserving the order of these components for the obtained HI or LHI values. To allow for a condition deviance point to be identified, the author will use a threshold that is defined as $threshold = \tilde{\mu} + 3\sigma$, where $\tilde{\mu}$ is the discrepancy signal median and σ is the discrepancy signal standard deviation. This threshold approach is considered a hypotheses test with $H_0 : P = \tilde{\mu}$ and $H_a : P \neq \tilde{\mu}$ with a p value of 0.003. To formulate this hypothesis test, the 99.7% confidence interval is used to develop deviance bounds on the average of the healthy discrepancy signals and a condition deviance point is detected when any discrepancy signal average exceeds these bounds. To associate between once-off anomalous instances and instances of damage the author will identify condition deviance points as those whose five-point-ahead average from a point of deviance are greater than the threshold. This five-point-ahead average can be considered an anti-causal filter $h(t)$ where $h(t) = \frac{\mu_{HI}(t+1)+\mu_{HI}(t+2)+\mu_{HI}(t+3)+\mu_{HI}(t+4)+\mu_{HI}(t+5)}{5}$. It is important to note that this is not a fault detection methodology that is being proposed, but rather a performance quantification methodology that is used to compare the various HIs and LHIs obtained from the models considered in this work. The use of an anti-causal filter introduces the assumption that the future state of the HI or LHI can be accessed, which is impossible if these techniques are implemented in real-time.

For the phenomenological model dataset, the author will demonstrate the performance of the available HIs and LHIs for both fault cases respectively. The performance will be initially quantified using *PCA*, to investigate whether additional model complexity is required. *PCA* will be used to quantify the *temporal preservation* approach with a focus on the chosen model window length, L_w. The window length investigated will be $L_w = 512$ and 4096, given that a shaft speed of approximately $10Hz$ and a sampling frequency of $25kHz$, the ratio $\frac{F_s}{f_s}$ is equal to 2500. The author also used a CCR of 95% for the *PCA* models unless indicated otherwise.

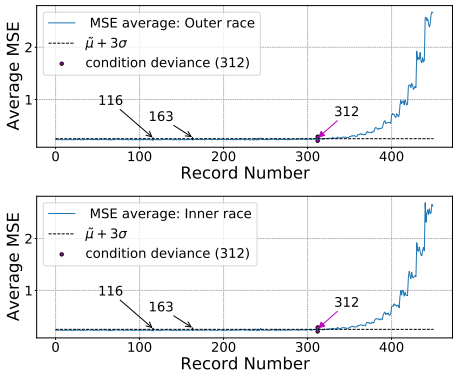
4.3.1 PCA Response

In figure 4.3(a) and (b), the $HI^{(1)}$ discrepancy signal average are shown for the two considered window lengths. The choice of window length clearly affects the reconstruction log-likelihood, with a larger window length resulting in a very similar record average while the smaller window length produces changes in record average. This is directly attributed to the interaction present between the pre-processing methodology and the model window length, as larger window lengths produce segments that always capture the fault. The relationship between Figure 4.3(a)-(b) and (c)-(d) is that the latter comprises alternative discrepancy signal statistics to the average to highlight how model window length and discrepancy measures may not give a full picture into the HI or LHI relationship with the fault present. The effect of the model window length on the discrepancy signal statistics shown in Figures 4.3(c) and (d) is noticeable, with the shorter window lengths returning segments that are in the same range as the healthy data as shown by the min-max range. The inner race fault median for a window length of $L_w = 512$ also shows how the type of fault affects the response of the discrepancy signal. This highlights the implicit assumptions between the partitioning scheme applied to vibration data and the model response, with a link to the fault frequency. It is also important to note here that the author chose to restrict the captured variance to 80% for the model with a window length of $L_w = 4096$. This was done as a 95% CCR produced noticeable discrepancies between the training and validation

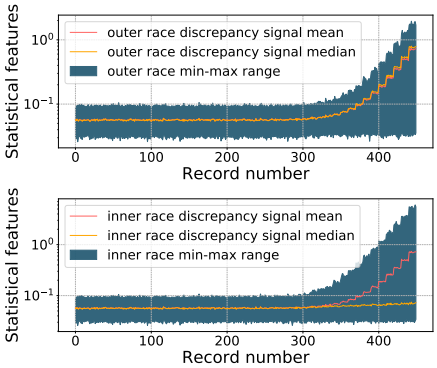
datasets.



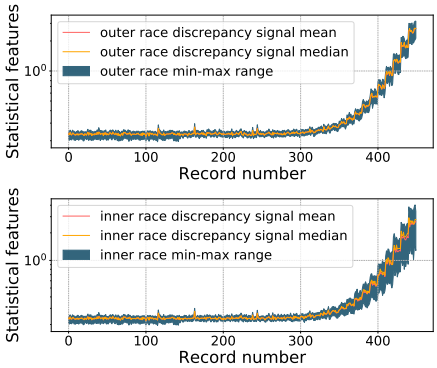
(a) $HI^{(1)}$: $L_w = 512$



(b) $HI^{(1)}$: $L_w = 4096$



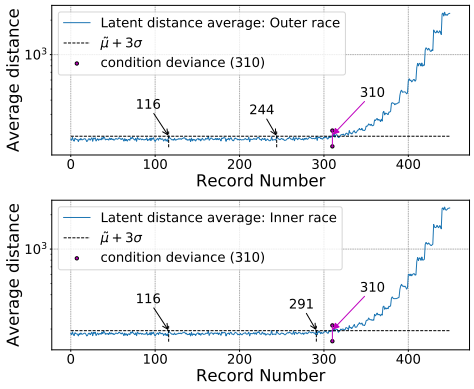
(c) Statistical features: $L_w = 512$



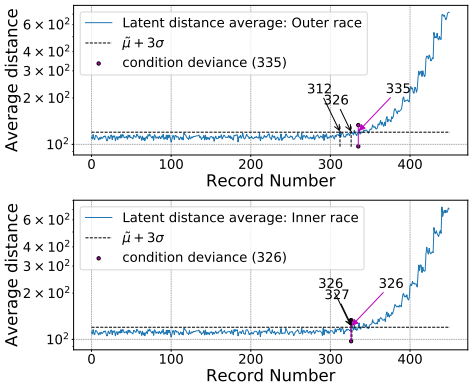
(d) Statistical features: $L_w = 4096$

Figure 4.3. The $HI^{(1)}$ and discrepancy signal statistics for *PCA* models trained on phenomenological model data that differ in model window length L_w. Figures 4.3(a) and (b) detail the $HI^{(1)}$ response while Figures 4.3(c) and (d) detail the statistical features. Notice how the choice of model window length affects discrepancy signal min-max range and median, with the model for $L_w = 512$ clearly encountering healthy signal segments.

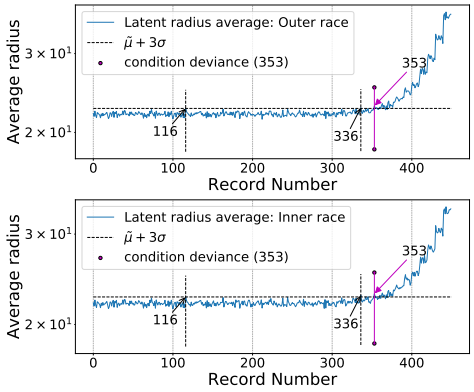
Figure 4.4 shows the response from the three *LHI*s made available through the *temporal preservation* approach, for the window lengths of interest. All three *LHI*s are responsive to damage for the $L_w = 512$ model while the larger window length produces poor responses from $LHI^{(2)}$. It is clear that $LHI^{(3)}$ is the best performing metric as it produces the earliest identifiable condition deviance point. It is clear that the latent manifold is responding differently to the different faults, with a noticeable difference in $LHI^{(3)}$ magnitude for a shorter window length shown in Figure 4.4(e). In comparison of the results for the different window length, there is an effect on the condition deviance detection point, with the larger window length producing points that occur consistently later for the $LHI^{(2)}$ and $LHI^{(3)}$ metrics. The latent metric that is least indicative of damage is $LHI^{(2)}$, which indicates that the latent manifold response to damage favours changes in latent manifold velocity over off-manifold path projections. The latent metrics allow one to discover manifold intuition that was previously unavailable and allow for physical interpretation due to the inclusion of the time component.



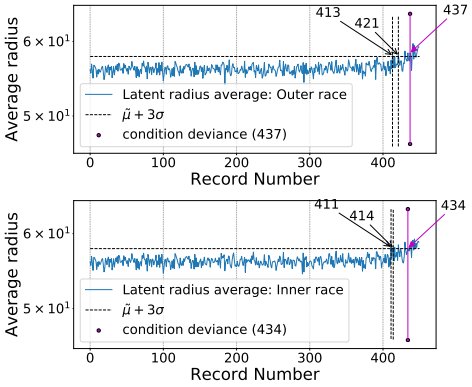
(a) $LHI^{(1)}$: $L_w = 512$



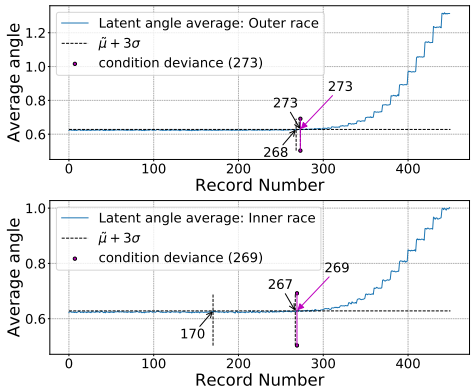
(b) $LHI^{(1)}$: $L_w = 4096$



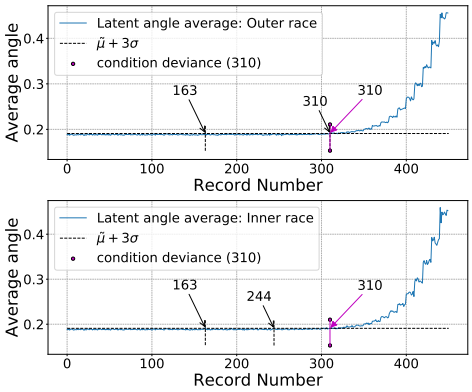
(c) $LHI^{(2)}$: $L_w = 512$



(d) $LHI^{(2)}$: $L_w = 4096$



(e) $LHI^{(3)}$: $L_w = 512$



(f) $LHI^{(3)}$: $L_w = 4096$

Figure 4.4. The three *LHI* responses for *PCA* models with different model window lengths L_w trained on phenomenological data. The discrepancy signal mean was used as a discrepancy metric.

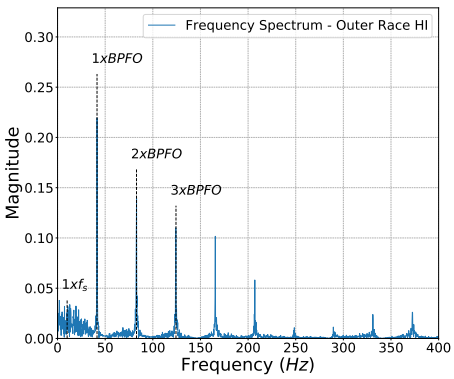
To further emphasise the benefits that one obtains through the *temporal preservation* approach, the author chose to illustrate the $HI^{(1)}$ discrepancy signal frequency spectrum content for each record for the two faults considered. For a shorter window length, it was found that a clearer distinction between signal segments and faulty segments could be isolated, which is beneficial for the Fourier analysis of signals. This is quantified by examining the difference in the magnitudes between Figures 4.5(a)-(b) and Figures 4.5(c)-(d), where it is clear that Figures 4.5(c) and (d) have a lower magnitude and is less refined. In figure 4.5, the frequency content of the final faulty record and the content through each record is shown. Figure 4.5(a) and (b) show the presence of the two bearing faults considered, with clear amplitudes at the fault frequencies and harmonics thereof. In Figure 4.5(c) and (d), the frequency content in the $HI^{(1)}$ discrepancy signal for each record is plotted and shown. The author normalised the frequency by the shaft speed so that the amplitudes could be shown with respect to shaft orders. It is clear, from a signal processing perspective, that the exploitation of the time component introduces changes in the HI that corresponds to the introduction of the fault.

4.3.2 VAE Response

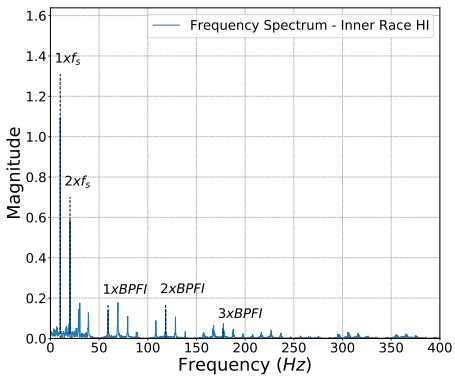
In Figure 4.6, the average of the discrepancy signals for $HI^{(1)}$ are shown for the deterministic and stochastic parametrisations of the VAE model for a window length of $L_w = 512$. It is evident to note is that the VAE_2 model produces stronger $HI^{(1)}$ responses to damaged segments which is attributed to the learnt output variance effect on the HI, as it quantifies the expected deviation in a reconstructed feature. It is clear that the VAE model reconstruction response is indicative of damage, a result attributed to the simplicity of the phenomenological model dataset. The discrepancy signal average of the outer race and inner race fault HI is notably different, which indicates that the faults produces differences in deviation magnitude. The condition deviance point identified by the model for the different fault cases is in-line with the points obtained from PCA, indicating that the HI is performing at a satisfactory level.

In Figure 4.7, the area-under-the-curve (AUC) and classification rate for the $HI^{(1)}$ discrepancy signals are given for two PCA and VAE_1 models that differ in window length. The objective here is to further quantify how the binary classification approach often employed to validate unsupervised deep learning models is highly dependant on the window length. It is noted here that the AUC was determined for a threshold that varied from the smallest LL magnitude from the validation set to the condition deviance threshold. It is clear to note that for a window length of $L_w = 512$, the classification performance is poor for the outer race fault and even worse for the inner race fault. This is attributed to the presence of both healthy signal segments and unhealthy signal segments in an unhealthy signal, which alters the definition of classification as the discrepancy signal elements are no longer all indicative of damage. In Figure 4.7(b), a window length of $L_w = 4096$ ensures that all segments processed with the *temporal preservation* approach contain damage and a 100% classification accuracy can be obtained. In Figures 4.7(a) and (c), performance differences can be noted between the type of fault present, which is attributed to the loading zone modulation present in inner race faults.

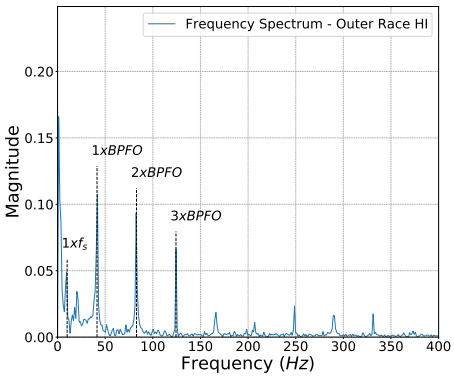
In Figure 4.8, the three LHI responses from a VAE_1 and VAE_2 model for a window length of $L_w = 512$ are shown. It is clear here that the $LHI^{(2)}$ response is very poor for both models and the best performing metric is $LHI^{(3)}$, a result that is aligned with what was seen in the PCA case. The VAE_2 model also provides a clearer and more indicative response, with improvements to both $LHI^{(1)}$ and $LHI^{(3)}$. It is clear that the learnt variance affects the learnt manifold, with $LHI^{(2)}$ identifying as an inferior metric that provides no indication of damage. The physical interpretation here is that the learnt manifold does not place anomalous instances far off the manifold but rather increases the distance travelled within the manifold. It is also clear that records 39, 116 and 163 are clear threshold deviance records



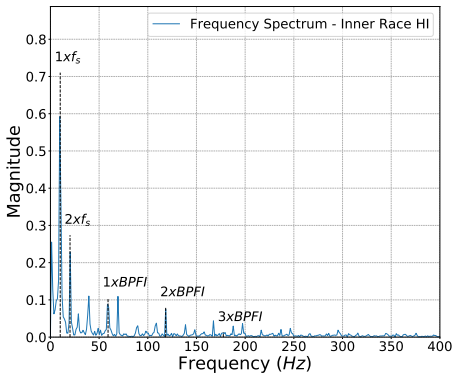
(a) Frequency spectrum: outer race fault



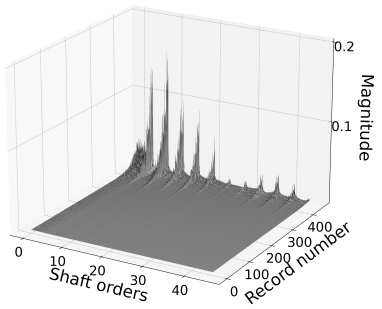
(b) Frequency spectrum: inner race fault



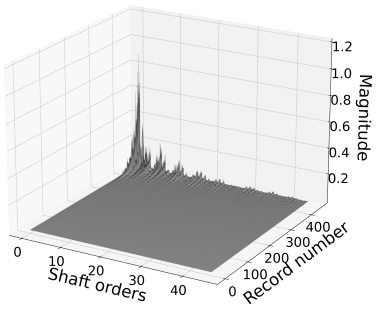
(c) Frequency spectrum: outer race fault



(d) Frequency spectrum: inner race fault

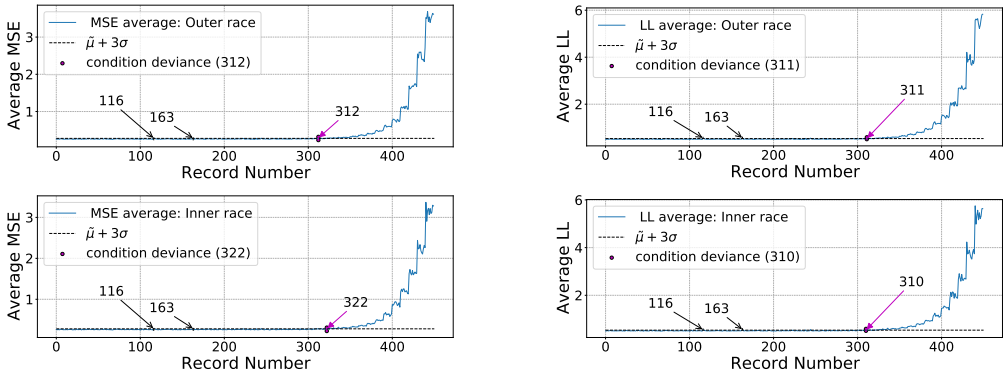


(e) Record-Frequency spectrum: outer race fault



(f) Record-Frequency spectrum: inner race fault

Figure 4.5. The frequency and record-frequency spectra of the $HI^{(1)}$ discrepancy signal from a *PCA* model with window length $L_w = 512$ and $L_w = 4096$ for the outer race and inner race fault data of the phenomenological model dataset. Figure 4.5(a), (b), (c) and (d) were developed using the final signal record in the dataset for the outer and inner race cases, with (a), (b), (e) and (f) developed from the $L_w = 512$ model while (c) and (d) were developed from the $L_w = 4096$ model.

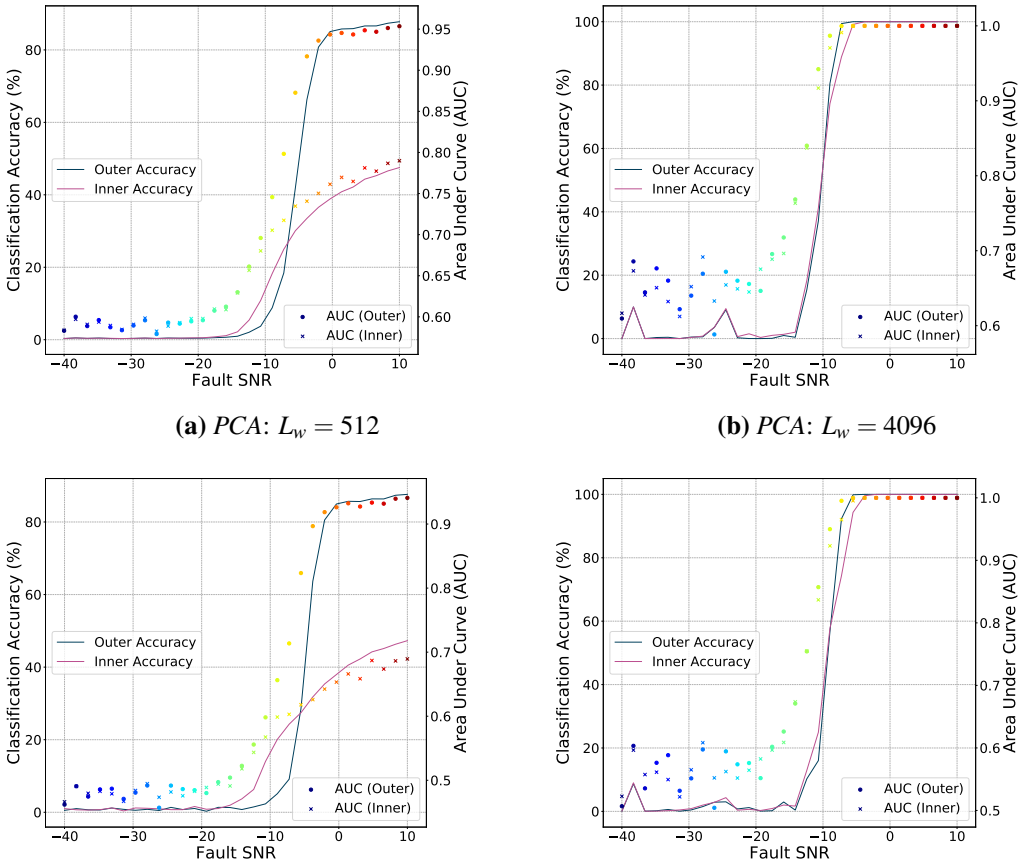


(a) $HI^{(1)}$: VAE_1 **(b)** $HI^{(1)}$: VAE_2

Figure 4.6. The discrepancy signal average of the $HI^{(1)}$ response for VAE_1 and VAE_2 models with a window length of $L_w = 512$. Note that in 4.6(b) the notation is now a log-likelihood average, which was done to indicate that a learnt output variance was used.

which is related to the shaft speed of the signals, as these records all have the largest speeds seen in Figure 4.2. It is clear that these are not the only threshold deviance records, but the variation in the *LHI* responses can be directly attributed to the shaft speed. The performance of the *VAE* models can now be compared to that of *PCA* and it is clear that by all accounts, *PCA* is the superior model in terms of the latent manifold response to damage. This is attributed to two potential reasons, the first is that this dataset is simple and thus a linear model can correctly represent and track damage without the need for increased model complexity. This may imply that the *VAE* models may attempt to linearise their in-built non-linearity and the result is a weakened manifold response. The second reason that is attributed to is the potential power of the *VAE* decoder, which may induce a latent manifold that is less informative. This is aligned with some known issues that plague *VAE*s, as detailed in Chen et al. (2017) and Zhao et al. (2017).

The author would now like to draw the reader's attention to a consistency between the classification and the *LHI* responses, with a detectable change around approximately $-10dB$ or from record three hundred and thirty onwards. This change point is important as this is the exact *SNR* of the random gear component, which indicates that the fault is only detected by the model once the bearing component dominates this component. If a Fourier analysis is then conducted using the $LHI^{(3)}$ response for two VAE_2 models that differ in window length on the inner race fault data, interesting results are obtained. In Figure 4.9, it is clear that there are two clear spikes that are consistent through each spectrum and these spikes are at shaft orders of 20 and 40 respectively. The two frequency components are the gear mesh frequency and its harmonic, indicating that there is some oscillation in the *LHI* at the dominant deterministic signal component. This indicates that the manifold may capture the dynamics of the system in the latent manifold and that these dynamics affect the traversal through the manifold. This is powerful as it shows that the properties of the data space and the latent space are shared, indicating that the manifold is capturing the intrinsic properties of the data. This also highlights the benefit of the proposed *temporal preservation* approach and the *LHI*s, as they introduce a level of intuition that was previously unnoticed in latent variable models.



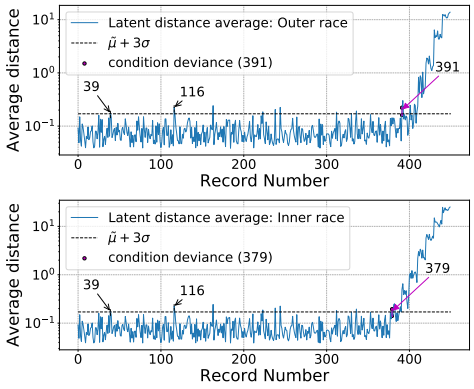
(a) *PCA*: $L_w = 512$ **(b)** *PCA*: $L_w = 4096$

(c) VAE_1: $L_w = 512$ **(d)** VAE_1: $L_w = 4096$

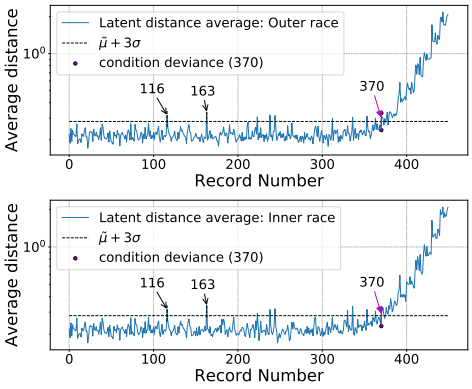
Figure 4.7. The *AUC* and classification accuracy obtained from two formulations of the *PCA* and VAE_1 model that differ in model window length. Notice the clear change in classification performance due to a simple assumption of model window length. Note that the $L_w = 512$ model does perform well but a level of interpretation exists that clearly needs to be built into the response analysis.

4.3.3 GAN-based Response

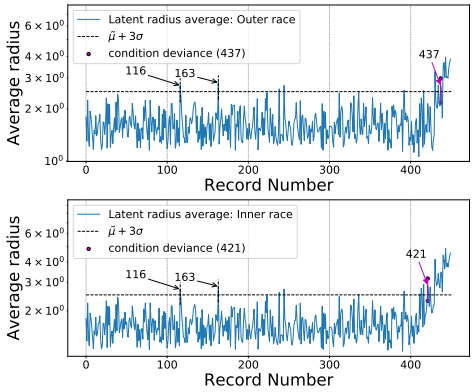
In Figure 4.10, the corresponding $HI^{(2)}$ and $HI^{(3)}$ response under the *temporal preservation* approach is shown for $RY-GAN$ and $DLS-GAN$ models with a window length of $L_w = 512$. The objective here was to highlight to the reader that this approach is not limited to the *LHI*s used, but can also be equally extended to the *HI*s. This implies that the usage of the *temporal preservation* approach can be readily used as a drop-in replacement for most model evaluation methods. The discrepancy signal mean was used as a discrepancy metric, with a clear response to damage exhibited from the reconstruction log-likelihood, data discriminator and the latent critic. It is clear that the *GAN*-based model is performing on a comparative level to *PCA*. The latent critic also identified condition deviance points early than the data discriminator, a result attributed to poor discriminator training. An indication of poor GAN training is the stability point for the data discriminator, which should produce an value around 0.5. Clearly, in Figures 4.10(a) and (b), this is not the case. This is attributed to the presence of the L_2 and GAN methodologies in the training scheme which optimise in juxtaposition to one another and often the L_2 loss will dominate training. It is clear, however, that in the presence of significant damage the discriminator detects a movement from the healthy data manifold and begins to classify this data as fake by tending to zero.



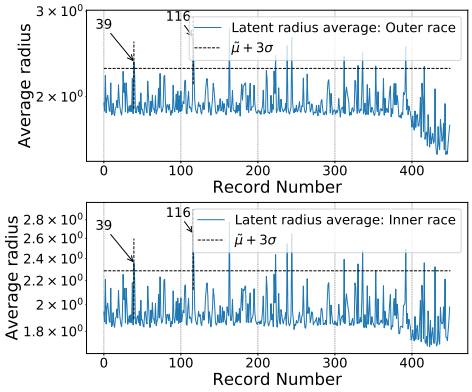
(a) $LHI^{(1)}$: VAE_1



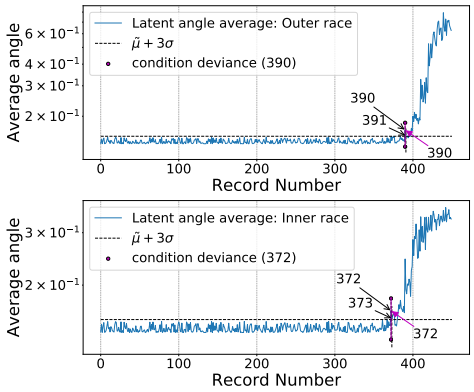
(b) $LHI^{(1)}$: VAE_2



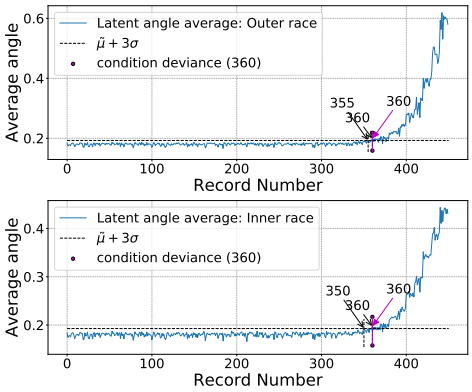
(c) $LHI^{(2)}$: VAE_1



(d) $LHI^{(2)}$: VAE_2

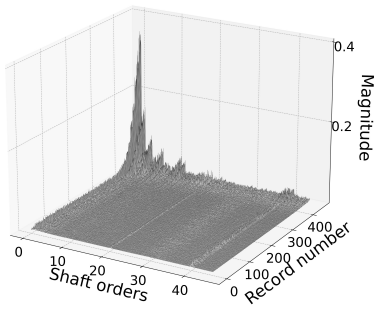


(e) $LHI^{(3)}$: VAE_1

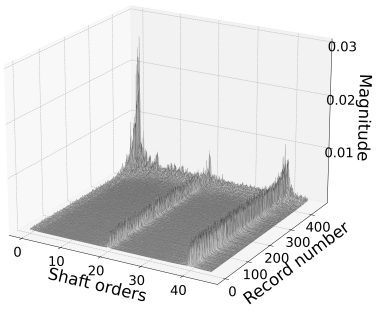


(f) $LHI^{(3)}$: VAE_2

Figure 4.8. The three *LHI* responses for VAE_1 and VAE_2 models with a window length of $L_w = 512$ trained on the phenomenological model data.



(a) Record-frequency spectrum: $L_w = 512$

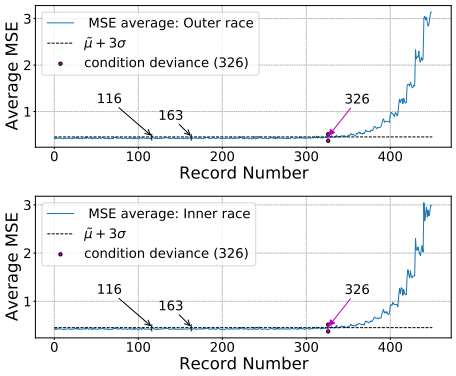


(b) Record-frequency spectrum: $L_w = 4096$

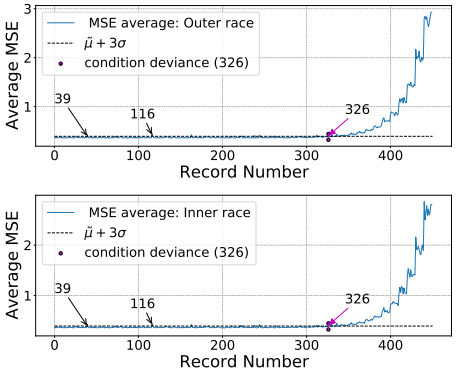
Figure 4.9. The $LHI^{(3)}$ record-frequency spectra for the VAE_2 models with a window length of $L_w = 512$ and $L_w = 4096$ for the inner race fault. Notice the clear presence of the gear mesh component and its harmonic at 20 and 40 shafts orders respectively.

To investigate how the latent manifold is responding to healthy and unhealthy data, Figure 4.11 shows the data discriminator and latent critic frequency spectra through time for the other race fault data. It is clear from both Figure 4.11(a) and (b) that there is some natural discrepancy signal oscillation at the frequency of the gear mesh frequency and its harmonics. It is clear that the data discriminator response in Figure 4.11(a) that the random gear and gear mesh frequencies dominate the earlier records and the fault frequencies dominate the later records. Figure 4.11(b) shows how the latent manifold is aware of the shaft speed, with a strong frequency component at the gear mesh frequency and its harmonics in relation to the fault frequencies. The presence of this component is strong in the latent critic response which indicates that the model is sensitive to the shaft speed and this sensitivity is amplified in the latent manifold. This also highlights for the reader how the latent manifold is greatly affected by the model formulation and that one can gain knowledge by looking at both the learnt latent manifold and the data manifold. Ultimately, the latent critic is a powerful element as it introduces latent understanding, however further interpretability can be obtained from the *LHI*s.

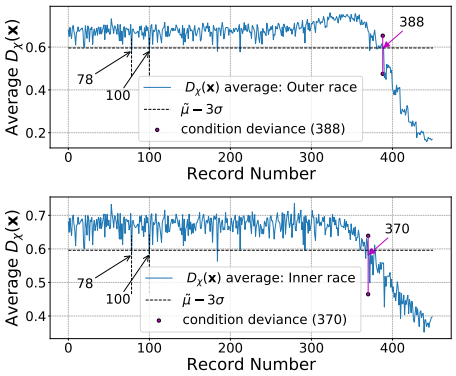
In Figure 4.12, the three *LHI*s are shown and it is clear that the $RY-GAN$ and $DLS-GAN$ methods have learnt similar latent manifolds. It is clear from Figures 4.12(a), (b), (e) and (f) that $LHI^{(1)}$ and $LHI^{(3)}$ produce clear condition deviance points, with $LHI^{(3)}$ identifying as the more sensitive metric. The condition deviance points for $LHI^{(3)}$ are competitive with those shown for *PCA* in Figure 4.4, which further indicates how the latent metrics proposed in this work augment the information available to the user. The latent critic was found to fluctuate at a frequency of the gear mesh component and the author confirmed that although the three latent metrics also contained some of this information, it was most clear in $LHI^{(2)}$. This highlights that the latent critic may be more responsive to changes in the distance from the origin, an expected response given its training objective. However, changes in the manifold velocity and trajectory are also interpretable and carry information related to damage. The magnitude of the three *LHI*s also provide some indication that the two faults are interpreted by the model differently, however this is a purely qualitative observation that can be made in comparison of the faults. One cannot directly quantify the fault present unless the analysis if further augmented with external information such as the fault frequency which requires knowledge of the shaft speed and bearing characteristics. However, this does highlight that deep learning can immediately benefit from simple signal processing techniques, as taking the Discrete Fourier Transform (DFT) of a discrepancy



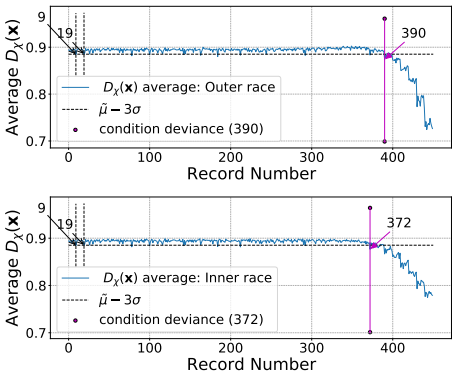
(a) $HI^{(1)}$: $RY-GAN$



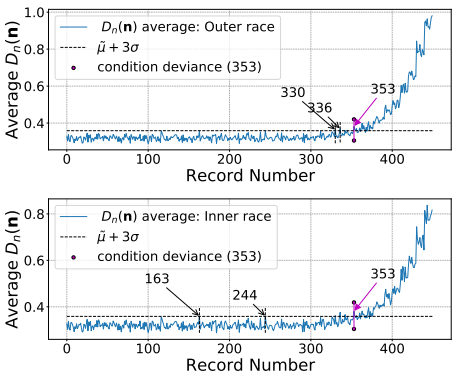
(b) $HI^{(1)}$: $DLS-GAN$



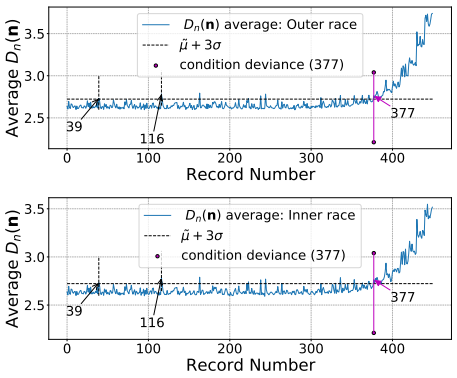
(c) $HI^{(2)}$: $RY-GAN$



(d) $HI^{(2)}$: $DLS-GAN$

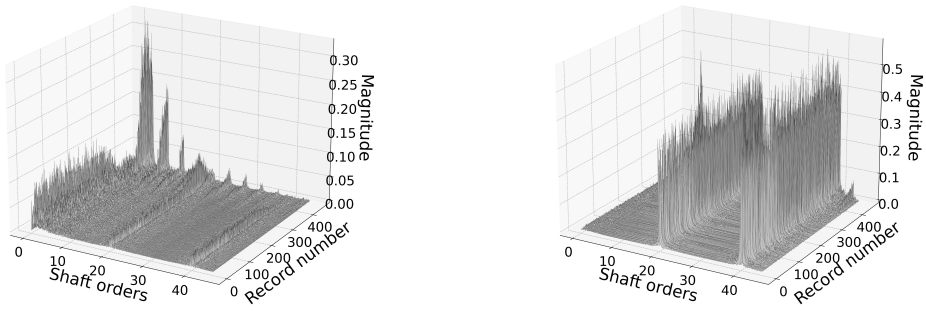


(e) $HI^{(3)}$: $RY-GAN$



(f) $HI^{(3)}$: $DLS-GAN$

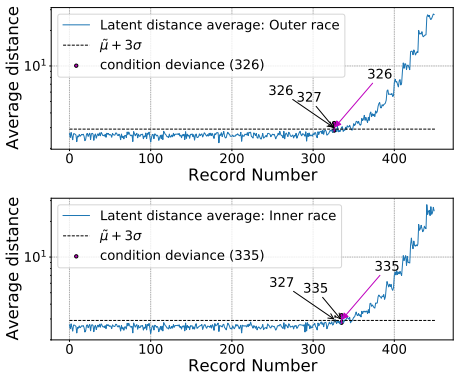
Figure 4.10. The responses from the data discriminator and the latent critic for the $RY-GAN$ and $DLS-GAN$ models with a window length of $L_w = 512$ for the outer and inner race fault data.



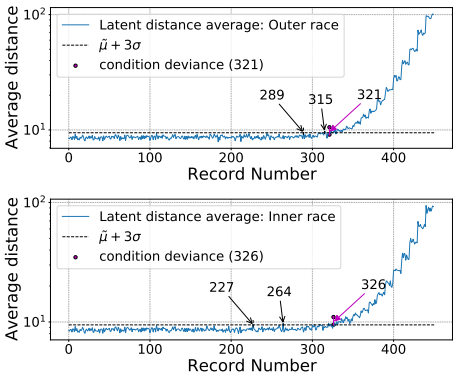
(a) Record-frequency spectrum: $HI^{(2)}$ **(b)** Record-frequency spectrum: $HI^{(3)}$

Figure 4.11. The record-frequency spectra of the data discriminator and latent critic response from a $RY-GAN$ model with a window length of $L_w = 512$ for the outer race fault data. Notice the clear presence of the gear mesh component.

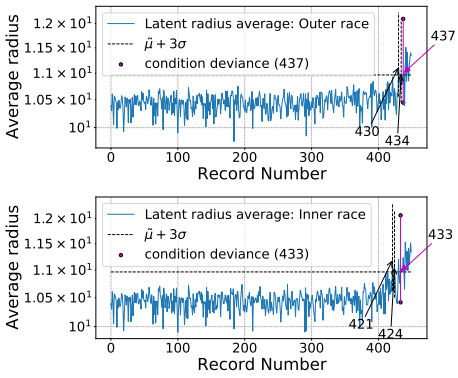
signal is trivial and the evaluation of frequency content is common in signal processing techniques under stationary operating conditions.



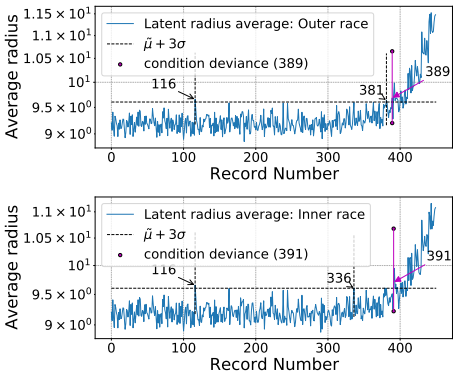
(a) $LHI^{(1)}$: $RY-GAN$



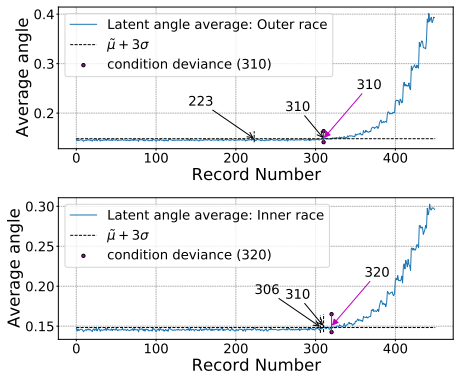
(b) $LHI^{(1)}$: $DLS-GAN$



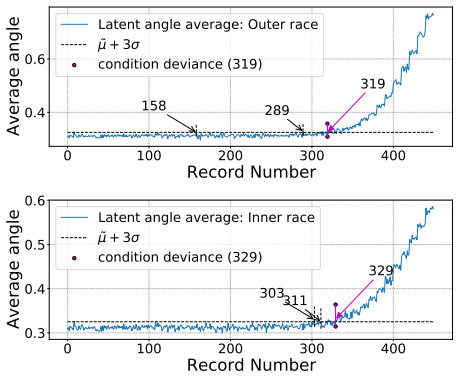
(c) $LHI^{(2)}$: $RY-GAN$



(d) $LHI^{(2)}$: $DLS-GAN$



(e) $LHI^{(3)}$: $RY-GAN$



(f) $LHI^{(3)}$: $DLS-GAN$

Figure 4.12. The latent manifold response for $RY-GAN$ and $DLS-GAN$ models with a window length of $L_w = 512$. Notice the strong condition deviance through the latent manifold metrics.

4.3.4 Dataset Consolidation

To conclude this dataset, the author will represent the identified condition deviance points for the various models considered in this work through Table 4.1. The decision was made to use the *temporal preservation* approach to determine the discrepancy signal mean, as was done for the various *LHI* responses. This was done to ensure that the *HI* and *LHI* responses are quantifiable in the same domain of analysis.

Table 4.1. The obtained threshold condition deviance point from the first phenomenological model dataset for both fault types when investigating the HI's. Note that IC_1 is the abbreviation used for inconclusive results.

Model type and characteristics		**Health indicator condition deviance point for outer race ‖ inner race fault**					
Model used	**Window length**	**HI$^{(1)}$**	**HI$^{(2)}$**	**HI$^{(3)}$**	**LHI$^{(1)}$**	**LHI$^{(2)}$**	**LHI$^{(3)}$**
PCA	$L_w = 512$	**310** ‖ **309**	N/A	N/A	**310** ‖ **310**	353 ‖ 353	**273** ‖ **269**
	$L_w = 4096$	**312** ‖ **312**	N/A	N/A	335 ‖ **326**	437 ‖ 434	310 ‖ **310**
VAE_1	$L_w = 512$	**312** ‖ **322**	N/A	N/A	391 ‖ 379	437 ‖ 421	390 ‖ 372
	$L_w = 4096$	**312** ‖ **312**	N/A	N/A	IC_1 ‖ IC_1	IC_1 ‖ IC_1	406 ‖ 411
VAE_2	$L_w = 512$	**311** ‖ **310**	N/A	N/A	370 ‖ 370	IC_1 ‖ IC_1	360 ‖ 360
	$L_w = 4096$	**326** ‖ **326**	N/A	N/A	430 ‖ 426	IC_1 ‖ IC_1	400 ‖ 409
$\beta - TC - VAE_1$	$L_w = 512$	**312** ‖ **322**	N/A	N/A	411 ‖ 391	IC_1 ‖ IC_1	383 ‖ 380
	$L_w = 4096$	**312** ‖ **312**	N/A	N/A	IC_1 ‖ IC_1	IC_1 ‖ IC_1	421 ‖ 421
$\beta - TC - VAE_2$	$L_w = 512$	**311** ‖ **310**	N/A	N/A	368 ‖ 370	IC_1 ‖ IC_1	373 ‖ 380
	$L_w = 4096$	**330** ‖ **326**	N/A	N/A	**300** ‖ 381	IC_1 ‖ 439	366 ‖ 354
$RY - GAN$	$L_w = 512$	**326** ‖ **326**	386 ‖ 368	353 ‖ 353	**326** ‖ 335	437 ‖ 433	**310** ‖ **320**
	$L_w = 4096$	350 ‖ 353	396 ‖ 396	363 ‖ 363	424 ‖ 419	353 ‖ 360	IC_1 ‖ IC_1
$DLS - GAN$	$L_w = 512$	**326** ‖ **326**	388 ‖ 379	377 ‖ 377	**321** ‖ **326**	389 ‖ 391	**319** ‖ **329**
	$L_w = 4096$	**325** ‖ **326**	361 ‖ 367	410 ‖ 408	339 ‖ 350	407 ‖ 390	340 ‖ 355

In the analysis of Table 4.1, it is clear that *PCA* is a strong performing method on this dataset, with clear fault identification through all *HI*s. The choice of window length appears to clearly affect the *VAE* model and the $\beta - TC - VAE$ variant, which highlights that there are careful considerations that must be made when selecting a window length. It is clear that the *GAN*-based model offers some immediate improvements over the *VAE* models, with the exception of $HI^{(1)}$. The difference is $HI^{(1)}$ is attributed to the addition of the *GAN* component, which makes some attempt to capture the noise and random gear component. The *LHI* metrics proposed in this work offer improved model interpretability and the fortunate result of responsive behaviours in all three *LHI* metric, while $LHI^{(2)}$ was often weaker. The best performing *LHI* can be identified to be $LHI^{(3)}$, which highlights how the *temporal preservation* approach can offer significant insights into how a model handles anomalous data.

The strong performance of *PCA* indicates that this dataset can be captured by a linear latent variable model. This fact shows that it is important to quantify dataset complexity by, at the minimum, considering how fault detection performance compares to *PCA*. If *PCA* produces adequate performance responses, one would have to justify why the addition of model complexity should feature.

Chapter 5 IMS Dataset Analysis

5.1 Chapter Abstract

In this chapter, the author presents the IMS dataset and the performance investigation of the different models considered in this work. There are four key concepts that are key to this investigation:

1. The model window length L_w affects model diagnostic performance
2. The latent manifold is interpretable under the *temporal preservation* approach
3. The amount of training data can greatly affect the response results
4. Model performance must be compared to fully highlight the benefits of complex methods

The reader is asked to keep these concepts in mind when going through the various results, as each dataset offers insights into each of these points. For a detailed collection of the model architectures, learning rates, stopping conditions and hyper-parameters please refer to Appendix B.5. The models used on this dataset are: PCA, the VAE_1 and VAE_2 models, the $\beta - TC - VAE_1$ and $\beta - TC - VAE_2$ models, the $RY - GAN$ model and the $DLS - GAN$ model.

5.2 Dataset Introduction

The IMS dataset, Qiu et al. (2007), is a well-used bearing failure dataset in literature due to the natural fault degradation characteristics of the data. The IMS dataset consists of three run-to-failure tests in which different faults occurred naturally through time. In the reference paper of Qiu et al. (2006), the dataset was introduced and the experimental set-up was detailed. Figure 5.1 presents a schematic of the set-up. Gousseau et al. (2016) presented an analysis using various signal processing techniques as well as potential conclusions that can be made for each of the run-to-failure tests such that a unified perspective of each test set could be established.

5.2.1 Dataset Description

The IMS endurance test rig consisted of four bearings on a shaft coupled to an AC motor through rub belts. The bearings were force lubricated using a circulation system and each test was stopped once a significant amount of metal debris was detected on a magnetic plug. The test rig characteristics are four double row Rexnord ZA-2115 bearings, two PCB 353B33 High Sensitivity Quartz ICP accelerometers per bearing, a shaft speed of $2000rpm$ and a radial load of $26.69kN$.

The three test datasets in the IMS dataset consist of many one-second measurements throughout the lifespan of the test. Each measurement was made in intervals of ten minutes, with an exception existing in the first dataset whereby the initial forty-three measurements were obtained in five-minute

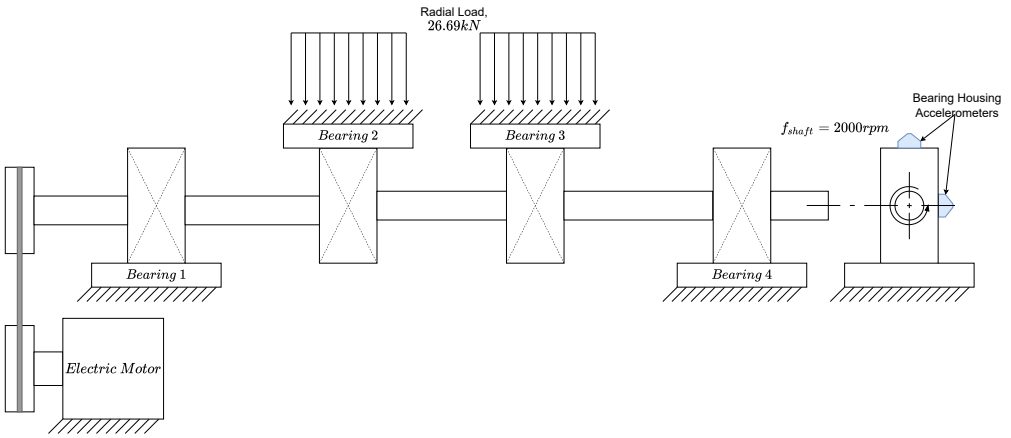


Figure 5.1. The IMS experimental set-up (Qiu et al., 2007). Notice the presence of four bearings with two accelerometers per bearing.

intervals. Table 5.1 presents a succinct summary of the dataset size and visually determined fault cases for each test set. Due to the stationary operating conditions in this dataset, the fault frequencies are known. However, there appears to be some discrepancy in the literature about the sampling frequency of the accelerometers. Qiu et al. (2006) indicated that the sampling frequency was $20kHz$ with a data length of 20480 points. However, Gousseau et al. (2016) indicated that the sampling frequency may be $20.48kHz$, with their investigation proving to support their claim. Under the assumption of a sampling frequency of $20.48kHz$, the fault frequencies are given in Table 5.2 alongside the bearing characteristics of the Rexnord ZA-2115 bearings (Qiu et al., 2006).

Table 5.1. IMS Bearing test set properties

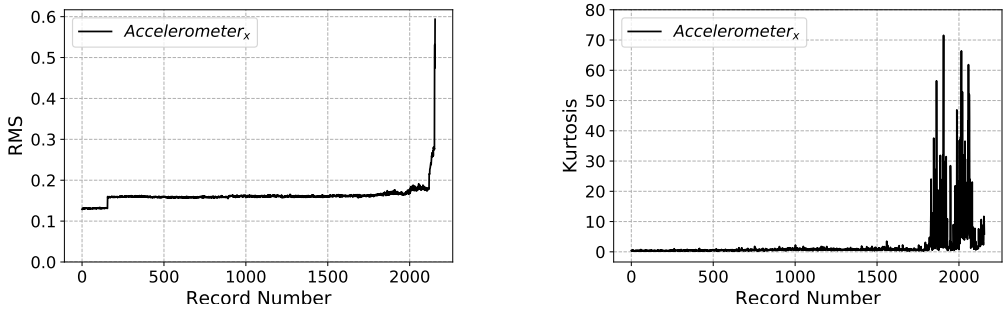
Dataset Number	Number of Channels	Set Duration	Signal records available	Failure Case
1	8	355.75 hrs	2156	Inner race fault (B3), Roller element defect (B4), outer race defect (B4)
2	4	158 hrs	984	Outer race fault (B1)
3	4	741.3 hrs	4448	Debated in literature

Table 5.2. IMS dataset bearing and fault characteristics.

	Characteristic	Value
Rexnord ZA-2115	Pitch diameter	$71.5mm$
	Roller element diameter	$8.4mm$
	Contact angle	15.17°
	Number of rolling elements	16
Frequencies of interest	Shaft Speed	$33.3Hz$
	Ball Pass Frequency Outer race (BPFO)	$236Hz$
	Ball Pass Frequency Inner race (BPFI)	$297Hz$
	Ball Spin Frequency (BSF)	$278Hz$
	Ball Cage Frequency/Fundamental Train Frequency (BCF/FTF)	$15Hz$

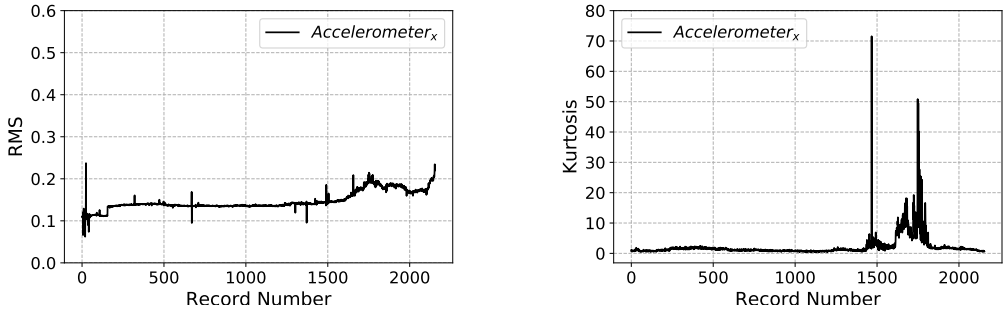
One may note in Table 5.1 that the author stated that test set three has a failure case which was debated in the literature. Gousseau et al. (2016) could not detect the presence of damage, while the IMS dataset Readme document stated that a fault had occurred. Due to this discrepancy, the author will not analyse

this dataset. One may also note in Table 5.1 how there are eight channels per bearing for test set one while for sets two and three there are only four channels. Due to this change, the author chose to limit the analysis to the $x-$ channels for test set one and will only analyse data from the bearings of interest, bearings three and four. As a basic analysis, one may compute simple statistical features from each record in a test set, where these statistics are the *RMS* and kurtosis of the signals, as was done in Qiu et al. (2006). One difference here is the author chose to leave the horizontal axis in integer increments corresponding to record number, as it allows one to see changes in the system that less obvious when viewing the axis under measurement interval. Figures 5.2 and 5.3 contain the computed statistical features for bearings three and four from test set one while Figure 5.4 shows the statistical features for bearing one from test set two.



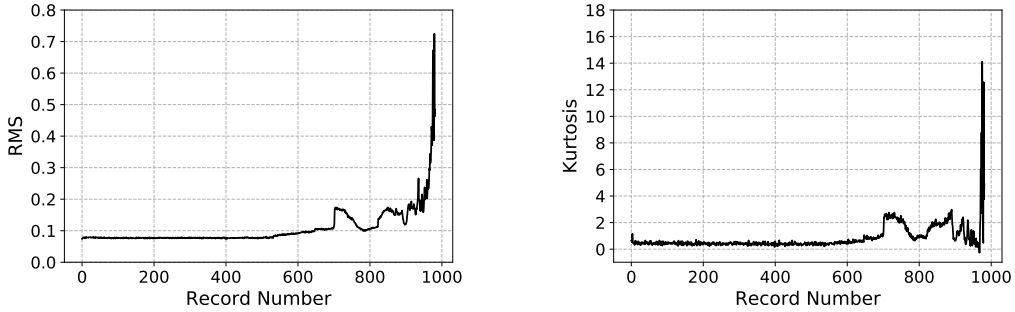
(a) Record *RMS* **(b)** Record kurtosis

Figure 5.2. The RMS and Kurtosis of bearing three data in channel five through all signal records for IMS test set one. Notice the discontinuous jump in RMS that occurs between record 155 and 156 in 5.2(a) and 5.2(b).



(a) Record *RMS* **(b)** Record kurtosis

Figure 5.3. The RMS and Kurtosis of bearing four data in channel seven through all signal records for IMS test set one. Notice the lack of discontinuous jump as was found in Figure 5.2.



(a) Record *RMS* **(b)** Record kurtosis

Figure 5.4. The RMS (5.4(a)) and Kurtosis (5.4(b)) of bearing one data in channels one through all signal records for IMS test set two. Notice the gradual increase in the region of record six-hundred and the drop around record eight-hundred, attributed to bearing self-healing.

5.3 Dataset Result Analysis

The IMS dataset analysis will focus on bearings three and four from the first dataset and bearing one from the second dataset. This was done as these bearings all had faults that developed throughout the experimental lifespan. For method comparability, four signal processing techniques shall be analysed, namely, $MED-SK-NES, SK-NES, CPW-NES$ and the SES, to provide insight into how unsupervised learning compares to fundamental and state-of-the-art signal processing methodologies. The analysis objective of this dataset is twofold, firstly to see how the different models perform in comparison to one another and secondly, how the latent manifold can augment model interpretability for *CBM*.

To allow for performance comparability, the author has chosen to follow approaches used by Abboud et al. (2019) and Schmidt et al. (2019a), whereby a threshold was defined as $threshold = \tilde{\mu} + 6\sigma$ where the median $\tilde{\mu}$ and standard deviation σ are obtained from the first N reference diagnostic metric measurements. For the IMS dataset six standard deviations were used by Abboud et al. (2019) to ensure that the condition deviance point was clear and this work will use the same approach to allow for comparative result response analysis. The decision to use the diagnostic metric median was made as the median is less susceptible to outliers and the threshold response to outliers can be captured in the standard deviation. For any diagnostic metric measurement greater than the threshold, the mean of the following five points was calculated to determine whether a point is a false positive or a point of condition deviance and shall be indicated as such. For the signal processing approaches used in this work, the first one hundred and two hundred records are considered as reference metrics for datasets one and two respectively. The decision to use one hundred records was made with consideration of the jump noted in the signal *RMS* shown in Figure 5.2(a). For any deep learning approaches, the records used will consist of those used for model training and validation.

For diagnostic metrics, the author predominately focuses on the discrepancy signal mean as the mean was found to be a sufficient metric for damage detection. Due to the operating condition present, little was gained from using the *temporal preservation* approach for the *HIs* as the shaft speed was sufficiently high. However, it is still necessary for the *LHI*s as increment continuity is required, with the exception of $LHI^{(2)}$. Note that the benefit of the *temporal preservation* approach for the discrepancy

metrics is not as significant here, as the ratio $\frac{F_s}{f_s} = \frac{20480}{33.3} = 615.02$ is very close to the lowest considered model window length of $L_w = 512$.

5.3.1 Dataset One: Bearing Three

For the first IMS dataset, the author conducted two parallel investigations focused on the amount of training data made available due to the clear jump around record one hundred and fifty-six shown in Figure 5.2(a). The purpose here is simple, under the decision to use 5% or 10% of the available records as training and validation data, referred to as *case one* and *case two* respectively, one may bias results to inadvertently capture or ignore the jump and the defined damage threshold and condition deviance point may suffer. In Table 5.3, these cases are given to make it clear to the reader how much data was used. Furthermore, the author will investigate, under these parallel cases, the assumed model window length as this parameter proved to be important to the interpretation of the model response in Section 4.3. The window lengths of interest are: $L_w = 512$ and $L_w = 4096$. As many models were analysed, it is infeasible to present all the results obtained for each case. Thus, the author will present results deemed interesting and after the response has been motivated, the remaining results will be collectively shown in Table 5.4.

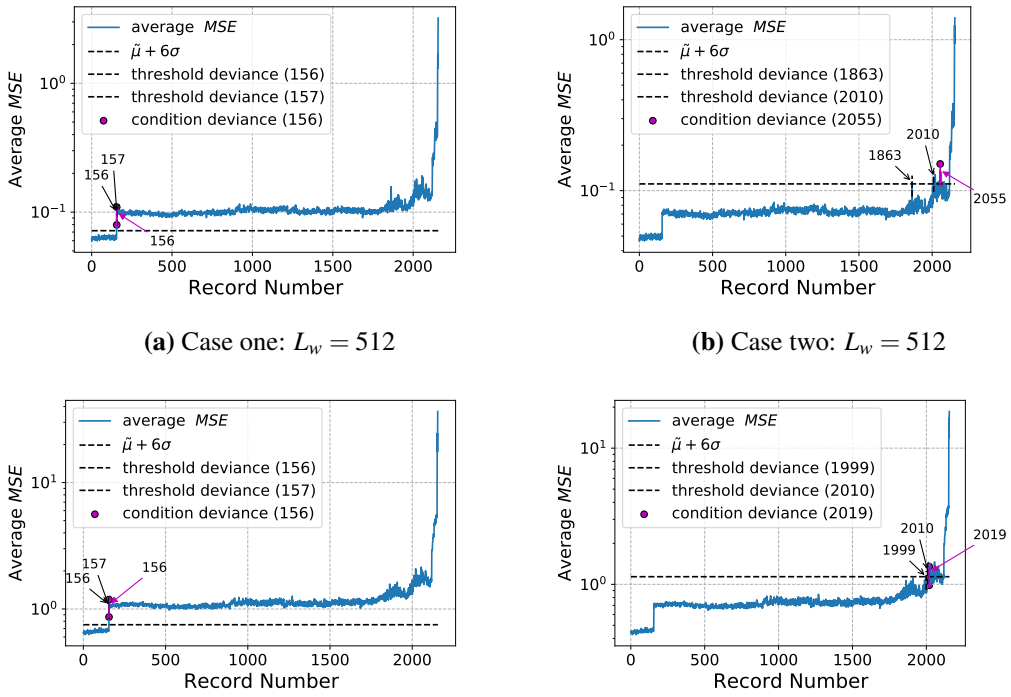
Table 5.3. The amount of healthy data used for the first IMS dataset.

Training data percentage	
Case one	*Case two*
5%	10%

5.3.1.1 PCA Model Response

In Figure 5.5, the author present the $HI^{(1)}$ response from case one and two for both considered window lengths using *PCA*.It is immediately clear that *case one* results in a condition deviance point at record one hundred and fifty-six while *case two* results in a condition deviance point in the region surrounding record two thousand, for both window lengths considered. Clear implications of the assumed percentage of training data are noted on the threshold deviance response, whereby in *case one* the condition deviance point gives a binary detection response while for *case two* this did not occur. The reconstruction in Figures 5.5(c) and (d) appear visually smoother in the region around record two-thousand which is attributed to the assumed window length. For larger window lengths, if the anomalous instances seen by the model are infrequent and short in occurrence in comparison to the window length, the *LL* will naturally be less indicative unless the anomalous component is significant. The *case two* response appears to highlight records surrounding record two thousand, which were noticeably impulsive in Figure 5.2(b). Finally, severe damage is seen to occur near the end of the experimental life-span, with significant growth in record average shown in Figure 5.5.

Figure 5.6 shows the response from the three *LHI*s for *PCA* models with a window length of $L_w = 512$ under *case one* and *case two*. Clearly, the latent manifold is responding to damage and it is interpretable. All three *LHIs* indicate the presence of damage albeit clear that $LHI^{(3)}$ is noticeably worse. The intuition here is that the manifold is responding by representing instances of damage approximately perpendicular to the manifold path and consequently, the distance between latent instance representations and the distance from the Euclidean origin increases. The latent angle still exhibits a mild response to damage which attributed to the lack of perfect orthogonal path projection.



(a) Case one: $L_w = 512$

(b) Case two: $L_w = 512$

(c) Case one: $L_w = 4096$

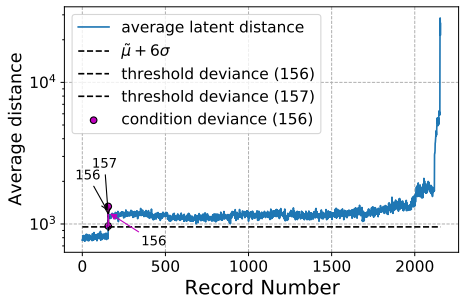
(d) Case two: $L_w = 4096$

Figure 5.5. The *PCA* model $HI^{(1)}$ response to all of the bearing three data from IMS dataset one for the two considered window lengths and training data cases.

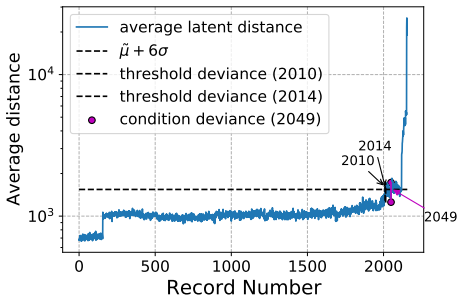
Furthermore, Figure 5.6 provides an intuition for the amount of training data used, where there is a clear condition deviance effect for this bearing data. One can note that for the *case two* model, it still captures the change around after one hundred and fifty-six in a different location on the manifold, which shows that the model latent manifold has an awareness to the fact that the records are different to the others. This may mean that there is some shift either induced by a fault or a change in system properties, with enough significance that the model cannot fully capture the change equally, as seen in Figures 5.6(b), (d) and (f). Clearly, the *PCA* models work extremely well on this bearing data which indicates that it is a simple dataset to detect damage on and additional model complexity may not be necessary. However, it is crucial that the complex models also perform well and thus, they will be analysed.

5.3.1.2 VAE Model Response

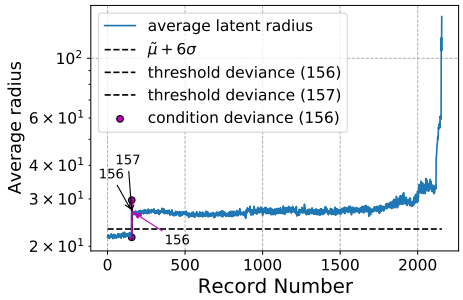
The results from the *VAE* models for bearing three data did not add any benefit over *PCA* and thus shall not be expanded completely. Under the VAE_2 model for *case two*, the jump at record one hundred and fifty-six disappears which indicates that the VAE_2 model learns a variance that can effectively capture the change, a curious result for the stochastic *VAE* model. The reason for this is attributed to the increased model complexity available to models trained with larger window lengths, which improves the expected model response given the training data. However, if the change is anomalous, it would be preferred that it be made known to the user. The author chose to analyse the $\beta - TC - VAE$ model in both its deterministic and stochastic form, however, it was noted that the response was similar to that shown for the *VAE* model. In fact, in reconstruction, the models performed equivalently with no clear jump after record one hundred and fifty-six for the $\beta - TC - VAE_2$ model trained under *case two*. Interested readers can refer to Figures E.1 and E.2 for a visual analysis of the *VAE* response.



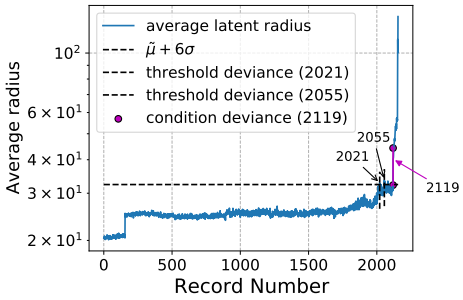
(a) $LHI^{(1)}$ using case one: $L_w = 512$



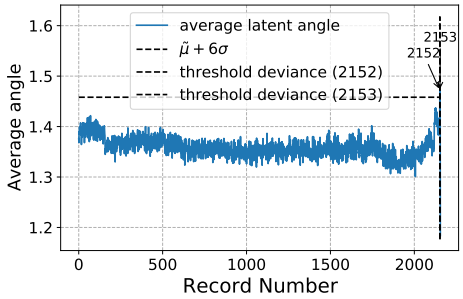
(b) $LHI^{(1)}$ using case two: $L_w = 512$



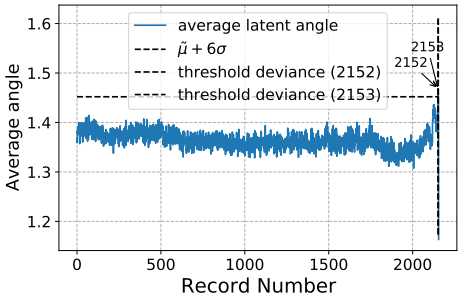
(c) $LHI^{(2)}$ using case one: $L_w = 512$



(d) $LHI^{(2)}$ using case two: $L_w = 512$



(e) $LHI^{(3)}$ using case one: $L_w = 512$



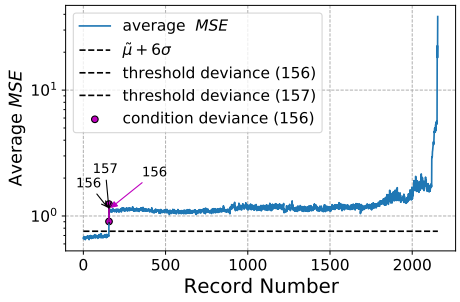
(f) $LHI^{(3)}$ using case two: $L_w = 512$

Figure 5.6. The *PCA* model *LHI* responses to all of the bearing three data from IMS dataset one for the two training data cases under a model window length of $L_w = 512$.

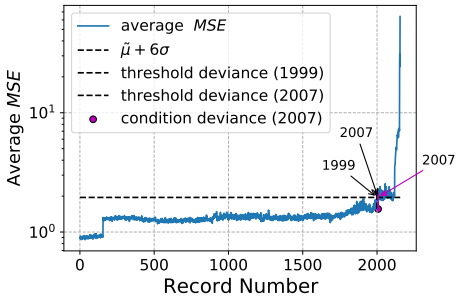
If one critically examines Figure E.1, it is clear to see that the addition of learning the model output variance is, in a fault diagnostic domain, unnecessary as the VAE_1 response is suitable. The only change where VAE_2 notably contributes is in the magnitude of the average, which is not a necessary requirement for this dataset. It is also interesting that the VAE_2 models respond to the records which have large kurtosis values, an indicator of impulsivity in a signal, as this implies that this impulsivity is highly unlike the healthy data which the model has seen and this is enhanced by the learnt variance in the *HI*. This can be attributed to the underlying assumption of a *VAE* where the output distribution is Gaussian, hence strong deviation from this is noticeable. This learnt variance also affects the learnt latent manifold, with clear differences noted between the VAE_1 and the VAE_2 response in Figure E.2.

5.3.1.3 GAN-based Model Response

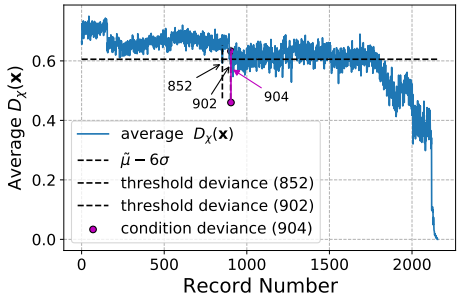
For the GAN-based methods, the author will present the response obtained from the $RY-GAN$ model for both *case one* and *case two* for a window length of $L_w = 512$ as these results provide the reader with a succinct summary of the response from the *GAN*-based models. The objective here is to indicate to the reader how the additional HIs available to the user respond and how the latent manifold responds under the GAN-based formulation.



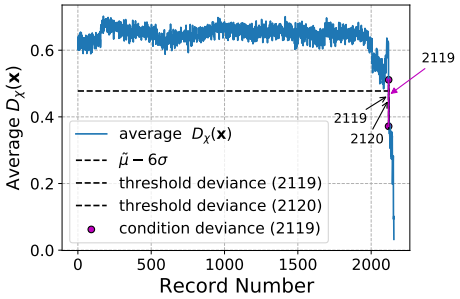
(a) $HI^{(1)}$: case one



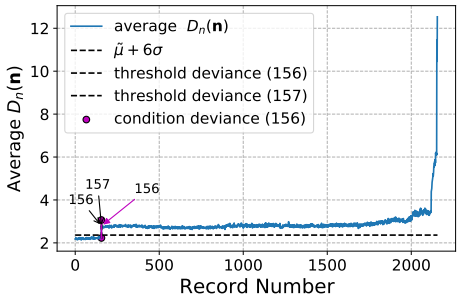
(b) $HI^{(1)}$: case two



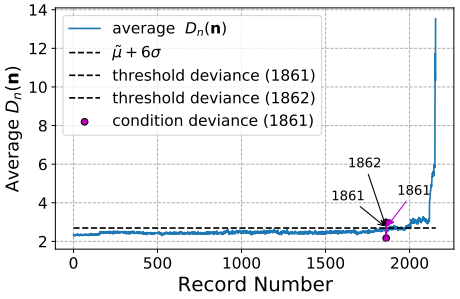
(c) $HI^{(2)}$: case one



(d) $HI^{(2)}$: case two



(e) $HI^{(3)}$: case one

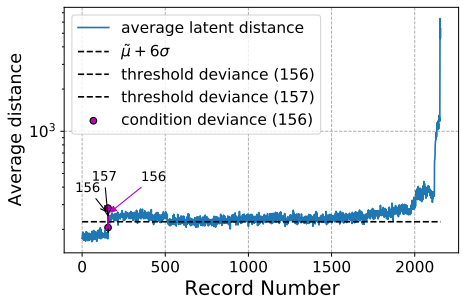


(f) $HI^{(3)}$: case two

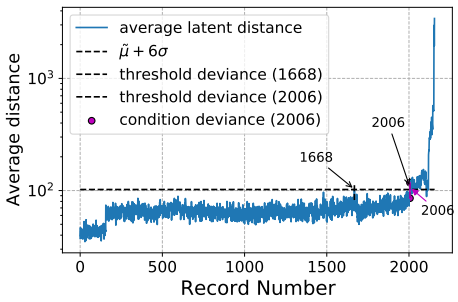
Figure 5.7. The mean response obtained from $HI^{(1)}$, $HI^{(2)}$ and $HI^{(3)}$ from a $RY-GAN$ model with a window length of $L_w = 512$ for the two cases on interest for bearing three from IMS dataset one.

In Figure 5.7, the response from the reconstruction log-likelihood, data discriminator and latent critic are shown and it is immediately noticeable how $HI^{(1)}$ and $HI^{(3)}$ are more responsive than $HI^{(2)}$. The $HI^{(2)}$ response performance can be attributed to the use of the L_2 objective function which operates in conflict to the *GAN* objective function. This is due to the difference in terms of what the generative

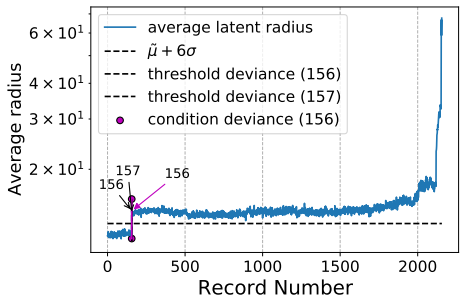
distribution $p(\mathbf{x}|\mathbf{z})$ is parametrised as, as the former is explicitly Gaussian while the latter is driven implicitly. The author does not believe that the use of the two in conjunction is beneficial to *GAN* training on vibration data and the part that suffers is the data discriminator. It is clear that $HI^{(3)}$ is a good *HI* as it is able to capture responses to damage and it appears to be less biased to the impulse data, attributed to the L_2 guidance and its effect on what information the encoder distribution captures. In the comparison between Figure 5.5 and Figure 5.7, it is clear that the $HI^{(1)}$ metric response is comparable, indicating that the *GAN*-based method produces satisfactory performance. The latent critic metric is also clearly indicative of damage and produces comparable responses to those obtained from *PCA*.



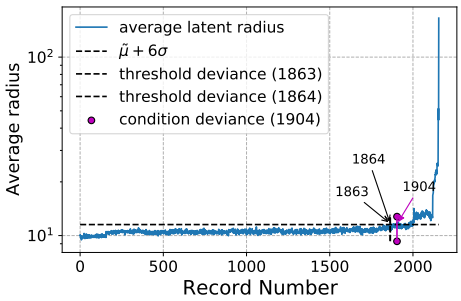
(a) $LHI^{(1)}$: case one



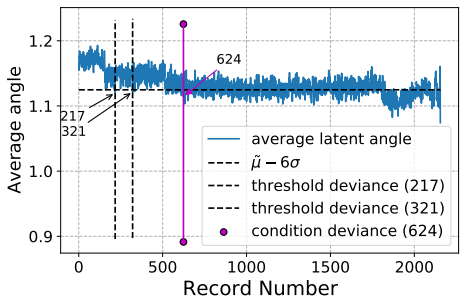
(b) $LHI^{(1)}$: case two



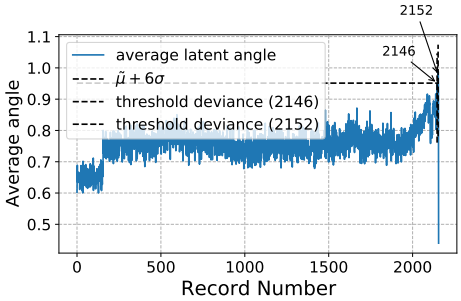
(c) $LHI^{(2)}$: case one



(d) $LHI^{(2)}$: case two



(e) $LHI^{(3)}$: case one



(f) $LHI^{(3)}$: case two

Figure 5.8. The *LHI* responses for a $RY - GAN$ model trained bearing three data for both 5% and 10% of the available data using a window length of $L_w = 512$.

In Figure 5.8, the *LHI* response for the two training data cases of the $RY - GAN$ are shown. $LHI^{(1)}$ and $LHI^{(2)}$ respond strongly to damage and are both indicative of the fault present. Figure 5.8(e) and (f) show $LHI^{(3)}$, with a weakened response and this is carried not only through both cases but it is

synonymous with how the *PCA* and the *VAE* models respond on the latent manifold. It is not unexpected that one *LHI* suffers while others perform well, as the three *LHI*s are linked in interpretation, with the resulting poor slow angle change attributed to the data following the same path through the latent space and the encoder mapping unseen anomalous instances far from the manifold but along the same trajectory. One can note how the response from the $RY-GAN$ model appears to be smoother for $LHI^{(1)}$ and $LHI^{(2)}$ which may be attributed to the built-in disentanglement allowing the **s** latent component to capture the deterministic part of the signal while **n** captures the rest of the information present in the signal. Figure 5.8(d) provides a less noticeable change at record one hundred and fifty-six and its response appears to be akin to the latent critic response shown in Figure 5.7(d).

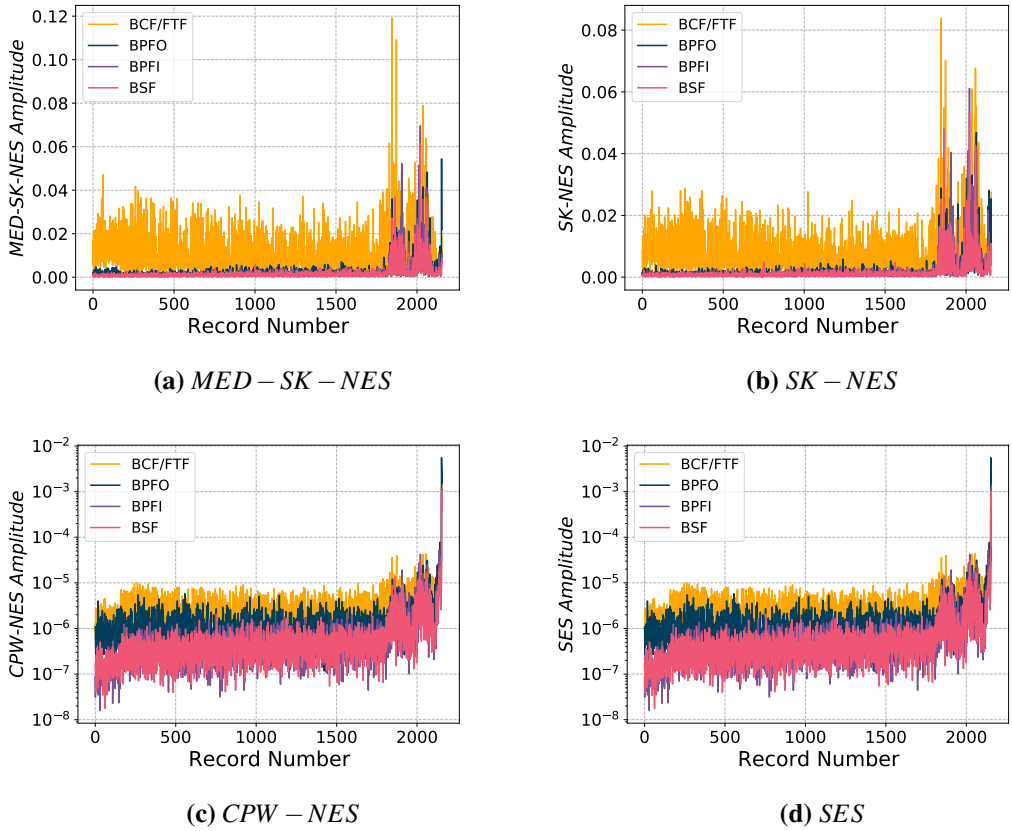
5.3.1.4 Signal Processing

For the next segment of this report, it is required to demonstrate how the four signal processing approaches respond on the third bearing data for the first dataset. This will show how the use of latent variable models is competitive with the state-of-the-art signal processing techniques. Note that the *SES* is used as a performance baseline technique, as it is often beneficial to utilise pre-processing techniques such as *MED-SK* or *CPW* filtering before analysing the *SES*. Figure 5.9 shows the various signal processing responses at the frequency amplitudes of interest for the third bearing. It is clear to note that the $MED-SK-NES$ and $SK-NES$ methods, shown in Figure 5.9(a) and (b) respectively, do not provide any clear indication of damage with a noisy *BCF* component. This noise, however, is attributed to the SNR noise floor and hence it is un-interpretable. The $CPW-NES$ and *SES* methods seem to provide better results, shown in Figure 5.9(c) and (d) respectively, with a clear drastic fault frequency progression in the final records. Oddly, these methods do not correctly identify which component is responsible for the damage, with all fault frequencies showing rapid growth in frequency magnitude. Furthermore, it is non-trivial to identify a point of condition deviance when using $MED-SK-NES$ and $SK-NES$ and as such, no threshold was defined. It is possible to do so for the $CPW-NES$ approach and the *SES* approach but it is limited to the final records. One can note a gradual growth in magnitude at record one hundred and fifty-six but this change, however, does not compare with the latent variable model responses.

5.3.1.5 Result Consolidation and Conclusion

To consolidate and represent the results obtained for this bearing data, Table 5.4 is provided such that the results for models trained under *case one* and *case two* for two input lengths considered can be interpreted and summarised. It must be noted here that some results can be classified into three inconclusive (IC) indicators, namely, IC_1, IC_2 and IC_3. The term IC_1 was given to any health indicator that was deemed a failure. IC_2 is an indicator used for *case two* models where the change at record one hundred and fifty-six affected the results in a manner that caused the health indicator condition deviance approach to be inconclusive but still indicative of anomalous data. Finally, IC_3 was reserved for the GAN-based approaches to indicate cases where the data discriminator training was unsatisfactory. The latter was included to highlight the flaws associated with the current training approach for the GAN-based methods which was attributed to the inclusion of both the L_2 and GAN training schemes in the model. As with the phenomenological model, the author chose to use the discrepancy signal mean from the *temporal preservation* approach for the results in Table 5.4 and all those that follow for the IMS dataset analysis, hence some results may differ to the figures shown.

In the analysis of Table 5.4, it is clear that the choice of the amount of training data greatly affects the condition deviance point that the threshold detects. It is common for *case one* models to isolate record one hundred and fifty-six. There are three approximate ranges that one can group the results of Table 5.4 into, with the first range around record one hundred and fifty-six, the second around the impulse signal band near record one thousand eight hundred to two thousand and the third is when the rapid



(a) $MED-SK-NES$ **(b)** $SK-NES$

(c) $CPW-NES$ **(d)** SES

Figure 5.9. The four considered signal processing approaches frequency amplitude at the four frequencies of interest for the first channel of bearing three from IMS dataset one. Notice the clear lack of fault progression when the $MED-SK-NES$, (a), and $SK-NES$, (b), approaches are used.

Table 5.4. The obtained threshold condition deviance point from the first IMS dataset for bearing three when investigating the HIs. Note that IC_1 is the abbreviation used for inconclusive, IC_2 refers to a case where the change at record 155 affects result performance and IC_3 refers to a poorly trained discriminator

Model type and characteristics		**Health indicator condition deviance point from case one ‖ case two**					
Model used	**Window length**	**HI**$^{(1)}$	**HI**$^{(2)}$	**HI**$^{(3)}$	**LHI**$^{(1)}$	**LHI**$^{(2)}$	**LHI**$^{(3)}$
PCA	$L_w = 512$	156 ‖ 2055	N/A	N/A	156 ‖ 2049	156 ‖ 2119	IC_1 ‖ IC_1
	$L_w = 4096$	156 ‖ 2019	N/A	N/A	156 ‖ 2049	156 ‖ 2076	1664 ‖ 2120
VAE_1	$L_w = 512$	156 ‖ 2019	N/A	N/A	196 ‖ 2119	196 ‖ IC_2	594 ‖ IC_1
	$L_w = 4096$	156 ‖ 2119	N/A	N/A	2055 ‖ 2121	2134 ‖ 2121	2119 ‖ IC_1
VAE_2	$L_w = 512$	156 ‖ 1843	N/A	N/A	637 ‖ 2121	177 ‖ 1843	1842 ‖ 2131
	$L_w = 4096$	156 ‖ 2019	N/A	N/A	2133 ‖ 2135	IC_1 ‖ IC_1	1739 ‖ IC_1
$\beta-TC-VAE_1$	$L_w = 512$	156 ‖ 2010	N/A	N/A	156 ‖ 2127	182 ‖ IC_2	594 ‖ IC_2
	$L_w = 4096$	156 ‖ 2119	N/A	N/A	2120 ‖ IC_2	2120 ‖ IC_2	IC_1 ‖ IC_2
$\beta-TC-VAE_2$	$L_w = 512$	156 ‖ 1843	N/A	N/A	1985 ‖ 2119	1819 ‖ 1843	2010 ‖ 2119
	$L_w = 4096$	156 ‖ 2006	N/A	N/A	2120 ‖ 2120	2134 ‖ IC_1	2119 ‖ IC_2
$RY-GAN$	$L_w = 512$	156 ‖ 2007	904 ‖ 2119	156 ‖ 1861	156 ‖ 2006	156 ‖ 1904	624 ‖ IC_1
	$L_w = 4096$	156 ‖ 2010	156 ‖ 2010	284 ‖ 2007	284 ‖ 2120	637 ‖ 1905	IC_1 ‖ 2136
$DLS-GAN$	$L_w = 512$	156 ‖ 2019	IC_3 ‖ IC_3	156 ‖ 2009	162 ‖ 2006	156 ‖ 2049	IC_1 ‖ IC_1
	$L_w = 4096$	156 ‖ 2010	156 ‖ IC_3	1903 ‖ IC_2	637 ‖ 2006	239 ‖ IC_2	2123 ‖ IC_1

failure occurs from record 2100. These bands allow one to home in on where exactly the condition deviance point is and why the model has chosen the point. It is also clear that the decision of the amount of training data greatly affects the condition deviance point with a large number of IC_2 cases. With regards to window length, it is clear to see that the larger window length tends to favour the rapid failure point through the *LHI*s. The larger window length alters the performance of the *LHI*s, with $LHI^{(2)}$ often performing poorly and this is exaggerated by the amount of training data used. When one compares the *VAE* models to that of the *GAN*-based, the additional model design complexity aids in model performance and produces more consistent results. *PCA* also appears to be a highly competitive method which shows that this bearing data is constrained in a manner that is interpretable through a linear latent variable model. One clear positive, when one compares the signal processing results to the latent variable model results, is that the latent variable models out-perform signal processing in result interpretability and in result conclusiveness. The presence and response to damage can be isolated, as opposed to the signal processing case where there is some discrepancy and the true fault is unclear.

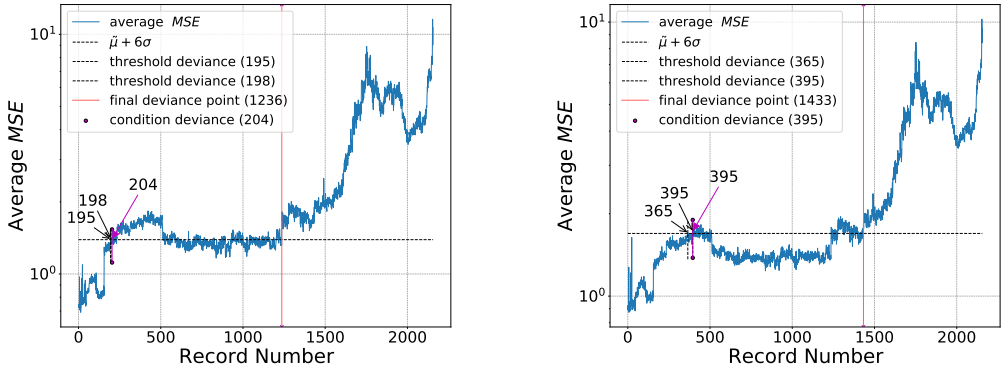
5.3.2 Dataset One: Bearing Four

For the fourth bearing, the analysis process will remain predominantly the same as that shown for bearing three with a slight focus on the interesting response results that were noted by the author. As with bearing three, the author chose to investigate the effect of the assumed amount of training data and difference in metric response based on a *case one* or *case two* approach was investigated for this bearing data. The author chose to not show the *PCA* response for this bearing, as its response was similar to that found from the *VAE* and *GAN* models.

The response results from this bearing data under the *case one* and *case two* training data options offered an interesting analysis as there appears to be a region of self-healing present in the response. This caused *case one* results to often identify a early condition deviance point but between records five hundred and one thousand the response would heal and go under the threshold. To aid in response quantification, a final deviance point is identified as the point that represents the final record that crosses the threshold. The final deviance point can be interpreted as the record from which all other record *HI* or *LHI* discrepancy average metrics are classified as damaged.

5.3.2.1 VAE Model Response

For the *VAE* discussion in this work, the author chose to show the results from training case one and will rather illustrate how the latent manifold changed as a result of the model window length. In Figure 5.10, the response from $HI^{(1)}$ under the two window lengths can be found, with it clear that both VAE_1 models respond well to damage and that the damage progression is clear. There are identifiable points where sudden jumps in mean occur and after an initial period of growth, there is a decline prior to the sudden growth from the discrepancy signal average around record one thousand five hundred. The images in Figure 5.10 also provide one with a clear indication of how the threshold is robust to fault progression but it is still a binary method of fault identification. In the presence of apparent self healing, the response may require some human interpretation to quantify the damage growth present. It is a non-trivial task to identify a clear binary point of fault diagnosis in vibration data but the process is greatly simplified if the health indicator responds strongly to damage amidst all other machine conditions and factors, which is the underlying purpose of vibration-based condition monitoring. It is clear in Figure 5.10 that the addition of more training data would ultimately be problematic as the threshold variance will automatically capture the jump at record one hundred and fifty six. The final condition deviance point is also a function of the training data percentage, with the *case one* response identifying an earlier final deviance point.

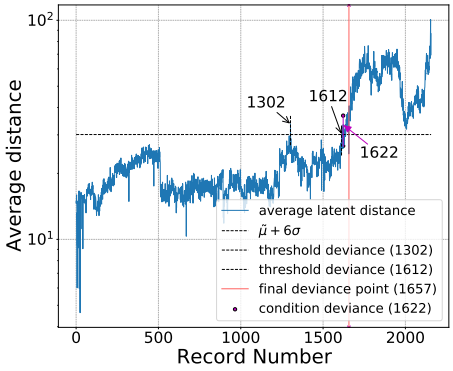


(a) $VAE_1 : L_w = 512$ **(b)** $VAE_1 : L_w = 4096$

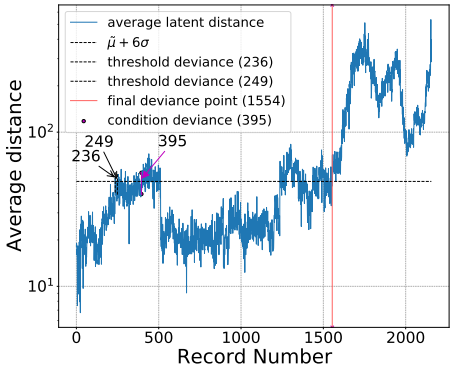
Figure 5.10. The VAE_1 response for two different window lengths for data from bearing four of the first IMS dataset using 5% of the available data for training.

In Figure 5.11, the three *LHI*s obtained using the *temporal preservation* approach from VAE_1 models with different window lengths for case one are shown. It is evident that, for the *VAE* models, the latent manifold manifests differently based on the window length. One can note how the robust metric now appears to be $LHI^{(2)}$ for $L_w = 512$, with the other *LHI*s showing strong off-manifold responses to the healthy data that has been identified as anomalous training instances. The presence of these anomalous instances is interesting as the model reconstruction did not identify these points. The reason for these anomalies will be explored shortly, but it is clear that the *LHI*s used in this work allow for clear result augmentation and give the user a better picture into the data they analyse. It is clear for $LHI^{(3)}$, as shown in Figures 5.11(e) and (f), the response is poor for a shorter window length while improved for a larger window length, at the expense of a strong response from $LHI^{(2)}$ with Figure 5.11(d) exhibiting an objectively slower response. The final deviance record identified in Figure 5.11 appears to be change based on latent metric and window length, which is attributed to the presence of anomalous training instances in the data.

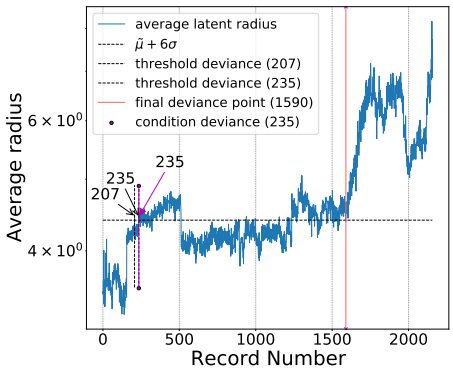
The change in performance of the *LHI*s for different window lengths was also found to be consistent across the deterministic and stochastic *VAE* models and the $\beta - TC - VAE$ models. This consistent response indicates that, for data from the fourth bearing from the first IMS dataset, the learnt manifold from a larger window length responds strongly to the presence of damage and does so by pushing anomalous data instances far from the healthy manifold with significant alterations to the path travelled through the manifold. This may be a result of the information present in signals with longer window lengths, as larger segments will naturally have more frequency content. This additional content may fundamentally change the learnt manifold as more information is to be captured in the manifold to produce adequate signal reconstruction. It may also be a function of how faults interact with the window length and the shaft frequency, as the ratio $\frac{f_s}{f_{shaft}}$ is a point between the two considered window lengths and thus the interaction between the two models and the fault may be different based on the fault frequency. It can be easily reasoned that the larger window length potentially has more faults in a given window and thus its response may be more significant through all latent components as the remaining healthy structure in the observed signal has been reduced.



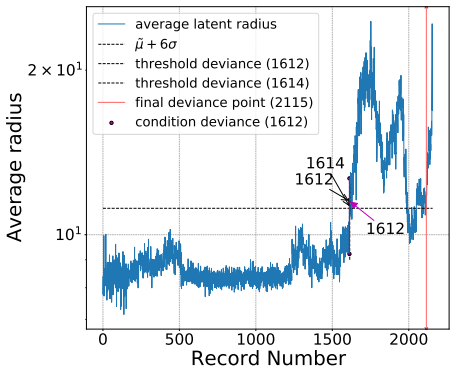
(a) $LHI^{(1)}$: $L_w = 512$



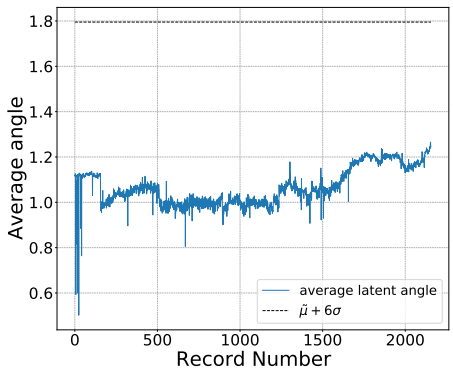
(b) $LHI^{(1)}$: $L_w = 4096$



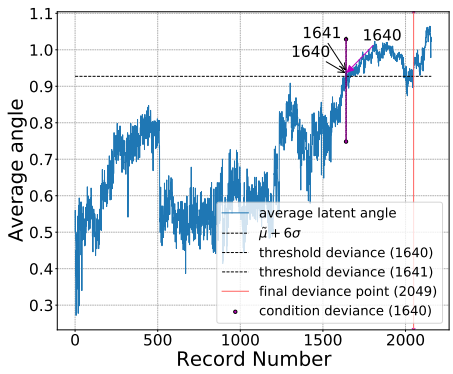
(c) $LHI^{(2)}$: $L_w = 512$



(d) $LHI^{(2)}$: $L_w = 4096$



(e) $LHI^{(3)}$: $L_w = 512$



(f) $LHI^{(3)}$: $L_w = 4096$

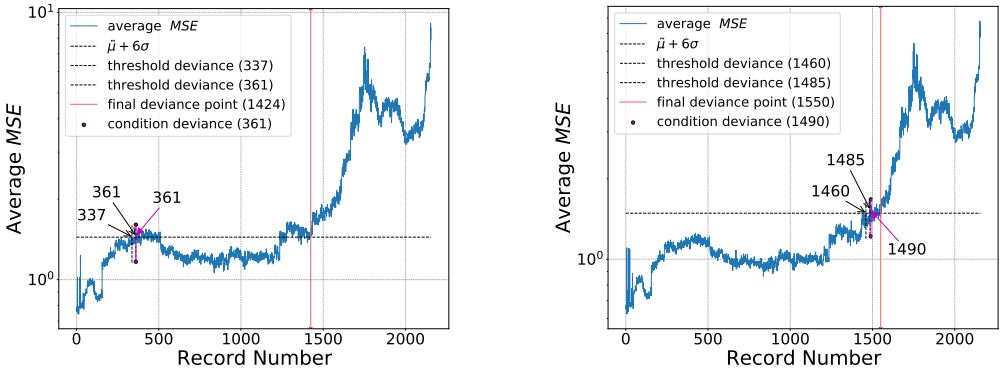
Figure 5.11. The three *LHI*'s for a VAE_1 model trained using two different window lengths using a case one approach to bearing four from IMS dataset one.

5.3.2.2 GAN-based Model Response

For the discussion of the performance of the GAN-based methods, the author will present the $RY-GAN$ results from *case one* and *case two* under a window length of $L_w = 512$. The purpose here is two-fold, the reconstruction log-likelihood, data discriminator and latent critic of the two methods must be analysed under the assumption of the amount of data provided for training and a latent manifold response investigation must be conducted. In Figure 5.12, the data discriminator and latent critic response is given and the data discriminator is clearly responding to damage, however it is still somewhat weakened in performance. The author does still believe that there are problems with the model formulation from the *GAN* side can be attributed to the sub-optimal performance. This problem is the presence of the L_2 objective function which drives the model towards Gaussian distributions while the GAN loss allows for distribution flexibility but in competition with the L_2 loss and ultimately the exploitation of this flexibility is not fully utilised. The amount of training data does affect the performance of the $RY-GAN$ on the data discriminator, as seen in Figure 5.12(b) where the condition deviance point occurs later. With regards to the latent critic, it is clear that the latent space is responding to the anomalous records that have been noted in the other model results shown on this dataset, a response that, albeit unpleasant, is not unexpected when one considers the training objective of the latent critic. Its objective is to enforce that the latent manifold is an isotropic Gaussian and if these points deviate from the other data shown to the model and is learnt as such, the latent critic will indicate these latent representations to be anomalous. This has a significant effect on the case two response shown in Figure 5.12(d) where the jump at record one hundred and fifty-six is significant enough to alter the threshold deviance significantly. It is also clear that record twenty-four is problematic with its average going past the threshold in Figures 5.12(c) and (d).

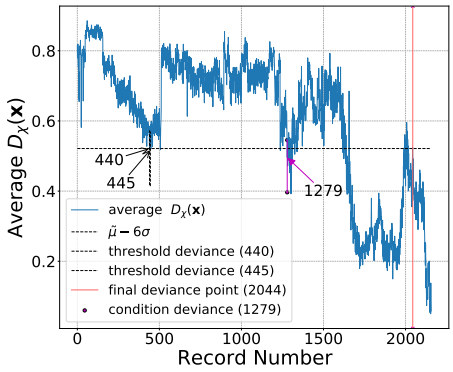
From the identified final deviance points in Figures 5.12(a)-(d), it is clear that for the reconstruction log-likelihood and data discriminator the performance is equivalent. This is attributed to the lack of anomalous instance segmentation in the data manifold, which ensures that the threshold is not affected by metric anomalies in the healthy data. The benefit of identifying final deviance records is clear, as the condition deviance point neglects the presence of bearing self-healing. The results from the *case one* metrics do identify earlier final deviance points, with this attributed to the jump at record one hundred and fifty-five affecting the threshold.

In Figure 5.13, the three *LHI*s as shown from the $RY-GAN$ model, wherein the latent anomalies are evident and having a clear effect on the all three latent metrics. The best metric is $LHI^{(1)}$ as it shows less response to the outliers present in the data and is capable of identifying a clear final deviance point. The use of the final deviance point is also clear between the *case one* and *case two* results in Figures 5.13(a) and (b). The difference between the condition deviance points is significant but less so for the final deviance point, which indicates that less training data is useful for detecting the initial presence of the fault but less so for identifying the final deviance point. All three latent metrics also show a response to damage but unfortunately the severe deviance of the healthy data anomalies has shifted the threshold to a region where fault diagnosis is not possible. The result of the *case one* and *case two* investigation is also evident in this dataset, where a threshold of larger magnitude was developed for all case two results. It is clear from Figures 5.13(e) and (f) that the latent angle has a more gradual response to damage and that the outliers in the data increase the threshold to an extent where no condition deviance point can be identified. It is noticeable from Figure 5.13, in comparison to Figure 5.10, that the latent manifold response is similar to the other models used in this work. This further emphasises how the connection between the models through the L_2 objective function has a strong influence on the latent space, regardless of how the latent manifold is constrained. This shows

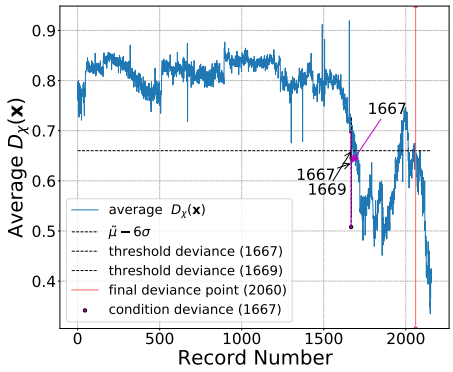


(a) $HI^{(2)}$: case one

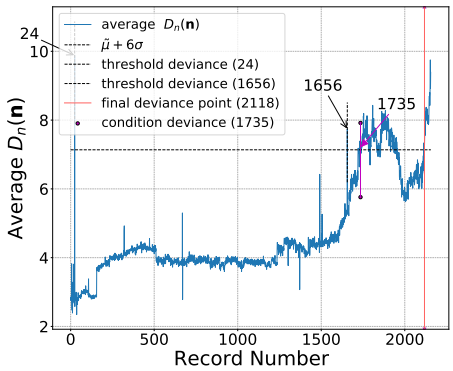
(b) $HI^{(2)}$: case two



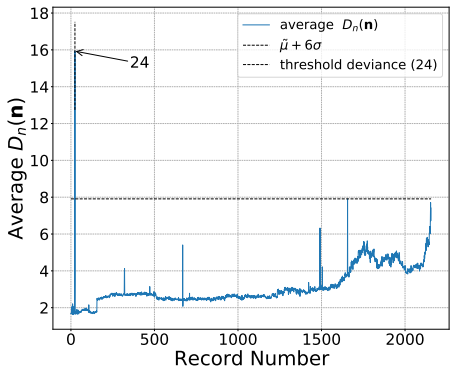
(c) $HI^{(2)}$: case one



(d) $HI^{(2)}$: case two



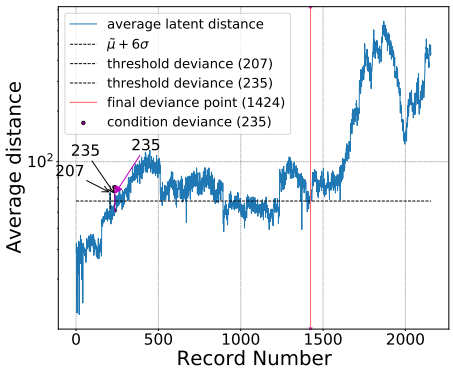
(e) $HI^{(3)}$: case one



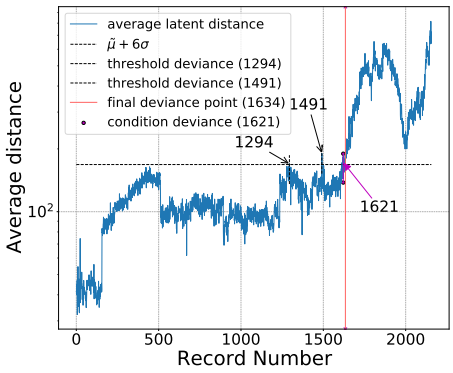
(f) $HI^{(3)}$: case two

Figure 5.12. The $RY-GAN$ model response through $HI^{(1)}$, $HI^{(2)}$ and $HI^{(3)}$ for the two training cases considered for data from bearing four from the first IMS dataset under a window length of $L_w = 512$.

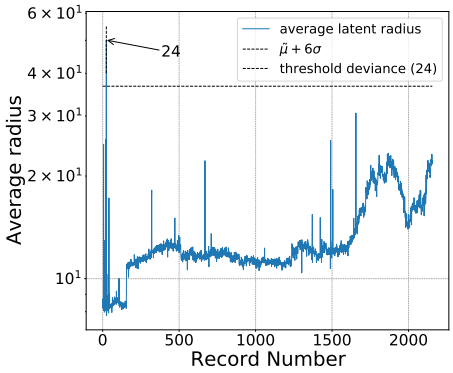
that the L_2 loss is influential in model training, which is not unexpected but its reach extends into similarities in how the latent manifold is constructed between methods.



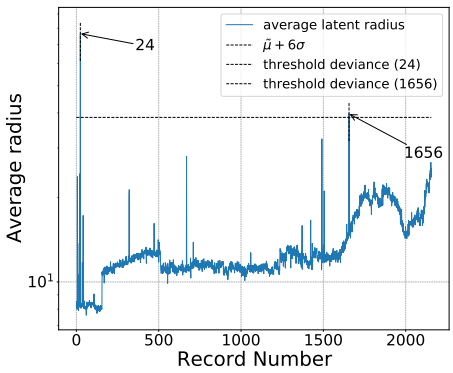
(a) $LHI^{(1)}$: case one



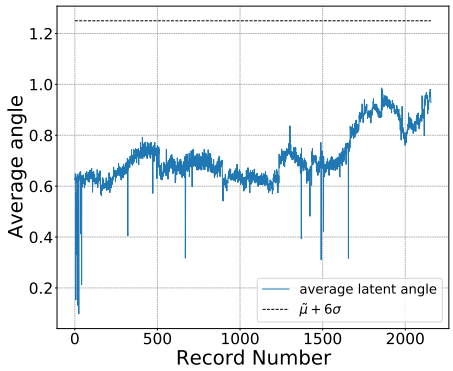
(b) $LHI^{(1)}$: case two



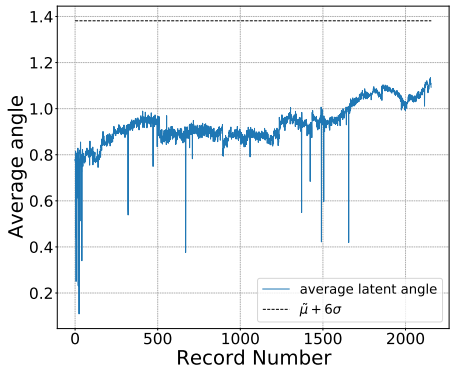
(c) $LHI^{(2)}$: case one



(d) $LHI^{(2)}$: case two



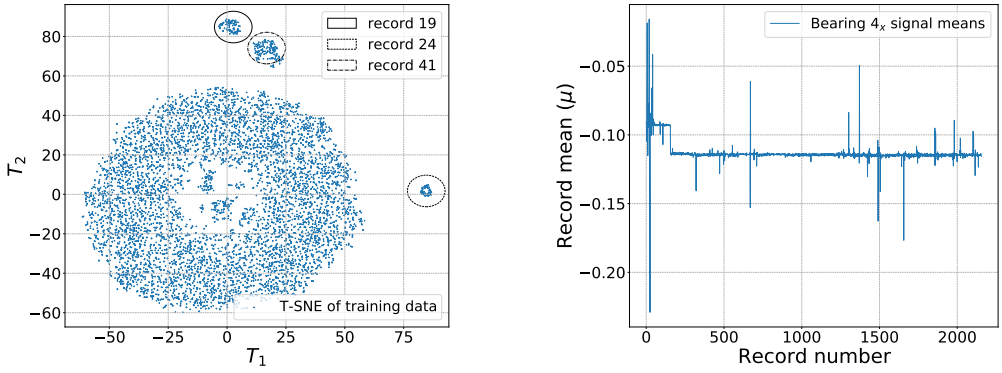
(e) $LHI^{(3)}$: case one



(f) $LHI^{(3)}$: case two

Figure 5.13. The various LHI's obtained from a $RY-GAN$ model trained on the two training data cases on interest for bearing four data from IMS dataset one with a model length of $L_w = 512$.

To investigate the anomalous records, the author chose to look at the signal statistics and the latent representation of the signal in the **n** space using $T-SNE$ to visualise the training data in its higher



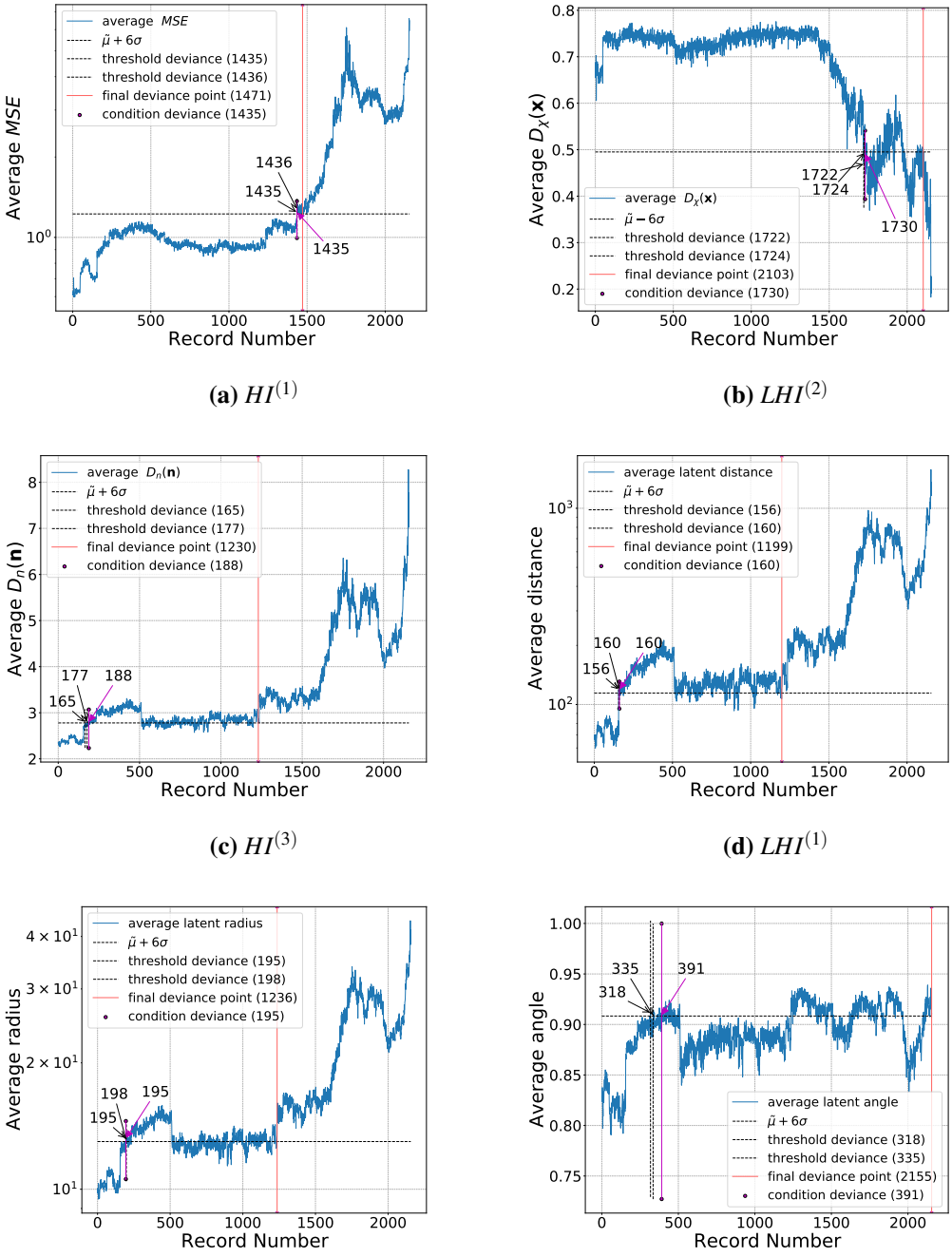
(a) $T-SNE$ for **n**

(b) Record mean

Figure 5.14. A $T-SNE$ latent **n** visualisation, Figure 5.14(a), and the record mean, Figure 5.14(b), for the training data used for a $RY-GAN$ model trained on the first 5% of the vibration data. It was noted for Figure 5.14(a) that records 6, 8, 19, 36 and 41 were all anomalous, however the clusters were centred around the records noted in the Figure.

dimensional space to two dimensions (van der Maaten and Hinton, 2008). In Figure 5.14(a), it is clear to note that there are obvious outliers present in the data, where these outliers have been identified and labelled for the reader. To investigate why this is the case, the author chose to analyse the statistical properties of all of the records obtained for the fourth bearing, to determine if any explanation can be obtained. In Figure 5.14(b), the record mean is shown for all records and it is clear to see that there is some noticeable fluctuation in the signal mean, which may be indicative of a sensor malfunction. It is clear, as shown in Figure 5.3, how the average with the record mean has clear impulses which contradicts that shown in Figure 18(b) in the work of Qiu et al. (2006). This indicates that Qiu et al. (2006) may have calculated an RMS with a removal of the record mean, a typical pre-processing technique in signal processing. The record mean is a statistical feature of the data and it is clear that the models used in this work are sensitive changes in the mean, which, in itself, shows that the models have recovered this information. To clarify to the reader how the results in this work can be immediately improved, the $HI^{(3)}$ and the three LHIs for a $RY-GAN$ model with a window length of $L_w = 512$ was trained on 5% of the data and these results are shown in Figure 5.15. It is clear that the identified condition deviance point is now more consistent between metrics, which indicates that the metrics are distinctive and are able to detect the presence of anomalous data instances.

The removal of the mean as a feature from the data immediately improves the latent results in fault detection but these anomalous instances have now been lost, a dangerous repercussion of manually removing information from the signals. This result speaks to the use of deep learning for anomaly detection and how just looking for anomalies in signal reconstruction is not always suitable, as the latent space may provide information that is not captured in the signal reconstruction. The latent critic is powerful as it provides one with a metric to quantify and possibly detect anomalous instances but, one has to analyse the latent manifold in a deeper manner to quantify why the latent critic responds in the way it does. This is where the latent metrics proposed in this work interpret the latent manifold as they are not restricted to a specific model and are not required to be built into the model training procedures in some way, which is a powerful notion for broad applicability.



(a) $HI^{(1)}$ **(b)** $LHI^{(2)}$

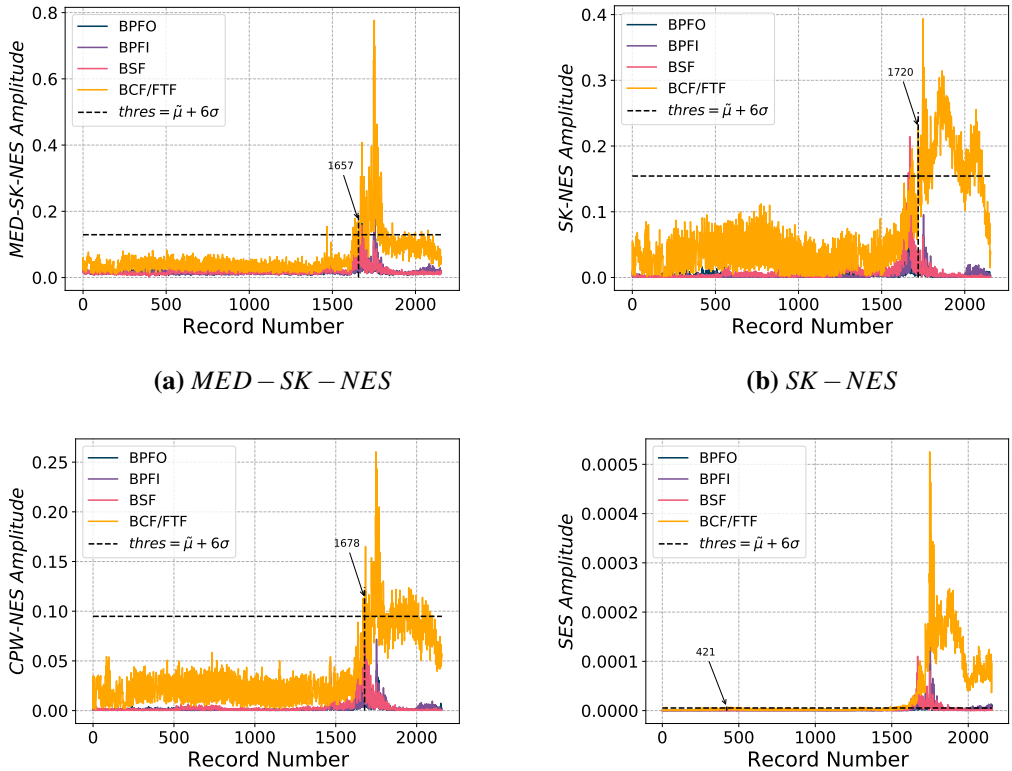
(c) $HI^{(3)}$ **(d)** $LHI^{(1)}$

(e) $LHI^{(2)}$ **(f)** $LHI^{(3)}$

Figure 5.15. The latent critic and *LHI* responses for a $RY-GAN$ model trained on bearing four data with the mean for each record removed. Notice the clear improvement for all cases presented in this figure in comparison to that shown in Figures 5.12 and 5.13.

5.3.2.3 Signal Processing

Figure 5.16 shows the four signal processing response amplitudes at the fault frequencies of interest for the fourth bearing. A clear growth in the FTF component is noticeable throughout all the methods and they all follow a similar trend. There also appears to be two large spikes on the other components which can also be noted in the signal kurtosis shown in Figure 5.3(b). It is interesting that this growth be noted in the FTF, as this type of fault was not indicated in the original write up of this dataset. Due to this discrepancy, the author has thus chosen to present the $CPW-NES$ and SES spectrum for record two thousand such that it is clear that this result is not fictitious. Figure 5.17 demonstrates the presence of the FTF in both spectra and one can note how the component is dominant and shows a presence of its third harmonic. Booyse et al. (2020) showed a similar growth in the FTF when using a technique called the Improved Envelope Spectrum, detailed in Abboud et al. (2019). For the signal processing results obtained, the best performing method is that of the SES with a condition deviance point identified at record 1528.



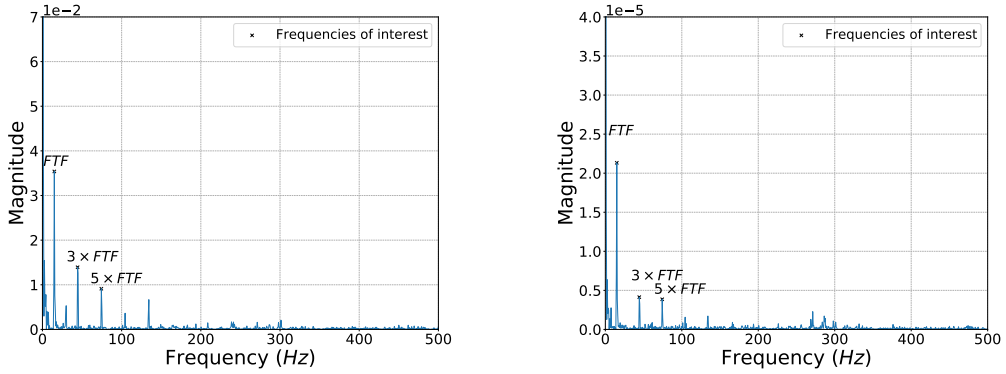
(a) $MED-SK-NES$ **(b)** $SK-NES$

(c) $CPW-NES$ **(d)** SES

Figure 5.16. The four considered signal processing approaches frequency amplitude at the four frequencies of interest for the first channel of bearing four from IMS dataset one. Notice the amplitude progression in the FTF throughout all methods.

5.3.2.4 Result Consolidation and Discussion

In Tables 5.5 and 5.6, the multiple model results from the fourth bearing data from the IMS dataset are presented and as such can now be interpreted. In Tables 5.5 the condition deviance points are given while in Tables 5.6 the final deviance points are given. There are model health indicators that have been identified as IC_2, a condition that indicates that the anomalous records resulted in a threshold



(a) $CPW - NES$ spectrum **(b)** SES spectrum

Figure 5.17. The resulting spectrum for record 2000 for IMS dataset one obtained through CPW, (a), and the SES, (b). Notice the clear spike at the FTF in both cases, with clear harmonics.

that was ineffective for condition deviance detection. Tables 5.5 and 5.6 also indicates the effect of training on 5% or 10% of the data, with the latter producing condition deviance points that occur later. One can note the impact of window length on the latent manifold, with the larger window length cases producing responses in $LHI^{(3)}$ for the VAE-based models. The addition of the $\beta - TC - VAE_1$ model does not seem to add much benefit to this dataset, with similar performance results obtained to that of a VAE. The GAN-based models also appear to be heavily influenced by the anomalous records, as pointed out in the previous sections of this work. It was also found that the $DLS - GAN$ was a poorer model to train from a data discriminator perspective, with the $RY - GAN$ model producing better responses.

Table 5.5. The obtained threshold condition deviance point from the first IMS dataset for bearing four when investigating the HIs. Note that IC_1 is the abbreviation used for inconclusive, IC_2 refers to a case where the anomalous signals offset the threshold substantially and IC_3 refers to a poorly trained discriminator

Model type and characteristics		**Health indicator condition deviance point from case one ‖ case two**					
Model used	**Window length**	**HI**$^{(1)}$	**HI**$^{(2)}$	**HI**$^{(3)}$	**LHI**$^{(1)}$	**LHI**$^{(2)}$	**LHI**$^{(3)}$
PCA	$L_w = 512$	1509 ‖ 1559	N/A	N/A	235 ‖ 1490	1667 ‖ 1656	IC_1 ‖ IC_1
	$L_w = 4096$	1277 ‖ 1514	N/A	N/A	193 ‖ 1555	IC_2 ‖ 1681	IC_1 ‖ IC_1
VAE_1	$L_w = 512$	204 ‖ 1277	N/A	N/A	1622 ‖ 1279	235 ‖ 1636	IC_1 ‖ IC_1
	$L_w = 4096$	395 ‖ 1607	N/A	N/A	395 ‖ 1302	1612 ‖ 1721	1640 ‖ 1649
VAE_2	$L_w = 512$	235 ‖ 1467	N/A	N/A	1649 ‖ 1657	1640 ‖ 1632	IC_1 ‖ IC_1
	$L_w = 4096$	1597 ‖ 1559	N/A	N/A	364 ‖ 1613	1612 ‖ 421	1806 ‖ IC_1
$\beta - TC - VAE_1$	$L_w = 512$	195 ‖ 1491	N/A	N/A	361 ‖ 1640	403 ‖ 1667	IC_1 ‖ IC_1
	$L_w = 4096$	355 ‖ 1596	N/A	N/A	1302 ‖ 1612	1657 ‖ 1699	1622 ‖ 1681
$\beta - TC - VAE_2$	$L_w = 512$	235 ‖ 1467	N/A	N/A	1657 ‖ 1640	1619 ‖ 1657	IC_1 ‖ IC_1
	$L_w = 4096$	439 ‖ 1555	N/A	N/A	456 ‖ 1609	1619 ‖ 406	2045 ‖ IC_2
$RY - GAN$	$L_w = 512$	361 ‖ 1490	1279 ‖ 1667	1735 ‖ IC_2	235 ‖ 1621	IC_2 ‖ IC_2	IC_2 ‖ IC_2
	$L_w = 4096$	361 ‖ 1552	156 ‖ IC_2	2145 ‖ IC_2	IC_1 ‖ 1635	IC_2 ‖ IC_2	IC_2 ‖ IC_1
$DLS - GAN$	$L_w = 512$	1277 ‖ 1491	IC_3 ‖ IC_3	IC_2 ‖	160 ‖ 1277	1678 ‖ 1749	IC_1 ‖ IC_1
	$L_w = 4096$	235 ‖ 1542	IC_3 ‖ IC_3	1294 ‖ 1663	235 ‖ 1237	1236 ‖ 1294	IC_2 ‖ IC_2

Table 5.6. The obtained final condition deviance point from the first IMS dataset for bearing four when investigating the HI's. Note that IC_1 is the abbreviation used for an inconclusive metric result.

Model type and characteristics		Health indicator condition deviance point from case one ‖ case two					
Model used	**Window length**	**HI**$^{(1)}$	**HI**$^{(2)}$	**HI**$^{(3)}$	**LHI**$^{(1)}$	**LHI**$^{(2)}$	**LHI**$^{(3)}$
PCA	$L_w = 512$	1528 ‖ 1568	N/A	N/A	1405 ‖ 1577	2091 ‖ 1667	IC_1 ‖ IC_1
	$L_w = 4096$	1433 ‖ 1550	N/A	N/A	1236 ‖ 1605	2155 ‖ 2115	IC_1 ‖ IC_1
VAE_1	$L_w = 512$	1236 ‖ 1590	N/A	N/A	1657 ‖ 1612	1590 ‖ 2049	IC_1 ‖ IC_1
	$L_w = 4096$	1433 ‖ 1612	N/A	N/A	1554 ‖ 1657	2115 ‖ 2155	2049 ‖ 2123
VAE_2	$L_w = 512$	1414 ‖ 1552	N/A	N/A	2119 ‖ 2119	2113 ‖ 2090	IC_1 ‖ IC_1
	$L_w = 4096$	1607 ‖ 1606	N/A	N/A	1577 ‖ 2119	2139 ‖ IC_1	1993 ‖ IC_1
$\beta - TC - VAE_1$	$L_w = 512$	1236 ‖ 1596	N/A	N/A	1657 ‖ 2116	1605 ‖ 2127	IC_1 ‖ IC_1
	$L_w = 4096$	1418 ‖ 1605	N/A	N/A	1612 ‖ 2116	2118 ‖ IC_1	IC_1 ‖ 2143
$\beta - TC - VAE_2$	$L_w = 512$	1427 ‖ 1553	N/A	N/A	2117 ‖ 2119	2109 ‖ 2113	IC_1 ‖ IC_1
	$L_w = 4096$	1433 ‖ 1606	N/A	N/A	1606 ‖ 2116	2143 ‖ 2155	IC_1 ‖ IC_1
$RY - GAN$	$L_w = 512$	1424 ‖ 1550	2044 ‖ 2060	2118 ‖ IC_1	1424 ‖ IC_1	IC_1 ‖ IC_1	IC_1 ‖ IC_1
	$L_w = 4096$	1432 ‖ 1577	1372 ‖ IC_1	2145 ‖ IC_1	IC_1 ‖ 2008	IC_1 ‖ 2155	IC_1 ‖ IC_1
$DLS - GAN$	$L_w = 512$	1433 ‖ 1550	IC_1 ‖ IC_1	2155 ‖ IC_1	1540 ‖ 2045	2124 ‖ 2146	IC_1 ‖ IC_1
	$L_w = 4096$	1367 ‖ 1554	IC_1 ‖ IC_1	1629 ‖ 2123	1542 ‖ 2044	1611 ‖ 2049	IC_1 ‖ IC_1

In Table 5.6, there appears to be a more consistent deviance point recognised between the *case one* and *case two* models, with less significant differences. As with Table 5.5, the record mean was still a feature and thus certain metrics responded poorly, but Figure 5.15 provides a clarification into how removing the mean improves results. It is clear from Table 5.6 that all methods appear to be performing adequately, with $HI^{(1)}$ as the seemingly optimal metric, however this again is a function of the record mean and its effect on the threshold. It is clear that the underlying physical and dynamical properties of this dataset are constrained in such a manner that a linear latent variable model such as *PCA* can easily detect and trend damage. The addition of model complexity does seem to improve the response of $HI^{(1)}$, however the *LHI*s used in this work add sufficient interpretability to the various models considered. The additional final condition deviance point aids in identifying the robustness of the metric, as it is now clear that a point can be identified from which all following records are damaged.

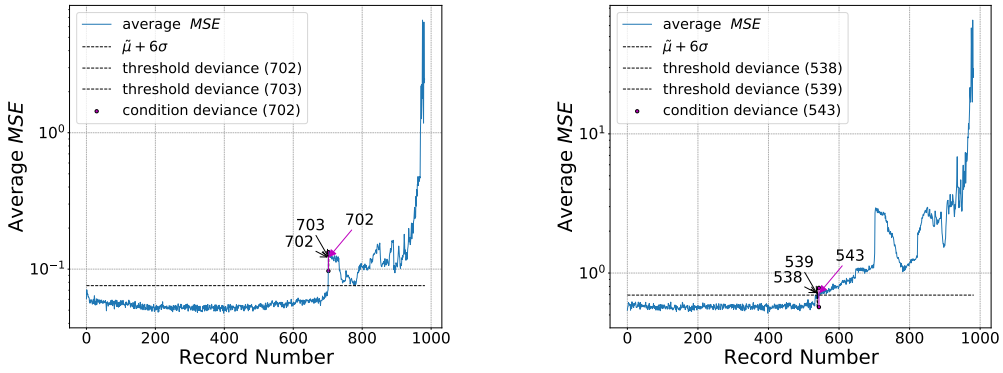
5.3.3 Dataset Two - Bearing One

For the second dataset, the first bearing was analysed as it was the bearing that was found to exhibit fault characteristics. In this investigation, the author chose to look at two window lengths to determine if any potential benefit or effect of using different model window lengths. As before, the *HI* and *LHI* results from the different models will be presented, where necessary, and the model performance will be discussed.

5.3.3.1 PCA Model Response

For *PCA*, the author demonstrate the model response for both window lengths for $HI^{(1)}$ and the three *LHI*s. Figure 5.18 shows the average of the reconstruction LL and for a window length of $L_W = 512$, the response to damage occurs later than that for a window length of $L_w = 4096$. Notably, in Figure 5.18(a) the response is less indicative around record five hundred. If one examines Figure 5.19, it is clear to see that the latent manifold response to damage for the two window lengths are similar and that the presence of damage is noticeable. All three *LHI*s appear to be responding to damage which is not only unexpected but also shows that this dataset's manifold is highly responsive. In the presence of an anomalous instance, the manifold not only deviates from the path in distance and radius to induce a change in velocity, but breaks the conservation of the path trajectory indicating a volatile path deviance phenomenon. This highlights the ability of the proposed latent metrics as they augment model intuition and introduce physical interpretation into how the model handles anomalous data without examining

the data in higher dimensional spaces. The self-healing phenomenon of the bearing can be noted here, whereby from record seven hundred a deviation back towards the healthy manifold is present until rapid failure occurs after record eight hundred. $LHI^{(3)}$ crosses the threshold in this region, indicating that the model tries to conserve the path trajectory but cannot perfectly re-obtain it.



(a) $L_w = 512$ **(b)** $L_w = 4096$

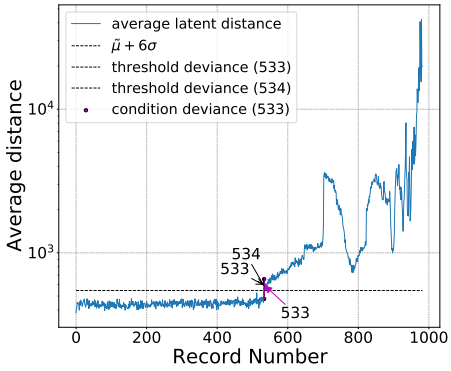
Figure 5.18. The reconstruction normalised log-likelihood, $HI^{(1)}$, obtained under a PCA model for two different window lengths. Notice the clear improvement in PCAs condition deviance point for the larger window length.

The author also found that the performance of the *VAE* models did not offer any significant improvement other than having a better response in $HI^{(1)}$ for a window length of $L_w = 512$. Interested readers can view Figures E.3 and E.4 if they wish, however these results are purely supplementary.

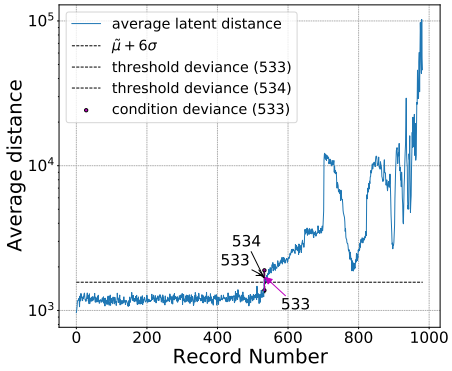
5.3.3.2 GAN-based Model Response

For the performance of the GAN-based models, the author will present a focus on $HI^{(2)}, HI^{(3)}$ and the three *LHI*s as these are the five key areas in which is it expected that these models provide increased performance. This does not imply that $HI^{(1)}$ is not useful but rather it is the less interesting *HI* for the GAN-based models as all other models considered in this work have access to it. For the majority of the results shown here, a visual interpretation for $L_w = 512$ was sufficient. However an interesting latent response was found for $RY - GAN$ for a window length of $L_w = 4096$ and as such will be discussed.

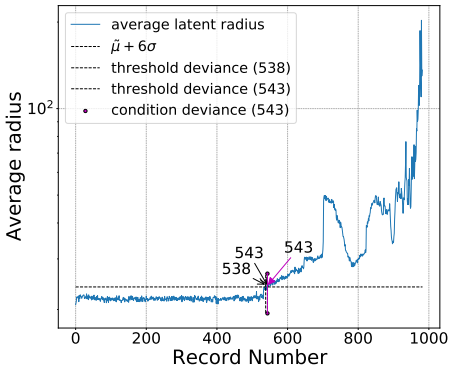
In the analysis of Figure 5.20, one can find the favourable responses obtained from both the data discriminator and the latent critic from both the $RY - GAN$ and the $DLS - GAN$ models. In Figure 5.20(a), the $RY - GAN$ model has learnt a better data discriminator, with a clear response to damage when compared to the $DLS - GAN$ data discriminator shown in Figure 5.20(b). It is interesting to note here that both data discriminators represent healthy data with a likelihood of approximately 0.8, an indication that there has been some unsatisfactory model representation of the true data distribution and as a result, the discriminator has a higher value assigned to the healthy data. This response was also noted in for the previously considered datasets. This can be attributed, again, to the L_2 objective function involved in the model optimisation, with the model failing to capture the signal noise floor as white noise is in-compressible and as a result, the discriminator attempts to enforce that this noise be learnt. This may explain the rapid manner in which the data discriminator moves from healthy to



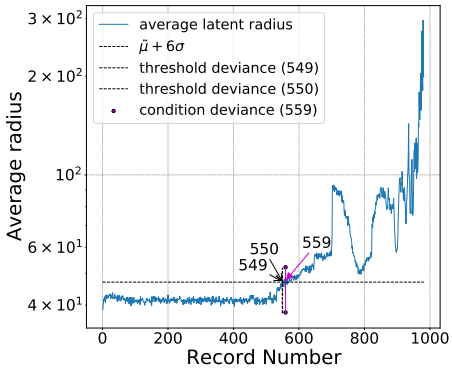
(a) $LHI^{(1)} : L_w = 512$



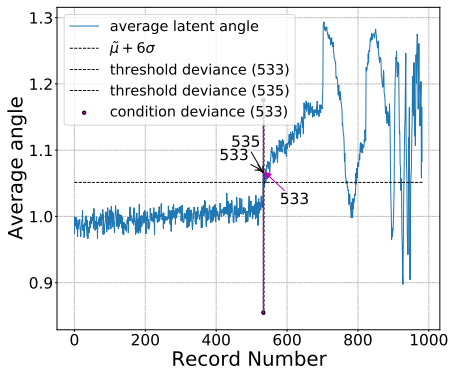
(b) $LHI^{(1)} : L_w = 4096$



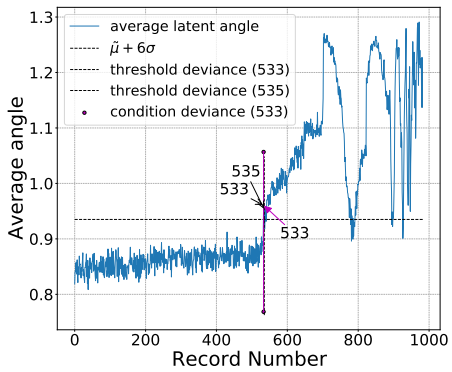
(c) $LHI^{(2)} : L_w = 512$



(d) $LHI^{(2)} : L_w = 4096$

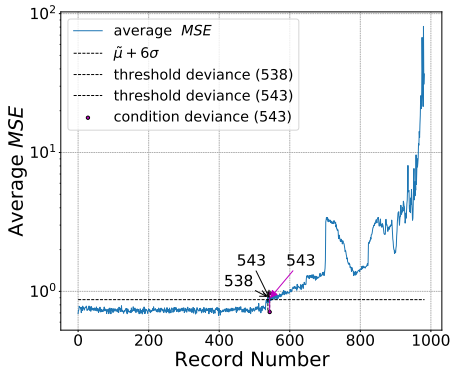


(e) $LHI^{(3)} : L_w = 512$

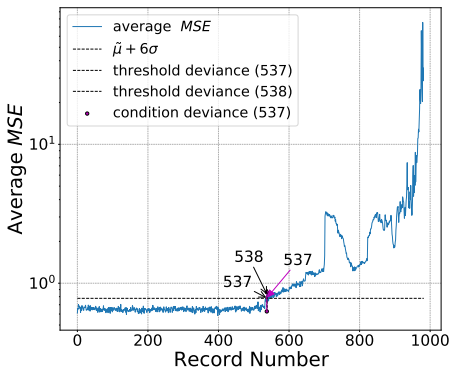


(f) $LHI^{(3)} : L_w = 4096$

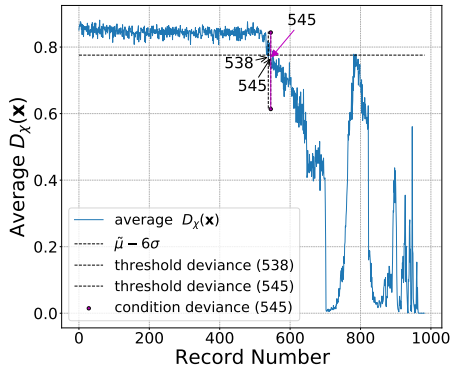
Figure 5.19. The three *LHI*s obtained using a PCA model for the two window lengths of interest for the first bearing from the second IMS dataset. Notice the strong response to damage that is identifiable through the latent manifold under the *temporal preservation* approach.



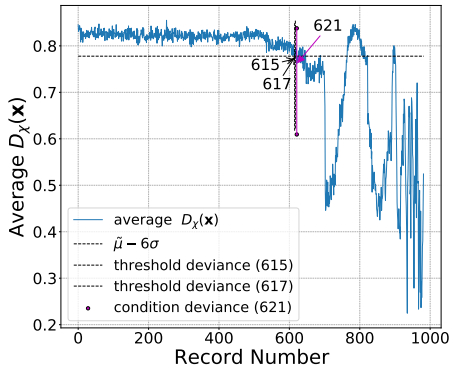
(a) $HI^{(1)}$: $RY-GAN$



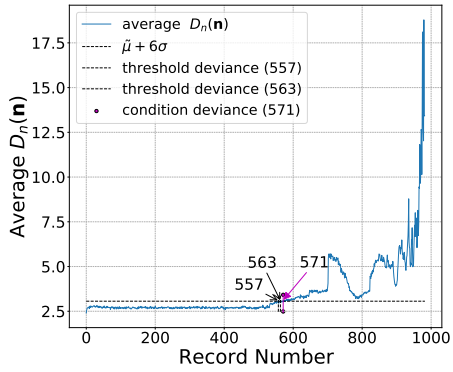
(b) $HI^{(1)}$: $DLS-GAN$



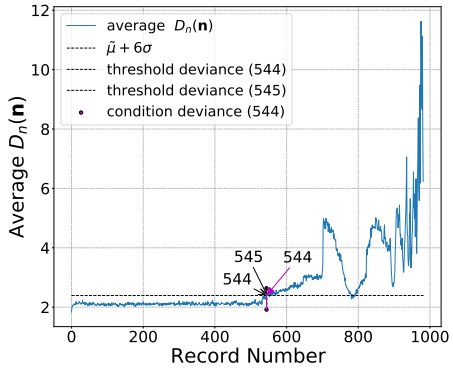
(c) $HI^{(2)}$: $RY-GAN$



(d) $HI^{(2)}$: $DLS-GAN$



(e) $HI^{(3)}$: $RY-GAN$



(f) $HI^{(3)}$: $DLS-GAN$

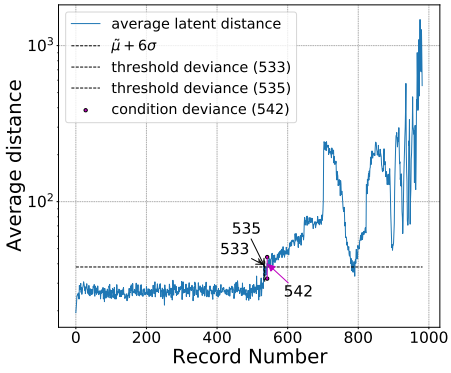
Figure 5.20. The reconstruction log-likelihood, data discriminator and latent critic results obtained from models trained on data from bearing one of the second IMS dataset for the two GAN-based methods considered in this work. Figures 5.20(a), (c) and (e) show the results for the $RY-GAN$ formulation while Figures 5.20(b), (d) and (f) show the results from the $DLS-GAN$ formulation. For these results the window length was kept at $L_w = 512$.

unhealthy. The author did try many model formulations to capture this noise but was unsuccessful. For the latent critic response, it is clear from Figure 5.20(c) and (d) that there is indeed a latent manifold response to damage and both critics provide satisfactory performance. The latent critic is a powerful addition as it is defined over the latent manifold, which, prior to the work of Baggeröhr (2019), was largely un-interpretable as no formulation had used the critic as a proxy metric on the latent space. It does appear that the latent critic identifies a condition indicator point that is worse than those noted for previous models, however this can be attributed to the slight change in the average of the first few records in Figure 5.20(c) and (d).

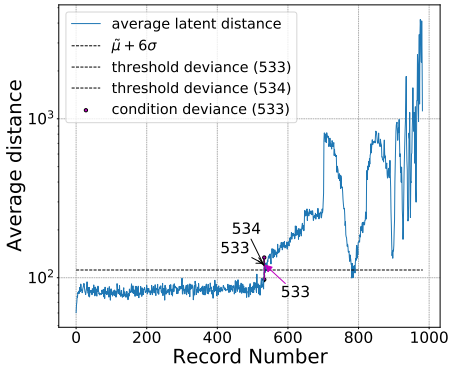
In Figure 5.21, it is clear to note that the latent space is responding to damage and it is interpretable, a result that is hinted at by the latent critic. It is clear that the performance is aligned with that obtained through *PCA* and the *VAE*s, which shows that the models are working. One interesting case is that, as was the case with the *VAE* results shown in Figure E.4 and by extension Figure 5.19 for *PCA*, the *LHI*s all respond in a similar manner. This can be attributed to the L_2 objective function, which is still included in the *GAN*-based formulations used in this work. The excellent performance of the *LHI*s on this dataset speaks to the applicability of the *temporal preservation* approach and the augmentation of information from the latent space obtained from latent-variable models. By performing simple tweaks to the data analysis process and exploiting the time element that it introduces, one can introduce model interpretation in the data space and the latent space.

An interesting result found by the author is the difference between the $RY-GAN$ models for the two different window lengths considered. For the larger window length, the latent manifold was found to respond with strong latent radius responses at the expense of the latent distance. This provides some intuition into how the window length affects the learnt latent manifold, with the preservation of the typical velocity over which the data travels through the latent space while the distance from the Euclidean centre is altered. To drive this point, the author used $T-SNE$ to reduce the latent N-space representation of the data into an interpretable *2-dimensional* equivalent, shown in Figure 5.22(a). It is clear to see here that as one moves through the samples, at some point there is a clear jump in the manifold to regions that are unlike those on which the healthy data lies. If one interprets the increase in radius at the expense of the latent distance, the latent angle must exhibit some response to preserve the latent distance and it manifests in a reduction of inter-instance angle shown in Figure 5.22(b). One can immediately see how the *LHI*s offer significant model interpretation and introduce model response conclusions to anomalous data.

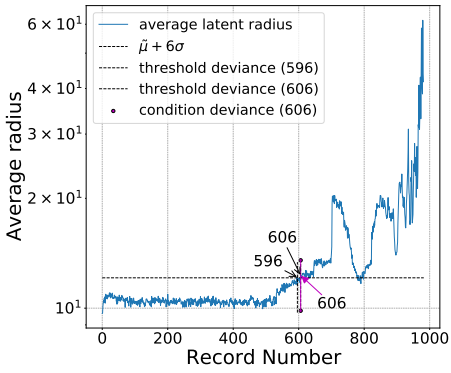
For the sake of concreteness, the author believes it is poignant to end the discussion with an analysis of how damage manifests in the **s** and **n** latent components as this portion of the model is not intuitive and well discussed in previous work. Additionally, the use of the $DLS-GAN$ was also introduced in this work to try and enforce latent component separation. In Figure 5.23, a *two-dimensional* $T-SNE$ visualisation of the **s** and **n** space for both the $RY-GAN$ and $DLS-GAN$ is given. It is clear that the $RY-GAN$ provides a less evident manifestation to damage in the **s** space as opposed to the **n** space, while only the significantly damaged instances at the end of the experimental life-span are mapped to the edges of the manifold. The $DLS-GAN$ response, however, indicates a stronger manifestation of damage than that found from the $RY-GAN$, with Figure 5.23(d) presented a more evident development at the outer edge of the manifold and it is akin to that seen in the **n** space shown in Figure 5.23. This indicates that albeit the $DLS-GAN$ method was used to avoid this very scenario, it seems to rather aid it. The $DLS-GAN$ was formulated to produce independent latent components with the hope that the **c** and **s** capture the relevant information and **n** capture the residual information. It is then expected that signals with anomalous components manifest their response in **n**. However, based on the placement



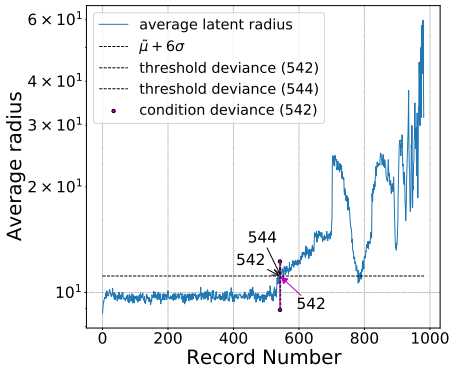
(a) $LHI^{(1)} : RY - GAN$



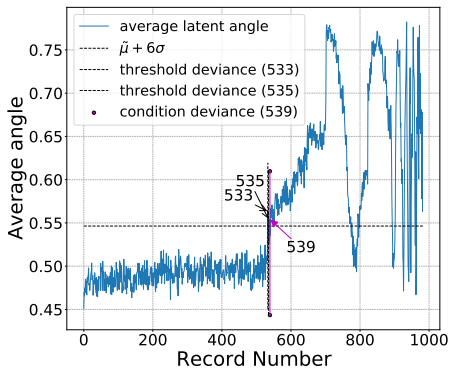
(b) $LHI^{(1)} : DLS - GAN$



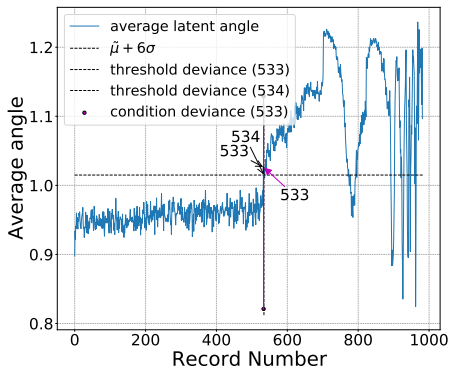
(c) $LHI^{(2)} : RY - GAN$



(d) $LHI^{(2)} : DLS - GAN$

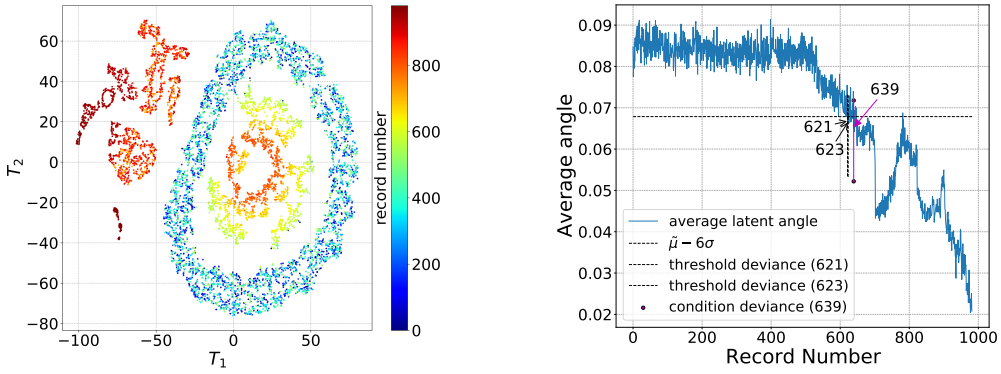


(e) $LHI^{(3)} : RY - GAN$



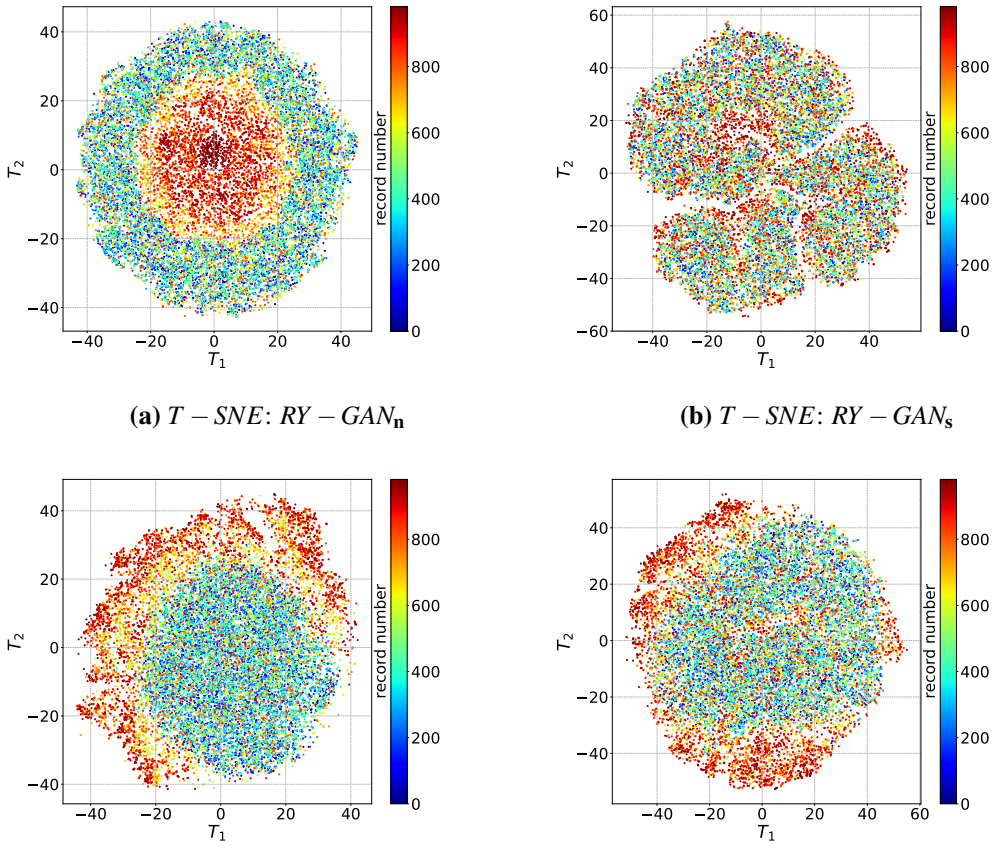
(f) $LHI^{(3)} : DLS - GAN$

Figure 5.21. The three *LHI* responses obtained using the $RY - GAN$ and $DLS - GAN$ methods for the first bearing of the second IMS dataset. Note that this visualisation was obtained for a window length of $L_w = 512$.



(a) $T-SNE$ Visualisation **(b)** $LHI^{(3)}$

Figure 5.22. A $T-SNE$ enabled latent **n** space and $LHI^{(3)}$ visualisation, Figures 5.22(a) and 5.22(b) respectively, for a $RY-GAN$ model trained on a window length of $L_w = 4096$ for data from the first bearing from IMS dataset two. Note that the colour-bar axis corresponds to record number.



(a) $T-SNE$: $RY-GAN_{\mathbf{n}}$ **(b)** $T-SNE$: $RY-GAN_{\mathbf{s}}$

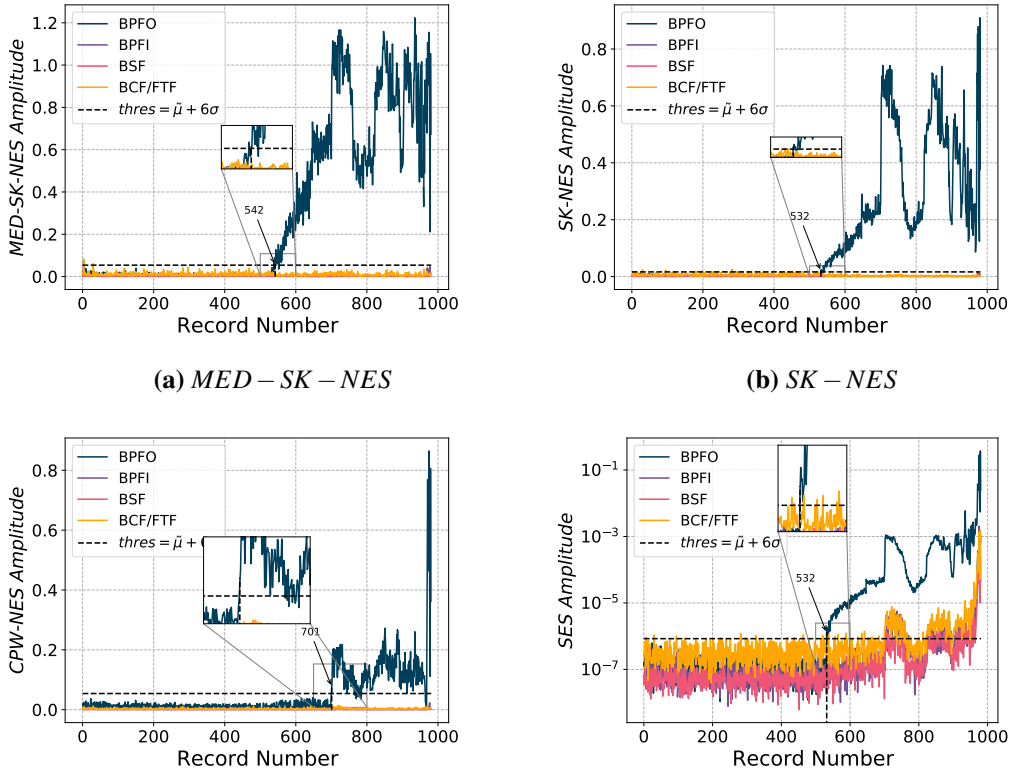
(c) $T-SNE$: $DLS-GAN_{\mathbf{n}}$ **(d)** $T-SNE$: $DLS-GAN_{\mathbf{s}}$

Figure 5.23. A $T-SNE$ visualisation of all the available data for the **n** and **s** manifold for the $RY-GAN$ and $DLS-GAN$ models trained with a window length of $L_w = 512$ on the first bearing data from the second IMS dataset. Note that the colour used corresponds to the record number.

of anomalous instances on the outer ranges of Figures 5.23 (c) and (d), it appears that damage is manifesting in the component that should only capture the generative factors of the data.

5.3.3.3 Signal Processing

For this dataset, it is important that clear comparability be shown for any deep learning approach through the identification of a condition deviance point using signal processing techniques. As before, four signal processing approaches will be used and the results obtained will be discussed. From inspection of Figure 5.24, three of the four methods perform extremely well with the exception of the $CPW-NES$ approach which only detects a condition deviance point in later records. It is interesting to note how the addition of MED almost seems to hinder the detectability, with a worsened condition deviance point identified in Figure 5.24(a) when compared to Figure 5.24(b). It is the author's inclination after spending some time with MED that the algorithm still needs some further development as MED was found to be problematic in optimisation of the kurtosis. Literature does detail methods to aid in this, however, it is clear that using $SK-NES$ is sufficient and better in fault diagnostic performance. The performance of the SES is also powerful as it is the simplest method used and highly efficient in implementation. However, the SES does induce some frequency discrepancy, with the other fault frequencies following a similar fault trend, albeit reduced in frequency magnitude. The condition deviance point at record five hundred and fifty two is in agreement with literature, highlighting the power of these signal processing techniques on constant operating condition bearing fault data.



(a) $MED-SK-NES$ **(b)** $SK-NES$

(c) $CPW-NES$ **(d)** SES

Figure 5.24. The four considered signal processing approaches frequency amplitude at the four frequencies of interest for the first channel of bearing four from IMS dataset two. Notice the clear instance of condition deviation in all cases.

5.3.3.4 Result Consolidation and Conclusion

To fully consolidate all results presented for IMS dataset two, Table 5.7 has been included for an analysis of the performance of the different *HI*s for the various models and window lengths considered. Some noticeable results will now be discussed and some conclusions will be drawn on the performance of the considered approaches. It is clear to note that the performance obtained from the different latent variable models is on par with the state-of-the-art signal processing techniques. This is an excellent result as it shows that latent variable models are comparable and that they allow for an excellent insight into fault detection applications, with many elements of the model in agreement into the presence of damage. Based on the results shown, it can be said that deep learning is better than signal processing in its ability to detect damage, with both the data space and latent space providing clear evidence of damage. There is also a strong indication that latent manifold analysis is applicable and that the manifold is responsive to faulty data instances that are detectable.

Table 5.7. The obtained threshold condition deviance point from the second IMS dataset for bearing one when investigating the available HIs and LHIs. Note that IC_1 is the abbreviation used for results deemed inconclusive by the author.

Model type and characteristics		**Health Indicator condition deviance point**					
Model used	**Window length**	**HI**$^{(1)}$	**HI**$^{(2)}$	**HI**$^{(3)}$	**LHI**$^{(1)}$	**LHI**$^{(2)}$	**LHI**$^{(3)}$
PCA	$L_w = 512$	702	N/A	N/A	**533**	**538**	**533**
	$L_w = 4096$	**543**	N/A	N/A	**533**	579	**533**
VAE_1	$L_w = 512$	**533**	N/A	N/A	578	648	**554**
	$L_w = 4096$	579	N/A	N/A	702	IC_1	622
VAE_2	$L_w = 512$	**545**	N/A	N/A	550	702	556
	$L_w = 4096$	647	N/A	N/A	586	IC_1	621
$\beta - TC - VAE_1$	$L_w = 512$	**533**	N/A	N/A	**549**	703	**539**
	$L_w = 4096$	**545**	N/A	N/A	933	IC_1	639
$\beta - TC - VAE_2$	$L_w = 512$	**543**	N/A	N/A	**542**	606	**539**
	$L_w = 4096$	589	N/A	N/A	702	975	639
$RY - GAN$	$L_w = 512$	**543**	**545**	571	**542**	606	**539**
	$L_w = 4096$	**537**	594	**543**	958	**548**	639
$DLS - GAN$	$L_w = 512$	**537**	621	**544**	**533**	**542**	**533**
	$L_w = 4096$	567	578	571	549	647	**542**

Through the analysis of Table 5.7, it is clear to see that most models respond similarly although differing in parametrisation and design. It is clear to see that $LHI^{(2)}$ is a poor condition indicator for *VAE* and $\beta - TC - VAE$ models trained with a larger window length. The performance of $HI^{(1)}$ is weakened when parametrising the *VAE* and $\beta - TC - VAE$ models with some learnt data variance. This is a surprising result as it was expected that the variance aid in explaining model condition deviance but this does appear to not be the case. With regards to the GAN-based methods, it is clear to see that all cases perform well through all condition indicators with the exception of $HI^{(2)}$ for the $DLS - GAN$ model with a window length of $L_w = 512$. This is attributed to the sub-optimal data discriminator training that is present due to the GAN loss and the L_2 loss. Table 5.7 indicates that the model window length does affect the latent manifold construction during training, with a change in the responsiveness of different *LHI*s as a function of window length. One may also note that *PCA* is a strong candidate

for fault detection on this bearing data, with great performance obtained through its available *HI*s. This is attributed to the simple operating and machine conditions present in this dataset. *PCA* is also a method that is not considered when talking about generative latent-variable models however it is clear that a constrained linear latent transform parametrisation is sufficient for damage detection on this dataset. This also introduces and highlights the fact that method complexity may not always be better, however there is still a place for explicitly constructing disentanglement in latent variable models as it may open doors into data interpretability and causality. The performance of the *LHI*s also indicate that the *temporal preservation* approach offers insights into model performance that has been previously neglected in vibration-based condition monitoring research. These insights best seen in the latent manifold response of latent variable models and the are beneficial as typically the latent manifold is neglected but it is clear that it holds important and interesting information.

5.3.4 IMS Consolidation

The IMS dataset has allowed for an in-depth analysis of different models and their respective performance results. It was found that *PCA* was sufficient for representing the data from the considered dataset in a latent manifold, with condition deviance points that are often competitive with the other models used and with the signal processing results. It is clear that the models themselves may offer some interpretation through the three *HI*s, however the latent manifold is evidently responsive and can be interpreted. The next frontier that must be explored is the domain of time-varying operating conditions as this is a relevant and important region of vibration-based condition monitoring.

Chapter 6 Gearbox Dataset Analysis

6.1 Chapter Abstract

In this chapter, the author presents the gearbox dataset and the performance investigation of the different models considered in this work. There are three key concepts that are key to this investigation:

1. The latent manifold is interpretable under the *temporal preservation* approach
2. Model complexity and the progression thereof needs to be highlighted and understood for applicability
3. Model performance must be compared to fully highlight the benefits of complex methods

The reader is asked to keep these concepts in mind when going through the various results, as each dataset offers insights into each of these points. For a detailed collection of the model architectures, learning rates, stopping conditions and hyper-parameters please refer to Appendix B.5. The models used on this dataset are: PCA, the VAE_1 and VAE_2 models, the $\beta - TC - VAE_1$ and $\beta - TC - VAE_2$ models, the $RY - GAN$ model and the $DLS - GAN$ model.

6.2 Dataset Introduction

The C-AIM experimental gearbox dataset contains healthy and unhealthy vibration data for a gearbox set-up that contains a single gear tooth fault. This dataset has been extensively analysed in the works of Schmidt et al. (2018) and Schmidt and Heyns (2020), and contains time-varying operating conditions.

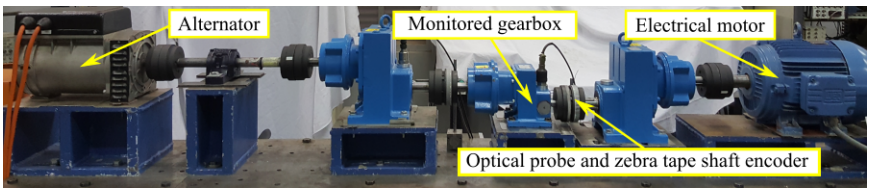


Figure 6.1. The C-AIM test gearbox experimental setup.

The experimental set-up, as depicted in Figure 6.1, consists of an electrical motor, three helical gearboxes of which one is a step-down and two are step-up gearboxes and an alternator to dissipate the system's rotational energy. The centre gearbox in Figure 6.1 is the one that was used for testing and as such, a tri-axial accelerometer was mounted to the bearing casing of the back of the test gearbox with

a proximity probe and an optical probe on the output and input shafts of the test gearbox respectively. The experimental data were measured using an Oros OR35 data acquisition system. Table 6.1 contains important properties of the data obtained in this dataset.

Table 6.1. A table showing the experimental gearbox dataset properties for the test gearbox, accelerometer and tachometer.

Characteristic	Value
Gear teeth	37
Pinion teeth	20
Gear Ratio	1.85
Accelerometer sampling frequency	$25.6kHz$
Tachometer sampling frequency	$51.2kHz$
Tachometer pulse [input || output]	88 || 1 pulses per revolution
Sampling duration	20 *seconds*

Schmidt et al. (2018) state that the system operating conditions were chosen based on a simplification of operating conditions seen in a gearbox from a bucket-wheel excavator and wind turbine application. Figure 6.2 show the speed profile obtained for a randomly selected healthy signal for the input and output shafts respectively. Due to geometrical imperfections at the butt joint of the shaft encoder seen by the optical probe, the Bayesian Geometry Compensation (BGS) algorithm proposed by Diamond et al. (2016) was used.

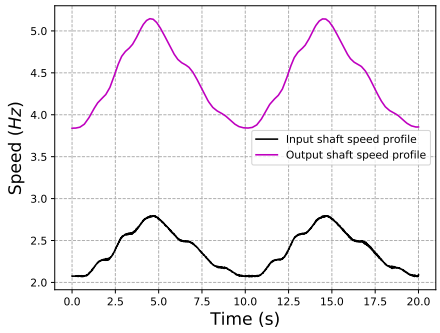
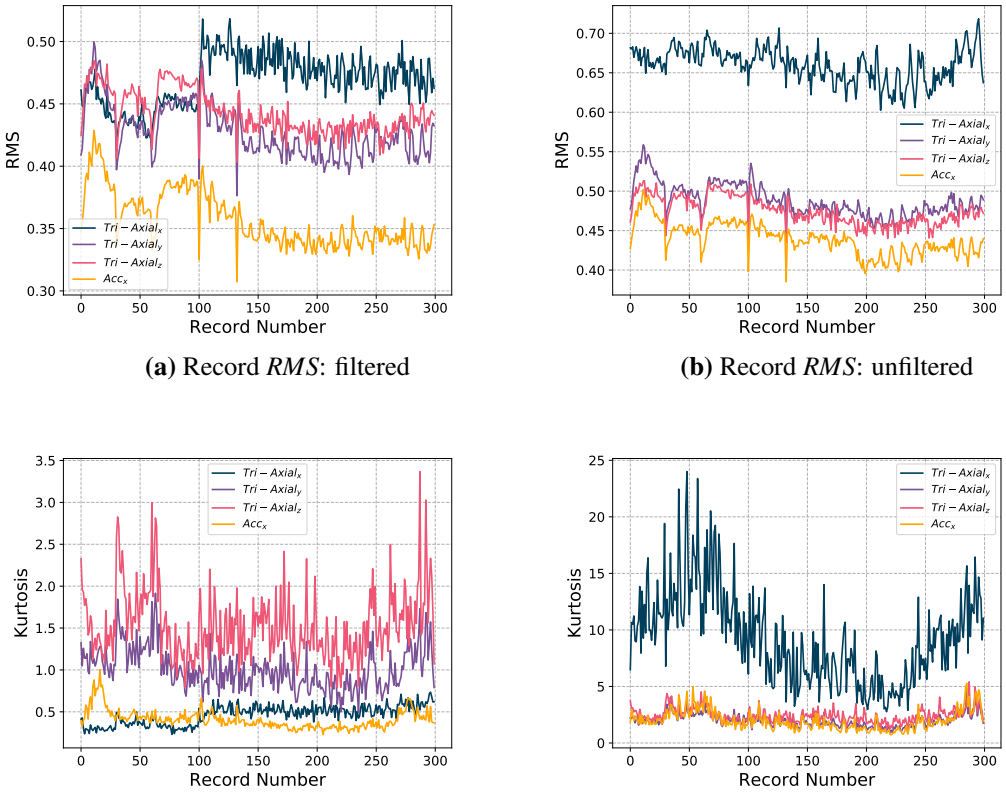


Figure 6.2. The input and output test gearbox speed profiles. Note that BGS was used for the input shaft speed while the output speed was obtained from the once per revolution tachometer and thus required no use for BGS (Diamond et al., 2016).

In this dataset, a total of one hundred healthy signals were obtained and were recorded in a relatively short space of time, within approximately two days from the first record to the last. The test gearbox was then disassembled and a slot was seeded into the root of a single gear tooth, with the slot along the entire width of the tooth, 50% of the tooth thickness deep and at a height of $0.3mm$. The gearbox was then re-assembled and left to run for approximately twenty days until gear tooth failure occurred. From this, two hundred unhealthy signals were recorded throughout the tooth failure life-cycle. As a basic analysis, the author chose to calculate the signal *RMS* and kurtosis, as was done for the IMS dataset, to

illustrate the difficulty one may face when analysing this dataset. This is shown in Figure 6.3 and the main component of interest is the axial component of the tri-axial accelerometer, $tri-axial_x$. A third order low-pass Butterworth filter was used to filter the data at $3200Hz$ as there are impulsive signal components that complicate the analysis procedure. Notice the clear fluctuations in the *RMS*, attributed to changes in the surrounding environment temperature. Figure 6.3(d) shows the clear presence of an impulsive component through the Kurtosis, while for the filtered case in Figure 6.3(c) this component is less noticeable.



(a) Record *RMS*: filtered

(b) Record *RMS*: unfiltered

(c) Record kurtosis: filtered

(d) Record kurtosis: unfiltered

Figure 6.3. The record *RMS* and kurtosis for the low-pass filtered and unfiltered versions of the gearbox dataset. The low-pass filtering was conducted at a cut-off frequency of $3200Hz$ with a third order Butterworth filter.

6.3 Dataset Result Analysis

For the analysis of the gearbox dataset, an initial assumption was made to investigate a filtered and unfiltered version of the data, as the accelerometer signals from the test gearbox had impulsive components in both the healthy and unhealthy data. This would make it possible to compare the performance of the methods without impulsive components in the training data and with impulsive components in the training data. In this investigation, the author chose to train the data using the common pre-processing strategy detailed in Section 3.3, whereby no complex pre-processing was done and the models were trained only on the raw data. However, it was found that the discrepancy signal average result interpretability was biased for the gear tooth fault present, which is a infrequent fault as it only occurs once per shaft revolution. This differs to the previously analysed datasets as bearing

faults typically occur in proportions greater than one to the shaft speed. The decision was made to look into the synchronous average of the *time-continuous* results for the health and latent health metrics using Equation (1.10).

The TSA processing step typically consists of first obtaining the points where one revolution starts and ends, for which a tachometer is required. It is key to note here that the author did not use the tachometer during model training, but only in model evaluation which resulted in the order tracked discrepancy signal. A linear interpolation method was used to re-sample the signals in a revolution into N_s points and the author chose to align each synchronous signal with the start of a revolution fixed to the point where the tachometer crossed the butt-joint on the zebra tape shaft encoder (ZTSE). This was done to aid in synchronous average interpretability and visualisation and was achieved using the Bayesian Geometry Compensation algorithm developed by Diamond et al. (2016). The author will demonstrate how result analysis can be improved with the synchronous average, with an initial example for each of the filtered datasets. This type of analysis is deemed *signal processing assisted* deep learning, a hybrid between the two research spheres. It can be seen as applied deep learning that hinges of powerful techniques from signal processing. For the discrepancy signal average, a threshold defined on three standard deviations from the median will be analysed, denoted as $threshold = \tilde{\mu} \pm 3\sigma$, with sign dependent on the type of health indicator. This threshold was chosen as three standard deviations was deemed to be a sufficient indicator of a strong deviation from the typical response value. To further enhance the condition deviance analysis, a point will be identified, if possible, based on the mean of five points ahead from a point that crosses the threshold. The purpose here is to ensure that once-off anomalies are not flagged as condition deviance points.

It is important for the reader to note here that the gearbox dataset consists of two separate experimental datasets that are combined to produce a healthy and unhealthy dataset. This change occurs at record 100, as the case might exist where a *HI* or *LHI* detects a change between these datasets but is unable to perform fault trending or detect fault degradation.

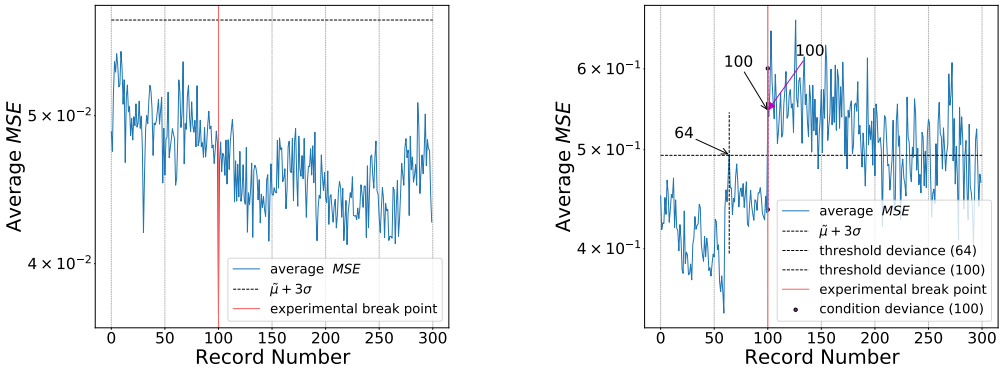
6.3.1 Filtered Gearbox Dataset

For the filtered version of the test gearbox dataset, an analysis will be conducted into how models perform with and without the integration of the *temporal preservation* analysis approach combined with the use of the synchronous average. For the former, all one has available for scrutiny is the discrepancy signal average and one must try perform fault inference on this metric. However, for the latter, one can look into both the average and the synchronous average as the time component of the dataset is preserved. For the *temporal preservation* analysis, only the first ten seconds of each signal was processed, due to evaluation time constraints, memory limitations and the fact that one operating condition cycle occurs at a frequency of $0.1Hz$.To filter out the impulsive components in the training data, a third order Butterworth filter was used at a cut-off frequency of $3200Hz$.

6.3.1.1 *PCA* Model Analysis

The author will present the reconstruction health indicator as well as the latent health indicators for two applications of *PCA*, one that uses all of the modes that capture 95% of the variance and one application that drops the first five Principal Components (PCs), to investigate the effect of dropping data information that contribute to the main sources of data variance. This will poorly favour signal reconstruction but may offer an insight into how the latent manifold responds to dominant signal information restriction. Figure 6.4 shows the discrepancy signal average of $HI^{(1)}$ discrepancy signal using the standard deep learning analysis procedure. It is directly noticeable that the *HI* obtained from a *PCA* model is a poor method under the discrepancy signal average, however, it is argued here that the average is an insufficient metric for this dataset due to the type of fault exhibited and the data

impulses. Figure 6.4(a) shows how using all of the PCs seem to be a poor choice as the model cannot detect the jump around record one hundred, which is where the tooth fault was seeded. This is not the case however for when one drops the first five principal components, which is interesting as this may indicate that the variance captured in the first PCs captures information that relates more to the operational state of the dataset than to the development of damage.



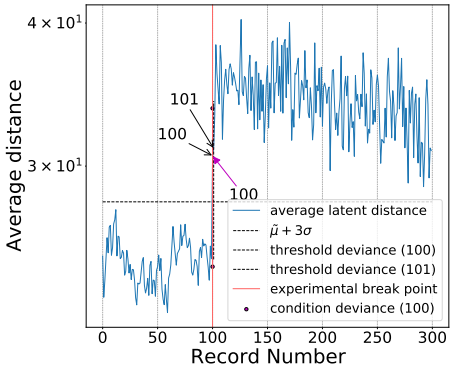
(a) All PCs. **(b)** Dropped five PCs.

Figure 6.4. The *PCA* $HI^{(1)}$ discrepancy signal average for all retained PCs, 6.4(a), versus dropping the firsts five PCs, 6.4(b) for the filtered gearbox data.

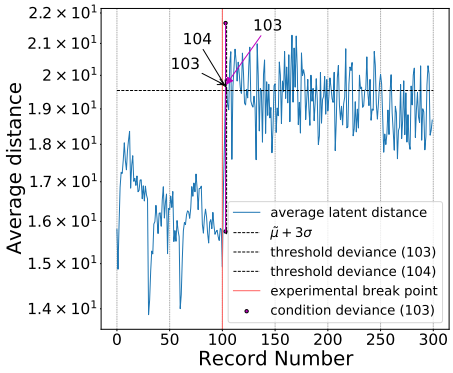
The latent metrics share the same result response, as shown in Figure 6.5. In Figure 6.5 a sudden change at record one hundred is detectable in the latent distance and radius, with a jump and then gradual progressive growth in the latent angle. There were some temperature variations in the surrounding environment of the gearbox test-rig, this resulted in changes to be observed in the vibration response of the system, which is ultimately also reflected in the first one hundred healthy records of $LHI^{(1)}$ and $LHI^{(2)}$, indicating model sensitivity to this parameter. For the model where five PCs were dropped, $LHI^{(3)}$ has a decreased diagnostic performance as it now does not detect the jump at record one hundred. This shows that one must take careful consideration into the preservation of PCs as for previous datasets the response was seen to be sufficient without dropping any PCs.

To investigate result interpretability, the synchronous average of the *HI*s and *LHI*s was obtained to clarify if the poor performance was just linked to result interpretability. Figure 6.6(a) shows the $HI^{(1)}$ synchronous average, whereby it is clear that there is some increase in magnitude around a shaft angle of approximately 140° from the ZTSE joint. The same response is not present in Figure 6.6(b) which is a direct result of dropping PCs as these PCs capture the dominant variance required for signal reconstruction. This does, however, demonstrate the potential of improved model interpretability under the *temporal preservation* approach, as the synchronous average process is not feasible under the standard model analysis approach.

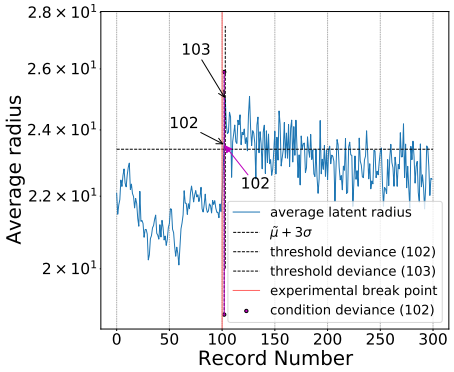
To further investigate the performance of *PCA*, the *LHI*s were analysed using the synchronous average, as shown in Figure 6.7. For the latent distance, $LHI^{(1)}$, is not interpretable for all PCs but its interpretability is improved when the first five PCs are dropped, as seen in Figures 6.7(a) and (b) respectively. For the latent radius, the improvements of dropping some PCs can be noted from the comparison of 6.7(c) and (d), with $LHI^{(2)}$ exhibiting a clear presence of damage and an improved



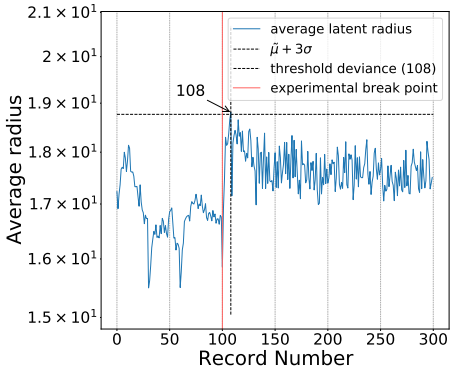
(a) $LHI^{(1)}$: all PCs.



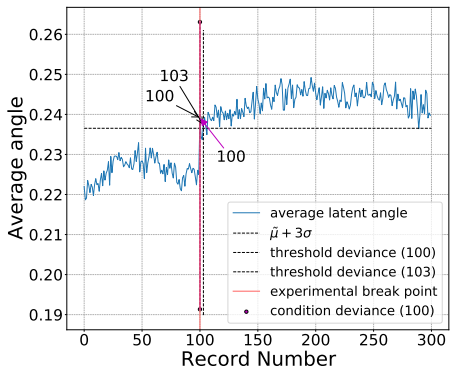
(b) $LHI^{(1)}$: dropped five PCs.



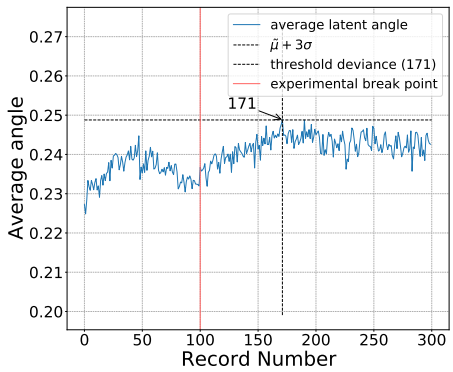
(c) $LHI^{(2)}$: all PCs.



(d) $LHI^{(2)}$: dropped five PCs.

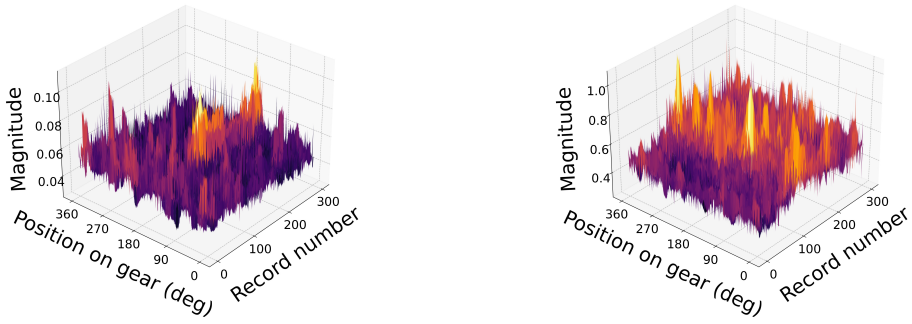


(e) $LHI^{(3)}$: all PCs.



(f) $LHI^{(3)}$: dropped five PCs.

Figure 6.5. The discrepancy signal average of the three *LHI* metrics using PCA with the *temporal preservation* analysis approach. 6.5(a), (c) and (e) refer to the case where all PCs are used while 6.5(b), (d) and (f) are for the case where 5 principal components are dropped.



(a) $HI^{(1)}$: all PCs.

(b) $HI^{(1)}$: dropped five PCs.

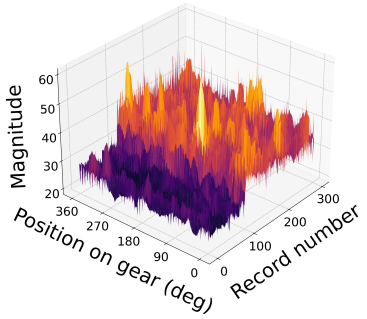
Figure 6.6. The $HI^{(1)}$ synchronous average using PCA with the *temporal preservation* analysis approach. Notice how dropping some PCs negatively favours the reconstruction ability of PCA.

synchronous average floor around the damaged tooth. In Figure 6.7(e) and (f), the latent angle can be seen to decrease in magnitude around the damaged tooth. Here again, the removal of the first five PCs appears to improve result consistency. One can interpret from the $LHI^{(2)}$ and $LHI^{(3)}$ response shown in Figure 6.7 is how these two components together provide some insight into how the latent space is responding to unhealthy data. When segments of the vibration signal that contain damage pass through the model, the latent representation casts the points off the manifold but tries to preserve the velocity of the traversal, as opposed to an orthogonal projection on the manifold to maintain the underlying trajectory.

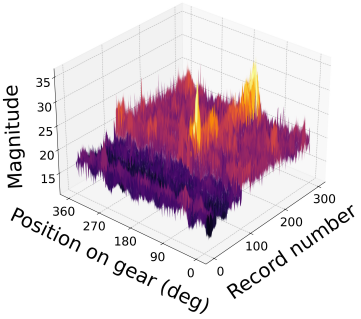
6.3.1.2 *VAE* Model Analysis

Having shown that the synchronous average improves result interpretability, the author preferred a result analysis of the synchronous average in the case of *VAE*s. However, the reconstruction average under the standard analysis approach shall be presented to provide some perspective of how assuming one learns a deterministic or stochastic *VAE* can affect results. In Figure 6.8, the reconstruction *HI* discrepancy signal average was analysed and it is clear to see that notable results were obtained. In Figure 6.8(a), a clear condition deviance threshold can be defined, however, due to the noisy average response from the healthy training data this threshold is undesirably late for damage detection. Figure 6.8(b) details the VAE_2 response, with an improved condition deviance threshold and some clear progressive damage growth occurring after record one hundred. It is also clear that the temperature effect is still clearly noticeable in the average of the healthy data, however the VAE_2 response is better able to capture and quantify these effects. It is also clear that after approximately record two hundred and fifty, there is a drop and then rapid increase through the final records and this is present in both models.

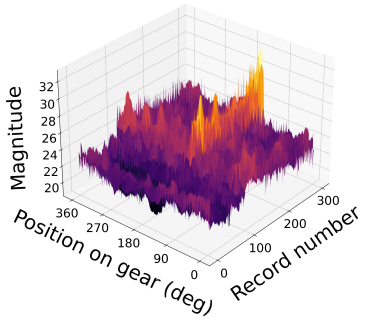
For the latent metrics of the *VAE* models, using the standard model evaluation procedure, it was found that metrics were highly uninformative to damage other than showing clear deviance around record one hundred. The latent distance and radius do not offer much insight while there does appear to be some progressive latent angle growth throughout the gearbox lifespan. However, as emphasised with *PCA*, the synchronous average will be examined and these results are shown in Figure 6.9 and Figure 6.10 for $HI^{(1)}$ and $LHI^{(1,2,3)}$ respectively.



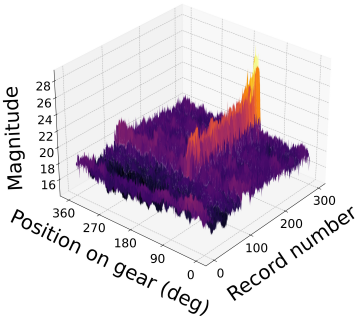
(a) $LHI^{(1)}$: all PCs.



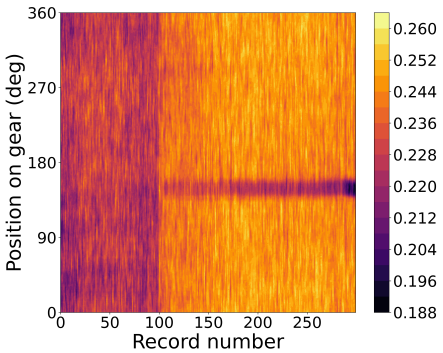
(b) $LHI^{(1)}$: dropped five PCs.



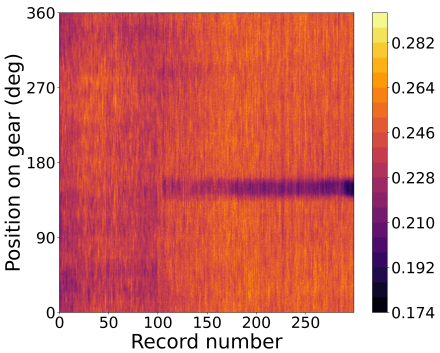
(c) $LHI^{(2)}$: all PCs.



(d) $LHI^{(2)}$: dropped five PCs.

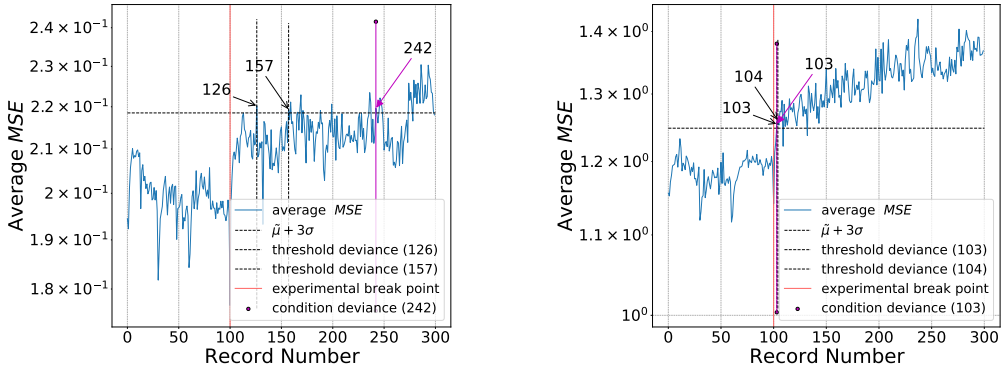


(e) $LHI^{(3)}$: all PCs.



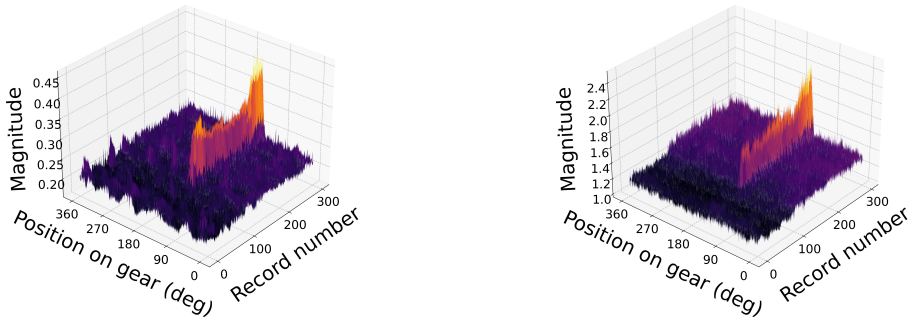
(f) $LHI^{(3)}$: dropped five PCs.

Figure 6.7. The three *LHI* metric under the synchronous average using PCA with the *temporal preservation* analysis approach. 6.7(a), (c) and (e) refer to the case where all PCs are used while 6.7(b), (d) and (f) are for the case where 5 principal components are dropped.



(a) $HI^{(1)}$: VAE_1. **(b)** $HI^{(1)}$: VAE_2.

Figure 6.8. The $HI^{(1)}$ discrepancy signal average for VAE_1 and VAE_2 models for the filtered gearbox dataset.



(a) $HI^{(1)}$: VAE_1. **(b)** $HI^{(1)}$: VAE_2.

Figure 6.9. The $HI^{(1)}$ discrepancy signal synchronous average for VAE_1 and VAE_2 models for the filtered gearbox dataset.

In Figure 6.9 it is clear that there are now clear improvements from *PCA* whereby the synchronous average of $HI^{(1)}$ is smoother and shows the presence of the gear tooth fault after the fault was seeded into the system. It is also clear that VAE_2 has some improvements over VAE_1 whereby the learnt variance introduces a drastic increase in fault magnitude, which may explain the damage growth in Figure 6.8(b), as the presence of damage is amplified in the *HI* due to the learnt variance. If we now look at the *LHI*s, as shown in Figure 6.10, it is clear that the latent distance, $LHI^{(1)}$, is a poor metric to use for damage detection and this is extended through both VAE_1 and VAE_2. However, the latent radius and distance show a clear presence of damage and allow for the same interpretability of manifold traversal as was given to *PCA*. It is important to note how *PCA* and the *VAE*s latent responses are in agreement in where the fault occurs and with how the damage manifests in the latent space. Due to similarities in model formulation, with a *VAE* is simply an unconstrained, non-linear version of *PCA*, it is expected that the latent response be similar in manifestation. One interesting note is that both VAE_1 and VAE_2 do not show the presence of damage in $LHI^{(1)}$ clearly whereas, as shown in Figure 6.7(b), *PCA* with some dropped PCs shows some presence of damage. Dropping latent

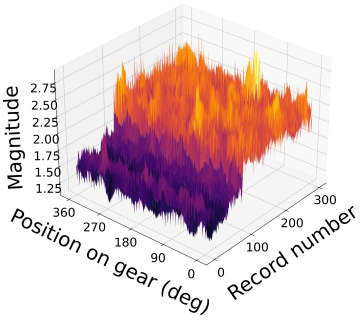
components in a *VAE* is a non-trivial task as there is no requirement for any latent hierarchy whereas *PCA* organises its PCs by variance captured. Additionally, latent disentanglement is implicit and thus the identification of generative factors or dominant components that one can adjust or suppress is non-trivial. It is not completely unjustifiable that if $LHI^{(3)}$ drops and $LHI^{(2)}$ increases one may lose information in $LHI^{(1)}$ as the model is favouring rapid jumps off the manifold as opposed to gradual shifts off the manifold. This strongly highlights the strength of the three proposed *LHI*s as they cover a large domain of manifold operations in an interpretable manner. It was also noted by the author that the $\beta - TC - VAE$ model responses were equivalent to that found from the *VAE*s regardless of parametrisation and offered little improvement.

6.3.1.3 *GAN* Model Analysis

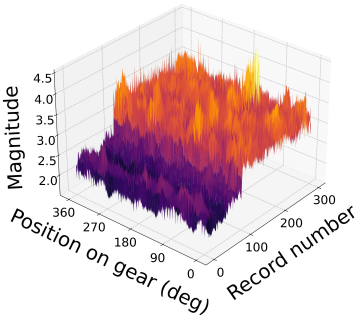
For the GAN-based model analysis, the author will present all three *HIs* and *LHIs* available to the $RY - GAN$ and the $DLS - GAN$ models using the discrepancy signal average to continue to motivate the use of the synchronous average. In Figure 6.11, the average of the $HI^{(1)}$ discrepancy signal is shown and this response is akin to those shown from the VAE_1 and $\beta - TC - VAE_1$ results previously. This is not unexpected as $HI^{(1)}$ comes from the auto-encoder framework but it is clear that the average of the training instances causes the threshold to be higher than the mean of the other records. The large threshold is due to the large variation in the healthy data $HI^{(1)}$ response which indicates that the behaviour is non-stationary and the standard deviation will capture some of those non-stationary effects. Both methods, however, do show a discrete jump at record one hundred and some drop and then rise after record two hundred and fifty. It is also clear that there are indeed strong temperature effect responses in the discrepancy signal average of the healthy data, with a growing and then dropping effect clearly evident.

If we now examine the $D_\chi(\mathbf{x})$ discrepancy signal average shown in Figure 6.12, it is interesting to note that the data discriminator appears to be responding to the data. However, it is not significant enough to say with certainty that a fault is present. The effects of the healthy data temperature variations are evident, with the condition deviance point at record eighty-one. This is not a fault point but rather the model response to the set-up temperature variation, which is interesting as it is a deviance from the learnt operating condition state of the set-up. The $DLS - GAN$ result also appears to be slightly worse, a frequent occurrence found through all attempted training runs. One improvement to the data discriminator is that the healthy data discrepancy signal mean for the $RY - GAN$ is around 0.5, which is the known stability point for *GAN* discriminator training. This is attributed to the low-pass filtering operation applied to the data and the fact that the decoder network does not have to try and capture the incompressible noise typically found in the vibration signals. This is a flaw that the author believes is holding the current GAN-based models back as the L_2 objective function and the discriminator objective function are competitive in the type of generative distribution $p(\mathbf{x}|\mathbf{z})$ they approximate. The L_2 loss drives one to capture the strongest elements of data variance and thus typically does not capture noise while the data discriminator is driven to force the model to capture this noise, albeit random and incompressible. For the previous datasets analysed, one cannot filter the data as the bearing faults manifest in a high-frequency resonance band and this trade-off may negatively affect the performance of the data discriminator and the model as the L_2 loss will dominate.

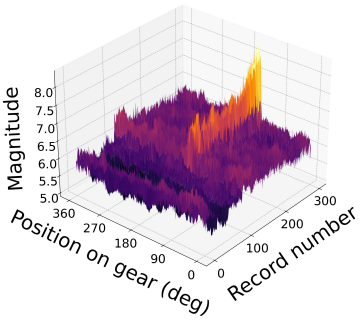
For the latent critic discrepancy signal average, as shown in Figure 6.13, it is clear that the latent space is responding to damage but the results are still somewhat un-interpretable. Both the $RY - GAN$ and the $DLS - GAN$ methods respond to record one hundred but they do not offer any other insights. The discrepancy measure does not exhibit any form of growth to the unhealthy data and appears to be quite flat. Again, a clear response to the temperature variation is present as there is some notable growth



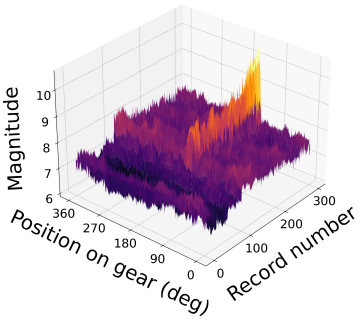
(a) $LHI^{(1)}$: VAE_1.



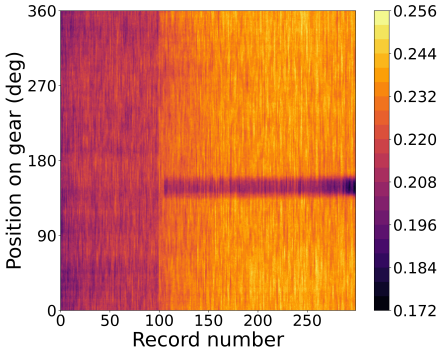
(b) $LHI^{(1)}$: VAE_2.



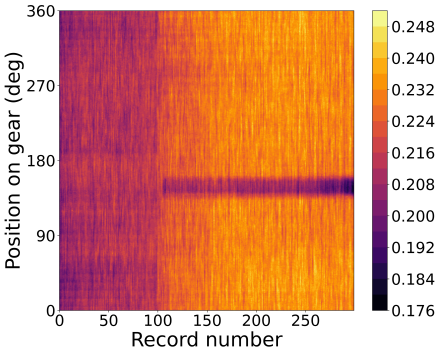
(c) $LHI^{(2)}$: VAE_1.



(d) $LHI^{(2)}$: VAE_2.

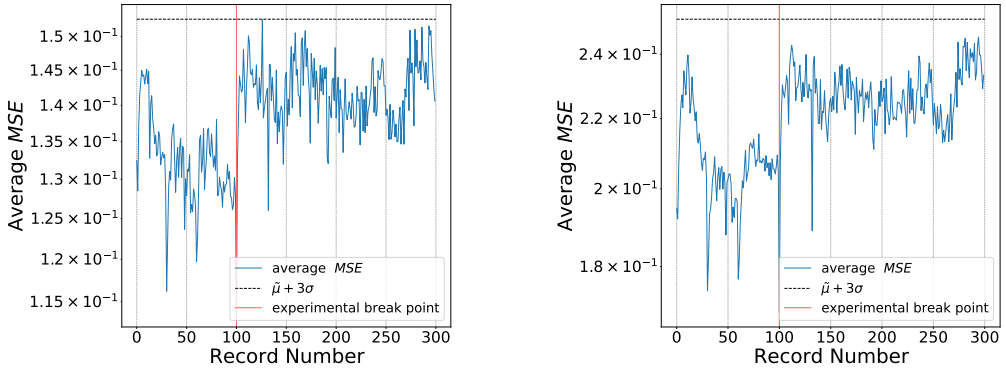


(e) $LHI^{(3)}$: VAE_1.



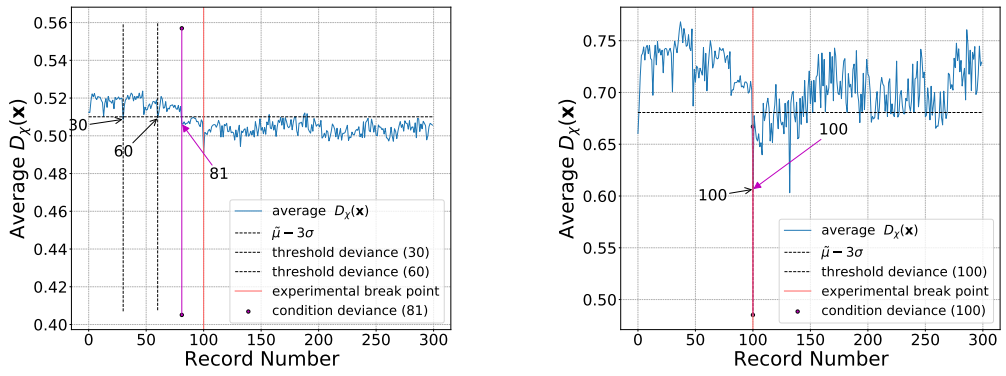
(f) $LHI^{(3)}$: VAE_2.

Figure 6.10. The *LHI* synchronous average responses from the VAE_1 and VAE_2 models for the filtered dataset.



(a) $HI^{(1)}$: $RY-GAN$. **(b)** $HI^{(1)}$: $DLS-GAN$.

Figure 6.11. $HI^{(1)}$ discrepancy signal average results using the $RY-GAN$ and $DLS-GAN$ methods from the filtered gearbox dataset under the standard processing approach.

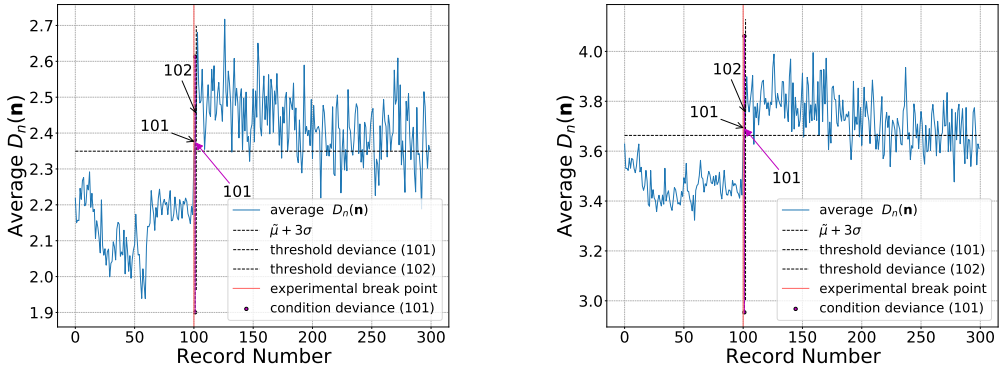


(a) $HI^{(2)}$: $RY-GAN$. **(b)** $HI^{(2)}$: $DLS-GAN$.

Figure 6.12. $HI^{(2)}$ discrepancy signal average results using the $RY-GAN$ and $DLS-GAN$ models from the filtered gearbox dataset under the standard processing approach. Figure 6.12(a) shows the $RY-GAN$ case and 6.12(b) shows $DLS-GAN$ case.

after record 50 for both methods.

Figure 6.14 shows the three HI synchronous average results for the filtered gearbox data. It is immediately clear that the reconstruction HI appears to be the most dominant out of the three HIs. One troubling result is that the data discriminator response, $HI^{(2)}$ in Figure 6.14(c) and (d), shows positive deviances to damage, which is unlike the known response one should obtain when using a data discriminator. One would expect that the response should tend to zero as the data discriminator represents $p(healthy|\mathbf{x})$. The signal impulsivity present in the training data is responsible for this, as it was not filtered out completely and thus the data discriminator has been trained poorly, to the extent where it recognises the gear tooth fault as healthy data. For the latent critic, it is interesting to note that there is a sign of fault deviance, with an anomaly around 140° from the $ZTSE$ butt joint. The latent critic is responding to damage and the synchronous average is responsible for uncovering this. The

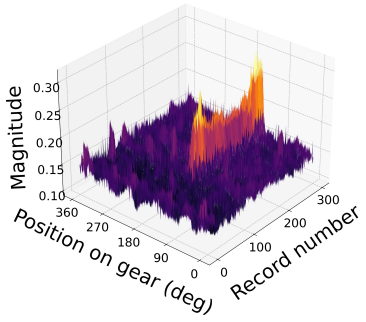


(a) $HI^{(3)}$: $RY-GAN$. **(b)** $HI^{(3)}$: $DLS-GAN$.

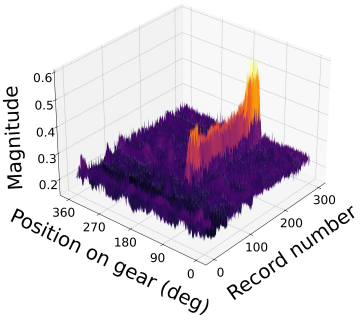
Figure 6.13. $HI^{(3)}$ discrepancy signal average results using the $RY-GAN$ and the $DLS-GAN$ methods from the filtered gearbox dataset under the standard processing approach.

latent critic response is not as clear as that shown for $HI^{(1)}$, which is attributed to the model inability to capture the impulsivity and the resulting manifestation thereof in the latent space. This may also indicate why $LHI^{(1)}$ is a poor metric and shows that the latent critic is considering the potential effects noted in the LHIs together.

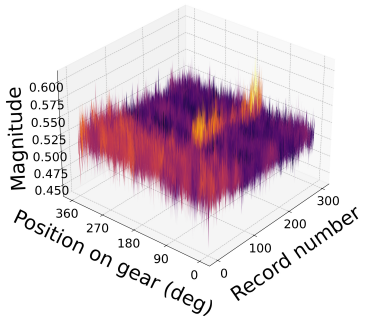
Figure 6.15 shows the response for the three LHIs using the synchronous average. It is clear that the latent distance is a poor metric on this dataset and that the latent radius and angle are strong candidates for fault detection. It is also clear to note that this response is akin the that of the VAEs and PCA, which is not a surprise given that the L_2 loss features in this model and that the latent space is constrained to be a factored Gaussian. Interestingly, the latent manifold is responding well while the latent critic, shown in Figure 6.14(a) and (b), responds less clearly. This indicates that the latent critic may be more receptive to latent distance deviances as a result of the impulses and that the deviance in radius and angle is insufficient to help identify the presence of the fault.



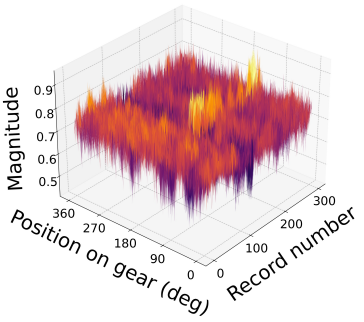
(a) $HI^{(1)}$: $RY - GAN$.



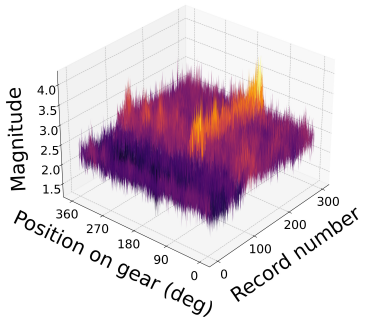
(b) $HI^{(1)}$: $DLS - GAN$.



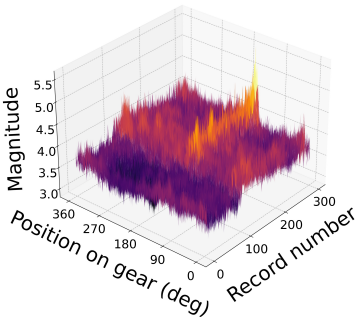
(c) $HI^{(2)}$: $RY - GAN$.



(d) $HI^{(2)}$: $DLS - GAN$.

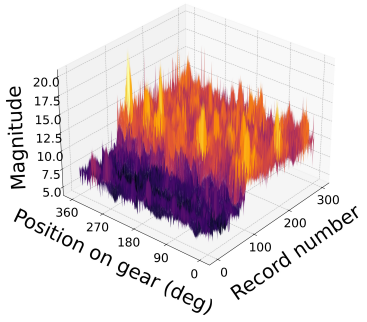


(e) $HI^{(3)}$: $RY - GAN$.

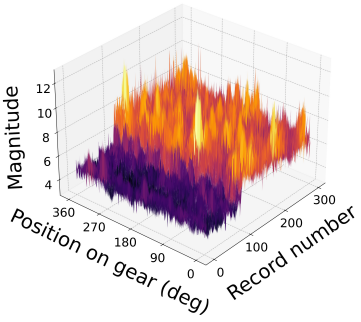


(f) $HI^{(3)}$: $DLS - GAN$.

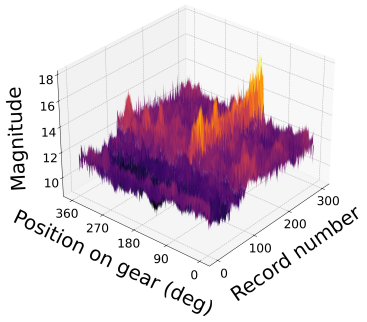
Figure 6.14. The three HI metrics obtained from the $RY - GAN$ (Figure 6.14(a), (c) and (e)) and $DLS - GAN$ (Figure 6.14(b), (d) and (f)) models analysed using the synchronous average.



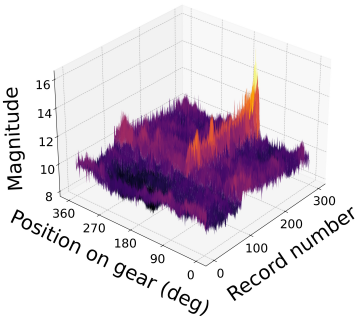
(a) $LHI^{(1)}$: $RY - GAN$.



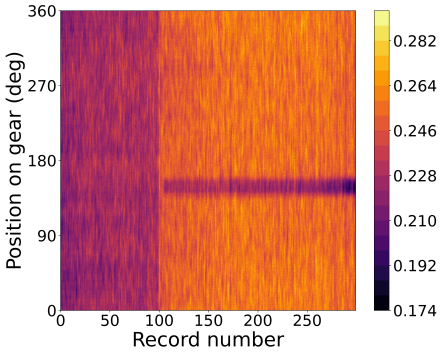
(b) $LHI^{(1)}$: $DLS - GAN$.



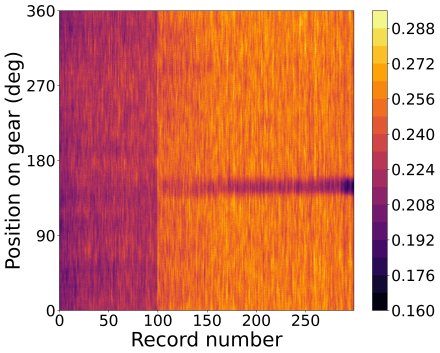
(c) $LHI^{(2)}$: $RY - GAN$.



(d) $LHI^{(2)}$: $DLS - GAN$.



(e) $LHI^{(3)}$: $RY - GAN$.



(f) $LHI^{(3)}$: $DLS - GAN$.

Figure 6.15. The three *LHI* metrics analysed using the synchronous average under the $RY - GAN$ (Figure 6.15(a), (c) and (e)) and $DLS - GAN$ (Figure 6.15(b), (d) and (f)) methodologies.

6.3.2 Unfiltered Gearbox Dataset

The next step in this dataset analysis is to analyse the unfiltered version of this dataset. The objective is to provide a true reflection of how deep learning performs when a dominant component of healthy and unhealthy data is an impulsive component. For the unfiltered data, often poor results were obtained from the discrepancy signal average and the synchronous average alike. Thus, for the sake of conciseness, some of the positive and negative responses will be shown, where applicable. As the L_2 objective function is integral to the considered models, it is not unreasonable to expect that the impulsive components, which are highly non-Gaussian, may not be represented by the models and thus results may break down.

The addition of the impulsive components is a good indicator of the current state of deep learning and how the objective functions used must be carefully analysed to match the field use-case. This requirement is evident when one considers the depth of research applied to vibration-based condition monitoring and the properties of vibration signals. For unsupervised deep learning to be truly competitive, it may be a requirement that one build in some domain knowledge to improve the model performance to a point where it is competitive on datasets consisting of complex vibration components and various operating conditions.

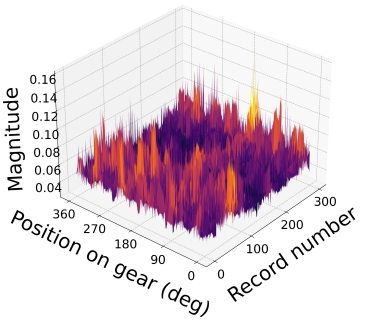
6.3.2.1 *PCA* Model Analysis

For *PCA*, the author chose to investigate two versions of *PCA*, namely, one model that uses all PCs and another model that neglects the first twenty PCs. If one examines $HI^{(1)}$ in Figures 6.16(a) and (b), it is clear that the inclusion of the impulse almost completely broke down the reconstruction ability of the model and the reduced number of PCs further emphasises this. It is interesting to note that $LHI^{(2)}$ appears to be slightly better at identifying damage and gives some indication of the presence of the tooth fault. $LHI^{(3)}$ also exhibits some notable damage presence. However, it is clear to see in Figure 6.16(e) how the latent angle drop is not without regions that rise and drop through time. The cause of this phenomenon is the surrounding temperature variation present in the data, which is problematic as it implies that $LHI^{(3)}$ deviations are not limited to the impulses. It is interesting to note that by dropping twenty PCs, $LHI^{(3)}$ seems to improve using PCA, with less clear temperature variations.

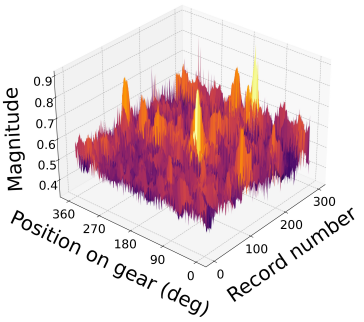
To investigate why *PCA* responds in this manner, the author chose to perform a small analysis into the latent content of *PCA* for the filtered data case and the unfiltered data case. It is possible to do so as PCA uses the eigenvectors of the covariance matrix which are orthogonal to one another and organised hierarchically by eigenvalue magnitude. Figure 6.17 shows the latent frequency content obtained through the *temporal preservation* analysis approach for all of the modes used in the *PCA* model for the filtered and unfiltered versions of the gearbox dataset. It is clear that the latent manifold captures the frequency content of a signal and that result is available when the *temporal preservation* analysis approach. It is clear to note how *PCA* tends to capture the central frequency in the frequency content bands first and then filters the remaining content down through the PCs. The first PCs tend to capture the frequency content around 400, 1700 and $8000Hz$, indicating that the content in the frequency bands dominate the vibration data. This type of analysis can also provide one with insight into what frequency content is lost in the signal reconstruction using *PCA*.

6.3.2.2 *VAE* Model Analysis

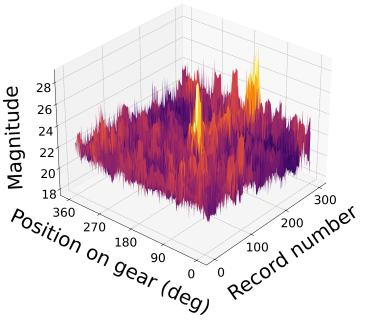
For the analysis of the performance of the *VAE* models, the results from $HI^{(1)}$, $LHI^{(2)}$ and $LHI^{(3)}$ are shown in Figure 6.18. If one analyses Figures 6.18(a) and (b), one immediately notices the difference between the deterministic and stochastic parametrisations in $HI^{(1)}$, with VAE_1 exhibiting no response



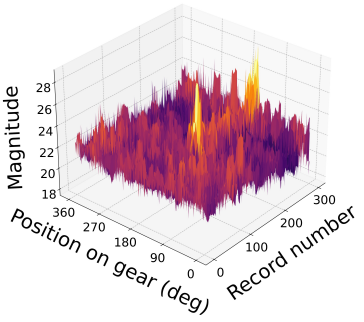
(a) $HI^{(1)}$: all PCs.



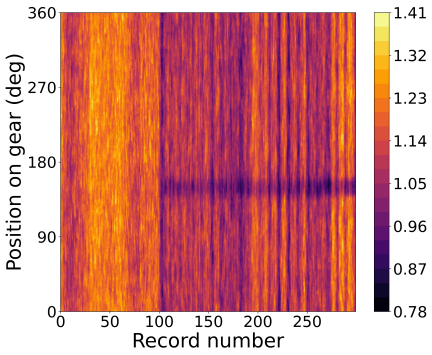
(b) $HI^{(1)}$: dropped twenty PCs.



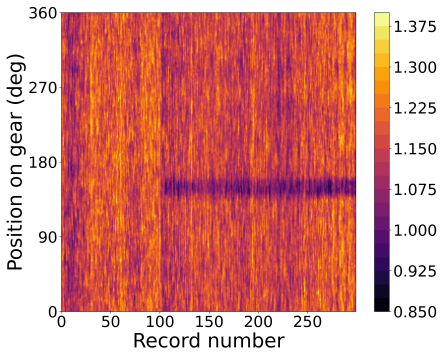
(c) $LHI^{(2)}$: all PCs.



(d) $LHI^{(2)}$: dropped twenty PCs.

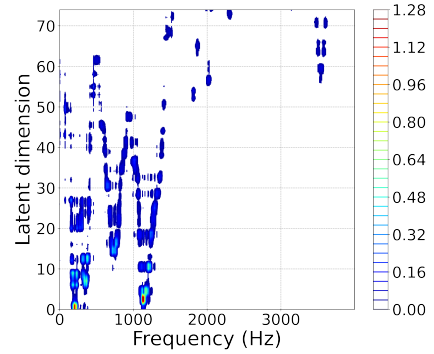


(e) $LHI^{(3)}$: all PCs.

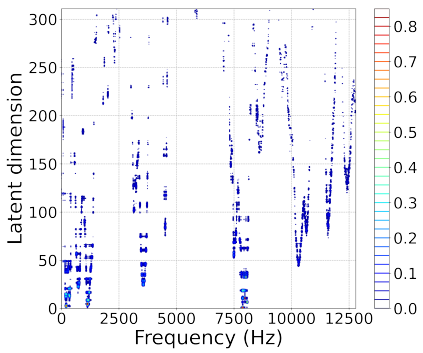


(f) $LHI^{(3)}$: dropped twenty PCs.

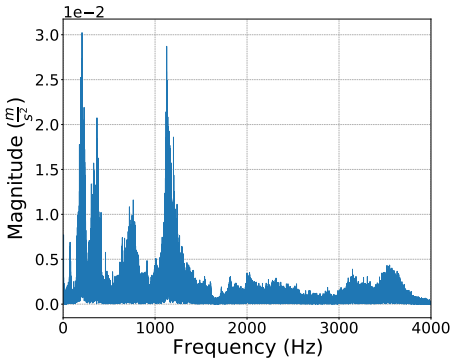
Figure 6.16. $HI^{(1)}$ and $LHI^{(2,3)}$ metrics using *PCA* for all PCs and twenty dropped PCs analysed using the synchronous average. (a), (c) and (e) show the metrics for the all PC case while (b), (d) and (f) show the metrics for the dropped mode case.



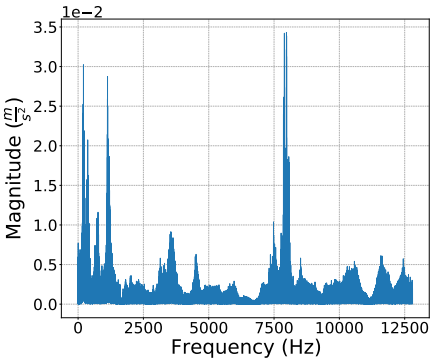
(a) Filtered latent spectrum.



(b) Unfiltered latent spectrum.



(c) Filtered signal frequency spectrum.



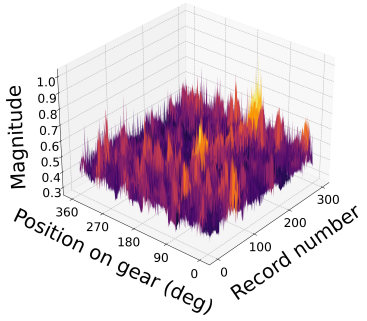
(d) Unfiltered signal frequency spectrum.

Figure 6.17. The latent frequency content of the *PCA* model for the first signal record using the filtered and unfiltered versions of the gearbox dataset. Figure 6.17(a) and (c) shows the filtered latent and signal content while Figure 6.17(b) and (d) shows the unfiltered latent and signal content. Notice the significant difference in the number of modes required to capture 95% of the data variance.

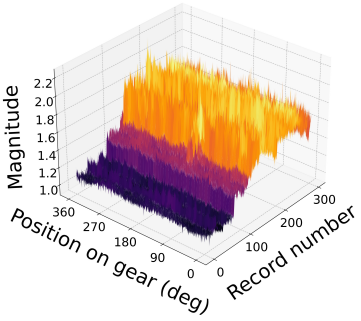
to damage while VAE_2 shows some rapid growth. However, it is clear that this growth is broad and not limited to one tooth increment, which is a clear indicator that albeit learning the variance helps in identifying anomalous instances, the impulsive components are dominating the discrepancy signal response. This is interesting as it highlights that although the variance can aid in detecting anomalous instances, the averaging procedure in the synchronous average is failing as the impulses are increasing the floor of the average. It is clear that although the signal reconstruction *HI* is failing, the latent space is responding to damage strongly. VAE_1 exhibits a response to damage through the latent radius while VAE_2 demonstrates true manifold expressiveness in the response obtained from this model. It is clear that $LHI^{(2)}$ is indicating the presence of the tooth fault and its manifestation is by placing deviance orthogonal from the learnt manifold. $LHI^{(3)}$ also exhibits some clear response to damage alongside some clear variations in record response attributed to the temperature effects present in the data.

The manifold response to damage from a *VAE* is a crucial enhancement of unsupervised deep learning on assets subject to time-varying operating conditions, as it highlights that there is a requirement for model complexity over and above *PCA* and that one needs to analyse the latent manifold clearly to analyse model response to anomalous data. By incorporating the time attribute present in vibration

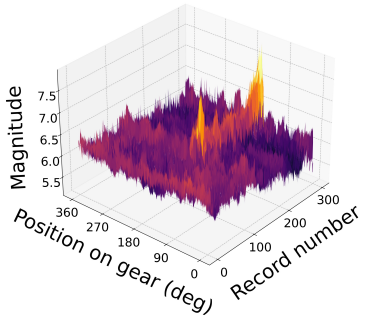
data, one can utilise simple signal processing methods to improve the interpretability of model results. It is also clear that there are issues with the model objective function formulation, with the presence of data impulses causing poor result performance.



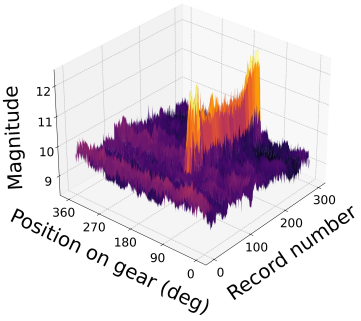
(a) $HI^{(1)}$: VAE_1.



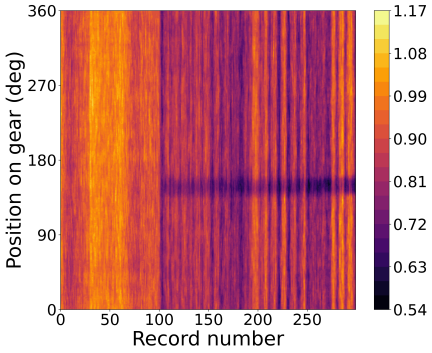
(b) $HI^{(1)}$: VAE_2.



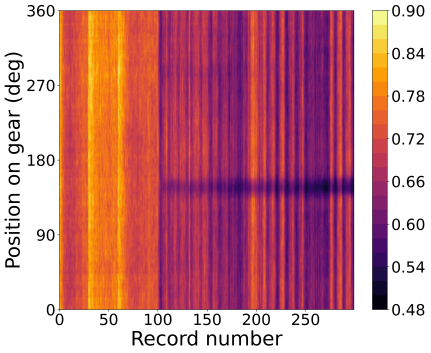
(c) $LHI^{(2)}$: VAE_1.



(d) $LHI^{(2)}$: VAE_2.



(e) $LHI^{(2)}$: VAE_1.

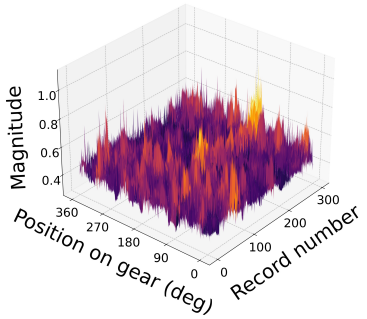


(f) $LHI^{(2)}$: VAE_2.

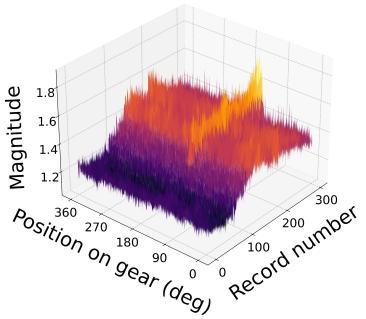
Figure 6.18. $HI^{(1)}$ and $LHI^{(2)}$ metrics using *VAEs* analysed using the synchronous average. (a) and (c) show the metrics for the deterministic *VAE* while (b) and (d) show the metrics for the stochastic *VAE*.

6.3.2.3 $\beta - TC - VAE$ Model Analysis

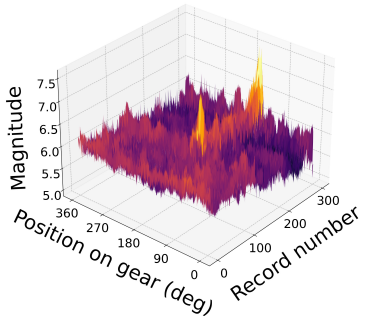
For the response from the $\beta - TC - VAE$ models, the objective is to see how the latent manifold and reconstruction HI respond to damage under the deterministic and stochastic model parametrisations. In Figure 6.19, only the response from $HI^{(1)}$ and $LHI^{(2)}$ as the other LHIs were found to be equivalent to that shown in Figure 6.18. It is clear to see how the model parametrisation affects the results with an improved $HI^{(1)}$ response at the expense of $LHI^{(2)}$ for the stochastic case, shown in Figure 6.19(b) and (d) respectively. The trade-off between signal reconstruction and the latent manifold is present as $\beta - TC - VAE_2$ indicates damage in the reconstruction HI to the detriment of the latent radius. The presence of damage is detectable in the latent manifold albeit less indicative as the response shown in Figure 6.18(d).



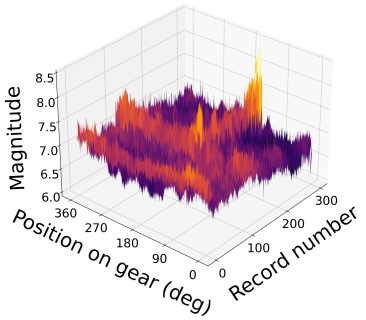
(a) $HI^{(1)}$: $\beta - TC - VAE_1$.



(b) $HI^{(1)}$: $\beta - TC - VAE_2$.



(c) $LHI^{(2)}$: $\beta - TC - VAE_1$.



(d) $LHI^{(2)}$: $\beta - TC - VAE_2$.

Figure 6.19. $HI^{(1)}$ and $LHI^{(2)}$ metrics using the $\beta - TC - VAE$ model analysed using the synchronous average. (a) and (c) show the metrics for the deterministic $\beta - TC - VAE$ while (b) and (d) show the metrics for the stochastic $\beta - TC - VAE$.

6.3.2.4 *GAN* Model Analysis

For the analysis of the *GAN*-based methods, the author chose to showcase the responses from the $RY - GAN$ method as it proved to be the most successful method between the two models considered. In Figure 6.20, all of the HIs and LHIs are shown. It is clear to note that the similarities between the response from $HI^{(1)}$ and those shown for VAE_1 and $\beta - TC - VAE_1$ are all highly similar, a by-product of the shared L_2 objective function. The $HI^{(2)}$ response indicates that the data discriminator is a

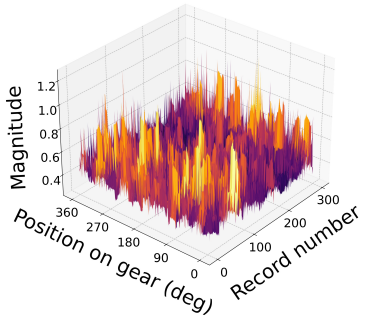
poor performance indicator, a result attributed to the by-product from the *GAN* and the *AE* training scheme trade-off. One good result is that the latent critic is responding to damage, as noted in Figure 6.20(c). The latent manifold is exhibiting some response to damage that is measurable, with the model demonstrating an awareness of anomalous data. From the three *LHI*s, it is clear that the latent radius and angle are exhibiting the presence of damage and that this damage is trackable through the *temporal preservation* approach. The latent manifold is responding in a way that is synonymous with that obtained from the *VAE* models, an attribute of the shared L_2 objective function. The L_2 objective function appears to dominate the construction of the latent manifold, albeit that disentanglement is built-in and the latent regularisation method was different.

6.3.3 Signal Processing Results

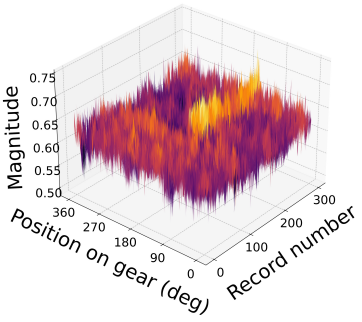
For model performance quantification, it was necessary to analyse how different signal processing techniques perform on the dataset. For this to occur, the author chose three versions of the dataset, namely, a band-pass filtered version of the dataset, a low-pass filtered version of the dataset and the unfiltered version of the dataset. The signal processing techniques used consisted of those shown previously with the addition of the Fast Kurtogram developed by Antoni (2007). Some methods, by design, are not well suited to gear tooth fault detection and as such, only sufficiently performing methods will be presented. Each signal was order-tracked to account for the time-varying operating conditions present in the data, such that the order spectrum could be analysed. For the band-pass filtered version of the dataset, a band of $[200, 700]\,Hz$ was used.

The *SES* and the $SK-NES$ methods produced the best results, while the Kurtogram, $MED-SK-NES$ and $CPS-NES$ approaches failed in one manner or another. $MED-SK-NES$ did work for the band-pass filtered case however, sporadic impulses appeared in some records, a result not uncommon to *MED*. In the figures presented in Figure 6.21, the frequency amplitude of the first, second and impulse orders were tracked for each version. The band-pass filtered version of the dataset produced the best performing signal processing results. Both the *SES* and the $SK-NES$ approaches present some presence of damage for the low-pass filtered case however it is clear that it is not as clear as the responses obtained from the various deep-learning approaches considered in this work. Furthermore, the presence of the impulse in the data at approximately 5.71 orders is dominant in the unfiltered data, with both methods failing significantly. The general shape of the impulse order magnitude from the unfiltered signal processing approaches shown in Figures 6.21(e) and (f) is interesting. It is akin to that seen in the record average for the $HI^{(1)}$ discrepancy signal for the unfiltered dataset, which further serves as an indication that the impulse in the data causes the widespread $HI^{(1)}$ response failure. Figure 6.22 highlights this response alongside the discrepancy signal order spectrum for a *PCA* model that uses all PCs.

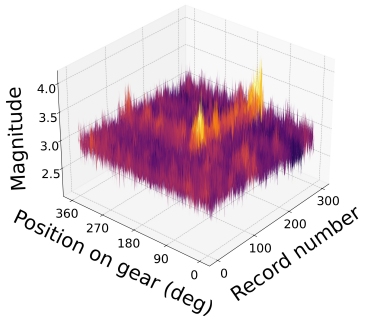
To allow for the deep learning approaches to be compared on the band-pass filtered dataset, the author investigated the performance of a *PCA* model on this data. Figure 6.23 shows the results in this regard and notably, the band-passed version of the dataset is simple to analyse with *PCA* responding strongly to damage. It is interesting to note how the *LHI*s have changed in their response, where previously the data did not respond in the latent distance whereas now it the latent angle that does not respond to damage. One may now interpret how the model utilises information when signal complexity is increased or decreased. As complexity increases, the model does not learnt a latent manifold that can drastically change the traversal velocity for anomalous instances. The model then indicates anomalous instances through the simple properties of the manifold such as the euclidean distance from the origin. In the alternative case of reduced complexity, the latent manifold velocity can be easily changed by placing points far from each other and far from the manifold. It is also clear how *PCA* responds in its



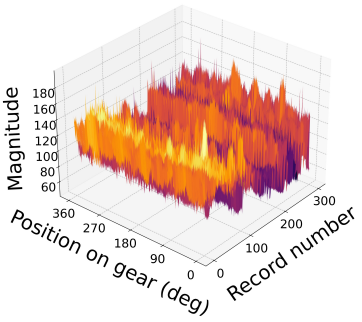
(a) $HI^{(1)}$.



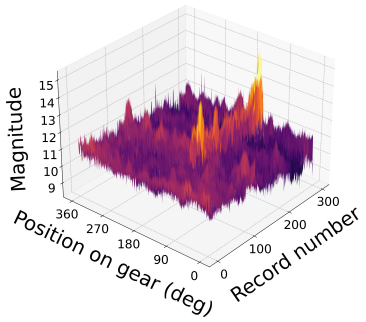
(b) $HI^{(2)}$.



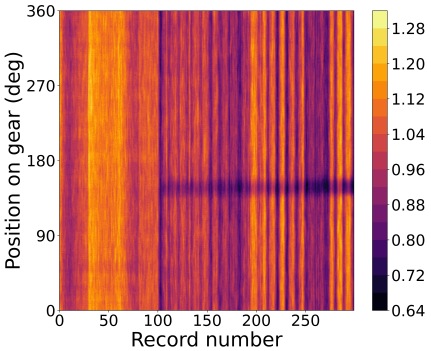
(c) $HI^{(3)}$.



(d) $LHI^{(1)}$.

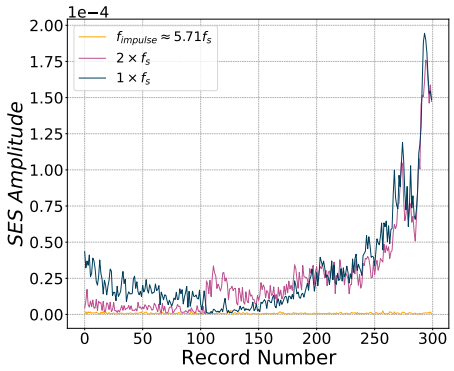


(e) $LHI^{(2)}$.

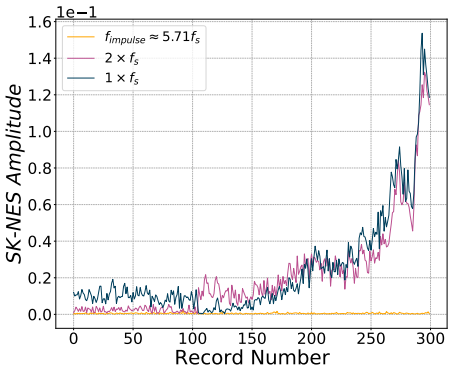


(f) $LHI^{(3)}$.

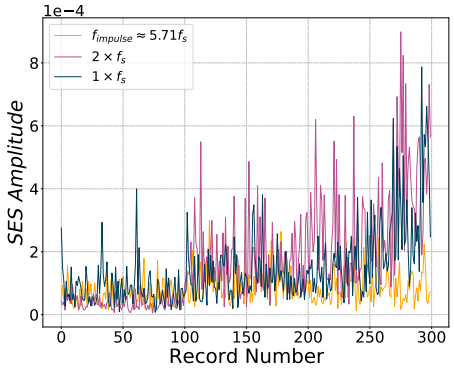
Figure 6.20. The response from the three HIs and the three LHIs using the $RY-GAN$ model analysed using the synchronous average.



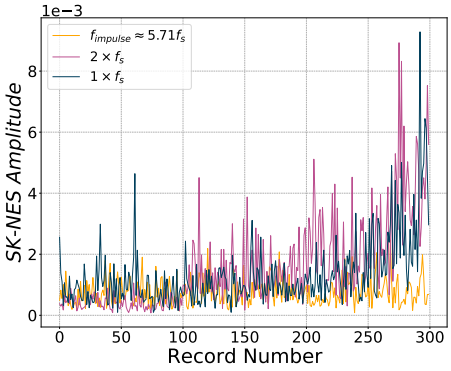
(a) *SES*: band-pass filtered



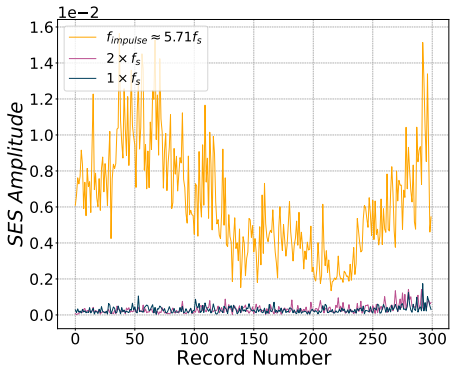
(b) $SK-NES$: band-pass filtered



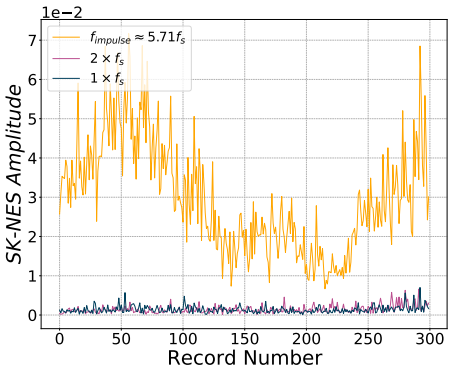
(c) *SES*: low-pass filtered



(d) $SK-NES$: low-pass filtered

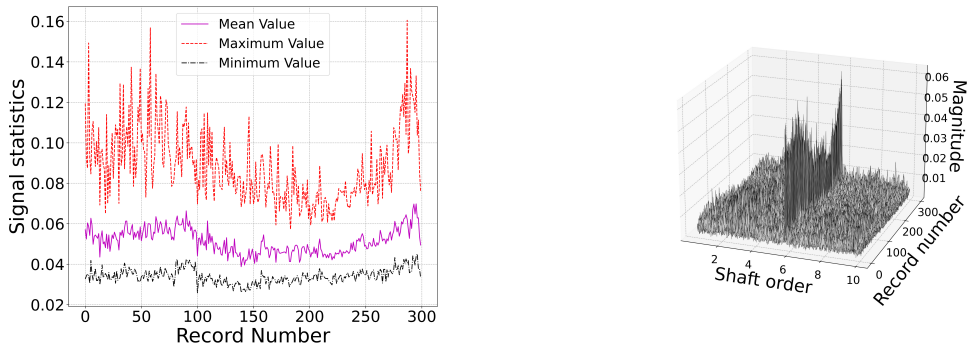


(e) *SES*: unfiltered



(f) $SK-NES$: unfiltered

Figure 6.21. The frequency amplitude responses from the *SES* and $SK-NES$ processing techniques for signals from three versions of the original gearbox dataset. Notice how the gradual addition of the impulsive component worsens the results, with Figure 6.21(e) and (f) showing clear impulsive components.



(a) Record statistics

(b) $HI^{(1)}$ order spectrum

Figure 6.22. The mean, maximum and minimum features of $HI^{(1)}$ synchronous average for each record from a *PCA* model trained in the unfiltered dataset with all PCs used. Note the presence of the impulse at 5.71 orders and the trend of the statistics shown in comparison to Figure 6.21(e) and (f).

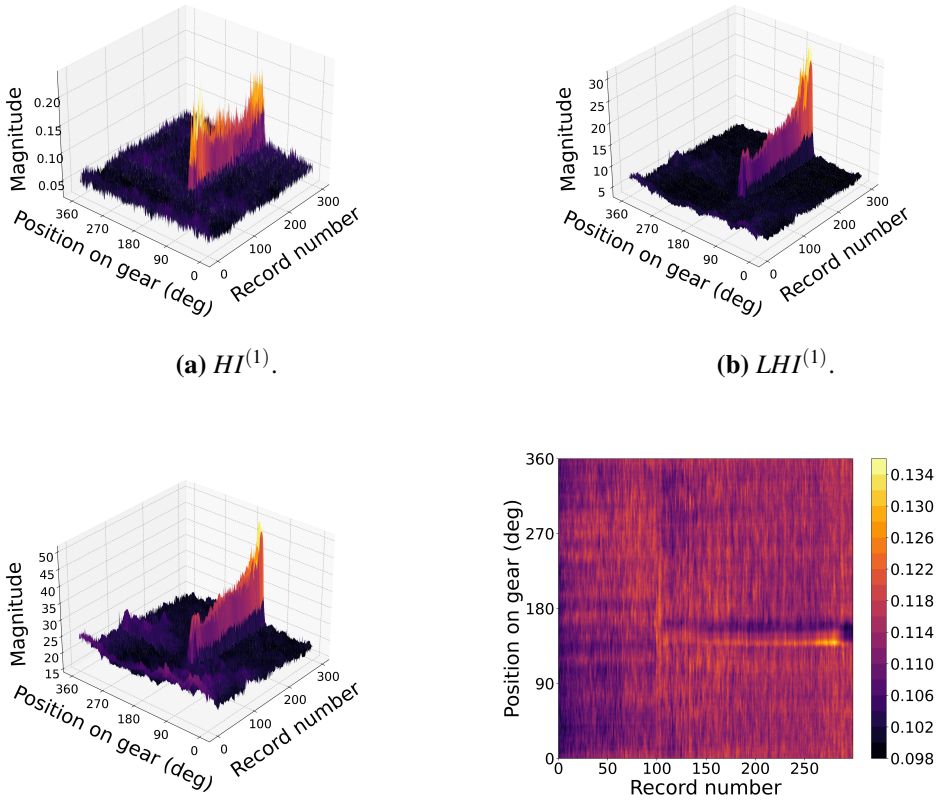
signal reconstruction HI, a clear indication that one can perform fault diagnostics using a simple linear latent variable model. The response under the other model formulations was also validated and was confirmed to be equivalent.

6.3.4 TSA Response Analysis

The next step required for model performance quantification is identify the condition deviance points from the TSA of HI and LHI response. To quantify damage in the synchronous average, an approach similar to that detailed in Schmidt et al. (2017) is applied. The process is to use a clustering algorithm to find two clusters in the synchronous average signal for a given record. One can then access to two means and two standard deviations, from which the means are organised by size under the assumption that when damage is present, the larger of the two means will represent this damage while the smaller one will represent the healthy portion. This process is then repeated for each synchronously averaged signal and allows one to track the growth in the synchronous average data. To alter this approach for the various HIs and LHIs considered, an alternative discrepancy signal to the synchronous average is proposed in this work. This discrepancy signal is the absolute of the difference between the synchronous average and the synchronous average median, given as

$$\tilde{HI}^{(i)} = |HI^{(i)} - \tilde{\mu}_{HI^{(i)}}|,\ i = 1,\ldots,6, \tag{6.1}$$

where $HI^{(i)}$ refers to any of the six available health or latent health indicators and $\tilde{\mu}_{HI^{(i)}}$ refers to the median of the synchronous average of the health indicator. Not all health indicators produce positive deviations around the fault and a HI alteration of this form will produce positive deviations. To identify a condition deviance point, the author has chosen to use the five ahead mean procedure with a threshold defined as $thres = \tilde{\mu} + 3\sigma$. The results of this approach are provided in Table 6.2. To visually motivate the results in Table 6.2, the results using a VAE_2 model on the filtered and standard gearbox dataset as shown in Figure 6.24 and Figure 6.25 respectively. Notice the condition deviance detection in Figure 6.24(b) but the poor growth in the larger k-means centre, a result not uncommon to the $LHI^{(1)}$ response.



(a) $HI^{(1)}$.

(b) $LHI^{(1)}$.

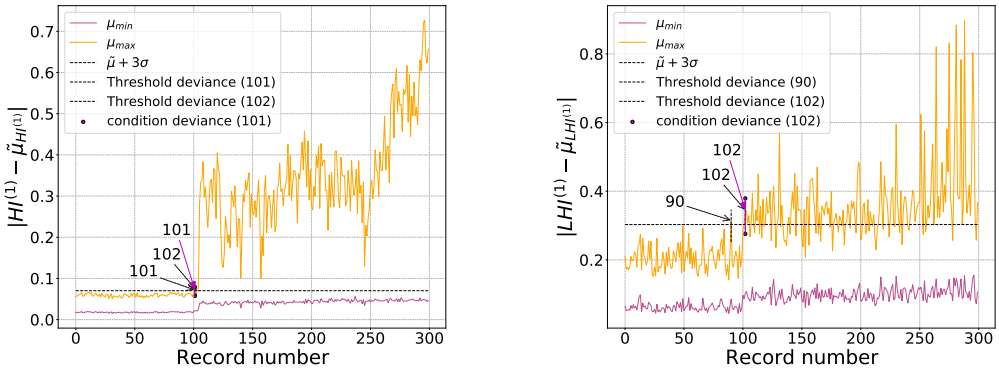
(c) $LHI^{(2)}$.

(d) $LHI^{(3)}$

Figure 6.23. The $HI^{(1)}$ and LHIs synchronous average responses using *PCA* for the band-pass filtered dataset.

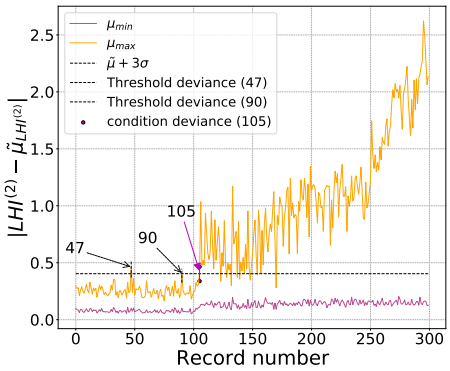
6.4 Conclusion

In the analysis of Table 6.2, two interesting points can be made by simply observing the results. The first is that, specifically for the unfiltered dataset, most methods seem to produce condition deviance points around record one hundred, barring *PCA*. This appears to be inconsistent with some figures shown previously, specifically for $LHI^{(1)}$ results. The subtraction of the synchronous average median causes the proposed metric to track relative deviances, with most metrics introducing detectable jumps around record one hundred potentially introducing a point of detection. The second observation is that *PCA* is consistently worse than the other metrics, which indicates that the additional model non-linearity and complexity is beneficial to this dataset. In the generation of Table 6.2, the author found that some metrics gave condition deviance points but when analysed, were poorly performing metrics and were responding to the change in condition attributed to the manually seeded fault. Table 6.2 shows these cases by underlining the result, with $LHI^{(1)}$ results often capable of detecting the change due to the fault but cannot track the growth thereof. The $DLS-GAN$ model performed poorly on the unfiltered dataset, with the author being unable to obtain any reasonable results. This failure is attributed to the use of both the L_2 and *GAN* objective functions and a potential *GAN* training failure, as detailed in Chapter 2. The change induced by the seeding of the fault was detectable but often this is where the analysis ended.

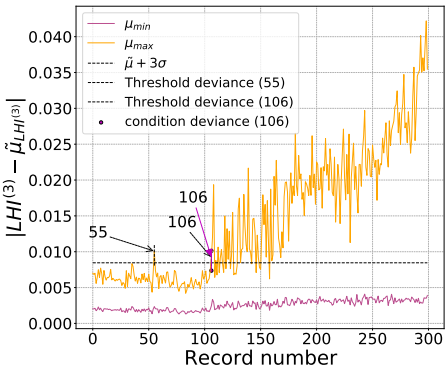


(a) $\tilde{HI}^{(1)}$.

(b) $L\tilde{H}I^{(1)}$.



(c) $L\tilde{H}I^{(2)}$.

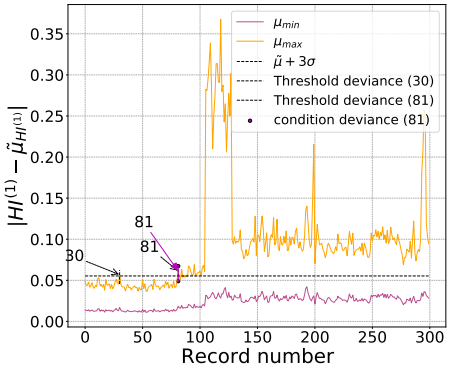


(d) $L\tilde{H}I^{(3)}$

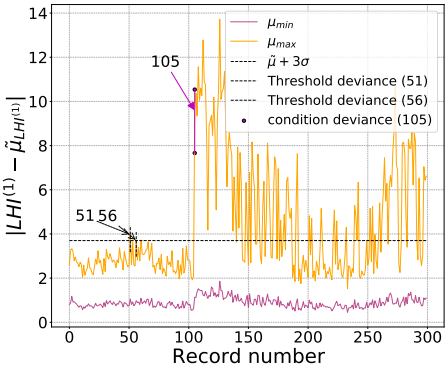
Figure 6.24. The $HI^{(1)}$ and LHI synchronous average $\tilde{HI}^{(i)}$ responses using VAE_2 for the filtered dataset.

Table 6.2. The obtained threshold condition deviance point from the gearbox dataset for the filtered and unfiltered versions of the dataset.

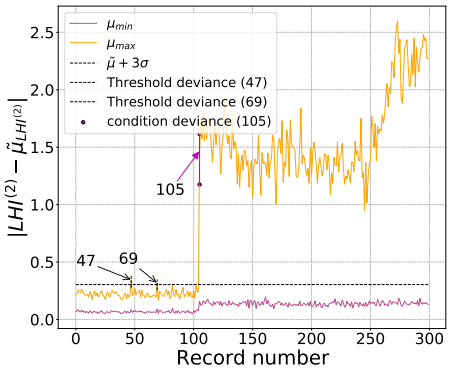
Model type	**k-means health indicator condition deviance point for: low-pass filtered ‖ unfiltered**					
	HI$^{(1)}$	**HI**$^{(2)}$	**HI**$^{(3)}$	**LHI**$^{(1)}$	**LHI**$^{(2)}$	**LHI**$^{(3)}$
PCA - all *PC*s	124 ‖ IC	N/A	N/A	102 ‖ IC	105 ‖ 110	172 ‖ 223
PCA - dropped *PC*s (-5 ‖ -20)	111 ‖ 285	N/A	N/A	101 ‖ IC	105 ‖ 110	176 ‖ 260
VAE_1	105 ‖ 267	N/A	N/A	107 ‖ IC	105 ‖ 105	105 ‖ 107
VAE_2	101 ‖ 81	N/A	N/A	102 ‖ 105	105 ‖ 105	106 ‖ 116
$\beta-TC-VAE_1$	105 ‖ 267	N/A	N/A	107 ‖ IC	105 ‖ 106	105 ‖ 107
$\beta-TC-VAE_2$	102 ‖ 102	N/A	N/A	102 ‖ IC	105 ‖ 106	106 ‖ 109
$RY-GAN$	105 ‖ IC	IC ‖ 102	103 ‖ 106	100 ‖ IC	105 ‖ 105	139 ‖ 104
$DLS-GAN$	105 ‖ IC	106 ‖ 106	113 ‖ 100	100 ‖ IC	106 ‖ 100	171 ‖ 157



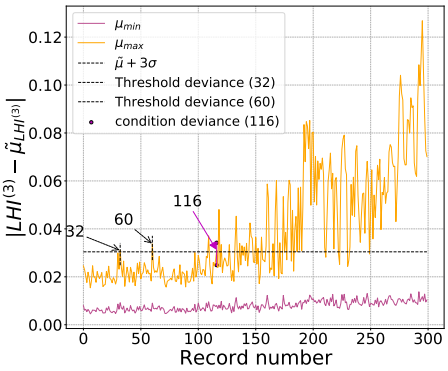
(a) $\tilde{HI}^{(1)}$.



(b) $\tilde{LHI}^{(1)}$.



(c) $\tilde{LHI}^{(2)}$.



(d) $\tilde{LHI}^{(3)}$

Figure 6.25. The $HI^{(1)}$ and LHI synchronous average $\tilde{HI}^{(i)}$ responses using VAE_2 for the low-pass unfiltered dataset.

Furthermore, the data discriminators for the *GAN*-based methods provided little in terms of fault detection. This was attributed to the impulsive data components and the model inability to capture this data due to its formulation. It can be concluded, for this dataset, that deep learning can be competitive with signal processing and that it is applicable for time-varying operating condition cases. However, the Gaussian assumptions built into the models will break down when the data is non-Gaussian and, as shown, will result in deep learning being unable to obtain results unless the impulses are accounted for in some way, which is why the synchronous average was crucial in result interpretation. Unsupervised latent variable models are competitive and capable of capturing speed profile variations but they are not at a place where they need to be, just yet.

Chapter 7 Conclusion and Recommendations

7.1 Conclusion

This research aimed to investigate the latent manifold for interpretability, responsiveness to damage and to analyse the performance of the proposed latent metrics. The importance of untangled latent manifolds and *temporal preservation* analysis of latent metrics indicates that previous investigations of unsupervised deep learning have approached these aspects suboptimally. Specifically, latent manifold response metrics were proposed alongside the *temporal preservation* analysis to highlight how the latent and data space can be used in conjunction for anomaly detection. The investigations considered and highlighted the interactions between model window length and fault frequency, with a focus on how the shaft speed is a crucial factor in this interaction. A variety of latent-variable models were considered, namely, *PCA*, *VAE*s, $\beta - TC - VAE$s and *GAN*-based methods such as the $DLS - GAN$ and the proposed $RY - GAN$. These models highlighted the importance of considering an ensemble of latent variable models to conduct PHM effectively.

The main contributions of this work are three-fold. Firstly, it shows that the standard approach to processing time-series data for data-driven models must be reformulated to utilise time. This allows for information in the *HI*s and *LHI*s to be fully realised and interpreted. Secondly, it shows that the latent manifold of latent variable models when untangled is interpretable, making them suitable to detect the presence of fault covariates in time-series data. It was empirically shown that latent variable models are strong competitors for identifying faults in time-series data. Analyses in gearbox fault applications under both stationary and time-varying operating conditions clearly demonstrated that one could detect a variety of faults using unsupervised data-driven techniques. Machine fault diagnostics highlighted the need to carefully assess the decision of which latent variable model to use for any given application, which opens up the possibility of using the multiple facets of a latent variable model, in a carefully considered manner, for PHM. The *temporal preservation* approach is a trivial reformulation of the data processing problem. Still, it can offer significant insights into the implicit model assumptions made during model design and into the traversal of time-series data through the latent manifold.

The investigations performed in this work demonstrated that latent variable models are capable of learning a latent manifold that is responsive to damage and that the three proposed *LHI*s allow for a physical interpretation of how a model responds when presented with anomalous data. This is the first application of latent metrics that are self-contained and not specific to any model type. The *LHI*s proposed are sufficiently flexible to allow for variations in fault and dataset type, with the investigations in this work yielding fruitful latent manifold responses. The difference between standard pre-processing and the proposed *temporal preservation* approach is trivial, requires no change in model training and is only used in model evaluation. Improvements include the ability to determine the type of fault in both stationary and time-varying operating condition cases. Deep learning, as it currently stands,

can be easily interpreted as a discrepancy analysis method and the *temporal preservation* approach allows one to augment the interpretation given to faults in vibration-based condition monitoring circumstances.

A study into the role of linear and non-linear latent variable models in PHM applications was considered. *PCA* often performed well on datasets with stationary operating conditions. In the datasets investigated, the response to damage was consistent through the latent-variable models, which provided a clear indication that the *HI*s and *LHI*s can be used in conjunction for damage detection. For the third dataset, it was shown that although non-linear latent variable models performed better than linear variable models, the contribution of the L_2 objective function introduced inappropriate responses. Specifically, the data impulses present in both the healthy and unhealthy data could not be effectively captured by the model and thus were immediately indicated as anomalous. This was shown by considering a filtered and unfiltered version of the dataset. The health and latent health metric responses were found to be mostly uninformative in their simple statistics, with a requirement of hinging off the benefits of time-synchronous averaging to uncover the fault. It can be noted that TSA was only performed post-training and is only feasible due to the *temporal preservation* approach. This does introduce the potential of combined deep learning and signal processing approaches, as typically these techniques are investigated in similar circumstances but with no interaction.

In the comparison between signal processing and deep learning techniques, it is clear that there is a strong case to be made for the use of latent variable models in gearbox fault diagnostics. The use of latent variable models in an unsupervised learning setting can offer significant improvements in the detection of anomalous instances in time-series data and this is a highly exploitable feature that can be used in the PHM field. The ability to take healthy machine data and detect deviations from the healthy data manifold and the latent manifold is powerful, as it completely negates the requirement for any faulty data. A lack of fault data means that the cost of implementing and deploying these models is significantly decreased along with the rise in computational power and the Internet of Things. The cost of monitoring an asset is decreased further under these considerations and can be employed in an online setting. The metrics obtained from latent variable models are also more intuitive to interpret and understand as opposed to those from signal processing techniques, which often require someone well versed with the implementation and technique to interpret the results. This is a direct result of the latent variable model frameworks, which reduces any problem to data and probability distributions. This generalises what is often used in signal processing, as the field is refined to approach particular types of issues. The downside of the deep learning techniques considered in this work is the presence of the L_2 objective function, which was shown to be problematic in the presence of impulsive signal components. The adversarial framework from *GAN*s offers a natural solution to this problem; however, model inference then becomes a non-trivial and complicated procedure.

In the presence of any additional faulty data, a unsupervised learning scheme can be employed to identify any fault classes, if required. The downside here is that this is an area where signal processing techniques offer significant advantages, as often these techniques can detect the type of fault by introducing expert knowledge into the problem and understanding the relationship between the energy content in a signal and the nature of faults. Signal processing also has access to in-depth knowledge about the nature of faults in time-series data, and this knowledge has been used to develop powerful techniques that can cover a wide variety of problems. Signal processing is a well-established field, which means that there is a high baseline that unsupervised data-driven techniques must overcome to be considered viable technology. However, this offers a natural merging between the two domains to capitalise on the benefits that each has to offer. Latent variable models provide a unique method to

uncover the presence of faults covariates in a signal. In contrast, signal processing offers services in fault classification and determining the exact type of fault present in the signal.

This benefit was realised during the analysis of the UP C-AIM gearbox dataset, where the latent variable models were used to develop the discrepancy signals for the different *HI*s and *LHI*s and *TSA*, a common signal processing technique, was used to uncover the exact location and type of fault. This example shows a unique method of combining the two fields, whereby deep learning techniques could be assisted by signal processing or vice versa. For the former combination, the use of signal processing techniques to pre-process the data could meaningfully benefit the performance of unsupervised data-driven methods as there are signal processing techniques that are known the improve the enhancement of fault covariates in a signal. There are a plethora of signal processing techniques that can be used, such as the Normalisation of the Amplitude Modulation caused by Varying Operating Conditions (NAMVOC) method proposed by Schmidt and Heyns (2020), for example. This technique can be used to improve the quality of the data shown to a latent variable model by reducing the amplitude modulation effects of time-varying operating conditions. For the latter combination, signal processing techniques can be applied to further evaluate the *HI*s and *LHI*s obtained under the *temporal preservation* approach to confirm the cause of any identifiable condition deviance points. This can manifest through record-frequency plots and the analysis of the evolution of known theoretical fault frequencies for the stationary operating condition case, an analysis technique that is demonstrated in Section 4.3, or through the order spectrum for time-varying operating condition cases. In hindsight, pursuing each field in isolation seems naive as the simple notion of using deep learning approaches in a discrepancy analysis framework can combine the two fields in a simple, intuitive and complementary manner. Ultimately, the goal of PHM is to ensure that fault can be detected, trended and isolated as early as possible, so utilising both fields to realise this goal can only benefit the machine fault diagnostics field.

Let us critically compare the results obtained from the signal processing techniques, linear latent variable model and the non-linear latent variable models. A variety of factors exist that must be considered during this comparison. The signal processing results from IMS dataset clearly highlighted the degradation of the bearing over time and could correctly identify the bearing component responsible for the fault. The difference between the linear and non-linear latent variable models was not that distinct, where *PCA* was a highly competitive option for fault diagnostics. The fact that *PCA* works well indicates that datasets with little to no change in operating condition throughout the experimental lifespan are constrained in such a manner that a linear transition function may be suitable uncover the presence of damage. In this case, the return on investment obtained from more complex non-linear formulations is minimal, as *PCA* is computationally efficient and is trivial to implement in the current computational state.

From the gearbox dataset, however, the addition of model non-linearity began to show as *PCA* became the model with consistently poor performance. This is attributed to the time-varying operating conditions, which make the input data space complex and non-linear. The non-linear latent variable models were then able to uncover the presence of the gear tooth fault in both the input and latent manifold. However, the downfall here was the impulsive signal components in the data. This is where a signal processing approach will tend to shine through, as the presence of impulsive components are detailed and investigated in the literature, with techniques that exist that can readily overcome them. This downfall, from a latent variable model perspective, was, however, a function of the explicitly assumed distributions rather than latent variable models in general. The distributions were limited to multivariate isotropic Gaussian distributions. The *GAN*-based methods were used as the adversarial

framework was proposed to offer improved flexibility in describing densities. However, it is clear from this work that this is not the case as the combination of this framework and the L_2 loss were not aligned towards the same goals.

7.2 Future work

For future work, the following avenues should be considered for deep learning research on vibration data:

- It is an absolute necessity that the L_2 loss is replaced for deep learning models trained on raw vibration data. It was shown and highlighted in detail in this work how this loss has substantial implications on the formulation of the latent manifold and forces the model to a place where specific vibration components cannot be learnt. Adversarial losses are a vital contributor to the solution as they make implicit data distribution assumptions. Booyse et al. (2020) did make this clear, however, a *GAN* does not offer any latent manifold exploration which was shown to be a key model element. *GAN* disentanglement has been implemented in literature, but often one cannot perform model inference.
- Normalising and auto-regressive flows are exciting aspects of unsupervised machine learning that have yet to be explored for PHM. The benefits of these approaches are that they exploit the change of variables theorem to obtain a series of invertible bijective mappings that, under certain design choices, allow one to easily transform data from a simple distribution to complex distribution and vice versa. Formulations such as the real-valued non-volume preserving (Real NVP) or Parallel Wavenet, proposed by Dinh et al. (2016) and Van Den Oord et al. (2018), maybe an interesting avenue of research. For vibration data, it is believed that a good starting point would be the forward KL divergence formulation detailed in Papamakarios et al. (2019) as we do not have access to $p(\mathbf{x})$ but only samples from this distribution.
- Currently, the formulations considered in this work explore disentanglement, but it was not made clear what advantages this offers into data causality and interpretation. This is still an avenue that needs to be investigated and clarified for time-series data.
- Formulations such as Independent Component Analysis may be an exciting avenue of work as it assumes that the data is non-Gaussian. This offers a direct departure from the L_2-loss. It may also be interesting, given the strong performance of *PCA* on datasets with stationary or quasi-stationary operating conditions, to investigate methods such as kernel *PCA*.
- This work highlighted the importance of the model window length; however, at no point was an optimal window length identified. It may be required for future work that this choice be made for the user in a transparent and conducive manner for deep learning-based PHM.
- It may be required that deep learning models be designed using domain-specific knowledge to enhance model performance for PHM further. If the primary objective of the research is to obtain work that is readily applicable to the industry, then the use of domain knowledge is vital for the development of work that utilises and exploits signal processing knowledge and can only improve the field.
- The threshold condition deviance approach used in this work was used to provide a consistent comparison platform for the various *HI*s and *LHI*s. This approach, however, is far from ideal and can be further explored to determine whether improved health indicator condition inference techniques can be developed.

References

Abboud, D., Antoni, J., Sieg-Zieba, S., and Eltabach, M. Envelope analysis of rotating machine vibrations in variable speed conditions: A comprehensive treatment. *Mechanical Systems and Signal Processing*, 84:200–226, 2017. ISSN 10961216. doi: 10.1016/j.ymssp.2016.06.033. URL http://dx.doi.org/10.1016/j.ymssp.2016.06.033.

Abboud, D., Elbadaoui, M., Smith, W. A., and Randall, R. B. Advanced bearing diagnostics: A comparative study of two powerful approaches. *Mechanical Systems and Signal Processing*, 114: 604–627, 2019. ISSN 10961216. doi: 10.1016/j.ymssp.2018.05.011. URL https://doi.org/10.1016/j.ymssp.2018.05.011.

Abboud, D., Antoni, J., Eltabach, M., and Sieg-Zieba, S. Angle\time cyclostationarity for the analysis of rolling element bearing vibrations. *Measurement: Journal of the International Measurement Confederation*, 75:29–39, 2015. ISSN 02632241. doi: 10.1016/j.measurement.2015.07.017. URL http://dx.doi.org/10.1016/j.measurement.2015.07.017.

Akcay, S., Atapour-Abarghouei, A., and Breckon, T. P. GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11363 LNCS, pages 622–637, may 2019. ISBN 9783030208929. doi: 10.1007/978-3-030-20893-6_39. URL http://arxiv.org/abs/1805.06725.

An, J. and Cho, S. SNU Data Mining Center 2015-2 Special Lecture on IE Variational Autoencoder based Anomaly Detection using Reconstruction Probability. 2015. URL https://pdfs.semanticscholar.org/0611/46b1d7938d7a8dae70e3531a00fceb3c78e8.pdf.

Antoni, J. Fast computation of the kurtogram for the detection of transient faults. *Mechanical Systems and Signal Processing*, 21(1):108–124, 2007. ISSN 08883270. doi: 10.1016/j.ymssp.2005.12.002.

Antoni, J. Cyclostationarity by examples. *Mechanical Systems and Signal Processing*, 23(4):987–1036, 2009. ISSN 08883270. doi: 10.1016/j.ymssp.2008.10.010.

Antoni, J. The infogram: Entropic evidence of the signature of repetitive transients. *Mechanical Systems and Signal Processing*, 74:73–94, 2016. ISSN 10961216. doi: 10.1016/j.ymssp.2015.04.034. URL http://dx.doi.org/10.1016/j.ymssp.2015.04.034.

Antoni, J. and Borghesani, P. A statistical methodology for the design of condition indicators. *Mechanical Systems and Signal Processing*, 114:290–327, 2019. ISSN 10961216. doi: 10.1016/j.ymssp.2018.05.012. URL https://doi.org/10.1016/j.ymssp.2018.05.012.

Antoni, J. and Randall, R. B. The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines. *Mechanical Systems and Signal Processing*, 20(2):308–331, 2006. ISSN 08883270. doi: 10.1016/j.ymssp.2004.09.002.

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

Baggeröhr, S. A Deep Learning Approach Towards Diagnostics of Bearings Operating under Non-Stationary Conditions. 2019.

Barber, D. and Agakov, F. The IM algorithm : A variational approach to information maximization. *Advances in Neural Information Processing Systems*, (2), 2004. ISSN 10495258.

Barszcz, T. and Jabłoński, A. A novel method for the optimal band selection for vibration signal demodulation and comparison with the Kurtogram. *Mechanical Systems and Signal Processing*, 25 (1):431–451, 2011. ISSN 08883270. doi: 10.1016/j.ymssp.2010.05.018.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2013.50.

Bishop, C. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 1 edition, 2006. ISBN 0387310738.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. ISSN 1537274X. doi: 10.1080/01621459.2017.1285773.

Booyse, W., Wilke, D. N., and Heyns, S. Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing*, 140:106612, 2020. ISSN 10961216. doi: 10.1016/j.ymssp.2019.106612. URL https://doi.org/10.1016/j.ymssp.2019.106612.

Borghesani, P., Pennacchi, P., Randall, R. B., and Ricci, R. Order tracking for discrete-random separation in variable speed conditions. *Mechanical Systems and Signal Processing*, 30:1–22, 2012. ISSN 08883270. doi: 10.1016/j.ymssp.2012.01.015. URL http://dx.doi.org/10.1016/j.ymssp.2012.01.015.

Borghesani, P., Pennacchi, P., Randall, R. B., Sawalhi, N., and Ricci, R. Application of cepstrum pre-whitening for the diagnosis of bearing faults under variable speed conditions. *Mechanical Systems and Signal Processing*, 36(2):370–384, 2013. ISSN 08883270. doi: 10.1016/j.ymssp.2012.11.001.

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., and Schoelkopf, B. From optimal transport to generative modeling: the VEGAN cookbook. pages 1–15, 2017. URL http://arxiv.org/abs/1705.07642.

Bouvrie, J. Notes on Convolutional Neural Networks. 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=FE6CFFA9E9606F72DAA6B9AF040C392C?doi=10.1.1.70.1419{&}rep=rep1{&}type=pdf{%}5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.1419.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in \textbeta-VAE. (Nips), 2018. URL http://arxiv.org/abs/1804.03599.

Capdessus, C., Sidahmed, M., and Lacoume, J. L. Cyclostationary processes: application in gear faults early diagnosis. *Mechanical Systems and Signal Processing*, 14(3):371–385, 2000. ISSN 08883270. doi: 10.1006/mssp.1999.1260.

Chapelle, O., Schölkopf, B., and Zien, A. *Semi-Supervised Learning*. Number November. MIT Press, Cambridge, UNITED STATES, 2006. ISBN 9780262255899. URL http://ebookcentral.proquest.com/lib/pretoria-ebooks/detail.action?docID=3338523.

Chen, T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, (NeurIPS), 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. jun 2016. URL http://arxiv.org/abs/1606.03657.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. Variational lossy autoencoder. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–17, 2017.

Coats, M., Sawalhi, N., and Randall, R. Extraction of tacho information from a vibration signal for improved synchronous averaging. *Annual Conference of the Australian Acoustical Society 2009 - Acoustics 2009: Research to Consulting*, 2009.

Di Mattia, F., Galeone, P., De Simoni, M., and Ghelfi, E. A Survey on GANs for Anomaly Detection. jun 2019. URL http://arxiv.org/abs/1906.11632.

Diamond, D. H., Heyns, P. S., and Oberholster, A. J. Online shaft encoder geometry compensation for arbitrary shaft speed profiles using Bayesian regression, 2016. ISSN 10961216.

Ding, F. and Luo, F. Clustering by Directly Disentangling Latent Space. 2019. URL http://arxiv.org/abs/1911.05210.

Ding, J., Ren, X., Luo, R., and Sun, X. An Adaptive and Momental Bound Method for Stochastic Learning. 2019. URL http://arxiv.org/abs/1910.12249.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. may 2016. URL http://arxiv.org/abs/1605.08803.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial Feature Learning. (2016):1–18, may 2016. URL http://arxiv.org/abs/1605.09782.

Dowson, D. C. and Landau, B. V. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 10957243. doi: 10.1016/0047-259X(82)90077-X.

Dubey, S. R., Chakraborty, S., Roy, S. K., Mukherjee, S., Singh, S. K., and Chaudhuri, B. B. diffGrad: An Optimization Method for Convolutional Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. ISSN 21622388. doi: 10.1109/TNNLS.2019.2955777.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. New York, 2 edition, 2001. ISBN 9781118586006.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially Learned Inference. pages 1–18, jun 2016. URL http://arxiv.org/abs/1606.00704.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization. *Uncertainty in Artificial Intelligence - Proceedings of the 31st Conference, UAI 2015*, pages 258–267, 2015.

Endo, H. and Randall, R. B. Enhancement of autoregressive model based gear tooth fault detection technique by the use of minimum entropy deconvolution filter. *Mechanical Systems and Signal Processing*, 21(2):906–919, 2007. ISSN 08883270. doi: 10.1016/j.ymssp.2006.02.005.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and van de Meent, J.-W. Structured Disentangled Representations. 2018. URL http://arxiv.org/abs/1804.02086.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the Manifold Hypothesis. 2013. ISSN 0894-0347. doi: 10.1090/jams/852. URL http://arxiv.org/abs/1310.0425.

Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W. J., and Ducoffe, M. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92(April):103678, 2020. ISSN 09521976. doi: 10.1016/j.engappai.2020.103678. URL https://doi.org/10.1016/j.engappai.2020.103678.

Fyfe, K. and Munck, E. Analysis of computed order tracking [machine vibration analysis]. *Mechanical Systems and Signal Processing*, 11(2):187–205, 1997. ISSN 0888-3270. doi: 10.1006/mssp.1996.0056. URL http://www.engineeringvillage.com/blog/document.url?mid=inspec{_}base905641210{&}database=ins.

Gao, Z., Cecati, C., and Ding, S. X. A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6):3757–3767, 2015. ISSN 02780046. doi: 10.1109/TIE.2015.2417501.

Gardner, W. A., Napolitano, A., and Paura, L. Cyclostationarity: Half a century of research. *Signal Processing*, 86(4):639–697, 2006. ISSN 01651684. doi: 10.1016/j.sigpro.2005.06.016.

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

Goodfellow, I., Bengio, Y., and Courville, A. The Deep Learning Book. *MIT Press*, 521(7553):785, 2017. ISSN 1432122X. doi: 10.1016/B978-0-12-391420-0.09987-X.

Goodfellow, I. J. On distinguishability criteria for estimating generative models. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, pages 1–6, 2015.

Gousseau, W., Antoni, J., Girardin, F., and Griffaton, J. Analysis of the rolling element bearing data set of the center for intelligent maintenance systems of the University of Cincinnati. In *13th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies, CM 2016/MFPT 2016*, 2016.

Gretton, A., Borgwardt, K., Rasch, M. J., Scholkopf, B., and Smola, A. J. A Kernel Method for the Two-Sample Problem. may 2008. ISSN 1049-5258. URL http://arxiv.org/abs/0805.2368.

Gryllias, K. C. and Antoniadis, I. A. A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments. *Engineering Applications of Artificial Intelligence*, 25(2):326–344, 2012. ISSN 09521976. doi: 10.1016/j.engappai.2011.09.010.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. Improved Training of Wasserstein GANs. mar 2017. URL http://arxiv.org/abs/1704.00028.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7. URL http://link.springer.com/10.1007/978-0-387-84858-7.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):6627–6638, 2017. ISSN 10495258.

Heyns, T., Godsill, S. J., De Villiers, J. P., and Heyns, P. S. Statistical gear health analysis which is robust to fluctuating loads and operating speeds, 2012a. ISSN 08883270.

Heyns, T., Heyns, P. S., and De Villiers, J. P. Combining synchronous averaging with a Gaussian mixture model novelty detection scheme for vibration-based condition monitoring of a gearbox. *Mechanical Systems and Signal Processing*, 32:200–215, 2012b. ISSN 08883270. doi: 10.1016/j.ymssp.2012.05.008. URL http://dx.doi.org/10.1016/j.ymssp.2012.05.008.

Heyns, T., Heyns, P. S., and Zimroz, R. Combining discrepancy analysis with sensorless signal resampling for condition monitoring of rotating machines under uctuating operations. *International Journal of Condition Monitoring*, 2(2):52–58, dec 2012c. ISSN 20476426. doi: 10.1784/204764212804729714. URL http://openurl.ingenta.com/content/xref?genre=article{&}issn=2047-6426{&}volume=2{&}issue=2{&}spage=52.

Heyns, T., Heyns, P., and de Villiers, J. Combining synchronous averaging with a Gaussian mixture model novelty detection scheme for vibration-based condition monitoring of a gearbox. *Mechanical Systems and Signal Processing*, 32:200–215, oct 2012d. ISSN 08883270. doi: 10.1016/j.ymssp.2012.05.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0888327012002221.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, jul 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL https://www.mitpressjournals.org/doi/abs/10.1162/neco.2006.18.7.1527.

Hoang, D. T. and Kang, H. J. A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing*, 335:327–335, 2019. ISSN 18728286. doi: 10.1016/j.neucom.2018.06.078.

Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://www7.informatik.tu-muenchen.de/{~}hochreit{%}0Ahttp://www.idsia.ch/{~}juergenhttps://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735.

Jardine, A. K., Lin, D., and Banjevic, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7): 1483–1510, 2006. ISSN 08883270. doi: 10.1016/j.ymssp.2005.09.012.

Jedliński, Ł. and Jonak, J. Early fault detection in gearboxes based on support vector machines and multilayer perceptron with a continuous wavelet transform. *Applied Soft Computing Journal*, 30: 636–641, 2015. ISSN 15684946. doi: 10.1016/j.asoc.2015.02.015.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(2):175–193, 1906. ISSN 0001-5962. doi: 10.1007/BF02418571. URL http://projecteuclid.org/euclid.acta/1485887155.

Jiang, G., He, H., Yan, J., and Xie, P. Multiscale Convolutional Neural Networks for Fault Diagnosis of Wind Turbine Gearbox. *IEEE Transactions on Industrial Electronics*, 66(4):3196–3207, 2019a. ISSN 02780046. doi: 10.1109/TIE.2018.2844805.

Jiang, Q., Jia, M., Hu, J., and Xu, F. Machinery fault diagnosis using supervised manifold learning. *Mechanical Systems and Signal Processing*, 23(7):2301–2311, 2009. ISSN 08883270. doi: 10.1016/j.ymssp.2009.02.006. URL http://dx.doi.org/10.1016/j.ymssp.2009.02.006.

Jiang, W., Cheng, C., Zhou, B., Ma, G., and Yuan, Y. A Novel GAN-based Fault Diagnosis Approach for Imbalanced Industrial Time Series. pages 1–6, 2019b. URL http://arxiv.org/abs/1904.00575.

Khan, S. and Yairi, T. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018. ISSN 10961216. doi: 10.1016/j.ymssp.2017.11.024. URL https://doi.org/10.1016/j.ymssp.2017.11.024.

Kim, H. and Mnih, A. Disentangling by factorising. *35th International Conference on Machine Learning, ICML 2018*, 6:4153–4171, 2018.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. pages 1–15, 2014. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. (Ml):1–14, dec 2013. URL http://arxiv.org/abs/1312.6114.

Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 4(January):3581–3589, 2014. ISSN 10495258.

Koehn, P. Combining Genetic Algorithms and Neural Networks: The Encoding Problem. (December): 1–67, 1994.

Kumar Singh, B., Verma, K., and S. Thoke, A. Investigations on Impact of Feature Normalization Techniques on Classifier's Performance in Breast Tumor Classification. *International Journal of Computer Applications*, 116(19):11–15, 2015. doi: 10.5120/20443-2793.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, apr 1998.

LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35288-1. doi: 10.1007/978-3-642-35289-8. URL http://link.springer.com/10.1007/978-3-642-35289-8.

Lee, J. Y. and Nandi, A. K. Extraction of impacting signals using blind deconvolution. *Journal of Sound and Vibration*, 232(5):945–962, 2000. ISSN 0022460X. doi: 10.1006/jsvi.1999.2778.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1-2):314–334, 2014. ISSN 08883270. doi: 10.1016/j.ymssp.2013.06.004. URL http://dx.doi.org/10.1016/j.ymssp.2013.06.004.

Lei, N., An, D., Guo, Y., Su, K., Liu, S., Luo, Z., Yau, S. T., and Gu, X. A Geometric Understanding of Deep Learning. *Engineering*, jan 2020a. ISSN 20958099. doi: 10.1016/j.eng.2019.09.010. URL https://linkinghub.elsevier.com/retrieve/pii/S2095809919302279.

Lei, Y., Lin, J., He, Z., and Zi, Y. Application of an improved kurtogram method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, 25(5):1738–1749, 2011. ISSN 08883270. doi: 10.1016/j.ymssp.2010.12.011. URL http://dx.doi.org/10.1016/j.ymssp.2010.12.011.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., and Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018. ISSN 10961216. doi: 10.1016/j.ymssp.2017.11.016. URL https://doi.org/10.1016/j.ymssp.2017.11.016.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587, 2020b. ISSN 10961216. doi: 10.1016/j.ymssp.2019.106587. URL https://doi.org/10.1016/j.ymssp.2019.106587.

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks. *32nd International Conference on Machine Learning, ICML 2015*, 3:1718–1727, 2015.

Liao, L. and Köttig, F. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1):191–207, 2014. ISSN 00189529. doi: 10.1109/TR.2014.2299152.

Lin, Z., Thekumparampil, K. K., Fanti, G., and Oh, S. InfoGAN-CR: Disentangling Generative Adversarial Networks with Contrastive Regularizers. jun 2019. URL http://arxiv.org/abs/1906.06034.

Liu, H., Zhou, J., Xu, Y., Zheng, Y., Peng, X., and Jiang, W. Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. *Neurocomputing*, 315:412–424, 2018a. ISSN 18728286. doi: 10.1016/j.neucom.2018.07.034. URL https://doi.org/10.1016/j.neucom.2018.07.034.

Liu, R., Yang, B., Zio, E., and Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, 2018b. ISSN 10961216. doi: 10.1016/j.ymssp.2018.02.016. URL https://doi.org/10.1016/j.ymssp.2018.02.016.

Locatello, F., Bauer, S., Lucie, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:7247–7283, 2019.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. nov 2017. URL http://arxiv.org/abs/1711.05101.

Luo, L., Xiong, Y., Liu, Y., and Sun, X. Adaptive gradient methods with dynamic bound of learning rate. In *7th International Conference on Learning Representations, ICLR 2019*, number 2018, pages 1–19, feb 2019. URL http://arxiv.org/abs/1902.09843.

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

Martens, J. New insights and perspectives on the natural gradient method. 2014. ISSN 1542-8818. doi: 10.1128/jmbe.v3.63. URL http://arxiv.org/abs/1412.1193.

Martin, A. Vibration monitoring of machines. *Technical review - Kjær & Brüel*, (1), 1987. ISSN 00072621.

Matsubara, T., Hama, K., Tachibana, R., and Uehara, K. Deep Generative Model using Unregularized Score for Anomaly Detection with Heterogeneous Complexity. 14(8):1–10, 2018. URL http://arxiv.org/abs/1807.05800.

McCulloch, W. and Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 3:115–133, 1943.

McDonald, G. L. and Zhao, Q. Multipoint Optimal Minimum Entropy Deconvolution and Convolution Fix: Application to vibration fault detection. *Mechanical Systems and Signal Processing*, 82: 461–477, 2017. ISSN 10961216. doi: 10.1016/j.ymssp.2016.05.036. URL http://dx.doi.org/10.1016/j.ymssp.2016.05.036.

McFadden, P. D. and Smith, J. D. Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of Sound and Vibration*, 96(1):69–82, 1984. ISSN 10958568. doi: 10.1016/0022-460X(84)90595-9.

McInerny, S. A. and Dai, Y. Basic Vibration Signal Processing for Bearing Fault Detection. 46(601 114):1663–1666, 2003.

Mescheder, L., Nowozin, S., and Geiger, A. The numerics of GANs. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 1826–1836, may 2017. URL http://arxiv.org/abs/1705.10461.

Mescheder, L., Geiger, A., and Nowozin, S. Which Training Methods for GANs do actually Converge? jan 2018. URL http://arxiv.org/abs/1801.04406.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. Unrolled Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, pages 469–477, nov 2016. ISSN 10495258. URL http://arxiv.org/abs/1611.02163.

Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. pages 1–7, 2014. URL http://arxiv.org/abs/1411.1784.

Miyato, T. and Koyama, M. CGANs with projection discriminator. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, feb 2018. URL http://arxiv.org/abs/1802.05957.

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

Munck, E. D. S. and Fyfe, K. R. Computed order tracking applied to vibration analysis of rotating machinery. 19(4):57–58, 1991. doi: 10.7939/R3S17SX8G.

Muthoo, A., Osborne, M. J., and Rubinstein, A. *A Course in Game Theory.*, volume 63. 1996. ISBN 0262650401. doi: 10.2307/2554642.

Nagarajan, V. and Kolter, J. Z. Gradient descent GAN optimization is locally stable. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5586–5596, jun 2017. ISSN 10495258. URL http://arxiv.org/abs/1706.04156.

Niehaus, W. N., Schmidt, S., and Heyns, P. S. NIC Methodology: A probabilistic methodology for improved informative frequency band identification by utilizing the available healthy historical data under time-varying operating conditions. *Journal of Sound and Vibration*, 488:115642, 2020. ISSN 10958568. doi: 10.1016/j.jsv.2020.115642. URL https://doi.org/10.1016/j.jsv.2020.115642.

Odena, A., Olah, C., and Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. oct 2016. URL http://arxiv.org/abs/1610.09585.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. pages 1–60, 2019. URL http://arxiv.org/abs/1912.02762.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32(NeurIPS), 2019. ISSN 10495258.

Perarnau, G., van de Weijer, J., Raducanu, B., and Álvarez, J. M. Invertible Conditional GANs for image editing. (Figure 1):1–9, nov 2016. URL http://arxiv.org/abs/1611.06355.

Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):1–257, 2019. ISSN 19358245. doi: 10.1561/2200000073.

Qin, Y., Mitra, N., and Wonka, P. How does Lipschitz Regularization Influence GAN Training? 2018. URL http://arxiv.org/abs/1811.09567.

Qiu, H., Lee, J., Lin, J., and Yu, G. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289(4-5): 1066–1090, 2006. ISSN 10958568. doi: 10.1016/j.jsv.2005.03.007.

Qiu, H., Lee, J., Lin, J., Yu, G., and Services (2007), R. T. IMS, University of Cincinnati. Bearing Data Set, NASA Ames Prognostics Data Repository, 2007. URL http://ti.arc.nasa.gov/project/prognostic-data-repository.

Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. pages 1–16, nov 2015. URL http://arxiv.org/abs/1511.06434.

Ramasso, E. Investigating computational geometry for failure prognostics. *Int. Journal on Prognostics and Health Management*, 5:1–18, 2014. ISSN 21532648. URL https://www.phmsociety.org/sites/phmsociety.org/files/phm{_}submission/2014/ijphm{_}14{_}005.pdf.

Ramasso, E., Saxena, A., Ramasso, E., and Abhinav Saxena. Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset To cite this version : HAL Id : hal-01145003 Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset. 2015.

Randall, R. B., Sawalhi, N., and Coats, M. A comparison of methods for separation of deterministic and random signals. *International Journal of Condition Monitoring*, 1(1):11–19, 2011. ISSN 20476426. doi: 10.1784/204764211798089048.

Randall, R. B. and Antoni, J. Rolling element bearing diagnostics-A tutorial. *Mechanical Systems and Signal Processing*, 25(2):485–520, 2011. ISSN 10961216. doi: 10.1016/j.ymssp.2010.07.017.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of Adam and beyond. *ICLR 2018*, pages 1–23, 2018.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative Adversarial Text to Image Synthesis. may 2016. URL http://arxiv.org/abs/1605.05396.

Riedmiller, M. *Rprop - description and implementation details : technical report*. Inst. f. Logik, Komplexitat u. Deduktionssysteme, Karlsruhe SE - 2 Seiten, 1994.

Riedmiller, M. and Braun, H. Direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *1993 IEEE International Conference on Neural Networks*, TA - TT -, pages 586–591. 1993. ISBN 0780312007. doi: 10.1109/ICNN.1993.298623LK-https://UnivofPretoria.on.worldcat.org/oclc/5872191146.

Rosca, M., Lakshminarayanan, B., Warde-Farley, D., and Mohamed, S. Variational Approaches for Auto-Encoding Generative Adversarial Networks. 2017. URL http://arxiv.org/abs/1706.04987.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. *Advances in Neural Information Processing Systems*, 2017-Decem (2):2019–2029, 2017. ISSN 10495258.

Ruder, S. An overview of gradient descent optimization algorithms. pages 1–14, 2017. URL http://arxiv.org/abs/1609.04747.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. In: Rumelhart D E, McClelland J L et al. (eds.) Parallel Distributed Processing:

Explorations in the Microstructure of Cognition. *MIT Press, Cambridge, MA*, 1(V):318–362, 1986. URL https://apps.dtic.mil/docs/citations/ADA164453.

Sadowsky, J. The Continuous Wavelet Transform: A tool for Signal Investigation and Understanding. *Wavelets: An Elementary Treatment of Theory and Applications*, 15(4):27–48, 1994. doi: 10.1142/9789814503747_0003.

Sait, A. S. Rotating Machinery, Structural Health Monitoring, Shock and Vibration, Volume 5. (March 2011), 2011. doi: 10.1007/978-1-4419-9428-8. URL http://link.springer.com/10.1007/978-1-4419-9428-8.

Salakhutdinov, R. and Hinton, G. Deep Boltzmann machines. *Journal of Machine Learning Research*, 5(3):448–455, 2009. ISSN 15324435.

San Martin, G., López Droguett, E., Meruane, V., and das Chagas Moura, M. Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis. *Structural Health Monitoring*, 18(4):1092–1128, 2019. ISSN 17413168. doi: 10.1177/1475921718788299.

Saruhan, H., Sandemir, S., Çiçek, A., and Uygur, I. Vibration Analysis of Rolling Element Bearings Defects. *Journal of Applied Research and Technology*, 12(3):384–395, jun 2014. ISSN 16656423. doi: 10.1016/S1665-6423(14)71620-7. URL http://www.jart.icat.unam.mx/index.php/jart/article/view/201.

Sawalhi, N., Randall, R. B., and Endo, H. The enhancement of fault detection and diagnosis in rolling element bearings using minimum entropy deconvolution combined with spectral kurtosis. *Mechanical Systems and Signal Processing*, 21(6):2616–2633, 2007. ISSN 08883270. doi: 10.1016/j.ymssp.2006.12.002.

Sawalhi, N. The application of spectral kurtosis to bearing diagnostics. *Acoustics - Conference*, (November):393–398, 2004. URL http://www.acoustics.asn.au/conference{_}proceedings/AAS2004/ACOUSTIC/PDF/AUTHOR/AC040115.PDF.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10265 LNCS:146–147, 2017. ISSN 16113349. doi: 10.1007/978-3-319-59050-9_12.

Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. ISSN 13618423. doi: 10.1016/j.media.2019.01.010. URL https://doi.org/10.1016/j.media.2019.01.010.

Schmidt, S., Heyns, P. S., and de Villiers, J. P. A novelty detection diagnostic methodology for gearboxes operating under fluctuating operating conditions using probabilistic techniques. *Mechanical Systems and Signal Processing*, 100:152–166, feb 2018. ISSN 10961216. doi: 10.1016/j.ymssp.2017.07.032.

Schmidt, S. and Heyns, P. S. An open set recognition methodology utilising discrepancy analysis for gear diagnostics under varying operating conditions, 2019. ISSN 10961216.

Schmidt, S. and Heyns, P. S. Normalisation of the amplitude modulation caused by time-varying operating conditions for condition monitoring. *Measurement: Journal of the International Measurement Confederation*, 149:106964, 2020. ISSN 02632241. doi: 10.1016/j.measurement.2019.106964. URL https://doi.org/10.1016/j.measurement.2019.106964.

Schmidt, S., Stephan Heyns, P., and de Villiers, J. Discrepancy signal processing techniques for gearbox condition monitoring applications. *Proceedings of the First World Congress on Condition Monitoring (WCCM 2017)*, 2017.

Schmidt, S., Heyns, P. S., and Gryllias, K. C. A discrepancy analysis methodology for rolling element bearing diagnostics under variable speed conditions. *Mechanical Systems and Signal Processing*, 116:40–61, feb 2019a. ISSN 08883270. doi: 10.1016/j.ymssp.2018.06.026. URL https://linkinghub.elsevier.com/retrieve/pii/S0888327018303583.

Schmidt, S., Heyns, P. S., and Gryllias, K. C. A pre-processing methodology to enhance novel information for rotating machine diagnostics. *Mechanical Systems and Signal Processing*, 124: 541–561, 2019b. ISSN 10961216. doi: 10.1016/j.ymssp.2019.02.005. URL https://doi.org/10.1016/j.ymssp.2019.02.005.

Sharma, V. and Parey, A. A Review of Gear Fault Diagnosis Using Various Condition Indicators. *Procedia Engineering*, 144:253–263, 2016. ISSN 18777058. doi: 10.1016/j.proeng.2016.05.131. URL http://dx.doi.org/10.1016/j.proeng.2016.05.131.

Sheng, S. Wind Turbine Gearbox Reliability Database, Condition Monitoring, and Operation and Maintenance Research Update. Technical report, National Renewable Energy Laboratory, Colorado, 2016.

Shigley, J. E. and Mischke, C. R. *Mechanical Engineering Design: In SI Units*. McGraw-Hill, New York, NY, 10 edition, 2005.

Siddique, M. N. and Tokhi, M. O. Training neural networks: Backpropagation vs genetic algorithms. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 2673–2678, 2001.

Snyman, J. A. A new and dynamic method for unconstrained minimization. *Applied Mathematical Modelling*, 6(6):449–462, 1982. ISSN 0307904X. doi: 10.1016/S0307-904X(82)80007-3.

Snyman, J. A. and Wilke, D. N. *Practical mathematical optimization: Basic optimization theory and gradient-based algorithms*, volume 133. Springer, Cham, Switzerland SE - xxvi, 372 pages : illustrations ; 24 cm., second edi edition, 2018. ISBN 3319775855 9783319775852.

Sohn, K., Yan, X., and Lee, H. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 2015-Janua, pages 3483–3491, 2015.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised MAP Inference for Image Super-resolution. pages 1–17, oct 2016. URL http://arxiv.org/abs/1610.04490.

Stander, C. J. and Heyns, P. S. Transmission path phase compensation for gear monitoring under fluctuating load conditions. *Mechanical Systems and Signal Processing*, 20(7):1511–1522, 2006. ISSN 08883270. doi: 10.1016/j.ymssp.2005.05.009.

Tenenbaum, J. B. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, dec 2000. ISSN 0036807. doi: 10.1126/science.290.5500.2319. URL https://www.sciencemag.org/lookup/doi/10.1126/science.290.5500.2319.

Tipping, M. E. and Bishop, C. M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 61(3):611–622, 1999. ISSN 13697412. doi: 10.1111/1467-9868.00196.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–20, feb 2018. URL http://arxiv.org/abs/1711.01558http://arxiv.org/abs/1902.09323.

Tse, P. W. and Wang, D. The design of a new sparsogram for fast bearing fault diagnosis: Part 1 of the two related manuscripts that have a joint title as "two automatic vibration-based fault diagnostic methods using the novel sparsity measurement - Parts 1 and 2". *Mechanical Systems and Signal Processing*, 40(2):499–519, 2013a. ISSN 08883270. doi: 10.1016/j.ymssp.2013.05.024. URL http://dx.doi.org/10.1016/j.ymssp.2013.05.024.

Tse, P. W. and Wang, D. The automatic selection of an optimal wavelet filter and its enhancement by the new sparsogram for bearing fault detection: Part 2 of the two related manuscripts that have a joint title as "two automatic vibration-based fault diagnostic methods using the . *Mechanical Systems and Signal Processing*, 40(2):520–544, 2013b. ISSN 08883270. doi: 10.1016/j.ymssp.2013.05.018. URL http://dx.doi.org/10.1016/j.ymssp.2013.05.018.

Udell, M., Horn, C., Zadeh, R., and Boyd, S. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016. ISSN 19358245. doi: 10.1561/2200000055.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. Generative Adversarial Nets from a Density Ratio Estimation Perspective. 2016. URL http://arxiv.org/abs/1610.02920.

Urbanek, J., Barszcz, T., Strączkiewicz, M., and Jablonski, A. Normalization of vibration signals generated under highly varying speed and load with application to signal separation. *Mechanical Systems and Signal Processing*, 82:13–31, 2017. ISSN 10961216. doi: 10.1016/j.ymssp.2016.04.017.

Van Den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Van Den Driessche, G., Lockhart, E., Cobo, L. C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. Parallel WaveNet: Fast high-fidelity speech synthesis. In *35th International Conference on Machine Learning, ICML 2018*, volume 9, pages 6270–6278, 2018. ISBN 9781510867963.

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

Večeř, P., Kreidl, M., and Šmíd, R. Condition Indicators for Gearbox Condition Monitoring Systems - Večeř, Kreidl, Šmíd - 2005.pdf. 45(6):35–43, 2005.

Vu, H. S., Ueta, D., Hashimoto, K., Maeno, K., Pranata, S., and Shen, S. M. Anomaly Detection with Adversarial Dual Autoencoders. pages 1–12, 2019. URL http://arxiv.org/abs/1902.06924.

Wang, D. and Tsui, K. L. Theoretical investigation of the upper and lower bounds of a generalized dimensionless bearing health indicator. *Mechanical Systems and Signal Processing*, 98:890–901, 2018. ISSN 10961216. doi: 10.1016/j.ymssp.2017.05.040. URL http://dx.doi.org/10.1016/j.ymssp.2017.05.040.

Wang, D., Tse, P. W., and Tsui, K. L. An enhanced Kurtogram method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing*, 35(1-2):176–199, 2013. ISSN 08883270. doi: 10.1016/j.ymssp.2012.10.003. URL http://dx.doi.org/10.1016/j.ymssp.2012.10.003.

Wang, P., Ananya, Yan, R., and Gao, R. X. Virtualization and deep recognition for system fault classification. *Journal of Manufacturing Systems*, 44(April):310–316, 2017. ISSN 02786125. doi: 10.1016/j.jmsy.2017.04.012. URL http://dx.doi.org/10.1016/j.jmsy.2017.04.012.

Whitley, D. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994. ISSN 09603174. doi: 10.1007/BF00175354.

Wiggins, R. A. Minimum entropy deconvolution. *Geoexploration*, 16(1-2):21–35, apr 1978. ISSN 00167142. doi: 10.1016/0016-7142(78)90005-4. URL https://linkinghub.elsevier.com/retrieve/pii/0016714278900054.

Wilke, D. N., Kok, S., and Groenwold, A. Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimisation: Notes on Diversity. *International Journal for Numerical Methods in Engineering*, (November):2006, 2006. ISSN 0743-1619. doi: 10.1002/nme.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. pages 1–6, aug 2017. URL http://arxiv.org/abs/1708.07747.

Yann LeCun, Y. B. Convolutional networks for images, speech, and time series.MIT Press, Cambridge. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. Efficient GAN-Based Anomaly Detection. feb 2018. URL http://arxiv.org/abs/1802.06222.

Zhang, S., Zhang, S., Wang, B., and Habetler, T. G. Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review. *IEEE Access*, 8(January):29857–29881, 2020. doi: 10.1109/access.2020.2972859.

Zhang, W., Li, C., Peng, G., Chen, Y., and Zhang, Z. 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. ISSN 10961216. doi: 10.1016/j.ymssp.2017.06.022. URL http://dx.doi.org/10.1016/j.ymssp.2017.06.022.

Zhang, Z., Zhang, R., Li, Z., Bengio, Y., and Paull, L. Perceptual generative autoencoders. *Deep Generative Models for Highly Structured Data, DGS@ICLR 2019 Workshop*, 2019.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., and Gao, R. X. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019. ISSN 10961216. doi: 10.1016/j.ymssp.2018.05.050. URL https://doi.org/10.1016/j.ymssp.2018.05.050.

Zhao, S., Song, J., and Ermon, S. InfoVAE: Information Maximizing Variational Autoencoders. 2017. URL http://arxiv.org/abs/1706.02262.

Zhou, Y., Gu, K., and Huang, T. Unsupervised Representation Adversarial Learning Network: From Reconstruction to Generation. *Proceedings of the International Joint Conference on Neural Networks*, 2019-July, 2019. doi: 10.1109/IJCNN.2019.8852395.

Zhu, J., Nostrand, T., Spiegel, C., and Morton, B. Survey of condition indicators for condition monitoring systems. *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014*, pages 635–647, 2014.

Zimroz, R., Bartelmus, W., Barszcz, T., and Urbanek, J. Diagnostics of bearings in presence of strong operating conditions non-stationarity - A procedure of load-dependent features processing with application to wind turbine bearings, 2014. ISSN 08883270.

Appendix A Machine Learning

A.1 Chapter Abstract

The purpose of this section is to provide a literature background into machine learning practices and their associated objective functions, training schemes and data processing techniques.

A.2 Introduction

Machine learning is a topical region of research in present society. Machine learning and the use of ANNs originated from the research into finding mathematical representations of the neuron processing in the human brain. This initial research was conducted by McCulloch and Pitts (1943), however, at the time the technological capabilities were insufficient to realise the potential of the technology. However, with the rise in computational power, ANNs have since become a viable technology. *Supervised learning* is when one has access to both input data and their associated labels, denoted as $\mathbf{x}$ and $\mathbf{t}$ respectively. In the supervised framework one aims to learn the conditional distribution $p(\mathbf{t}|\mathbf{x})$. Here the label types can formulate the classification or regression frameworks, based on whether the labels are discrete or continuous. *Unsupervised learning* is when one does not have any data labels and tries to model the data distribution $p(\mathbf{x})$ (Hoang and Kang, 2019, Goodfellow et al., 2017).

A.3 Supervised Learning

There are two approaches to network training that seek to determine the optimal weights of a neural network, the first is the maximum likelihood learning approach while the second is the Kullback-Leibler (KL) divergence approach. The formulation of the former is more intuitive and will be explored in this literature review. However, any interested reader should look at the work of Martens (2014), who presents an interesting comparison between maximum likelihood training and the KL divergence.

Consider now the case where one has access to some data $\mathbf{x}$ and an associated target variable $\mathbf{t}$, the goal is to use this data to learn a function that can correctly predict a target label given an input. However, we need a way to capture the prediction uncertainty and to do this one can utilise probabilistic techniques. The mathematical formulation of a machine learning problem is as follows, given input features $\mathbf{x}$, obtain a function $f(\mathbf{x})$ that can represent a target $\mathbf{t}$, where the function $f(\mathbf{x})$ is parametrised by weights $\mathbf{w}$. One must then optimise this function such that its representation is optimal.

In a supervised learning probabilistic framework, given a random vector $\mathbf{x}$ and labels $\mathbf{t}$, the objective is to learn the conditional distribution $p(\mathbf{t}|\mathbf{x})$ of this data. We now make the decision to parametrise the conditional distribution with the parametric function $f(\mathbf{x})$, which varies based on the assumed distribution form. For a batch of N samples of $(\mathbf{x},\mathbf{t})$, under the assumption that the data is sampled

independently, we can construct a likelihood function in the form

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \prod_{i=1}^{N} p(\mathbf{t}_i|\mathbf{x}_i,\mathbf{w}_i). \tag{A.1}$$

It is now mathematically convenient to take the logarithm of this distribution, as gradients for products of probability values often tend to zero quickly. The objective now is to maximise the likelihood given a batch of samples, which requires the parametric function to be updated. It is also convenient to take the negative of this log-likelihood function to produce the objective function as

$$\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{N} \log p(\mathbf{t}_i|\mathbf{x}_i,\mathbf{w}_i), \tag{A.2}$$

which emits a method for obtaining an optimal fit to the condition distribution. In Martens (2014), it is shown that the KL divergence approach with a parametric conditional distribution $q(\mathbf{t}|\mathbf{x})$ results in the same objective function.

A.3.1 Regression

For regression, it is assumed that the target variable $\mathbf{t}$ is continuous $\mathbf{t} \in \mathbb{R}$, where it is assumed that a Gaussian distribution can be used for the approximate distribution that is parametrised with a neural network function $f(\mathbf{x},\mathbf{w})$, where this function is parametrised by unknown weights, as mentioned previously. The form of this distribution is given as

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \mathcal{N}(\mathbf{t}|f(\mathbf{x},\mathbf{w}),\beta^{-1}\mathbf{I}), \tag{A.3}$$

where β is the Gaussian noise precision. Consider the case now where a single target variable t is obtained for samples $\mathbf{x}$ that can be used as a training set. This reduces the Gaussian distribution from its multivariate form to its univariate form. This assumption is not made for complexity purposes as the difference is trivial but rather for illustration purposes. By using Equation (A.2), the negative logarithm over N training samples results in

$$-\sum_{\mathbf{x}} \log p(t|\mathbf{x},\mathbf{w}) = \frac{\beta}{2}\sum_{i=1}^{N}(f(\mathbf{x}_i,\mathbf{w})-t_i)^2 - \frac{N}{2}\log\beta + \frac{N}{2}\log(2\pi), \tag{A.4}$$

By neglecting the terms that are not a direct function of the weights, one can obtain an loss function that can be used to update the weights of the parametric function $f(\mathbf{x})$ (Bishop, 2006). The loss function in this case is

$$\mathcal{L}_i = \frac{1}{2}(f(\mathbf{x}_i,\mathbf{w})-t_i)^2. \tag{A.5}$$

For the multivariate case in which one has multiple target variables, the loss function can be developed to be in the form

$$\mathcal{L}_i = \frac{1}{2}\|f(\mathbf{x}_i,\mathbf{w})-\mathbf{t}_i\|_2^2, \tag{A.6}$$

where $\|\cdot\|_2$ is the L_2 norm. This loss function is known in literature as the squared error loss function. One can also take the mean of this to obtain the mean-squared error (MSE) (Bishop, 2006).

A.3.2 Classification

In a classification framework, one assumes that the target variable $\mathbf{t}$ is discrete $\mathbf{t} \in \mathbb{Z}$, where each dimension of the target variable refers to a specific class label in k classes. There are three dominant cases of classification, namely, binary classification, multi-class classification and multi-class binary classification, often referred to as multi-label classification in literature (Bishop, 2006).

A.3.2.1 Binary Classification

Binary classification, or two-class classification, assumes the conditional distribution to be a Bernoulli distribution parametrised as $p(t|\mathbf{u})$, where $t \in [0,1]$ and $\mathbf{u} = f(\mathbf{x},\mathbf{w})$ (Bishop, 2006). The parametrized distribution can be given as

$$p(t_i|\mathbf{x}_i,\mathbf{w}) = f(\mathbf{x}_i,\mathbf{w})^{t_i}(1-f(\mathbf{x}_i,\mathbf{w}))^{1-t_i}. \tag{A.7}$$

If one takes the negative logarithm of this distribution, as required by Equation (A.2), the result is

$$\mathcal{L}_i = -\left[t_i \log f(\mathbf{x}_i, \mathbf{w})) + (1 - t_i) \log(1 - f(\mathbf{x}_i, \mathbf{w}))\right], \tag{A.8}$$

which can be summed and normalised by the batch size in a neural network batch training setting, under the assumption that the data is *i.i.d.* This objective function is often known as the binary or sigmoid cross-entropy loss.

A.3.2.2 Multi-Class Classification

If one makes the assumption that the class labels are exclusive (the multi-class, single-label case) for a given input variable $\mathbf{x}$, then one can use the categorical distribution otherwise known as the generalised Bernoulli distribution, given as

$$p(\mathbf{t}_i|\mathbf{x}_i, \mathbf{w}) = \prod_{k=1}^{K} f_k(\mathbf{x}_i, \mathbf{w})^{t_{ik}}, \tag{A.9}$$

where in this case the target is a vector, where the vector is required to satisfy $\sum_{k=1}^{K} t_k = 1$. This is easily achieved with a softmax activation function on the output (Bishop, 2006). If one takes the negative logarithm, as required by Equation (A.2), the resulting loss function can be given as

$$\mathcal{L}_i = -\sum_{k=1}^{K} t_{ik} \log f_k(\mathbf{x}_i, \mathbf{w}), \tag{A.10}$$

which can be summed and normalised by the batch size in a neural network batch training setting, under the assumption that the data is *i.i.d.* This objective function is often known as the *softmax cross-entropy* loss.

A.3.2.3 Multi-Label Classification

In the multi-label setting (one input, multiple classes), we assume that there are K binary class labels that are independent. The distribution used in this case is a version of the Bernoulli distribution for multiple independent classes

$$p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}) = \prod_{k} f_k(\mathbf{x}_n, \mathbf{w})^{t_{nk}} (1 - f_k(\mathbf{x}_n, \mathbf{w}))^{1 - t_{nk}}, \tag{A.11}$$

where the target is now a vector and each element represents the probability of a given class. If one takes the negative logarithm of this distribution, as required by Equation (A.2), the result is

$$\mathcal{L}_i = -\sum_{k=1}^{K} \left[t_{ik} \log f_k(\mathbf{x}_i, \mathbf{w})) + (1 - t_{ik}) \log(1 - f_k(\mathbf{x}_i, \mathbf{w}))\right], \tag{A.12}$$

which can be summed and normalised by the batch size in a neural network batch training setting, under the assumption that the data is *i.i.d.*

A.4 Network Architecture

The motivation for the use of a neural network often stems from the original use of basis functions to capture any non-linearity present in the data. The issue with basis functions, however, is that to model higher-order non-linearity, dimensionality becomes an issue (Duda et al., 2001). A neural network is one that attempts to mimic the biological processes involved in the human brain, with a multilayer perceptron being the main network structure with the best practical value (Bishop, 2006). The simplest form of a neural network is the feed-forward neural network (FFNN) and an example is given in Figure A.1.

There are a few notational elements used in defining the basic process of an ANN which are given as follows: the parameters ω_{ji} are the network weights, ω_{j0} is the bias weights, a_j is known as the nodal activation and $h(\cdot)$ is an activation function. A node is considered to be a point where all variable

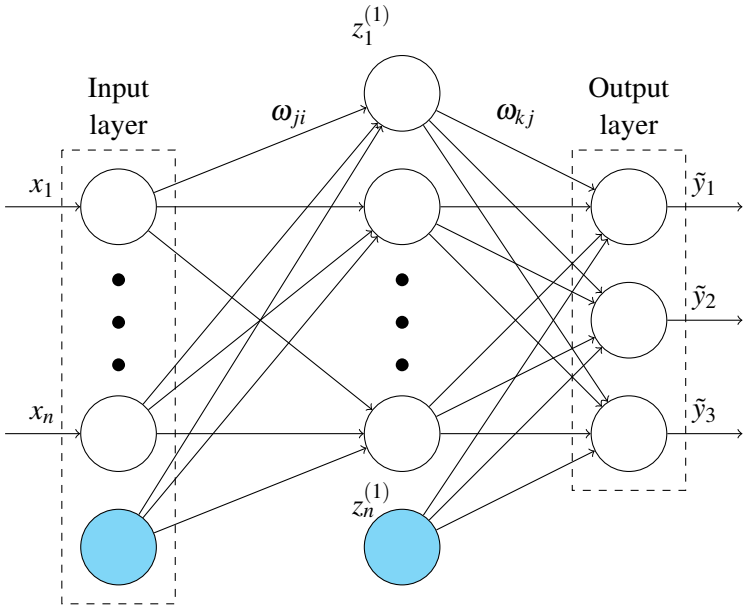


Figure A.1. A simple two-layer neural network. Bias units are denoted using blue shaded circles.

inputs to the node are combined to produce a nodal output. If we generate M linear combinations of the input features $x_1, x_2, \cdots x_m$ the result is

$$a_j = \sum_{i=1}^{M} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}, \tag{A.13}$$

where the superscript (1) indicates that it is the first layer of the neural network. The activation a_j is then passed through the activation function $h(\cdot)$ to give the variable z_j (Bishop, 2006). The variable z_j can then be moved through the second layer of the neural network, to the output nodes, if the network is a two-layer neural network such as the one shown in Figure A.1, where the activation is now

$$a_k = \sum_{j=1}^{J} \omega_{kj}^{(2)} z_j + \omega_{k0}^{(2)}, \tag{A.14}$$

where one can have an arbitrary number of output nodes for the network and the number is often problem-specific. The activation a_k must be passed through a final activation function to give the outputs of the network y_k. However, this final layer activation is problem-specific. The final form of a simple two-layer neural network is

$$y_k = h\left(\sum_{j=1}^{M} \omega_{kj}^{(2)} h\left(\sum_{i=1}^{N} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}\right) + \omega_{k0}^{(2)}\right). \tag{A.15}$$

The entire process followed here is known as forward propagation, as the feature vector $\mathbf{x}$ is forward multiplied through the network. If more hidden units are required, the process can be repeated as necessary (Bishop, 2006). The activation function $h(\cdot)$ can be any function, however, there are a few popular ones typically used in practice. An activation function is a scalar-to-scalar function that typically converts any input to a bounded output range (Bishop, 2006). It is preferable that the activation function emits an obtainable and continuous derivative and that it induces non-linearity in some way (Duda et al., 2001).

The linear activation function is the first available type, however, this activation function is not known to be useful for complex problems as a network with only linear activation can only learn a linear

representation from input to output. Non-linear activation functions are useful for complex problems, with three main activation functions being popular. These three functions are the sigmoid activation function, the hyperbolic tangent (tanh) activation function and the ReLU activation function. Due to the saturation of the sigmoid activation function that was once considered beneficial, ReLU is often preferred in literature as it does not make gradient descent techniques harder to implement due to potentially small gradients (Bishop, 2006). One drawback of the ReLU activation function is that due to the nature in which it applies activation, one cannot learn using typical back-propagation techniques if the activation on a node is zero. There are alternate forms of the ReLU activation function, such as the leaky ReLU, parametric ReLU, Softplus and absolute value ReLU which are all alterations of the generalised ReLU form

$$h(a_i) = \max(0, a_i) + \alpha_i \min(0, a_i), \tag{A.16}$$

where α_i is a slope parameter. An analytical form of ReLU exists through the softplus activation function (Goodfellow et al., 2017). Finally, an output activation unit that one can use for multi-class, single-label classification is that of the softmax activation function (Bishop, 2006).

An alternative form of standard FFNNs is that of a Convolutional Neural Network (CNN). To understand why one might use a CNN, one needs to be well acquainted with the property of invariances. For machine learning, the learnt parametric function should be invariant (to remain unchanged) to an input feature vector that has undergone some form of transformation. For signal processing techniques this means that, at least in the case of rotating machinery, a signal that undergoes time-domain translation should not be seen as a new signal to which the network has no understanding of what information it holds. If a network is invariant to this transformation and other types, then it becomes possible to learn the periodic nature of a vibration signal under stationary or even non-stationary conditions (Yann LeCun, 1995).

Bishop (2006) discusses these techniques in detail, however, the most useful invariance formulation is the CNN, proposed in Yann LeCun (1995). A convolutional layer in a neural network typically consists of three operating principles: local receptive fields, weight sharing and sub-sampling, where the latter is optional. A CNN implements the mathematical operation known as convolution, however sometimes in practice, one might implement the cross-correlation operation. These two operations appear to be very similar mathematically but have significant differences in implementation, with the convolution operation using the reflection of the filter. A convolutional unit in a network uses multiple filters that are convolved over an input to generate an output (Bouvrie, 2006). The mathematical notation of a general convolution operation is given as

$$\mathbf{z}_j^{(l)} = h\left(\sum_{i \in D_j} \mathbf{z}_i^{(l-1)} * \mathbf{w}_{ij}^l + b_j\right), \tag{A.17}$$

where l is the layer number, $*$ signifies a convolution operation, D_j refers to multiple inputs, the weights $\mathbf{w}_{ij}^l$ are what is known as a feature map of a filter and b_j is the bias that is added to a specific output. Typically, one will convolve i filters over the i input vectors or matrices and then repeat this process tor produce j outputs (Bouvrie, 2006). Thus there will be $i \times j$ filters in a filter layer, where each filter can be organised into banks and each bank size is equal to the number of inputs $\mathbf{z}_i^{l-1}$. A filter bank is then convolved and summed to produce the j^{th} output. One can then also incorporate a sub-sampling layer, such as average sampling or max sampling to reduce the dimensionality of the system. Sub-sampling also helps ensure that the filtered elements that contain information are being utilised in the next layer, as it might be the case that a filtered component carries no information and thus to continually pull its contribution is unnecessary (Wang et al., 2017).

A.4.1 Data Pre-processing

As is often required for most machine learning problems, one must pre-process the data before the data is used for any form of feature selection or classification. The reason for this is that data features are often obtained from a variety of different sources and will have noticeably different feature ranges. Feature normalisation is used to ensure that each feature is on an equivalent range such that the network is not immediately biased to a certain feature range due to magnitude. There are two main types of scaling often employed in literature, namely *min-max* normalisation and *z-score* normalisation. These two types are given as

$$\tilde{x}_i = \frac{[x_i - \min(x_i)][\max(x_{new}) - \min(x_{new})]}{\max(x_i) - \min(x_i)} + \min(x_{new}), \tag{A.18}$$

$$\tilde{x}_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{A.19}$$

where the former is the general form of *min-max* normalisation, and the latter is *z-score* normalisation. Often, literature performs unit scaling where a feature range is shifted to $[0,1]$. Equation (A.19) is the form used for *z-score* normalization, where the mean μ_i and standard deviation σ_i of feature i is used to shift the feature domain to one that has a zero mean and unit variance (Kumar Singh et al., 2015). For the case of vibration signals, *z-score* normalization is preferred as it retains the natural structure of a vibration signal data and one cannot say with certainty that the data range will never contain outliers.

A.5 Network Optimisation

To determine the optimal network parameters $\omega_{ji}^{(1)}$ and $\omega_{kj}^{(2)}$ induces an interesting analysis, as there are many weights. Before this formulation is presented, one needs to define minimisation. Minimisation is the process whereby an objective function, which is a function of the certain tunable parameters, is minimised such that an optimal parameter set can be obtained. For machine learning, this process may allow the network to accurately predict the representation of the target variable given any new input feature vector. This minimization is performed in the weight space of the objective function, where the aim is to determine the point where the gradient of the error function indicates a local minimum, given by

$$\nabla E(\mathbf{w}) = 0. \tag{A.20}$$

Numerous techniques attempt to find the minimum point within the weight space, with the most common and perhaps fundamental approach being gradient descent. Gradient descent attempts to find a local minimum, as to truly prove that your function is at a global minimum is highly complex and often not feasible for an iterative numerical method (Bishop, 2006, Snyman and Wilke, 2018). Gradient descent is where one assumes an initial starting vector $\mathbf{w}_0$ and iteratively takes steps, given as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \Delta\mathbf{w}^{(t)}, \tag{A.21}$$

where t is known as the iteration step counter. The gradient is used as an indication of where to move in the weight space, however, one does typically not take a unit step from any given gradient, as if the weight space is highly non-linear and large steps make it easy to move past the minimum. Instead, one uses a learning rate η that scales the iterative weight update process in the direction of the negative gradient, given as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla E(\mathbf{w})^{(t)}. \tag{A.22}$$

In machine learning, there are different approaches to the evaluation of the gradient. If one utilises the entire training set, then this is known as a batch method and the descent method is typically called batch gradient descent or steepest descent. An alternative approach is known as on-line gradient descent which can be seen as a sequential gradient descent technique. In this variation, an update for the weights is made for each data point in the training set. One can also utilise mini-batch sampling

where the batch size is not equal to the training set size N, which can be seen as a hybrid technique. The advantages to the on-line or hybrid gradient descent techniques are that they can handle data redundancy, reduce computational effort and can circumvent local minima induced in steepest descent techniques (Bishop, 2006).

Back-propagation is a technique that aims to determine the error gradient, through a clever implementation of the chain rule and gradient descent. There are two stages typically used in back-propagation, namely, the evaluation and determination of the error function gradient with respect to the weights are found. Gradient descent is then applied iteratively to determine the optimal weights for the network. In the first stage, an error is back-propagated through the application of chain rule through the network to ascertain the correct weight gradients and then in the second stage weight adjustments are made to ensure that the objective function is minimised (Bishop, 2006, LeCun et al., 2012).

Back-propagation, however, is the simplest form of gradient descent available and by no means is it the only method. Multiple alternative gradient descent methodologies have been proposed, such as the well-known Resilient Propagation (RPROP) method which operates by making weight space movements based on gradient sign analysis (Riedmiller and Braun, 1993, Riedmiller, 1994). The leapfrog method is a numerical optimisation technique inspired by the principles governing particle energy, proposed by Snyman (1982) and detailed in Snyman and Wilke (2018). Back-propagation with momentum is another famous method, which makes slight alterations by incorporating a previous iteration gradient as denoted in Duda et al. (2001) . Finally, the adaptive moment estimation method (Adam) which was proposed by Kingma and Ba (2014) is one of the more popular deep learning techniques of late. Ruder (2017) provides a succinct summary of the more recent gradient descent techniques, with it being noted that Adam is the recommended gradient descent technique for deep learning applications. The Adam algorithm is a local per weight adaptive gradient-based optimisation algorithm which aids convergence (Kingma and Ba, 2014).

Since Adam's proposal, there have been many suggested alternatives to help improve training. These variants include, but are not limited to, AMSGrad proposed by Reddi et al. (2018), DiffGrad proposed by Dubey et al. (2019), AdaMod and Adabound proposed by Ding et al. (2019), with its extension being AMSbound proposed by Luo et al. (2019). A preferred alternative to Adam is that of AdamW, which was formulated by Loshchilov and Hutter (2017) to utilise the proven successes of weight decay, and by equivalence, L_2 regularisation, in stochastic gradient descent approaches. An analysis and implementation description of Adam and AdamW is given in Appendix B.

However, gradient-based methods are not the only known methods used to optimise neural networks, with heuristic optimisation also being used. These methods typically do not utilise gradient information but rather use function evaluations of the loss function only. Well-known techniques are as follows: Genetic Evolutionary Algorithms as detailed in Siddique and Tokhi (2001), Whitley (1994) and Koehn (1994), Particle Swarm Optimisation as detailed in Wilke et al. (2006) and simulated annealing. In heuristic optimisation, often a population set of N networks are all individually evaluated and ranked using an objective function. The population is used to explore the weight space and iteratively changed to find a suitable minima, which may be global but it is not always guaranteed. These techniques offer a greater range of search in the weight space but often are computationally expensive by a factor of the population size N.

Appendix B Network Optimisation, GAN training schemes and Network Architectures

B.1 Chapter Abstract

The purpose of this section is to present and show the mathematics behind the Adam algorithm and its variant, AdamW. It is also to detail the training schemes of the $DLS-GAN$ and $RY-GAN$ models.

B.2 Adam and AdamW

The basic workings of the Adam method shall now be described (Kingma and Ba, 2014). There are two estimate vectors in Adam, that of the first moment estimate and the second moment estimate

$$\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1) \odot \nabla E(\mathbf{w}_t), \tag{B.1}$$

$$\mathbf{v}_{t+1} = \beta_2 \mathbf{v}_t + (1-\beta_2) \nabla E(\mathbf{w}_t) \odot \nabla E(\mathbf{w}_t). \tag{B.2}$$

The first and second moment are then bias corrected, where their form is now given as

$$\widehat{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1-\beta_1^t}, \tag{B.3}$$

$$\widehat{\mathbf{v}}_{t+1} = \frac{\mathbf{v}_{t+1}}{1-\beta_2^t}. \tag{B.4}$$

It is thus clear that there are two unknown parameters in the Adam algorithm, that of the exponential decay parameters β_1 and β_2. These parameters were recommended by Kingma and Ba (2014) to be initialised as $\beta_1, \beta_2 \in [0,1)$ and are recommended to be set as $\beta_1 = 0.9, \beta_2 = 0.999$. Finally, the Adam update applied recursively to the weights is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\widehat{\mathbf{m}}_{t+1}}{\sqrt{\widehat{\mathbf{v}}_{t+1}} + \varepsilon}, \tag{B.5}$$

where ε is a parameter that is typically set to 10^{-8} for numerical stability in the algorithm, with the purpose of the parameter to ensure that at no point a division by zero occurs. One can immediately note is that there is a clear local gradient normalisation property and a historical gradient property. The parameter β_1 controls the historical gradient emphasis, where the higher it is the more Adam will rely on the previously accumulated gradients as opposed to the current gradient at any time t. It is also clear to see that the gradient normalisation creates a local gradient for each dimension of the objective function space. The adaptation of Adam to AdamW is trivial and can be given as

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \left(\frac{\widehat{\mathbf{m}}_{t+1}}{\sqrt{\widehat{\mathbf{v}}_{t+1}} + \varepsilon} + \lambda \mathbf{w}_t \right), \tag{B.6}$$

where λ is the weight decay parameter. The pseudo-code for the Adam and AdamW method can be found in Algorithm 1.

Algorithm 1 The Adam Algorithm

Require: learning rate η
$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e^{-8}$
Initialise: first moment $\mathbf{m}_0 = \mathbf{0}$, second moment $\mathbf{v}_0 = \mathbf{0}$, t = 0
Compute loss function gradient: $\nabla E(\mathbf{w}_t)$
Compute the first moment update: $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1) \odot \nabla E(\mathbf{w}_t)$
Compute the second moment update: $\mathbf{v}_{t+1} = \beta_2 \mathbf{v}_t + (1-\beta_2) \nabla E(\mathbf{w}_t) \odot \nabla E(\mathbf{w}_t)$
Compute the bias corrected first moment: $\widehat{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1-\beta_1^t}$
Compute the bias corrected second moment: $\widehat{\mathbf{v}}_{t+1} = \frac{\mathbf{v}_{t+1}}{1-\beta_2^t}$
if Adam == True **then**
  Compute Updated weights: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\widehat{\mathbf{m}_{t+1}}}{\sqrt{\widehat{\mathbf{v}_{t+1}}} + \varepsilon}$
else if AdamW == True **then**
  Compute Updated weights: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \left(\frac{\widehat{\mathbf{m}}_{t+1}}{\sqrt{\widehat{\mathbf{v}}_{t+1}} + \varepsilon} + \lambda \mathbf{w}_t \right)$
end if
Store weights in $\mathbf{w}_{t_{Store}}$
Store the first moment in $\mathbf{m}_{t_{Store}}$
Store the second moment in $\mathbf{v}_{t_{Store}}$
Update the time parameter $t = t + 1$
return Updated weights

B.3 β-TC-VAE

Due to the issues with initially implementing this *VAE* method, the decision was made to present the β-TC-VAE thoroughly, so that future work can be done more efficiently, if required. Consider now the KL divergence from the VAE loss in Equation 2.22, which was given in mini-batch form. In Chen et al. (2018), they initialise the proof using

$$\frac{1}{N}\sum_{i=1}^{N} KL\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|p_{\boldsymbol{\theta}}(\mathbf{z})\right) = \mathbb{E}_{p(\mathbf{n})}[KL\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|p(\mathbf{z})\right)], \tag{B.7}$$

where $\mathbf{n}$ now makes reference to the entire dataset of training data $\mathbf{x}$. The decomposition can then be represented as follows, where the parametrisation elements $\boldsymbol{\phi}$ were dropped for brevity:

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{n})}[KL(q(\mathbf{z}|\mathbf{n})\|p(\mathbf{z}))] &= \mathbb{E}_{p(\mathbf{n})}[\mathbb{E}_{q(\mathbf{z}|\mathbf{n})}[\log q(\mathbf{z}|\mathbf{n}) - \log q(\mathbf{z}) - \log q(\mathbf{z}) + \log \prod_j q(\mathbf{z}_j) - \log \prod_j q(\mathbf{z}_j)]] \\
&= \mathbb{E}_{q(\mathbf{z},\mathbf{n})}\left[\log \frac{q(\mathbf{z}|\mathbf{n})}{q(\mathbf{z})}\right] + \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{\prod_j q(\mathbf{z}_j)}\right] + \mathbb{E}_{q(\mathbf{z})}\left[\sum_j \log \frac{q(\mathbf{z}_j)}{p(\mathbf{z}_j)}\right] \\
&= \mathbb{E}_{q(\mathbf{z},\mathbf{n})}\left[\frac{q(\mathbf{z}|\mathbf{n})p(\mathbf{n})}{q(\mathbf{z})p(\mathbf{n})}\right] + \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{\prod_j q(\mathbf{z}_j)}\right] + \sum_j \mathbb{E}_{q(\mathbf{z}_j)}\left[\log \frac{q(\mathbf{z}_j)}{p(\mathbf{z}_j)}\right] \\
&= KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n})) + KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j)) + \sum_j KL(q(\mathbf{z}_j||p(\mathbf{z}_j)).
\end{aligned} \tag{B.8}$$

Thus, the decomposed form of the KL divergence can be shown to be

$$\mathbb{E}_{p(\mathbf{n})}[KL(q(\mathbf{z}|\mathbf{n})\|p(\mathbf{z}))] = \underbrace{KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n}))}_{\text{Index-Code MI}} + \underbrace{KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j))}_{\text{Total Correlation}} + \underbrace{\sum_j KL(q(\mathbf{z}_j||p(\mathbf{z}_j))}_{\text{Dimension-wise KL}}, \tag{B.9}$$

where $\mathbf{z}_j$ is used to refer to the j^{th} latent variable (Chen et al., 2018). As noted in Equation B.9, there are three elements which can be referred to as the index code Mutual Information (MI), the Total Correlation (TC) and the dimension-wise KL divergence. The intuition between these three elements are: the index code MI can aid in enabling compact and disentangled latent space representations, the total correlation term can aid in finding independent latent factors in the data distribution and the dimension-wise KL divergence ensures that the latent dimensions do not deviate from the prior distribution. Chen et al. (2018) then argue that the existence of the TC term in the KL divergence is why VAEs can learn disentangled latent representations and give their objective function as

$$\mathcal{L}_{\beta\text{-}TC} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{n})p(\mathbf{n})}[\log p(\mathbf{n}|\mathbf{z})] + \alpha KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n})) + \beta KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j)) + \gamma \sum_j KL(q(\mathbf{z}_j)||p(\mathbf{z}_j)), \tag{B.10}$$

where α, β and γ are weighting parameters, with Chen et al. (2018) stating that one use $\alpha = \gamma = 1$ and modifying β. This is the final objective of the β-TC-VAE. However, there is a clear dependency here on the entire dataset $\mathbf{n}$, which can be problematic when datasets become large in size. One can however see that this form of the objective function cannot be easily implemented, thus what will follow is an expansion of the terms until an entire objective function can be presented. To do this, the author will break down each term individually, for brevity. Consider now the Index-code MI, which, following the proof of the KL divergence expansion, can be given as

$$\begin{aligned} KL(q(\mathbf{z},\mathbf{n})||q(\mathbf{z})p(\mathbf{n})) &= \mathbb{E}_{q(\mathbf{z},\mathbf{n})}\left[\log \frac{q(\mathbf{z}|\mathbf{n})}{q(\mathbf{z})}\right] \\ &= \mathbb{E}_{p(\mathbf{n})q(\mathbf{z}|\mathbf{n})}[\log q(\mathbf{z}|\mathbf{n}) - \log q(\mathbf{z})] \\ &= \mathbb{E}_{p(\mathbf{n})q(\mathbf{z}|\mathbf{n})}[\log q(\mathbf{z}|\mathbf{n})] - \mathbb{E}_{q(\mathbf{z},\mathbf{n})}[\log q(\mathbf{z})]. \end{aligned} \tag{B.11}$$

It is now important to understand that the marginal distribution $q(\mathbf{z})$ is obtained from $\mathbb{E}_{p(\mathbf{n})}[q(\mathbf{z}|\mathbf{n})]$, therefore indicating that the second term in the expansion of the index-code MI can readily be given as

$$\mathbb{E}_{q(\mathbf{z},\mathbf{n})}[\log q(\mathbf{z})] = \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})], \tag{B.12}$$

which is the notation adopted by Chen et al. (2018) in their proofs. This is a subtle but important result, whereby the expectation over the joint distribution $(\mathbf{z},\mathbf{n})$ can be reduced to an expectation over $q(\mathbf{z})$ due to $q(\mathbf{z})$ being the marginalisation of $q(\mathbf{z}|\mathbf{n})$. Consider now the expansion of the TC term, which can be given as

$$KL(q(\mathbf{z})||\prod_j q(\mathbf{z}_j)) = \mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z}) - \sum_j \log q(\mathbf{z}_j)\right]. \tag{B.13}$$

Finally, one can expand the dimension-wise KL term, which can be given, from the proof of the expanded KL, as

$$\sum_j KL(q(\mathbf{z}_j)||p(\mathbf{z}_j)) = \mathbb{E}_{q(\mathbf{z})}\left[\sum_j \log q(\mathbf{z}_j) - \sum_j \log p(\mathbf{z}_j)\right]. \tag{B.14}$$

It is clear to note that the expansion of these three terms all depend have elements on the expectation over $\mathbf{z}$, with a clear dependence of $q(\mathbf{z})$. Chen et al. (2018) proposed that one perform some form of mini-batch sampling, but state that purely computing the Monte Carlo approximation with a mini-batch, as is the case for the standard VAE, may underestimate certain components in the objective function. For the likelihood distribution $p(\mathbf{x}|\mathbf{z})$, using mini-batches is sufficient. However, for $q(\mathbf{z})$, a Monte-Carlo approximation will underestimate the term. Chen et al. (2018) proposes two methods to

estimate $q(\mathbf{z})$, the first being Mini-batch Weighted Sampling (MWS) and the second being Mini-batch Stratified Sampling (MSS). The operational principle of these methods are akin to leave one out cross-validation, whereby MWS acts as a pure cross validation approach and MSS acts summation of MWS for two different batches.

To begin the explanation of these sampling approaches, consider an estimation of $q(\mathbf{z}_n)$ for an sample from a given mini-batch of samples, where this mini-batch is denoted here as $\mathcal{B}_M = \{\mathbf{x}_i, \mathbf{x}_M\}$ and $\mathbf{x}_n \in \mathcal{B}_M$, with $M \leq N$. Let $\mathcal{B}_M$ be samples obtained without replacement from the dataset $\mathbf{n}$, an estimation of $q(\mathbf{z}_n)$ using MWS can be given as

$$q(\mathbf{z}_n) = \left[\frac{1}{NM} \sum_{m=1}^{M} q(\mathbf{z}_n|\mathbf{x}_m) \right], \tag{B.15}$$

where N is the dataset size and M is the batch size. Intuitively, MWS can be described as the likelihood of sample $\mathbf{z}_n$ under all samples in the mini-batch $\mathcal{B}_M$, normalised by the batch size and the dataset size. It is critical to note here that $\mathbf{z}_n$ is a sample obtained from the posterior distribution $q(\mathbf{z}|\mathbf{x}_n)$, which can be obtained using the *re-parametrisation* trick. MSS uses the same principle but can be seen as the sum of MWS for a mini-batch including $\mathbf{x}_n$ and MWS for a dataset not containing $\mathbf{x}_n$. Using MSS, an estimation for $q(\mathbf{z})$ can be given as

$$q(\mathbf{z}_n) = \frac{1}{N} q(\mathbf{z}_n|\mathbf{x}_n) + \frac{1}{M} \sum_{m=1}^{M-1} q(\mathbf{z}_n|\mathbf{x}_m) + \frac{N-M}{NM} q(\mathbf{z}_n|\mathbf{x}_M). \tag{B.16}$$

To then obtain approximations for the entire mini-batch, the expectation is taken over all $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$, for $\mathbf{x} \in \mathcal{B}_M$. Therefore, the final form for MWS and MSS can be denoted as

$$\mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] = \frac{1}{M} \sum_{i=1}^{M} \left[\log \left[\sum_{j=1}^{M} q(\mathbf{z}_i|\mathbf{x}_j) \right] - \log(NM) \right] \tag{B.17}$$

$$\mathbb{E}_{q(\mathbf{z})}\left[\log q(\mathbf{z})\right] = \frac{1}{M} \sum_{i=1}^{M} \left[\log \left[\frac{1}{N} q(\mathbf{z}_i|\mathbf{x}_i) + \frac{1}{M} \sum_{j=1}^{M-1} q(\mathbf{z}_i|\mathbf{x}_j) + \frac{N-M}{NM} q(\mathbf{z}_i|\mathbf{x}_M) \right] \right] \tag{B.18}$$

For the observant reader, one may have noted that this only solved one half of the problem and that the term $\mathbb{E}_{q(\mathbf{z})}\left[\sum_j q(\mathbf{z}_j)\right]$ has not been addressed. Fortunately, to compute this term one can use the same forms of MWS and MSS described above, with the summation over the latent dimensions occurring after the evaluation of $\log q(\mathbf{z}_n)$, rather than before. It is also important that one use the *log_sum_exp* operator to evaluation $\log q(\mathbf{z}_n)$ and $\sum_j \log q_j(\mathbf{z}_n)$, to reduce numerical instabilities obtained when evaluating likelihoods. Chen et al. (2018) stated that although MSS is unbiased, the results obtained did not differ substantially in implementation.

B.4 DLS-GAN and RY-GAN Training Algorithms

The purpose of this section is to present both of the training procedures required for the $DLS-GAN$ and $RY-GAN$. Algorithm 2 contains the DLS-GAN training procedure while Algorithm 3 contains the RY-GAN training procedure. These algorithms are given as the DLS-GAN approach used here differs slightly to that used by Ding and Luo (2019) and due to RY-GAN being proposed as an alternative. For these algorithms, let $\mathcal{H}(\cdot,\cdot)$ refer to the calculation of any cross-entropy loss, for notational simplicity. The MMD loss shall also not be expanded, as its analytical form can be found in Equation 2.57.

Algorithm 2 Model training for the DLS-GAN approach, adapted from Ding and Luo (2019).

Require: Initialise E_ϕ, G_θ, D_χ and D_ω, the number of classes k, continuous latent variable dimensionality D_s and noise latent variable dimensionality D_n, λ, λ_{AE} and $\beta_1 - \beta_6$ parameters.

Require: Training data $\mathbf{x}$, discriminator update iteration count M

while *not converged* **do**
  for $i = 1, \dots M$ **do**
    Sample $\mathbf{c}_s \sim p(\mathbf{c})$, a batch of size N of one-hot encoded vectors
    Sample $\mathbf{s}_s \sim p(\mathbf{s})$, a batch of size N of continuous samples from a isotropic Gaussian distribution
    Sample $\mathbf{n}_s \sim p(\mathbf{n})$, a batch of size N of noise samples from a isotropic Gaussian distribution
    Sample $\mathbf{x}_r \sim p(\mathbf{x})$, a batch of size N from the training data
    $\mathbf{z}_g \leftarrow (\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_s)$, assembled latent prior sample
    $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z}_g)$
    $\mathbf{z}_r \leftarrow \mathbf{c}_r, \mathbf{s}_r, \mathbf{n}_r \leftarrow E_\phi(\mathbf{x}_r)$
    $\varepsilon \sim \mathbb{U}(0,1)$, a sample from a uniform distribution
    $\tilde{\mathbf{n}} \leftarrow \varepsilon \odot \mathbf{n}_s + (1-\varepsilon) \odot \mathbf{n}_r$ ($\odot$ = element-wise product)
    $\nabla D_\chi \leftarrow \frac{1}{N} \nabla_\chi \left[\log D_\chi(\mathbf{x}) - \log(1 - D_\chi(\mathbf{x}_g)) \right]$
    $\nabla D_\omega \leftarrow \frac{1}{N} \nabla_\omega \left[-D_\omega(\mathbf{n}_s) + D_\omega(\mathbf{n}_r) + \lambda [\|\nabla_{\tilde{\mathbf{n}}} D_\omega(\tilde{\mathbf{n}})\|_2 - 1]^2 \right]$
    Update network parameters D_ς and D_ω
  end for
  Sample $\mathbf{c}_s \sim p(\mathbf{c})$, a batch of size N of one-hot encoded vectors
  Sample $\mathbf{s}_s \sim p(\mathbf{s})$, a batch of size N of continuous samples from a isotropic Gaussian distribution
  Sample $\mathbf{n}_s \sim p(\mathbf{n})$, a batch of size N of noise samples from a isotropic Gaussian distribution
  Sample $\mathbf{x}_r \sim p(\mathbf{x})$, a batch of size N from the training data
  $\mathbf{z}_g \leftarrow (\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_s)$, assembled latent prior sample
  $\mathbf{z}_r \leftarrow (\mathbf{c}_r, \mathbf{s}_r, \mathbf{n}_r) \leftarrow E_\phi(\mathbf{x}_r)$
  $\tilde{\mathbf{x}}_r \leftarrow G_\theta(\mathbf{z}_r)$
  $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z}_g)$
  $(\tilde{\mathbf{c}}_s, \tilde{\mathbf{s}}_s, \tilde{\mathbf{n}}_s) \leftarrow E_\phi(\mathbf{x}_g)$
  $\mathbf{z}' \leftarrow (\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_r)$, assemble combined latent representation
  $(\tilde{\mathbf{c}}_{s_2}, \tilde{\mathbf{s}}_{s_2}, \tilde{\mathbf{n}}_r) \leftarrow E_\phi(G_\theta(\mathbf{z}'))$
  $\nabla E_\phi \leftarrow \frac{1}{N} \nabla_\phi \Big[\lambda_{AE} \|\mathbf{x}_r - \tilde{\mathbf{x}}_r\|_2^2 - \beta_1 D_\omega(E_\phi^{\mathbf{n}}(\mathbf{x}_r)) + \beta_2 \|\mathbf{n}_r - \tilde{\mathbf{n}}_r\|_2^2 + \beta_3 \mathcal{H}(\mathbf{c}_s, \tilde{\mathbf{c}}_s)$
  $\quad + \beta_4 \mathcal{H}(\mathbf{c}_s, \tilde{\mathbf{c}}_{s_2}) + \beta_5 \|\mathbf{s}_s - \tilde{\mathbf{s}}_s\|_2^2 + \beta_6 \|\mathbf{s}_s - \tilde{\mathbf{s}}_{s_2}\|_2^2 \Big]$
  $\nabla G_\theta \leftarrow \frac{1}{N} \nabla_\theta \Big[-\log \frac{D_\chi(G_\theta(\mathbf{z}_g))}{1 - D_\chi(G_\theta(\mathbf{z}_g))} + \lambda_{AE} \|\mathbf{x}_r - \tilde{\mathbf{x}}_r\|_2^2 + \beta_2 \|\mathbf{n}_r - \tilde{\mathbf{n}}_r\|_2^2 + \beta_3 \mathcal{H}(\mathbf{c}_s, \tilde{\mathbf{c}}_s)$
  $\quad + \beta_4 \mathcal{H}(\mathbf{c}_s, \tilde{\mathbf{c}}_{s_2}) + \beta_5 \|\mathbf{s}_s - \tilde{\mathbf{s}}_s\|_2^2 + \beta_6 \|\mathbf{s}_s - \tilde{\mathbf{s}}_{s_2}\|_2^2 \Big]$
  Update network parameters E_ϕ and G_θ
end while

Algorithm 3 Model training for the RY-GAN approach.

Require: Initialise E_ϕ, G_θ, D_χ, D_ω and D_ς, the number of classes k, continuous latent variable dimensionality D_s and noise latent variable dimensionality D_n, λ, λ_{AE} and α parameters.

Require: Training data $\mathbf{x}$, discriminator update iteration count M

while *not converged* **do**

  for $i = 1, \dots M$ **do**

    Sample $\mathbf{c}_s \sim p(\mathbf{c})$, a batch of size N of one-hot encoded vectors

    Sample $\mathbf{s}_s \sim p(\mathbf{s})$, a batch of size N of continuous samples from a isotropic Gaussian distribution

    Sample $\mathbf{n}_s \sim p(\mathbf{n})$, a batch of size N of noise samples from a isotropic Gaussian distribution

    Sample $\mathbf{x}_r \sim p(\mathbf{x})$, a batch of size N from the training data

    $\mathbf{z}_g \leftarrow (\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_s)$, assembled latent prior sample

    $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z}_g)$

    $\mathbf{z}_r \leftarrow \mathbf{c}_r, \mathbf{s}_r, \mathbf{n}_r \leftarrow E_\phi(\mathbf{x}_r)$

    $\tilde{\mathbf{x}}_r \leftarrow G_\theta(\mathbf{z}_r)$

    $\varepsilon \sim \mathbb{U}(0,1)$, a sample from a uniform distribution

    $\tilde{\mathbf{n}} \leftarrow \varepsilon \odot \mathbf{n}_s + (1-\varepsilon) \odot \mathbf{n}_r$ ($\odot$ = element-wise product)

    $\nabla D_\chi \leftarrow \frac{1}{N} \nabla_\chi \left[\log D_\chi(\mathbf{x}) - \frac{1}{2} \left(\log(1 - D_\chi(\mathbf{x}_g)) + \log(1 - D_\chi(\tilde{\mathbf{x}}_r)) \right) \right]$

    Sample $\varepsilon \sim \mathbb{N}(\mathbf{0}, 0.3^2)$, white noise to be added to samples seen by D_ς

    $\nabla D_\varsigma \leftarrow \frac{1}{N} \nabla_\varsigma \left[\log D_\varsigma(\mathbf{c}_s) - \log(1 - D_\varsigma(\mathbf{c}_r)) \right]$

    $\nabla D_\omega \leftarrow \frac{1}{N} \nabla_\omega \left[-D_\omega(\mathbf{n}_s) + D_\omega(\mathbf{n}_r) + \lambda [\|\nabla_{\tilde{\mathbf{n}}} D_\omega(\tilde{\mathbf{n}})\|_2 - 1]^2 \right]$

    Update network parameters D_χ, D_ς and D_ω

  end for

  Sample $\mathbf{c}_s \sim p(\mathbf{c})$, a batch of size N of one-hot encoded vectors

  Sample $\mathbf{s}_s \sim p(\mathbf{s})$, a batch of size N of continuous samples from a isotropic Gaussian distribution

  Sample $\mathbf{n}_s \sim p(\mathbf{n})$, a batch of size N of noise samples from a isotropic Gaussian distribution

  Sample $\mathbf{x}_r \sim p(\mathbf{x})$, a batch of size N from the training data

  $\mathbf{z}_g \leftarrow (\mathbf{c}_s, \mathbf{s}_s, \mathbf{n}_s)$, assembled latent prior sample

  $\mathbf{z}_r \leftarrow (\mathbf{c}_r, \mathbf{s}_r, \mathbf{n}_r) \leftarrow E_\phi(\mathbf{x}_r)$

  $\tilde{\mathbf{x}}_r \leftarrow G_\theta(\mathbf{z}_r)$

  $\mathbf{x}_g \leftarrow G_\theta(\mathbf{z}_g)$

  $(\tilde{\mathbf{c}}_s, \tilde{\mathbf{s}}_s, \tilde{\mathbf{n}}_s) \leftarrow E_\phi(\mathbf{x}_g)$

  Sample $\varepsilon \sim \mathbb{N}(\mathbf{0}, 0.3^2)$, white noise to be added to samples seen by D_ς

  $\nabla E_\phi \leftarrow \frac{1}{N} \nabla_\phi \left[\alpha \lambda_{AE} \|\mathbf{x}_r - \tilde{\mathbf{x}}_r\|_2^2 - D_\omega(E_\phi^{\mathbf{n}}(\mathbf{x}_r)) - \log \frac{D_\varsigma(E_\phi^{\mathbf{c}}(\mathbf{x}_r) + \varepsilon)}{1 - D_\varsigma(E_\phi^{\mathbf{c}}(\mathbf{x}_r) + \varepsilon)} + \mathcal{L}_{MMD}(\mathbf{s}_s, \mathbf{s}_r) \right]$

  $\nabla G_\theta \leftarrow \frac{1}{N} \nabla_\theta \left[-\frac{1-\alpha}{2} \left(\log \frac{D_\chi(\mathbf{x}_g)}{1 - D_\chi(\mathbf{x}_g)} + -\log \frac{D_\chi(\tilde{\mathbf{x}}_r))}{1 - D_\chi(\tilde{\mathbf{x}}_r)} \right) + \alpha \lambda_{AE} \|\mathbf{x}_r - \tilde{\mathbf{x}}_r\|_2^2 + \mathcal{H}(\mathbf{c}_s, \tilde{\mathbf{c}}_s) + \|\mathbf{s}_s - \tilde{\mathbf{s}}_s\|_2^2 \right]$

  Update network parameters E_ϕ and G_θ

end while

B.5 Network Architectures and Parameters

In this work, Pytorch was used to optimise and train the models (Paszke et al., 2019). The system used to train and evaluate the models had an Intel i7-8750H processor and a Geforce RTX 960 graphics card. The decision was made to generalise the network architecture design based on the window length to simplify the analysis process and to reduce the analysis complexity for the results section of this work. This allows for a simple implementation as well as consistent network design referencing. For any network that used a convolutional layer, the decision was made to use a fixed stride and kernel size, with the padding being designed around these two components to ensure that equal feature map division could be found from one layer to the next. The convolutional layer design used in this work consists of:

- $L_{stride} = 4$
- $L_{kernel} = 32$
- $L_{padding} = \frac{L_{kernel} - L_{stride}}{2} = 14$

Under these properties, the network convolutional layer output size is enforced to undergo a reduction or expansion of a factor of four on the layer input size. To determine the output dimensionality of a convolutional layer, $\mathbb{R}^{(N_c \times L_{out}, 1)}$, one then needs to simply divide (or multiply if one uses de-convolutional layers) the number of channels by four times the number of convolutional layers that have been used at that point. To attach and layer that performs convolution to a fully connected layer, the decision was made to use a layer size of $L_{FC_1} = 800$ at the intermediary level and then a second fully connected layer to the final output dimensionality, where this could be a prescribed latent space dimensionality or a single node in the case of a discriminator or critic network. For a tabular visualisation of what these architectures may look like for an encoder, decoder or data discriminator network, please refer to Table B.1.

Table B.1. A table showing the basic network architecture for an encoder network for $L_{stride} = 4, L_{kernel} = 32$ and $L_{pad} = 14$. Note that N is the batch size and if one wishes to design a decoder, this table can simply be reversed. If one wishes to design a data discriminator, the final layer at depth level 5 can be a fully connected layer $\mathbb{R}^{800} \rightarrow \mathbb{R}^1$.

Network depth level	Layer Operator	Layer Dimensionality
0	-	$\mathbb{R}^{N \times 1 \times L_w}$
1	Convolution	$\mathbb{R}^{N \times 32 \times \frac{L_w}{4}}$
2	Convolution	$\mathbb{R}^{N \times 64 \times \frac{L_w}{16}}$
3	Convolution	$\mathbb{R}^{N \times 64 \times \frac{L_w}{64}}$
4	Fully-connected	$\mathbb{R}^{N \times 800}$
5	Fully-connected	$\mathbb{R}^{N \times Z_{latent}}$

For any latent discriminator or critic that is used in this work, a simple three layer fully-connected network is used where the dimensionality follows the process: $\mathbb{R}^{input} \rightarrow \mathbb{R}^{3000} \rightarrow \mathbb{R}^{3000} \rightarrow \mathbb{R}^1$. This decision was made arbitrarily and based of basic discriminator designs the authors noted in literature. In this work, the windowed partitioning scheme with a overlap percentage shall be used as it gives a good number of signal samples from a single vibration signal, with an overlap of 50% being arbitrarily

chosen. The next section of this section will detail the chosen network architectures for the different datasets analysed in this work. This information comprises of consistent and varied decisions and thus the authors will try to convey this information as clearly as possible to the reader. In Table B.2, the relevant activation functions for the different model components are given. Note that the author kept these activations consistent through the different datasets. The final element that allows for results reproduction is that of the latent dimensionality and hyper-parameters used for each dataset. This information is given in Tables B.3 and B.4. To train the models in this work, the Adam and AdamW methods were used for the VAE and *GAN*-based methods respectively, with parameters $\beta_1 = 0.6$, $\beta_2 = 0.999$. Instance noise was also used for the first three thousand epochs of the *GAN*-based method training. The $RY-GAN$ method also used generated samples from $Z_{latent} = [\mathbf{c}, \mathbf{0}, \mathbf{0}]$ to aim in enforcing that the decoder use the class variables. This additional component was linearly annealed in the first 2500 epochs and was added to the L_2 loss component.

Table B.2. A table showing the basic network activation functions that was used alongside the information in Table B.1.

Method	Hidden Layer Activations	Output Layer Activations
VAE: $\mathbf{z} \in \mathbb{R}^{Z_{latent}}$	ReLU for all hidden layers	μ : linear, σ^2: softplus (for both encoder and decoder)
$\beta-TC-VAE(\beta=1)$: $\mathbf{z} \in \mathbb{R}^{Z_{latent}}$	ReLU for all hidden layers	μ : linear, σ^2: softplus (for both encoder and decoder)
$RY-GAN$: $\mathbf{z} \in \mathbb{R}^{Z_{latent}=[\mathbf{c},\mathbf{s},\mathbf{n}]}$	Encoder: leaky ReLU ($a=0.2$) for all hidden layers Decoder: leaky ReLU ($a=0.2$) for all hidden layers $D_\chi(\mathbf{x})$: SN on all hidden layers, ReLU for all hidden layers $D_n(\mathbf{n})/D_c(\mathbf{c})$: ReLU for all hidden layers	Encoder ($[\mathbf{c},\mathbf{s},\mathbf{n}]$): softmax, linear, linear Decoder: linear $D_\chi(\mathbf{x})$: sigmoid $D_n(\mathbf{n})/D_c(\mathbf{c})$: Linear/Sigmoid
$DLS-GAN$: $\mathbf{z} \in \mathbb{R}^{Z_{latent}=[\mathbf{c},\mathbf{s},\mathbf{n}]}$	Encoder: leaky ReLU ($a=0.2$) for all hidden layers Decoder: leaky ReLU ($a=0.2$) for all hidden layers $D_\chi(\mathbf{x})$: SN on all hidden layers, ReLU for all hidden layers $D_n(\mathbf{n})/D_c(\mathbf{c})$: ReLU for all hidden layers	Encoder ($[\mathbf{c},\mathbf{s},\mathbf{n}]$): softmax, linear, linear Decoder: linear $D_\chi(\mathbf{x})$: sigmoid $D_n(\mathbf{n})/D_c(\mathbf{c})$: Linear/Sigmoid

Table B.3. The relevant latent dimensionality of the different models trained on the different datasets. Note that the input window length was not included here as its effect is noted in the form of the encoder and decoder given by Table B.1.

Latent variable model type:	Phenomenological model	IMS dataset	Gearbox Dataset
VAE_1 and VAE_2	$Z_{latent} = 100$	$Z_{latent} = 100$	$Z_{latent} = 50$
$\beta-TC-VAE_1$ and $\beta-TC-VAE_2$	$Z_{latent} = 100$	$Z_{latent} = 100$	$Z_{latent} = 50$
$RY-GAN$	$Z_{latent} = \mathbb{R}^{[10,10,128]}$	$Z_{latent} = \mathbb{R}^{[4,10,128]}$	$Z_{latent} = \mathbb{R}^{[10,10,125]}$
$DLS-GAN$	$Z_{latent} = \mathbb{R}^{[10,10,128]}$	$Z_{latent} = \mathbb{R}^{[10,10,128]}$	$Z_{latent} = \mathbb{R}^{[10,10,150]}$

Table B.4. The relevant hyper-parameters of the different models trained on the different datasets.

Latent variable model type:	Phenomenological model	IMS dataset	Gearbox Dataset
VAE_1 and VAE_2	$\eta = 1e^{-4}$	$\eta = 1e-4$	$\eta = 1e-4$
$\beta-TC-VAE_1$ and $\beta-TC-VAE_2$	$\eta = 1e^{-4}$	$\eta = 1e-4$	$\eta = 1e-4$
$RY-GAN$	$\eta = 5e^{-5}, \alpha = 0.5, \lambda_{AE} = 10$	$\eta = 5e^{-5}, \alpha = 0.5, \lambda_{AE} = 10$	$\eta = 5e^{-5}, \alpha = 0.5, \lambda = 10$
$DLS-GAN$	$\eta = 1e^{-5}, \lambda_{AE} = 60, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 1$	$\eta = 1e^{-5}, \lambda_{AE} = 80, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 1$	$\eta = 1e^{-5}, \lambda_{AE} = 40, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 1$

Appendix C Phenomenological Model Parameters

C.1 Chapter Abstract

The purpose of this chapter is to present the important parameters for the phenomenological data model. These parameters were chosen such that a simple model could be obtained while still ensuring that sufficient difficulty could be presented in the data.

C.2 Model Parameters

For the phenomenological data model, Table C.1 contains the modulation coefficients for the first order impulse responses of the phenomenological data, Table C.2 contains the deterministic gear components while Table C.3 contains the random gear component statistics. Table C.4 contains the variance of the noise and random gear components. The SNR that one can design each component for, to provide some scaling with respect to the noise floor using Equation 4.17, is given in Table C.5. The properties of the bearings used in this work and the respective fault frequencies are given in Table C.6. Note that for all of the z_i components that are modulated, the amplitude modulation function used in this work is

$$M_i(t) = \omega_{ref}^2(t). \tag{C.1}$$

Table C.1. A table showing the impulse response coefficients for Equation (4.5) for the three transmission paths being modelled.

Modulation Component	**Natural Frequency -** $f_{n,i}(Hz)$	**Damping Ratio -** ξ_i
$h_{g_d}(t)$	2000	0.05
$h_{g_r}(t)$	1300	0.05
$h_{b_r}(t)$	7000	0.05

Table C.2. A table showing the mesh coefficients for the deterministic gear excitation signal, whose form is given in Equation (4.6).

j	**1**	**2**	**3**
Amplitude - $a_{g_d}^{(j)}$	1	2	3
Phase - $\varphi_{g_d}^{(j)}$	0	0	0

Table C.3. A table showing the mesh coefficients for the random gear excitation signal, whose form is given in Equation (4.8).

j	**1**	**2**	**3**
Amplitude - $a_{g_r}^{(j)}$	1	2	3
Phase - $\varphi_{g_r}^{(j)}$	0	0	0

Table C.4. A table showing the variance of the distributed gear noise and the white noise, given in Equations (4.8) and 4.9 respectively.

Variance Component	**Value**
σ_n^2	0.01
$\sigma_{g_r}^2$	1

Table C.5. A table showing the pre-defined component SNR used in Equation (4.17).

Component Signal-to-Noise Ratio	**Value** (dB)
Deterministic gear	5
Random gear	-10
Bearing fault	$[-40, 10]$, linearly spaced

Table C.6. Phenomenological model bearing and fault characteristics. Notice the fault frequencies given proportional to the dataset shaft speed f_{shaft}.

	Characteristic	**Unit**
Roller Bearing	Pitch diameter	$6.35mm$
	Roller element diameter	$36mm$
	Contact angle	$0°$
	Number of rolling elements	10
	Gear teeth	20
	Gear ratio	1
Frequencies of interest	Sampling frequency (F_s)	$25kHz$
	Gear Mesh Frequency	$20\frac{Hz}{f_{shaft}}$
	Ball Pass Frequency Outer race (BPFO)	$4.12\frac{Hz}{f_{shaft}}$
	Ball Pass Frequency Inner race (BPFI)	$5.88\frac{Hz}{f_{shaft}}$
	Ball Spin Frequency (BSF)	$2.75\frac{Hz}{f_{shaft}}$
	Ball Cage Frequency/Fundamental Train Frequency (BCF/FTF)	$0.41\frac{Hz}{f_{shaft}}$

Appendix D MED-SK-NES: Derivation and Application

Minimum entropy deconvolution (MED) is a methodology that was proposed in the work of Wiggins (1978) as a means to obtain a signal filter that maximises the kurtosis of the filtered signal. The MED objective is to recover a filtered signal that captures the impulsive components of a signal, whereby the minimization of entropy makes reference to the enhancement of structural information in a signal, as a signal with higher entropy will tend to be more Gaussian, or more similar to white noise. MED utilises the kurtosis as a proxy to measure the entropy, as a higher kurtosis corresponds to a further deviation from a Gaussian-like state. Under the kurtosis proxy the objective is to determine a set of FIR filter coefficients that maximise the objective function $O_{MED}(\mathbf{h})$

$$O_{MED}(\mathbf{h}) = \frac{\sum_{n=0}^{N-1} y^4(n)}{\left[\sum_{n=0}^{N-1} y^2(n)\right]^2}, \tag{D.1}$$

where $y(n)$ is a signal that was filtered through a FIR filter $\mathbf{h}$ consisting of L coefficients which can be given as

$$y(n) = \sum_{l=1}^{L} \mathbf{h}(l)x(n-l). \tag{D.2}$$

One can then consider the vector field of O_{MED} with respect to each filter coefficient and determine the filter coefficients that result in a local maximum. This can be found through

$$\frac{dO_{MED}(\mathbf{h})}{d\mathbf{h}} = \mathbf{0}. \tag{D.3}$$

The expansion of this objective function shall be detailed in this work as the author felt that its reproducibility is key to future use of such methods. As a first step, consider the representation of $O_{MED}(\mathbf{h})$ as the ratio of two components, $v(\mathbf{h}) = \sum_{n=0}^{N-1} y^4(n)$ and $u(\mathbf{h}) = \sum_{n=0}^{N-1} y^2(n)$ such that $O_{MED}(\mathbf{h}) = \frac{v(\mathbf{h})}{u^2(\mathbf{h})}$. Using the quotient rule, the partial derivative of $O_{MED}(\mathbf{h})$ with respect to h_i can be found to be

$$\begin{aligned}
\frac{\partial O_{MED}(\mathbf{h})}{\partial h_i} &= \frac{u^2(\mathbf{h})\left[\sum_{n=0}^{N-1} 4y^3(n)\frac{\partial y(n)}{\partial h_i}\right] - v(\mathbf{h})\left[2\left(\sum_{n=0}^{N-1} y^2(n)\right)\left(2\sum_{n=0}^{N-1} y(n)\right)\right]}{\left[u^2(\mathbf{h})\right]^2}, \\
&\text{using } \frac{\partial y(n)}{\partial h_i} = x(n-i), \\
&= \frac{1}{u^4(\mathbf{h})}\left[u^2(\mathbf{h})\left[4\sum_{n=0}^{N-1} y^3(n)x(n-i)\right] - v(\mathbf{h})\left[4u(\mathbf{h})\sum_{n=0}^{N-1} y(n)x(n-i)\right]\right], \\
&= \frac{1}{u^4(\mathbf{h})}\left[4\sum_{n=0}^{N-1} y^3(n)x(n-i) - 4v(\mathbf{h})u(\mathbf{h})\sum_{n=0}^{N-1} y(n)x(n-i)\right].
\end{aligned} \tag{D.4}$$

One can then consider the case where the local maximum is to be found, by setting Equation D.4 to zero. The result of this is the relation

$$\frac{v(\mathbf{h})}{u^3(\mathbf{h})}\sum_{n=0}^{N-1} y(n)x(n-i) = \frac{1}{u^4(\mathbf{h})}\sum_{n=0}^{N-1} y^3(n)x(n-i), \tag{D.5}$$

from which the result can be shown to be

$$O_{MED}(\mathbf{h})u(\mathbf{h})\sum_{n=0}^{N-1} y(n)x(n-i) = \sum_{n=0}^{N-1} y^3(n)x(n-i), \tag{D.6}$$

which can be simplified further that noting that one can also expand the summation on the left as it is dependant on $y(n)$, resulting in

$$O_{MED}(\mathbf{h})u(\mathbf{h})\sum_{p=1}^{L}\mathbf{h}(p)\sum_{n=0}^{N-1} x(n-p)x(n-i) = \sum_{n=0}^{N-1} y^3(n)x(n-i), \tag{D.7}$$

It is immediately noticeable how this equation is a non-linear function of $\mathbf{h}$ and cannot be solved for analytically. It is also only the partial derivative towards ∂h_i, and thus needs to be expanded to account for the other filter coefficients. This process can be given in matrix form as $\underset{\sim}{\mathbf{A}}\mathbf{h} = \mathbf{g}$, where $\underset{\sim}{\mathbf{A}}$ is a Toeplitz auto-correlation matrix of dimensionality $\mathbb{R}^{L\times L}$, $\mathbf{h}$ are the filter coefficients and $\mathbf{g}$ is a column vector of dimensionality $\mathbb{R}^{L\times 1}$ containing the right hand side of Equation D.7. Due to the issues with respect to the analytical solution, a iterative process is applied whereby an initial set of filter coefficients $\mathbf{h}^{(0)}$ are assumed and the iterative algorithm

$$\mathbf{h}^{(t+1)} = \underset{\sim}{\mathbf{A}}^{-1}(\mathbf{h}^{(t)})\mathbf{g}(\mathbf{h}^{(t)}). \tag{D.8}$$

Lee and Nandi (2000) provides a detailed analysis of this method, referred to as the Objective Function Method (OFM), with a potential stopping function check, which can be given as

$$\mathbf{e} = \frac{\mathbf{h}^{(t+1)} - \mu_t\mathbf{h}^{(t)}}{\mu_t\mathbf{h}^{(t)}}, \tag{D.9}$$

where μ_t is given as

$$\mu_t = \sqrt{\frac{\mathbb{E}\left[(\mathbf{h}^{(t+1)})^2\right]}{\mathbb{E}\left[(\mathbf{h}^{(t)})^2\right]}}. \tag{D.10}$$

One may then use $\mathbb{E}[\mathbf{e}]$ as a threshold, with Lee and Nandi (2000) using a value of 0.01 such that iteration is terminated when $\mathbb{E}[\mathbf{e}] \leq 0.01$. Interestingly, literature defines a stopping condition not on the kurtosis but rather on the rate of change of the coefficients from one iteration to the next. Endo and Randall (2007) provides a discussion of this, stating that optimal kurtosis may not be optimal for fault diagnosis purposes. It was also noted that in their case typically two iterations were required to converge to a very small coefficient change.

In the work of Sawalhi et al. (2007), an Auto-Regressive residual MED-SK approach was used to filter the signal after which the squared envelope and envelope spectrum were analysed. During their investigation, an analysis was performed into the inclusion of MED and the benefits it may offer, by comparing an AR-SK result to a AR-MED-SK result. It was shown that with the enhancement of MED, a deeper understanding of the fault present could be obtained. Interestingly, the result was noticeable but it was clear that just an SK filter produced sufficient results. In the work of McDonald and Zhao (2017), flaws of the MED approach were presented and it was noted that MED tended to produce spurious impulses in the filtered signal which the authors attributed to the convolution operation and the zero-padding that is often involved in there. Alternative techniques were then proposed to improve the performance of MED.

The next element of the MED-SK-NES approach is that of the SK-based filter. Abboud et al. (2019) states that this method is used to further enhance signal impulsivity with respect to the background noise floor prior to the analysis with the SES. The SK is a widely used technique in signal processing, with works such as the Kurtogram, Antoni and Randall (2006), Antoni (2007), being heavily dependant on the usage of the SK. The SK is then forth-order normalised cumulant of the short-time Fourier transform of a signal $x(n)$, denoted here as $X(n,f)$, which can be given as

$$K_x(f) = \frac{\langle |X(n,f)|^4 \rangle}{\langle |X(n,f)|^2 \rangle^2} - 2, \tag{D.11}$$

where $\langle \cdot \rangle$ denotes the average operator over time index n, $\langle f(n) \rangle = \lim_{n\to\infty} \frac{1}{N} \sum_N f(n)$. Note that the SK definition here subtracts two as opposed to three as $X(n,f)$ is complex and thus the Gaussian spectral kurtosis is not longer three. To use the SK as a filter, one must use the square root of the spectral kurtosis as this filter is proportional to a Wiener filter (Sawalhi, 2004). This then defines a Wiener filter of the form

$$W(f) = k\sqrt{K_x(f)}, \tag{D.12}$$

where k is an arbitrary constant and can be set to one. One can then either perform frequency domain weighting or transform the SK filter to the time domain using the Inverse Fourier transform to obtain a impulse response that can be used as a filter. In this work the former approach is favoured over the latter. The final element of this analysis is the calculation of the Normalised Envelope Spectrum of the $\sqrt{SK}$-filtered MED signal $z(n)$. To do this, the Squared Envelope Spectrum is determined using the discrete time Fourier Transform (DTFT) of the squared Hilbert transform analytic signal

$$SES_z(\alpha) = |DTFT_{n\to\alpha}(|A\left[z(n)\right]|^2)|, \tag{D.13}$$

where $A[\cdot]$ is the complex analytic signal obtained from the Hilbert transform. The SES is then normalised by the DC offset of the spectrum to obtain the NES

$$NES_z(\alpha) = \left(\frac{SES_z(\alpha)}{SES_z(0)}\right)^2. \tag{D.14}$$

It is then expected that the $NES_z(\alpha)$ exhibit harmonics of a bearing fault frequency with side-bands offset by the modulation frequency. Figure D.1 contains a visual illustration of the full process in which one applies MED-SK-NES.

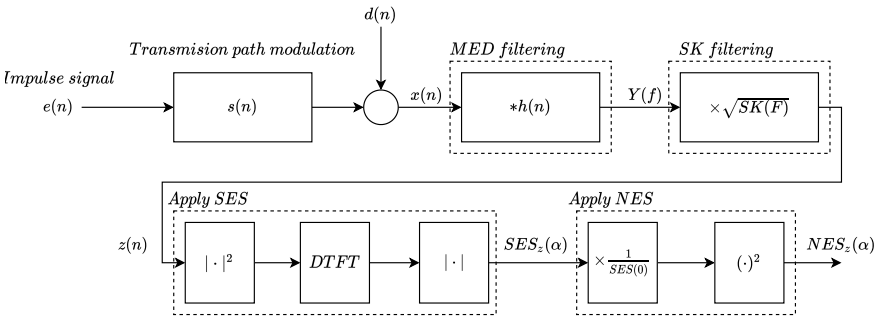
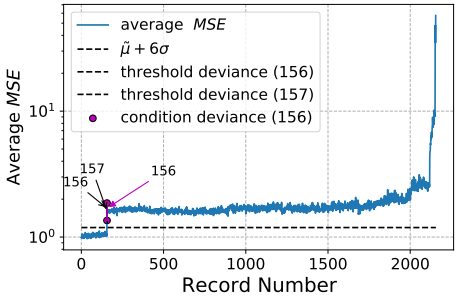


Figure D.1. A figure showing the MED-SK-NES process in full, adapted from Abboud et al. (2019).

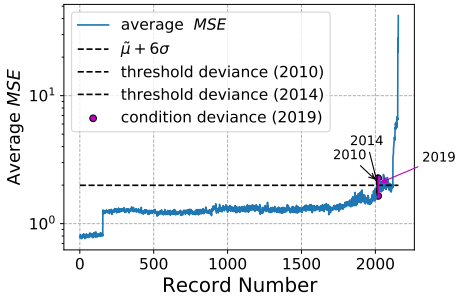
Appendix E Interesting Results

The purpose of this appendix document is to allow for interested readers to analyse other results that were excluded from the results section of this work.

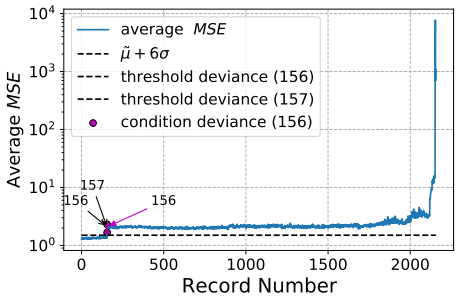
E.1 IMS: Bearing three, dataset one



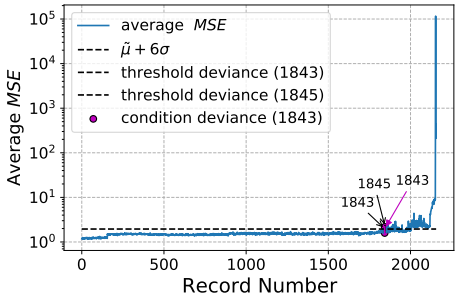
(a) Case one, VAE_1



(b) Case two, VAE_1



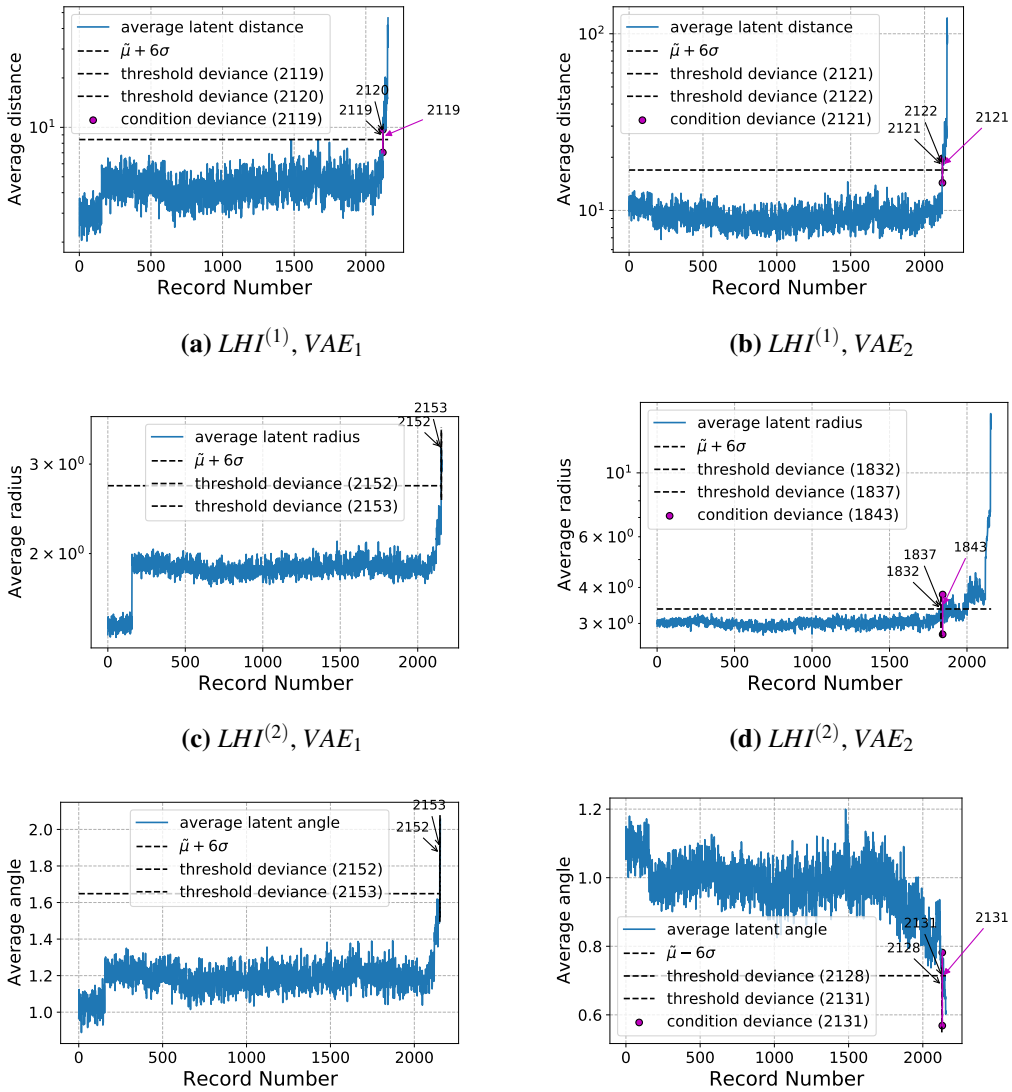
(c) Case one, VAE_2



(d) Case two, VAE_2

Figure E.1. The *HI* response obtained from VAE_1 and VAE_2 models trained using 5% and 10% of the data available for bearing three from IMS dataset one.

E.2 IMS: Bearing one, dataset two

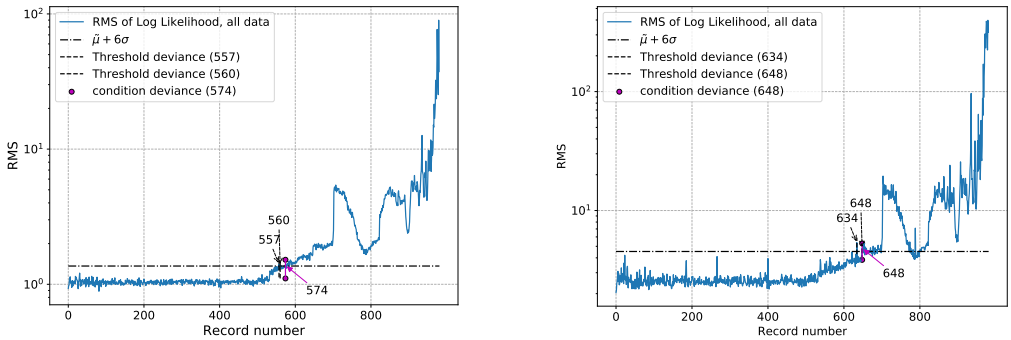


(a) $LHI^{(1)}, VAE_1$ **(b)** $LHI^{(1)}, VAE_2$

(c) $LHI^{(2)}, VAE_1$ **(d)** $LHI^{(2)}, VAE_2$

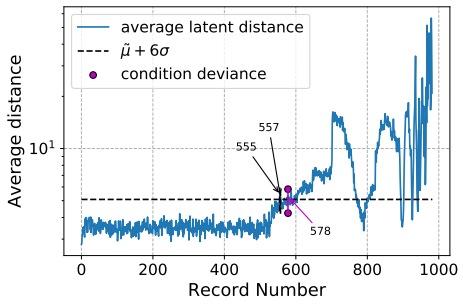
(e) $LHI^{(3)}, VAE_1$ **(f)** $LHI^{(3)}, VAE_2$

Figure E.2. The *LHI* responses from VAE_1 and VAE_2 models for the case where the model had access to 10% of the records available as training data. Notice the visible lack of jump around record 155 for the VAE_2 response in $LHI^{(1)}$ and $LHI^{(2)}$.

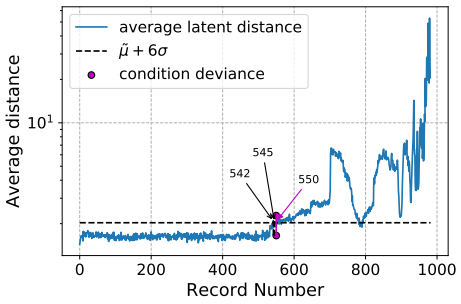


(a) $VAE_1, L_w = 512$ **(b)** $VAE_2, L_w = 512$

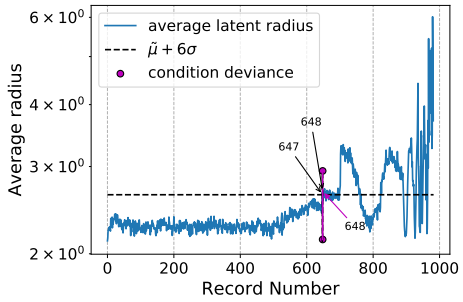
Figure E.3. The $HI^{(1)}$ result obtained using two different parametrisations of the *VAE* model under the same window length for the first bearing of the second IMS dataset. Figure E.3(a) shows the deterministic VAE while Figure E.3(b) shows the stochastic VAE



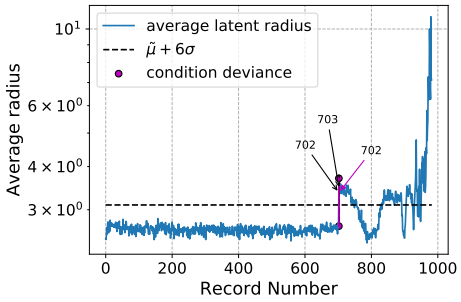
(a) $LHI^{(1)} - VAE_1, L_w = 512$



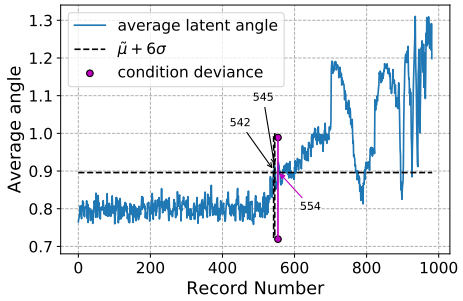
(b) $LHI^{(1)} - VAE_2, L_w = 512$



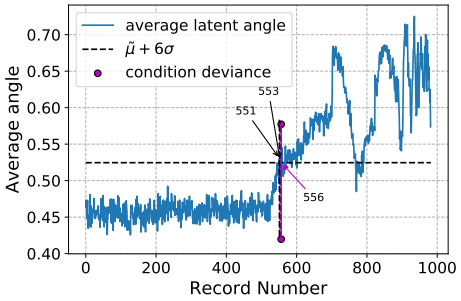
(c) $LHI^{(2)} - VAE_1, L_w = 512$



(d) $LHI^{(2)} - VAE_2, L_w = 512$



(e) $LHI^{(3)} - VAE_1, L_w = 512$



(f) $LHI^{(3)} - VAE_2, L_w = 512$

Figure E.4. The three *LHI*'s response curves obtained from the deterministic and stochastic VAE models trained on bearing one data from the second IMS dataset. Figures E.4(a), (c) and (e) refer to the deterministic results and Figures E.4(b), (d) and (f) refer to the stochastic results.