Deep digital twins for detection, diagnostics and prognostics

Wihan Booyse[a,b],*, Daniel N. Wilke[a] and Stephan Heyns[a]

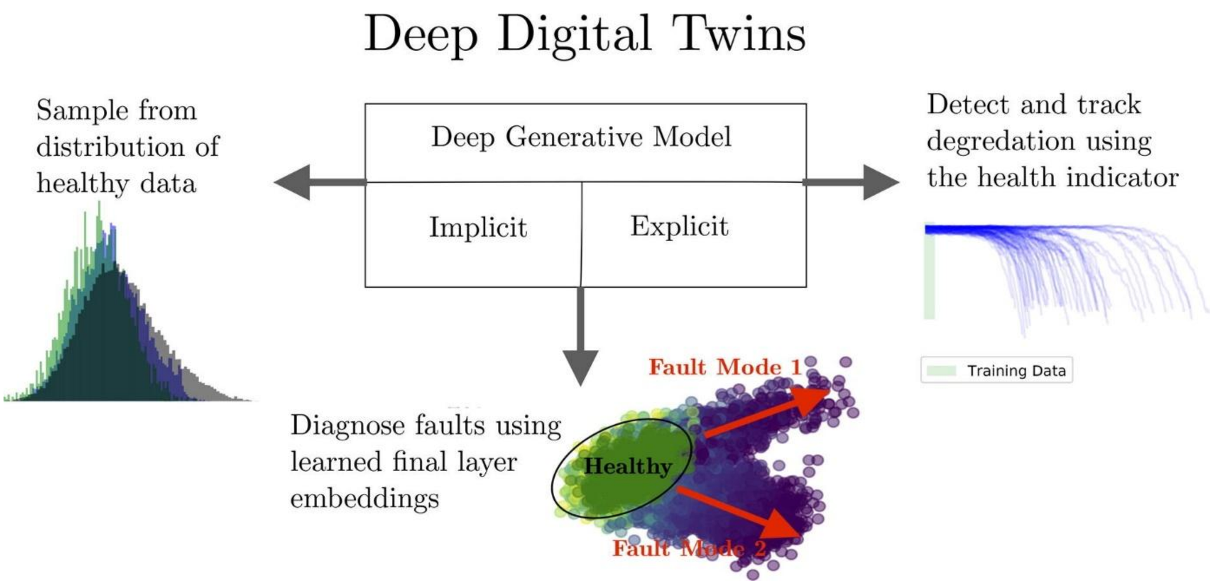
[a] Centre for Asset Integrity Management, University of Pretoria, South Africa
[b] Kriterion AI Research, South Africa

*Corresponding author. wihan@kriterion.ai

Highlights

- Deep generative models as deep digital twins.
- Simulate asset responses to operating conditions directly from data.
- Unsupervised health indicator from only healthy data.
- Track degradation over asset lifecycle.
- Suitable for non-stationary operating conditions.

Graphical abstract


Deep Digital Twins
Sample from distribution of healthy data
Deep Generative Model
Implicit
Explicit
Detect and track degradation using the health indicator
Training Data
Diagnose faults using learned final layer embeddings
Fault Mode 1
Healthy
Fault Mode 2

Abstract

A generic framework for prognostics and health monitoring (PHM) which is rapidly deployable to heterogeneous fleets of assets would allow for the automation of predictive maintenance scheduling directly from operational data. Deep learning based PHM implementations provide part of the solution, but their main benefits are lost when predictions still rely on historical failure data and case-by-case feature engineering. We propose a solution to these challenges in the form of a Deep Digital Twin (DDT). The DDT is constructed from deep generative models which learn the distribution of healthy data directly from operational data at the beginning of an asset's life-cycle. As the DDT learns the distribution of healthy data it does not rely on historical failure data in order to produce an estimation of asset health. This article presents an overview of the DDT framework and investigates its performance on a number of datasets. Based on these investigations, it is demonstrated that the DDT is able to detect incipient faults, track asset degradation and differentiate between failure modes in both stationary and non-stationary operating conditions when trained on only healthy operating data.

Keywords

Artificial intelligence; Deep learning; System health management; Predictive maintenance; Deep generative models; Digital twins

1. Introduction

As heavy industries such as power generation, manufacturing and mining continue to digitise at a rapid pace, the amount the data produced by industrial facilities is greater than ever before. In order to capitalise on the vast amount of process and plant data available in modern industrial complexes, it is critical that modern prognostics and health management strategies are designed for scalable deployment to heterogeneous fleets of assets under both stationary and non-stationary conditions.

One primary constraint to the practical deployment of data driven system health management is the requirement for large amounts of both healthy and unhealthy data in order to train supervised learning approaches [19]. This includes state-of-the-art deep learning architectures, that include Autoencoders, Variational

Autoencoders, ladder networks and Convolutional Neural Networks, that are mostly employed in a supervised setting using fully labelled healthy and unhealthy datasets or a semi-supervised setting using sparsely labelled healthy and unhealthy datasets [20]. Unfortunately, obtaining a large number of run to failure datasets of critical assets is problematic due to the long operational lifespans of critical assets. In addition, during the life-cycle of an asset, the timespan over which damage can be recorded is usually only a fraction of the total life. For example, an asset that fails due to a crack will spend 85–90% of its life in the healthy regime, 5–8% of its life in the damaged regime and 1–2% of its life in the near failure regime – as given by the Paris Law [42]. In addition it is problematic to align failure events and sensor data in a practical industrial setting as maintenance logs and sensor data usually exist in separate databases without clear connections between them. Moreover, supervised learning approaches are inapplicable to newly procured assets where there is no possibility to leverage historical data.

A recent review on the state of deep learning on system health management [20], revealed that although the field of data driven prognostics and health monitoring has begun to migrate towards using deep learning for system health monitoring, many implementations in literature are still constrained to specific equipment or applications. Khan [20] highlights that the current case-by-case methodology and ad hoc approach does not allow for the scalable deployment of deep learning implementations for system health management. Khan further motivates that in order for deep learning based implementations to gain widespread adoption it is necessary to develop a single framework which can systematically be extended to all aspects of system health management.

The development this framework is challenging due to the diverse nature of assets that need to be monitored as well as the uncertainties regarding the available data. Consequently, such a framework must:

1. Use only healthy (nominal) operational data for training.
2. Have the ability to be trained online—not require historical data from similar assets or fleets.
3. Be applicable to equipment operated under stationary and non-stationary conditions.
4. Have the ability to be applied at different asset resolutions—component, sub-system, system, fleet.

5. Be extendible to adding additional functionality as data becomes available.

As a step towards the development of a single framework for system health management—which falls under the general ambit of prognostics and health monitoring (PHM)—this paper proposes a strategy, based on unsupervised deep learning, to construct a Deep Digital Twin (DDT) of a real world asset from sensor data. The proposed DDT which is trained only on healthy data, aims to learn a probabilistic representation of the real world asset from which it is possible to sample "Generated" sensor readings given current operational conditions and observed healthy data. Therefore, this generative model is a probabilistic digital twin of the asset that is constructed from the dataset. The primary benefit of the DDT is that it is automatically able to produce a health indicator from only healthy data.

The focus of this study is to investigate whether the DDT does indeed conform to the requirements of a PHM framework for predictive maintenance and system health monitoring. To this end, it is necessary to evaluate whether the DDT is able to satisfy each of the three phases of PHM as outlined by Randall [37]:

1. *Detection:* The first phase of PHM requires that an implementation is able to detect anomalous or faulty behaviour.

2. *Diagnostics:* The second phase of PHM requires that an implementation is able to differentiate various types anomalous events (failure modes).

3. *Prognostics:* The third phase of PHM requires that an implementation is able to provide a measure of system health which allows for the tracking of degradation.

2. Deep Digital Twins

Traditionally, Digital Twins have relied on deterministic physics based simulations to approximate the system being monitored [38]. Unfortunately, for complex systems it is generally not possible to simulate each individual asset within a fleet of homogeneous assets due to manufacturing and material uncertainties [44]. Moreover, if digital twin technology is to be deployed to a heterogeneous fleet of assets, the implementation becomes significantly more complex as each unique asset class will require its own physics based model [44]. Consequently, simulation based digital twin technology is only economically feasible for high-cost and high-risk assets.

In order to circumvent the practical constraints of implementing physics based Digital Twins it is proposed to use deep learning to develop Digital Twins directly from the operational data of an asset, this will be known as a Deep Digital Twin (DDT). The primary goal of a DDT is to obtain a digital representation of the expected behaviour of a real world asset directly from data using a deep generative model. Once such a representation is obtained, it becomes possible to simulate system responses by sampling from the distribution $q(\mathbf{x}|\mathbf{y},\mathbf{z})$ of the DDT where $\mathbf{x}$ is the system response to the conditional (operational setting) $\mathbf{y}$ and the underlying latent variables which describe the system behaviour $\mathbf{z}$. Sampling from the DDT in this manner allows asset owners and operators to predict how the asset will respond in various hypothetical operational settings (engine speed, RPM, ambient temperature, feed rate, throttle position, etc.); as well as comparing the predicted behaviour of the asset to the ground truth. However, the primary motivation for the implementation of a DDT is inevitably the benefits it provides to the life-cycle management of an asset. DDTs are particularly useful for prognostics and health monitoring (PHM) due to their probabilistic nature. The probabilistic nature of a DDT allows asset owners and operators to interrogate the likelihood or likelihood-ratio of the current observed asset state given the historical healthy data. Consequently, a DDT is able to automatically produce a health indicator by means of either the likelihood or likelihood ratio.

Using deep generative methods for creating Deep Digital Twins, as opposed to simulation based approaches, do not explicitly simulate the physics of the entire system. Instead the Deep Digital Twin approach uses sensor data collected while the asset is operational to create a probabilistic model of the system.

DDT may be defined as:

- An implicit physics model of an asset learned from healthy asset data, requiring no explicit physics knowledge.
- A digital representation from which sensor values can be sampled under both stationary and non-stationary operational settings such as rotational speed, throttle and load.
- A data driven model which does not require any asset specific feature engineering.
- A probabilistic model which is able to automatically produce a health indicator which is a metric of the deviation from the healthy asset data.

In the Deep Digital Twin framework, all sensor measurements are treated as random variables denoted $\mathbf{x}$ and the control inputs of the system as random variables $\mathbf{y}$. The DDT assumes there exists some distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y}, \mathbf{z}^*)$ which perfectly describes the behaviour of the system given the control inputs and a number of unobserved latent variables $\mathbf{z}^*$—which represents additional hidden variables which are not observed directly but could be approximated given enough observations of $\mathbf{x}$ and $\mathbf{y}$. The goal of the DDT training is to learn some approximation $q(\mathbf{x}|\mathbf{y}, \mathbf{z})$ which closely matches the true distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y}, \mathbf{z}^*)$, where $\mathbf{z}^*$ represent the true unobserved latent variables and $\mathbf{z}$ are a learned approximation to $\mathbf{z}^*$, such that the DDT instance is representative of $p_{\text{data}}(\mathbf{x}|\mathbf{y}, \mathbf{z}^*)$. This allows for the interrogation of $p_{\text{data}}(\mathbf{x}|\mathbf{y}, \mathbf{z}^*)$ through $q(\mathbf{x}|\mathbf{y}, \mathbf{z})$, which is embedded in the DDT instance. In this study, we specifically focus on constructing DDT instances of assets' healthy state only, which allows us to infer when operating conditions deviate from the healthy state and estimate the probability of damage.

To ensure that the DDT instance is useful and reliable for PHM, it is important that:

1. The DDT instance has captured the entire distribution or manifold of the training dataset.

2. The DDT does not embed spurious information present in the dataset in the learned manifold, or assign high likelihood to nonsensical observations.

3. It is possible to test whether 1 and 2 are satisfied.

4. Track the degradation of an asset such that it is possible to infer the condition of an asset from the DDT.

The following paper will outline the implementation of a Deep Digital Twin for asset condition monitoring. Once trained, it is possible to sample system simulated responses from the distribution $q(\mathbf{x}|\mathbf{y}, \mathbf{z})$ which is assumed to be a good representation of the true data distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y}, \mathbf{z})$. In addition to the ability to simulate by sampling from $q(\mathbf{x}|\mathbf{y}, \mathbf{z})$ the DDT is also able to produce a non-parametric estimation of the current system health given observed values of $\mathbf{x}$ and $\mathbf{y}$.

3. Deep generative models as deep digital twins

Currently there exist two forms of deep generative models which may be suitable for implementing and constructing a DDT. Namely:

- Explicit probabilistic models
- Implicit probabilistic models

The primary difference between the explicit and implicit models is that explicit models provide an explicit parametrisation of the distribution of the observed variable $\mathbf{x}$ by specifying a log-likelihood function $\log(p_\theta(\mathbf{x}))$. On the other hand implicit probabilistic models do not assume a tractable form of the log-likelihood function but rather learn a likelihood ratio $r = \frac{p_{\text{data}}(\mathbf{x}|\mathbf{y},\mathbf{z})}{q(\mathbf{x}|\mathbf{y},\mathbf{z})}$. Consequently while explicit models are able to provide an estimation of the likelihood of an observation $p_\theta(\mathbf{x})$, implicit models are only able provide the ratio of the likelihood between the observed data distribution and the generated data distribution. Both the likelihood and likelihood ratio provide an indication of the chance that observed data belongs to the healthy data distribution.

3.1. Health indicator construction

The key advantage of the use of a deep generative model for the construction of a DDT is the automatic calculation of a health indicator. Namely:

- For explicit models it is proposed that a health indicator can be constructed by evaluating the likelihood of an observation $p_\theta(\mathbf{x})$ given the parametric model of healthy data.
- For implicit models it is proposed that the health indicator can be obtained by evaluating the likelihood-ratio between an observation $\mathbf{x}$ and the generated data distribution $q(\mathbf{x}|\mathbf{y},\mathbf{z})$

3.2. Explicit deep digital twins

In the explicit setting the DDT will be constructed using a Variational Autoencoder (VAE) as the deep generative model. Variational Autoencoders are explicit deep generative models which from part of a group of statistical models that perform approximate inference in latent variable models [22]. A latent variable model assumes that some set of observed variables $\mathbf{x}$ are generated by a random process which is dependent on some parameters $\boldsymbol{\theta}^\star$ and unobserved latent variables $\boldsymbol{z}$.

The two main goals of machine learning (and by extension deep learning) within the context of latent variable models are inference and learning. Inference is the evaluation of the posterior distribution

$$p_{\theta}(\mathbf{z}|\mathbf{x}, \theta^{\star}) = \frac{p_{\theta^{\star}}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})}{p_{\theta^{\star}}(\mathbf{x})} \tag{1}$$

of the latent variables $\mathbf{z}$ given the observed data $\mathbf{x}$ and some parameters θ. On the other hand, learning is the maximisation of the model evidence. That is, given some probability distribution p_{θ} parametrised by θ which approximates the true distribution $\theta^{\star}$, maximise the likelihood of the data x given the distribution parameters θ: Maximise the evidence,

$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z) \ \ dz\,. \tag{2}$$

In the context of VAEs, the estimated posterior $p_{\theta}(x|\mathbf{z})$ is parametrised by a neural network ($g_{\theta}(\mathbf{z})$), known as the decoder. From this perspective the VAE can be seen as a neural network architecture which contains two neural networks, an encoder ($f_{\phi}(\mathbf{x})$) network and a decoder ($g_{\theta}(\mathbf{z})$) as shown in Fig. 1. The task of the encoder network is to take the original input data of dimension D_{input} and map the data into a lower dimension D_{latent} to give an under-complete ($D_{latent} < D_{input}$) representation of the input data. The encoder takes the observed variables x and outputs the natural parameters (ξ) of the assumed prior distribution, where the parameters ξ are dependent on the values of ϕ as

$$\xi_{x_i} = f_{\phi}(x_i).$$

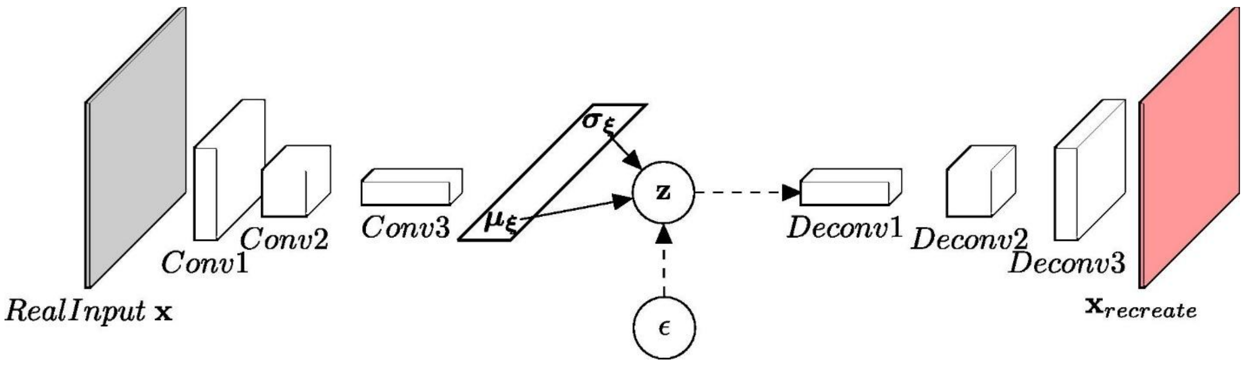


Fig. 1. Basic architecture of a Variational Autoencoder (VAE) with convolutional and de-convolutional layers, dashed lines denote sampling from a stochastic node. On the left side of the graph is the encoder network $q_{\xi}(\mathbf{z}|\mathbf{x}, \phi)$; On the right is the decoder network $p_{\theta}(\mathbf{x}|\mathbf{z})$.

For example, consider the case where $\boldsymbol{\xi}$ parametrises a Gaussian distribution

$$q_{\xi}(\mathbf{z}|\mathbf{x},\phi) = \mathcal{N}(\mathbf{z}|\mu_{\xi},\sigma_{\xi})$$

the components of $\boldsymbol{\xi}$ will be the mean and the variance of the latent variables for each data-point $\mathbf{x}_i$ to obtain $\boldsymbol{\xi}_{x_i} = (\boldsymbol{\mu}_{x_i}, \sigma^2_{x_i})$.

After the real data is passed through the encoder it becomes possible to sample the encoded latent variables from the posterior distribution $q_{\xi}(\mathbf{z}|\mathbf{x},\phi)$ using the statistical parameters μ_{ξ}, σ_{ξ} by means of the re-parametrisation trick

$$\mathbf{z} = \boldsymbol{\mu}_{\xi} + \boldsymbol{\epsilon}\boldsymbol{\sigma}_{\xi} \tag{3}$$

where,

$$\boldsymbol{\epsilon} = \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

After passing through the encoder, the posterior $\mathbf{z}$ samples are fed through the decoder which attempts to reconstruct the input given the latent representation.

Viewing the VAE in this manner results in a loss function which only consists of minimising the mean squared error (MSE) between the true input $\mathbf{x}$ and the recreated input $g_{\theta}(\hat{\mathbf{z}})$ (where $\hat{\mathbf{z}} = f_{\phi}(\mathbf{x})$). Unfortunately minimising the recreation error alone results in the VAE learning a non-smooth latent variable space which may have no overlapping support between learned clusters. This behaviour is undesirable as it does not constrain the latent variable space distribution. If the latent variable space's distribution is unconstrained it becomes possible for the VAE to simply memorise the data and assign each data point to an arbitrary location in the latent space, while disregarding any inherent structure in the data.

To circumvent this the VAE optimisation process does not simply consist of a MSE loss but also includes a regularisation term to ensure that the encoded prior distribution $q_{\xi}(\mathbf{z}|\phi)$ matches the assumed prior distribution $p(\mathbf{z})$ (usually a unit standard deviation factored Gaussian). This regularisation is applied by adding a penalty term which penalises dissimilarity between the encoded prior distribution $q_{\xi}(\mathbf{z}|\phi)$ and the assumed prior distribution $p(\mathbf{z})$. The penalty term penalises the VAE based on the KL divergence between $q_{\xi}(\mathbf{z}|\phi)$ and $p(\mathbf{z})$, is given by

$$D_{\mathrm{KL}}(q_{\xi}(\mathbf{z}|\phi)||p(\mathbf{z})) = \sum_{i=1}^{N} \frac{1}{N} \sum_{d=1}^{D_{latent}} \mu^2_{x_{id}} + \sigma^2_{x_{id}} - \log(\sigma_{x_{id}}) - 1, \tag{4}$$

for the standard case where $p(\mathbf{z})$ is assumed to be a factorised Gaussian with unity standard deviation and zero mean.

Including the KL penalty along with the MSE between the recreated input and the real input results in the following VAE loss function:

$$\mathscr{L}_{VAE} = ||g_\theta(f_\phi(\mathbf{x})) - \mathbf{x}||_2 + \sum_{i}^{N} \frac{1}{N} \sum_{d}^{D_{latent}} \mu^2_{x_{id}} + \sigma^2_{x_{id}} - \log(\sigma_{x_{id}}) - 1, \tag{5}$$

for the health indicator (HI) of the VAE the likelihood of the observed point under the learned parameters is obtained by Monte Carlo sampling to obtain the integral given in Eq. (2). This is done by encoding an observation into the latent space and sampling multiple simulated 'observations' from the posterior distribution and calculating the likelihood of the observed point under the learned probability distribution

$$HI_{VAE} = p_\theta(\mathbf{x}). \tag{6}$$

3.3. Implicit deep digital twins

In the implicit deep generative network setting the DDT will be constructed from a Generative Adversarial Network (GAN) as shown in Fig. 2. The GAN training process is cast as a two player non-cooperative game where each player has control over its own parameters (the weights of each network that constitute either the Generator (G) or Discriminator (D) as shown in Fig. 2). The goal of the game is for each player to minimise its own loss function $\mathscr{L}$ by tuning these parameters [12].

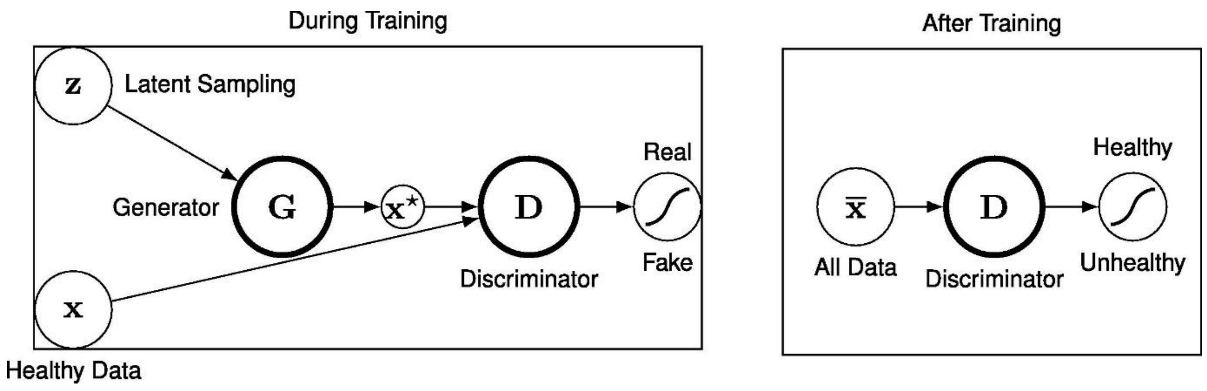


Fig. 2. Diagram demonstrating the GAN DDT. The left depicts the GAN training process and the right image demonstrates the process of producing a health indicator is shown.

In the GAN training framework, the two opposing networks D (Discriminator) and G (Generator) are each parametrised by a different set of weights and biases. Where D is parametrised by $\boldsymbol{\theta}$ and G is parametrised by $\boldsymbol{\phi}$. By changing $\boldsymbol{\theta}$ the optimiser associated with D aims to minimise some loss function $\mathscr{L}_D(\boldsymbol{\theta}, \boldsymbol{\phi})$. Similarly, by changing ϕ the optimiser associated with G aims to minimise its own loss function $\mathscr{L}_G(\boldsymbol{\theta}, \boldsymbol{\phi})$.

At each iteration G samples from a latent variable space $\mathbf{z}$ which is assumed to be a multivariate factorised Gaussian distribution with $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\sigma} = \mathbf{1}$. The samples from the latent variable space are passed through the parametric distribution q_ϕ, generating output samples $\mathbf{x}^\star$

$$\mathbf{x}^\star = G(\mathbf{z}),$$

which are fake examples conditioned on the sampled latent variable. As training progresses the divergence between the true data distribution p and the learned distribution q_ϕ becomes smaller as samples generated from q_ϕ begin to resemble samples from the true distribution p. During training the discriminator becomes better at identifying data that does not come from the true distribution p. Consequently, the discriminator may be seen as learning a discrepancy function, a function which assigns a high probability to points which are close to the manifold of real data and a low probability everywhere else. Thus the health indicator (HI), as shown in Fig. 2, is the output of the discriminator (D) of the GAN after learning the manifold of healthy data and equates to the density ratio between the observed point and the learned distribution

$$HI = D(\mathbf{x}). \tag{7}$$

The goal of GAN training is to reach Nash equilibrium [16], [39]. Nash equilibrium is defined as a game state in a non-cooperative game where it is impossible for any player to improve its score by only changing their own strategy [30]. In the context of a GAN, Nash equilibrium is a state of the game between the Generator and Discriminator at which the samples q_ϕ created by the Generator match the true distribution p and the Discriminator is unable to distinguish between data sampled from p and data sampled from q_ϕ as they are the same distribution [45]. This is outlined in Fig. 3(a)–(b), where given some operational data (two dimensional in the illustration) with numerous observations of p (black dots), the GAN, when training is successful, is able to learn a lower dimensional (one dimensional in the illustration) embedded probabilistic representation of the data

q_ϕ (green line). Observations that deviates from the learned manifold is associated with being less likely. Consider Fig. 3(c), where the data of two operational faults are indicated as cyan and magenta dots. As the operational faults have a lower probability of operating on the learned manifold, it becomes more evident that there is a high likelihood for a fault. However, should the GAN training have failed and only be representative of a subset of the healthy and normal operational data, we would misclassify normal operational data as having a low probability of lying on the manifold as depicted in Fig. 3(d). This is known as mode collapse, where essentially the generator, $G(\mathbf{z}, \phi)$, produces samples of limited diversity or even the same sample independent of the value of the random latent variable $\mathbf{z}$. It is therefore important to ensure and confirm that the GAN represents the entire healthy dataset and not just a subset. The techniques to establish whether mode collapse has occurred or not is covered in detail in the appendix.

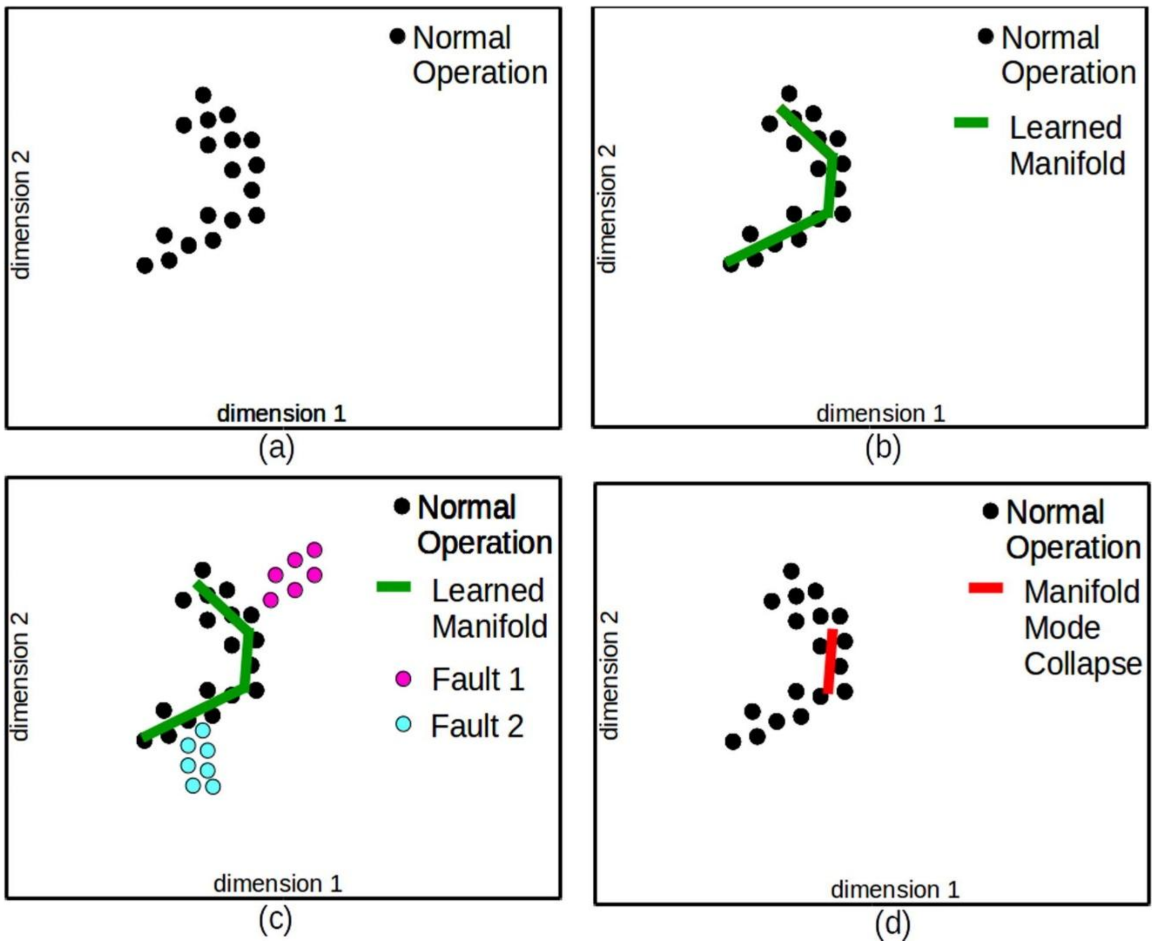


Fig. 3. (a) Normal operating data used to (b) train a GAN to resolve an underlying lower dimensional manifold that describes the data. A deviation from the learned manifold can be a result of the onset of operating conditions that deviate from the normal operating conditions due to (c) faults or (d) when the learned manifold only describes a subset of the normal operating conditions due to mode collapse.

One significant advantage of the GAN approach is that unlike most regression and system identification methods, it does not require the minimisation of a Mean Squared Error (MSE) loss. Instead the D approximates the density ratio between the observed data and the generated data and the G minimises some divergence between q_ϕ and p. These training dynamics allow the GAN to become very sensitive to even low magnitude deviations from the manifold. Moreover, since the GAN discriminator is based on a density ratio approach as opposed to a least squares approach, it is sensitive to the direction of the deviations from the manifold—not only the Euclidean distance of the deviations as in a least squares approach. This is clear in Fig. 3, where only an observation of the least squares loss between the manifold and either fault, directional information would be lost when computing the Euclidean distance. For the full details of the GAN implementation used in this work, please refer to the appendix.

4. Simulation based experiments

In order to assess the capabilities of the DDT approach for prognostics and health monitoring, two simulation based datasets were selected to evaluate the different capabilities of the DDT.

- *Gearbox:*
  - –Detection—Is the DDT able to detect unhealthy data when only trained on healthy data?
  - –Prognostics—Does the HI produced by the DDT decrease monotonously with an increase in damage?
  - –Robustness—Are the DDT's predictions (HI and Generated samples) robust to non-stationary data over continuously varying operating conditions?
  - –System Identification—Is the DDT able to generate realistic vibration spectra as a function of operating condition (RPM)
- *CMAPSS:*
  - –Detection—Is the DDT able to detect the inception of unhealthy behaviour from individual assets when trained on healthy fleet data?
  - –Prognostics—Does the health indicator produce a clear degradation curve over an assets operational life?

–Diagnostics— Is the DDT able to differentiate between different failure modes when producing the HI?

–Robustness—Are the DDT's predictions (HI and generated samples) robust to non-stationary data from various discrete operational settings?

–System Identification—Is the DDT able to generate realistic sensor data over a number of different operational settings?

4.1. Gearbox

The Gearbox data set is representative of the effect of a broken tooth within a one stage spur gear transmission. This gearbox model is presented in Chaari et al. [5], [6]. The model is highly non-linear due to the effect of tooth damage on the vibration of the system as investigated in Chaari et al. [5]. The data used in this investigation was provided by Schmidt [41]. No theoretical investigation was performed on the data, however, the correlation between the performance of fault identification algorithms on this simulated dataset and experimental gearbox data is well documented in Schmidt [41].

4.1.1. Data structure

Each data point within the dataset consists of an acceleration signal $(\mathbf{a}(t))$ (which is treated as the $\mathbf{x}$ parameter) sampled at 50 kHz for one second. The gearbox dataset is representative of the gearbox's acceleration response over a range of different torque and angular velocities. The data consists of healthy gearbox data and unhealthy gearbox data over a range of damage severity ranging from 1% damage to 20% damage.

Shown in Fig. 4 is the multiple degree of freedom (MDOF) non-linear simulation of a one stage spur gear transmission where shafts are treated as torsional springs and the bearing supports as standard springs. The non-linear spring approximation to the gear mesh stiffness $k_{gm}(t)$ is also clearly indicated between the gear and the pinion.

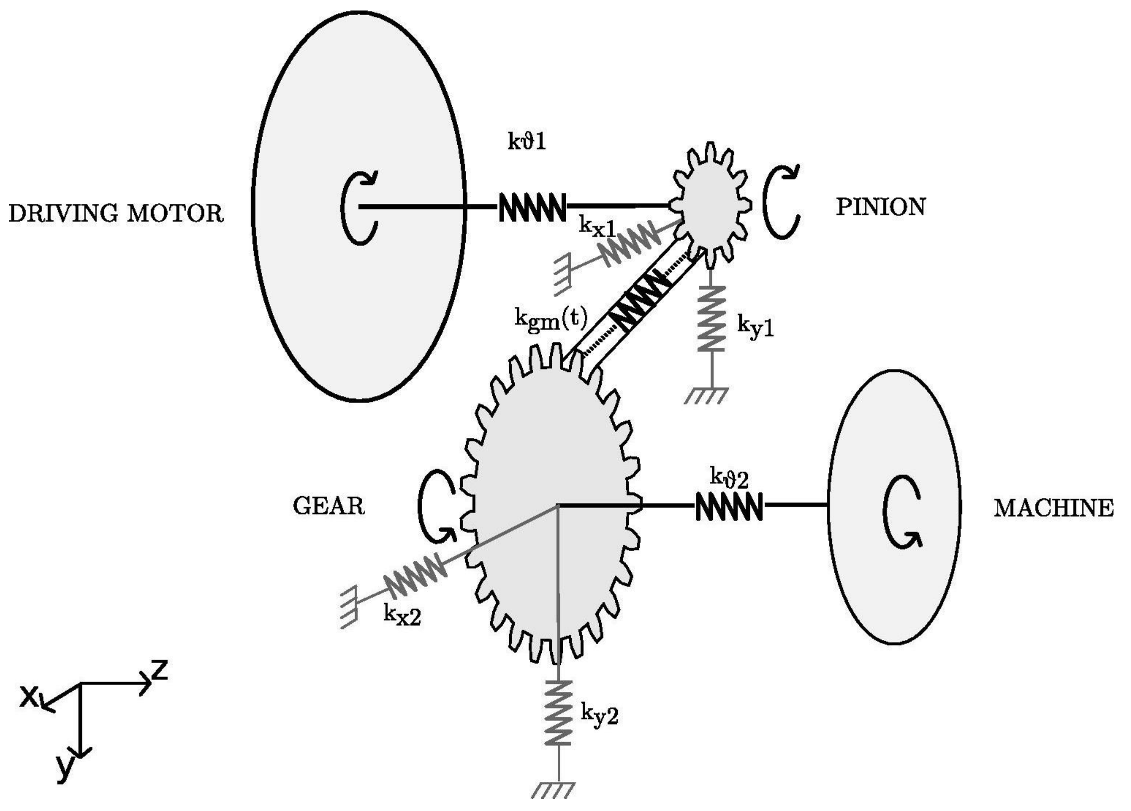


Fig. 4. Diagram of Gearbox Simulation.

4.1.2. Simulated damage

Damage was introduced by means of a reduction in the stiffness of the gear mesh. Fakher et al. [5] demonstrated that there exists a linear relationship between the depth of the crack on a damaged tooth and the gear mesh stiffness. In addition, there exists a critical crack length at which the gear tooth will fail [42]. As a result, the gear mesh stiffness can be seen as representative of the amount of damage in a system. The relationship between damage and gear mesh stiffness is given in [41] as

$$k_{gm}(t) = (1 - \alpha)\hat{k}_{gm}(t),$$

where α corresponds to the % fraction of damage (between 0 and 1) which is inversely proportional to the crack depth and $\hat{k}_{gm}(t)$ is the healthy gear mesh stiffness.

4.1.3. Data preprocessing

As the gearbox dataset consists of observations at various rotational speeds it is important to normalise the data with respect to the rotational speed, known as order tracking. Order tracking involves transforming the data in a manner which represents the inherent periodicity of the data. For a gearbox the inherent periodicity of the data is a function of angular displacement (ψ) as opposed to time (t). In order to perform order tracking it is assumed that the true rotational speed (ω^* which is treated as the conditional parameter $\mathbf{y}$) of the machine is not known exactly, but a noisy approximation (ω) is available. For the purposes of this investigation the rotational speed was sampled from a normal distribution

$$\omega = \mathcal{N}(\omega^*, 5)\ [RPM].$$

Using the known rotational speed and the fact that the speed is assumed to be constant over the observation period it is possible to calculate the angular displacement (ψ) as,

$$\psi = \frac{2\pi}{60}\omega t.$$

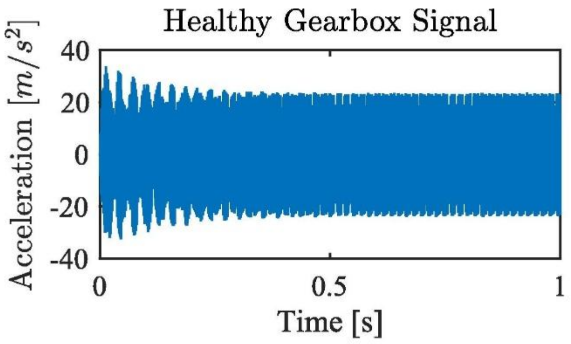
The maximum RPM which was considered was 1500 RPM, equivalent to 25 rotations per second, which corresponds to a maximum 2000 samples per shaft rotation. Consequently, after calculating the angular displacement the signal was re-sampled using linear interpolation at 2048 times per revolution.

After the order tracking process each 1 s sample was then subjected to a time synchronous averaging (TSA) procedure [37]. A TSA procedure involves averaging the signal over the recording period using a window, where the size of the window used for the averaging process is determined by the nature of the data being averaged. For example, in the case of a gearbox the size of the window should correspond to one revolution of the largest gear. The synchronous averaging procedure with an averaging window period Ψ is given by

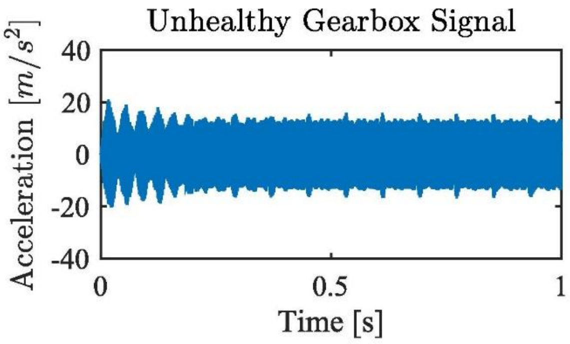
$$\mathbf{a}_{avg}(\psi) = \frac{1}{K}\sum_{k=0}^{K}\mathbf{a}(\psi + k\Psi), \tag{8}$$

where K refers to the number of observed rotations over the one second observation window. Evidently, $\Psi = 2\pi$ corresponds to a window of one revolution.

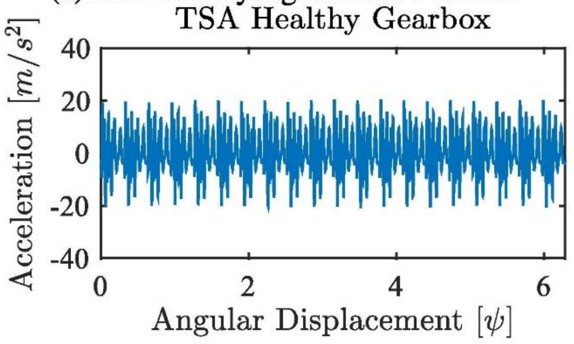
The effect of the order tracking and synchronous averaging procedure on the signal is given in Fig. 5. From Fig. 5 it is clear that the TSA highlights clear periodicity over a single rotation; as well as significantly compressing the data and damping the effects of transients in the signal before 0.25 s.



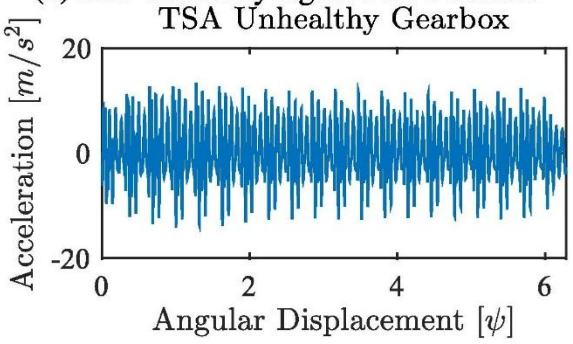
(a) Raw healthy signal over 1 second



(b) Raw unhealthy signal over 1 second



(c) TSA of healthy data over 1 shaft rotation



(d) TSA of unhealthy data over 1 shaft rotation

Fig. 5. Raw accelerometer readings before (a, b) and after (c, d) order tracking and TSA.

4.1.4. Gearbox results

In order to quantify the relationship between damage and the HI, it is necessary to investigate the behaviour of the two DDT implementations on data containing various levels of damage. To this end, the gearbox dataset was constructed with healthy data and unhealthy data of varying severity (1, 5, 10, 15, 20% damage).

This investigation into the performance of the DDT implementations under varying levels of damage aims to establish whether the GAN and the VAE are able to represent increasing levels of damage as a monotonically decreasing HI. This is a highly desirable property for condition monitoring as it allows the severity of the damage to be tracked using the output of the DDT under the assumption that the HI is a continuous metric representing the level of damage present, rather than a binary prediction of failure.

4.1.5. Convergence evaluation

In order to determine the similarity between the simulated and generated spectra we adopt the notion of spectral divergence as discussed in Georgiou [13] as a distance metric. This spectral divergence measure is given in Eq. (9), where S is the power spectral density (PSD) of a signal

$$\log(\rho(S_0, S_1)) = \log\left(\int_{-T/2}^{T/2} \frac{S_0}{S_1} dt\right) - \int_{-T/2}^{T/2} \log\left(\frac{S_0}{S_1}\right) dt. \tag{9}$$

In Georgiou [13] it is shown that Eq. (9) is analogous to the Kullback-Leibler divergence between the two spectra. Moreover, Georgiou [13] also showed that it is possible to calculate the geodesic d_g distance between two spectra and that this metric is analogous to the classical notion of standard deviation, which is evident from

$$d_g(S_0, S_1) = \sqrt{\int_{\pi}^{\pi} \log\left(\frac{S_0(\theta)}{S_1(\theta)}\right)^2 \frac{d\theta}{2\pi} - \left(\int_{\pi}^{\pi} \log\left(\frac{S_0(\theta)}{S_1(\theta)}\right) \frac{d\theta}{2\pi}\right)^2}. \tag{10}$$

Consequently, the metrics from Eqs. (9), (10) may be used to evaluate the degree to which the generated vibration data matches the observed data, and the histogram thereof may be used to detect mode collapse. As the gearbox dataset represents a gearbox at various rotational speeds it is possible to plot a cascade plot of the data which represents the PSD (*S*) as a function of rotational speed. The cascade plot for the simulated healthy and unhealthy data along with the generated samples for the GAN and the VAE (Fig. 6) clearly illustrates that both the GAN and the VAE learn the manifold of the healthy data to some extent. Most notably, both methods learn the relationship between the natural frequencies of the gearbox and RPM, this may be observed at around 50 orders and its harmonics where both algorithms capture the migration of the natural frequency line with changing RPM. From Fig. 6 it is evident that the GAN learns a much 'sharper' representation of the data manifold than the VAE. This is highlighted by the fact that the VAE does not capture the resonance which occurs at a normalised RPM of 0.16. The difference between the representation learned by the VAE and the GAN becomes more apparent when considering the plot of the difference between the generated healthy data for both algorithms and the real healthy data.

In Fig. 7, the VAE's failure to capture the resonance band at a normalised RPM of 0.16 is clearly visible around order 100 where the VAE significantly under-represents the magnitude of the vibrations. On the other hand the GAN seems to exaggerate the resonance band at a normalised RPM of 0.28. The expected spectral difference $\log(\rho(S_{fake}, S_{real}))$ as well as the expected geodesic distance $d_g(S_{fake}, S_{real})$ between the real healthy data and the generated GAN and VAE data is given in Table 1. It is clear that the GAN represents a better approximation of the real data manifold when compared to the VAE, as quantified by the significantly lower divergence variance from the real data by the GAN as compared to the VAE.

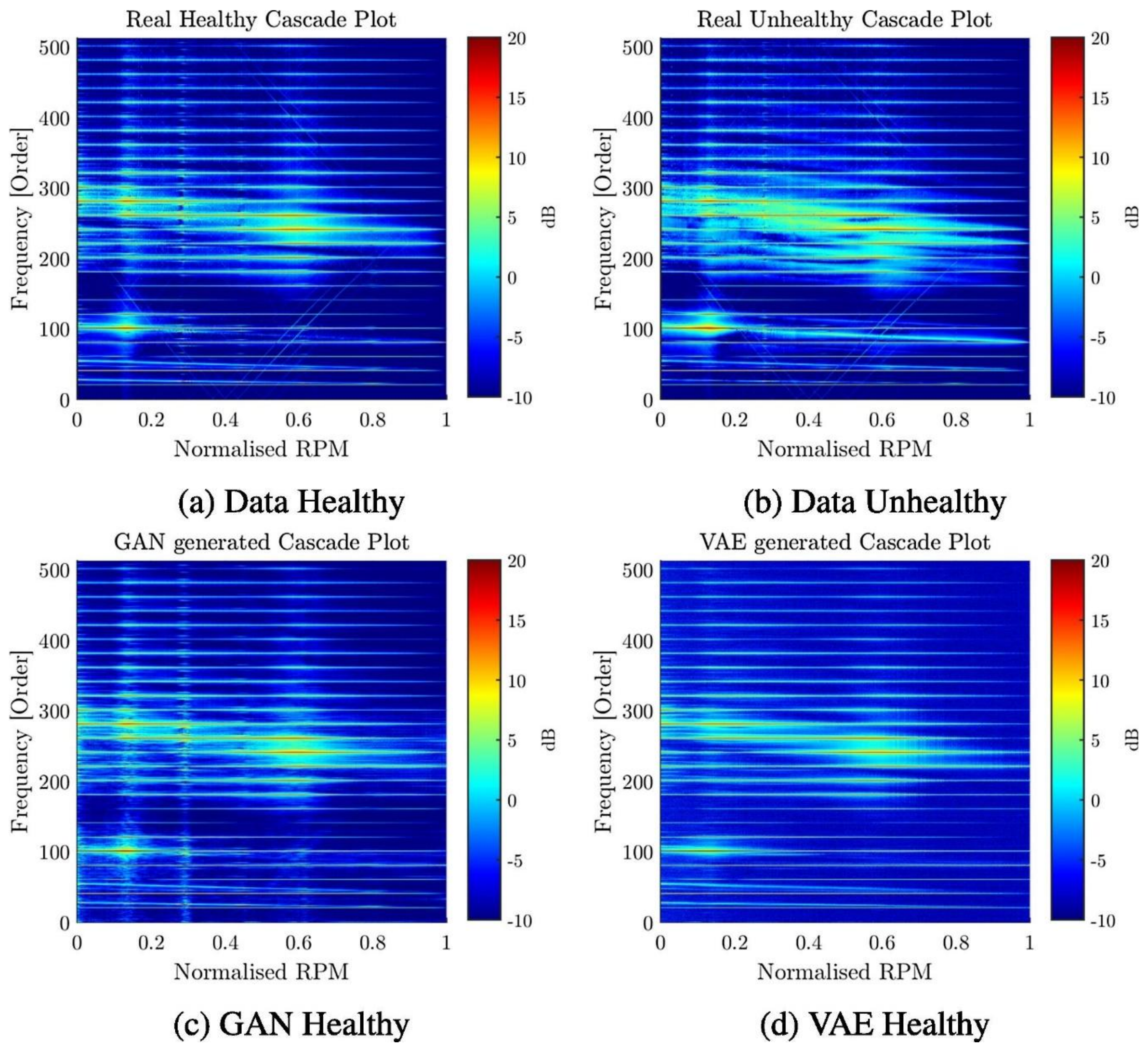


Fig. 6. Cascade plots of the gearbox PSD for the (a) healthy and (b) unhealthy data as well as the healthy generated data from the (c) GAN and the (d) VAE.

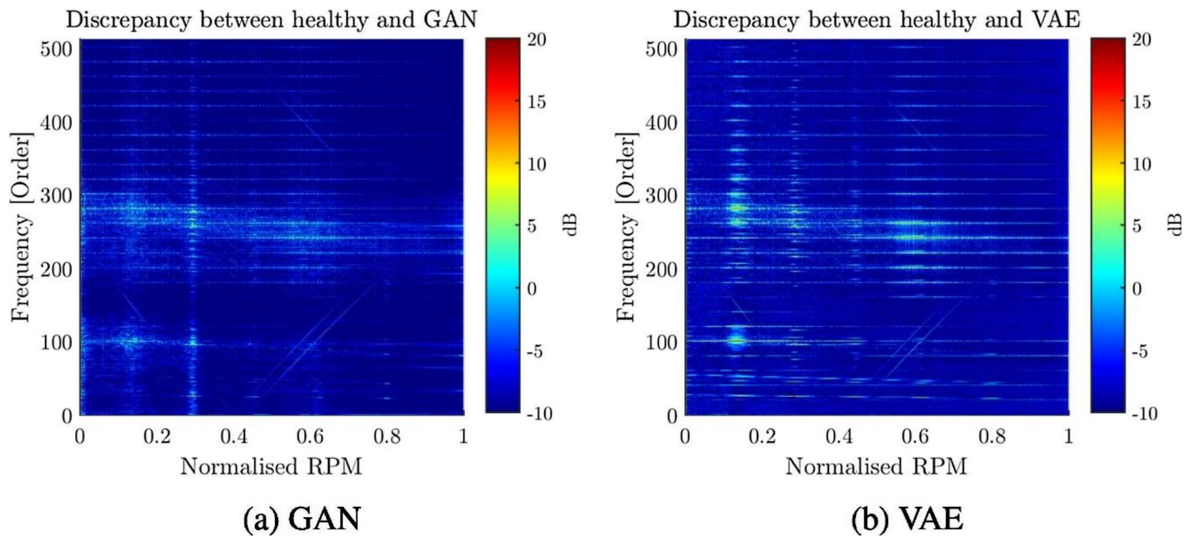


Fig. 7. Cascade plots of the difference between gearbox PSD for the simulated healthy data and the healthy generated data from the (a) GAN and the (b) VAE.

Table 1. Average Spectral Difference and Geodesic Distances PSD over the entire RPM range.

Algorithm	$\log(\rho(S_{fake}, S_{real}))$	$d_g(S_{fake}, S_{real})$
GAN	0.2574	0.7353
VAE	0.4166	0.94
Δ	38%	21.7%

In addition to evaluating the expected Spectral Differences and Geodesic Differences between the GAN and VAE for all RPMs, we also quantify both divergences in terms of their relationship to RPM with variances included as presented in Fig. 8. The expected values are depicted as dark orange and purple, whereas one standard deviation from the expected values are depicted as solid black lines. It is evident that the GAN outperforms the VAE at every RPM.

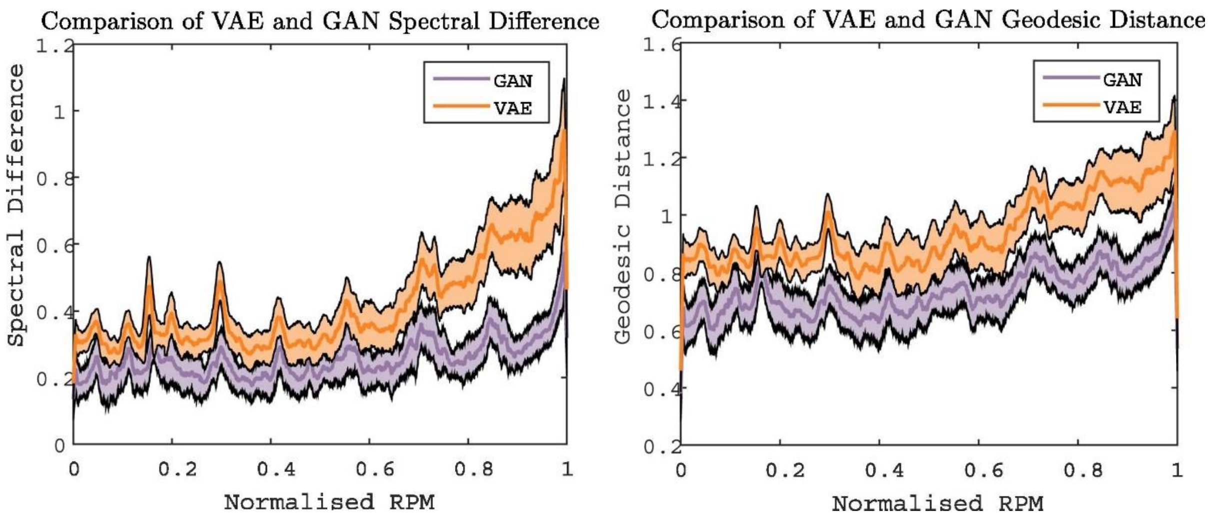


Fig. 8. (a) Spectral differences and (b) Geodesic distances of the GAN and VAE, as a function of RPM for the Gearbox dataset.

4.1.6. Comparison of GAN and VAE health thresholds

After training, both the GAN and the VAE's outputs on healthy data were used to determine the healthy/unhealthy threshold. This threshold was selected arbitrarily as the point at which either model would have a 10% false positive rate. This value corresponded to a GAN HI of 0.5842 and a $p_\theta(\mathbf{x})$ of 0.975. At this threshold the GAN significantly outperforms the VAE obtaining a true positive rate of 84.8% compared to the VAE's true positive rate of 61.4%. The cumulative distribution of the healthy data is given in Fig. 9a and b for the GAN and the VAE respectively.

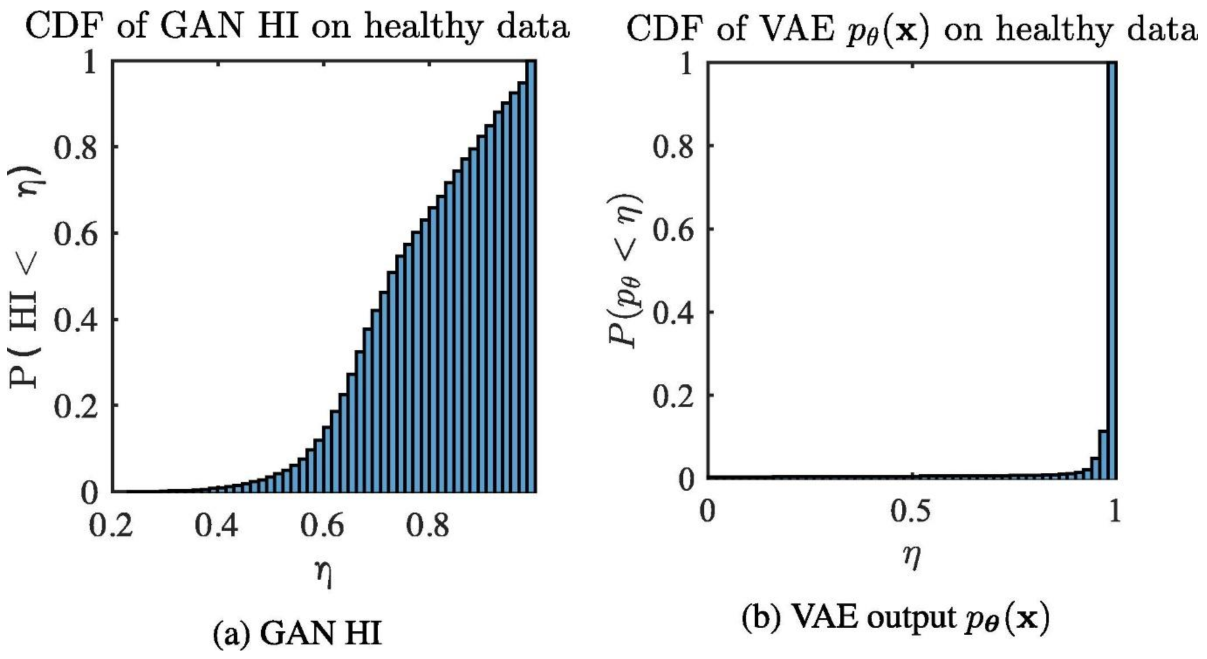


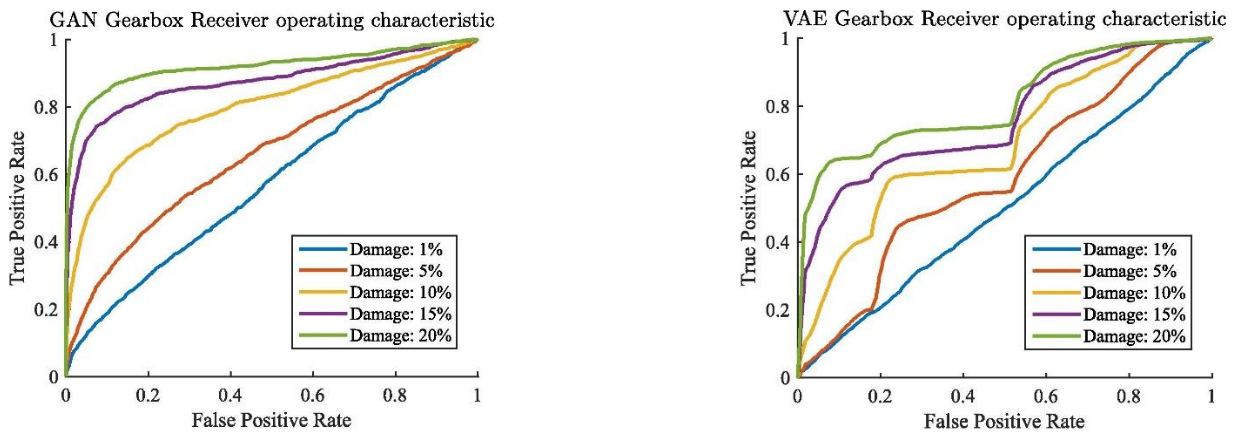
Fig. 9. Cumulative Distribution Function of DDT over the training data for both the GAN and VAE.

From the cumulative distributions given in Fig. 9a and b it is clear that the VAE assigns high probabilities to healthy data, the distribution p_θ generated by the VAE has a low variance with more than 90% of the training data having values of $p_\theta > 0.975$. As opposed to the VAE, the GAN has a much larger variance on its outputs. Using the threshold of a 10% false positive rate it is clear that the GAN significantly outperforms the VAE, particularly for the lower levels of damage.

4.1.7. Receiver operating characteristic curve comparison

Laslty we consider the practical performance differences between the GAN and VAE at various damage levels and threshold levels of $p_\theta(\mathbf{x})$, by plotting receiver operating characteristic (ROC) curves. ROC curves depict the relationship between the true positive rate (TPR) against the false positives rate (FPR) as $p_\theta(\mathbf{x})$ and damage is varied. The TPR indicates the probability of detection, while the FPR indicates the probability of a false alarm, essentially a horisontal straight line at TPR = 1 is the ideal ROC curve.

The GAN and VAE ROC curves at damage levels of 1, 5, 10, 15, 20% are depicted in Fig. 10. From Fig. 10 it is interesting to note that the VAE ROC has a significant increase in the TPR at a FPR of approximately 0.55. This suggests that some of the failure data while being clustered around one value of $p_\theta(\mathbf{x})$ is deeply embedded within the distribution of healthy data. This embedding of the unhealthy data within the distribution of healthy data seems to affect the performance of the VAE.



(a) GAN (b) VAE

Fig. 10. Receiver operating characteristic for the GAN and VAE.

4.1.8. Degradation tracking

In addition to comparing the performance of the VAE and the GAN for various levels of damage, the gearbox dataset was also constructed to test whether either model is able to track degradation. Effective degradation tracking requires that the output of the algorithm—be it the HI or $p_\theta(\mathbf{x})$—decreases monotonically as the level of damage increases. The relationships between the DDT health indicators (p_θ and GAN HI) and the level of damage are depicted in Fig. 11. From Fig. 11 it is clear that both models display a monotonically decreasing expected value of the output for increasing levels of damage. However, the GAN seems to maintain a constant variance over the levels of damage, as opposed to the VAE which displays a significant increase in the variance of its outputs as the damage increases. Maintaining a constant or decreasing variance as the damage increases is highly desirable, as this enhances identification potential of faults as well as the reliable tracking of asset degradation. On the other hand the increase in the VAE's variance for increased damage is problematic as samples drawn from the distribution of unhealthy data at a damage level of 0.2 overlaps significantly with samples drawn at lower damage levels. In other words, the amount of overlapping support between healthy distributions and unhealthy distributions does not decrease significantly as damage increases for the VAE, which is in contrast to the GAN where overlapping support between distributions for different damage levels decreases significantly.

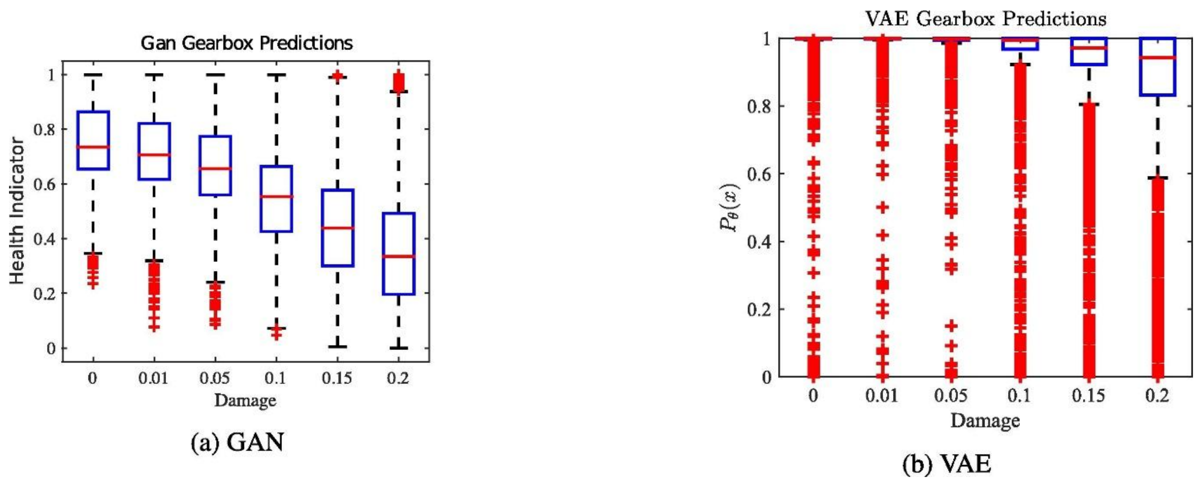


Fig. 11. Boxplot of GAN and VAE outputs as a function of the level of damage present in the system.

4.1.9. Discussion

The gearbox dataset was selected as a benchmark dataset to compare the explicit deep digital twin (VAE) to the implicit deep digital twin (GAN) as well as evaluating the generative, detective and prognostic abilities of the DDT under non-stationary operating conditions. From the investigation it is evident that although both implementations are able to learn the manifold of healthy data to some extent over the entire RPM range. The investigation into the behaviour of the DDT under varying levels of damage demonstrates that the implicit digital twin is significantly more sensitive to deviations from healthy behaviour compared to the VAE. It is suspected that this behaviour stems from the VAE's optimisation of the least squared error as opposed to the GAN's adversarial loss. In addition both DDT implementations seemed to be robust to the non-stationary nature of the data. Following the investigation on the Gearbox dataset, subsequent experiments will focus on demonstrating the capabilities and performance of the Implicit DDT.

Capability assessment:

- Detection—From Fig. 10 it is evident that both the implicit and explicit DDT are able to detect the presence of a fault when trained only on healthy data.

- Prognostics—From Fig. 11 it may be observed that both DDT health metrics increase proportionally to the severity of the fault.

- Robustness—As the faults detected in this investigation occurred at various RPMs it is clear that the learned health indicators are invariant to the operating condition.

- System Identification—From Fig. 6 it is evident that although the models are only trained on the acceleration data waveforms, the spectra of the sampled waveforms is consistent with the spectra of the real data. Albeit more so for the GAN than the VAE.

4.2. CMAPSS

4.2.1. Dataset overview

The turbofan dataset consists of simulated sensor data of the entire life-cycle of multiple turbofan assets. The data is generated using the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) developed by NASA. Each example within the dataset is a time series signal of various sensor data and operating condition data [36] which is measured periodically over the life-cycle of the turbofans.

4.2.2. Simulation

The dataset consists of a number of simulated turbofan engines each with varying internal initial conditions. This is to simulate real world engines which will invariably have different initial wear due to manufacturing differences. In addition, the sensor data is severely corrupted by noise. The data from the simulation which is included in the dataset is not vibration data but rather sensor data from the aircraft such as fan speed, pressures and turbofan temperatures. Each full run to failure trajectory of a turbofan is considered to be a single temporal observation which consists of a number of observations over time.

4.2.3. Dataset characteristics

Effectively, CMAPSS dataset contains 4 distinct datasets of varying complexity and failure characteristics. Datasets 1 and 3 represent only one operating condition. That is, all data was collected at constant altitude, Mach Number and throttle. Datasets 2 and 4 contain data from turbines at varying operating conditions throughout the life of the engine. In datasets 1 and 2 the only failure mode is high pressure compressor degradation whereas datasets 3 and 4 contain an additional failure mode—fan degradation. The characteristics of the datasets are summarised below in Table 2 [35].

Table 2. Characteristics of the CMAPSS datasets.

Dataset	No. of Failure modes	No. of Operating Conditions
#1	1	1
#2	1	6
#3	2	1
#4	2	6

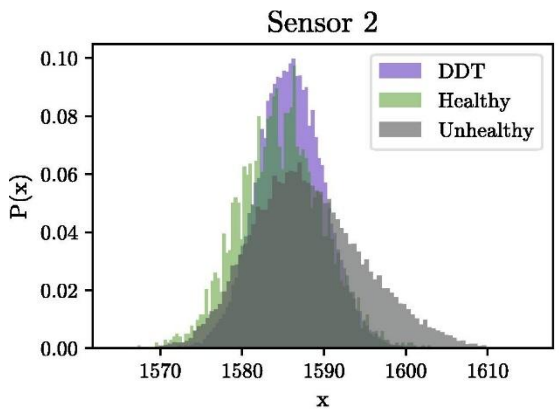
4.2.4. Investigation

For the CMAPSS dataset the implicit DDT implementation was evaluated on all four sub-datasets, encompassing each possible failure mode and operating condition. It is hypothesised that a GAN trained on the first part of the time series data will be able to represent the state of damage within the system as well as learning the behaviour of a healthy asset. To investigate this, a GAN was trained on the sensor values[1] of the first 20 cycles of the Turbofan dataset (first 5–10% of life), as these are assumed to be representative of healthy turbofans. This is opposed to conventional approaches that use full run to failure cycles in order to determine a HI [35], [36].

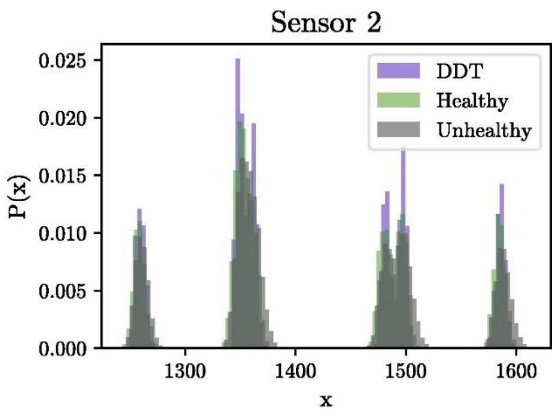
Once trained, it is possible to sample generated sensor responses from the GAN at various operating conditions. As the turbofans degrade over time the Health Indicator of the GAN should decrease once a fault manifests and continue to decrease as the turbofans degrade to the end of life.

4.2.5. System identification

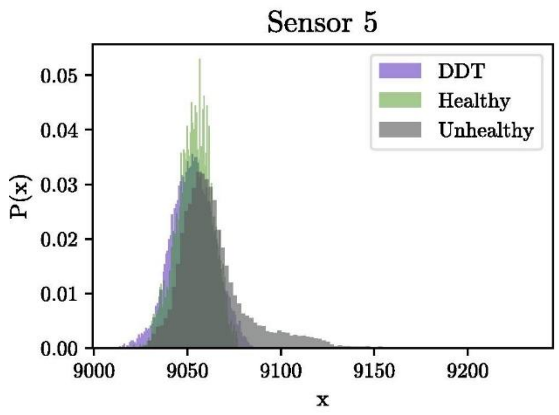
In order to determine whether the GAN is indeed able to capture the sensor response of the CMAPSS engines under the various operating conditions the simulated and real sensor values are compared in Fig. 12. From Fig. 12, which is a histogram of the sensor values over all times, it is evident that the GAN is able to learn the distribution of healthy data for engines with one or multiple operating conditions.



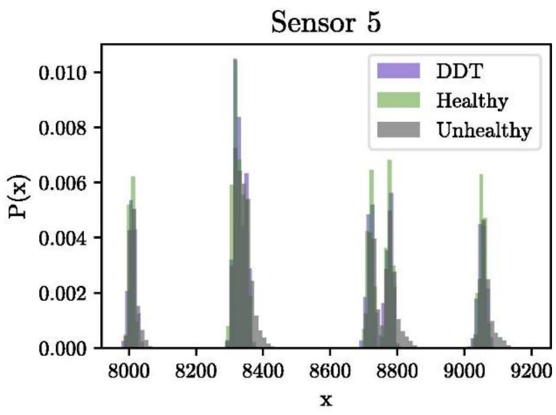
(a) Sensor 2 Response Dataset 3



(b) Sensor 2 Response Dataset 4



(c) Sensor 5 Response Dataset 3



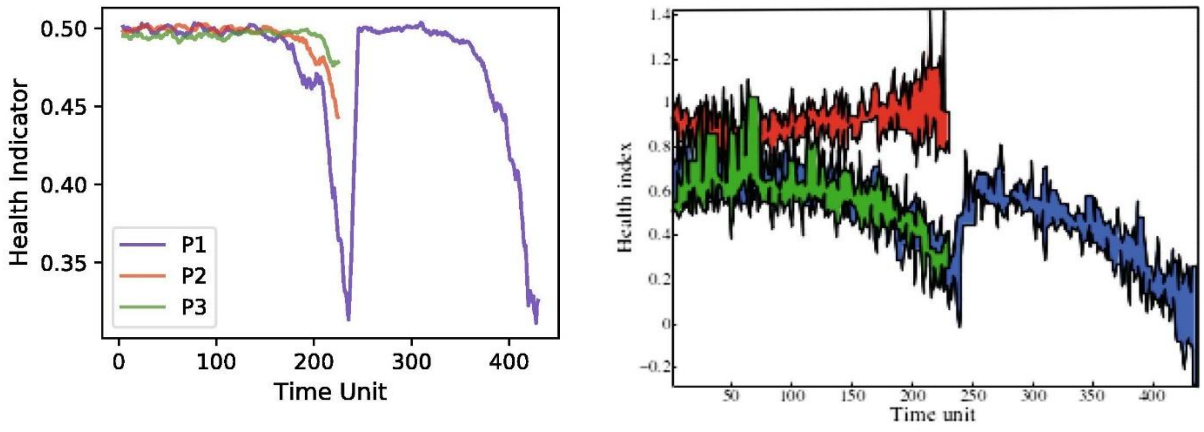
(d) Sensor 5 Response Dataset 4

Fig. 12. Histogram of sampled sensor values from GAN compared with real healthy and unhealthy sensor values from datasets 3 and 4.

4.2.6. Condition monitoring

In addition to learning the distribution of healthy data, the GAN is also able to provide a non-parametric HI which is indicative of the current level of degradation in the system through the discriminator. In order to evaluate the efficacy of the GAN in generating a HI, the health of the DDT may be compared to the HI produced by the RULCLIPPER algorithm of Ramasso [35] which currently is the best performing algorithm on the CMAPSS dataset. To this end, Figs. 1 and 9 in Ramasso [35] are recreated in Fig. 13, Fig. 14. In Ramasso [35] the three cases $P1$ (Run 178 Train Dataset #4,[2] $P2$ (Run 100 Train Dataset #4) and $P3$ (Run 1 Test Dataset #4 RUL:22) are shown in Fig. 13. Each were selected to demonstrate a pathological limitation of the health indicator used by the RULCLIPPER algorithm:

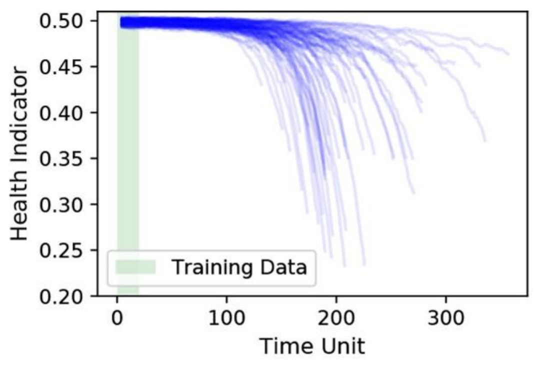
- **P1**: Ramasso [35] noted the "Healing" behaviour of the asset at $t \in [225, 275]$, which was attributed to a change in the operating conditions. The healing behaviour is to be expected as the authors of Saxena et al. [40], from which the dataset stems, explain that in the creation of the data the fault severity is a function of the operating conditions.
- **P2**: Ramasso [35] observed that the estimated HI for this engine increases as the engine degrades as opposed to decreasing. This behaviour is not observed for the GAN output and the HI correctly decreases as expected.
- **P3**: Ramasso [35] observed a large variance in the HI between $t \in [10, 75]$. This phenomenon was not observed in the HI generated by the GAN.



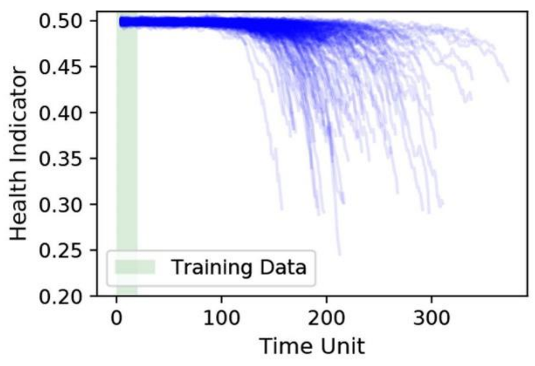
(a) GAN non-parametric Health Indicator (HI)

(b) Figure 1 From Ramasso (2014)

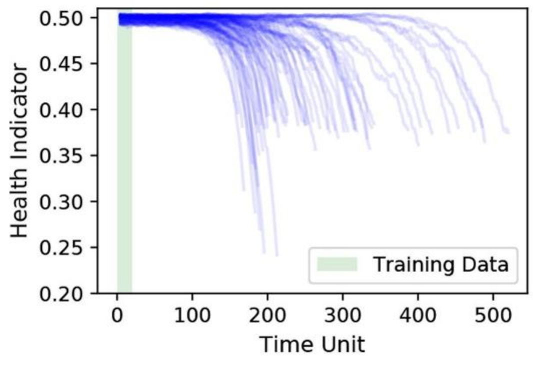
Fig. 13. Recreation of Fig. 1 from Ramasso [35] compared to the GAN Health Indicator (HI).



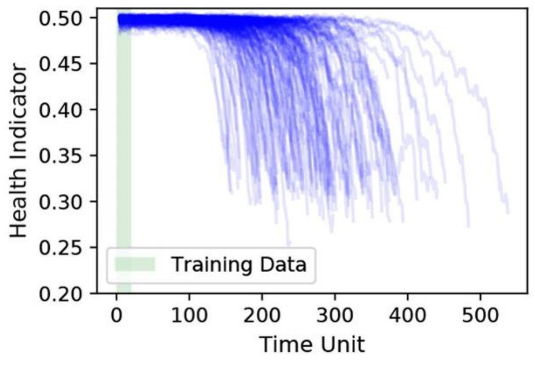
(a) GAN output for fleet #1



(b) GAN output for fleet #2 (Figure 9 From Ramasso (2014))



(c) GAN output for fleet #3



(d) GAN output for fleet #4

Fig. 14. Output of the GAN Health Indicator (HI) for all four CMAPSS datasets over engine life cycle. Data used for training the GAN is highlighted in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 9 of Ramasso [35] displays all the HI outputs for the fleet of dataset #2. It may be observed that the output of the GAN shown in Fig. 14 is visually similar despite the fact that it was only trained on 8% of the dataset as opposed to using the full dataset. One clear difference between the GAN output and the HI from Ramasso [35] is the behaviour of the HI in the beginning of life. The HI from Ramasso [35] is obtained via least squares minimisation in order to fit the health index to the functional form

$$\widehat{\mathbf{HI}} = 1 - \exp\left(\frac{\log(0.05)}{0.95T_i}t\right), \quad t \in [0.05T_i, 0.95T_i]. \tag{11}$$

This is opposed to the GAN HI, which does not assume a functional form for the HI. A limitation of the loss minimised by Ramasso [35] is that the model is constrained to generate a HI of a fixed functional form e.g. exponential as in Eq.

(11). This is problematic as the assumed functional form may not be representative of the true degradation pattern of the data.

The degradation pattern of the CMAPSS dataset is given in Saxena et al. [40], where the true health indicator $\widehat{\mathbf{HI}}$ is defined as:

$$\widehat{\mathbf{HI}} = \min(m_{Fan}, m_{HPC}, m_{HPT}, m_{EGT}), \tag{12}$$

where $[m_{Fan}, m_{HPC}, m_{HPT}, m_{EGT}]$ correspond to the stall margins of 4 components—the two failure modes from dataset #4 are Fan failure and HPC (High-Pressure Compressor). The failure modes are characterised by flow $f(t)$ and efficiency $e(t)$ parameters defined as:

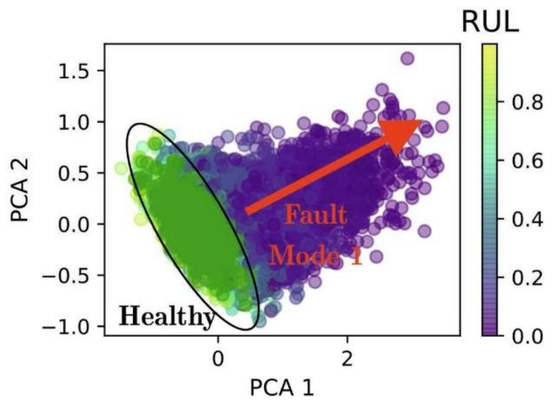
$$e(t) = 1 - d_e - \exp\left(a_e(t)t^{b_e(t)}\right) \tag{13}$$

$$f(t) = 1 - d_f - \exp\left(a_f(t)t^{b_f(t)}\right) \tag{14}$$

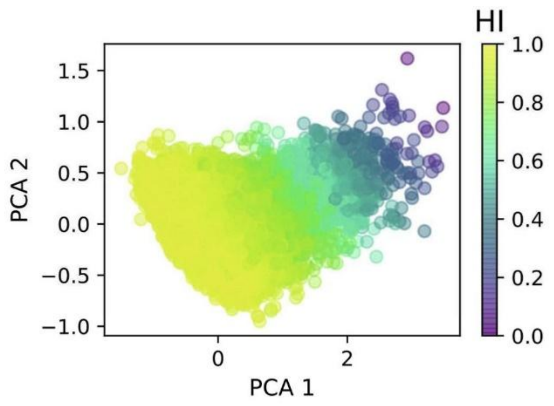
where d_e and d_f are random initial wear parameters defined by the generalised exponential wear equation, $w = Ae^{B(t)}$. It follows that degradation trajectories defined by Eqs. (13), (14) could produce a similar pattern to the trajectories shown in Fig. 14 as the $\widehat{\mathbf{HI}}$ produced by Eq. (12) would remain constant until the wear of the degrading component becomes larger than the initial wear of all the other components [14].

4.2.7. Data manifold

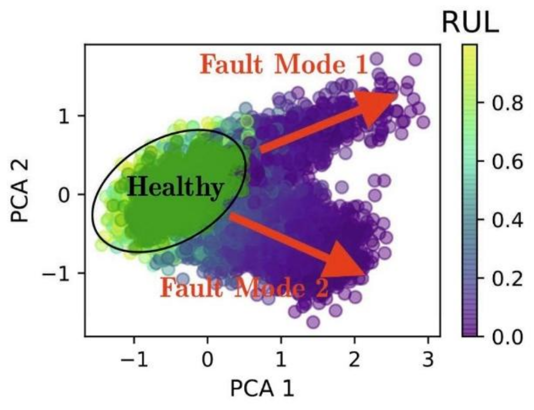
It is expected that the GAN is able to capture the manifold of healthy data and be able to represent the manner in which the observed data deviates from the learned manifold. In order to ascertain whether the GAN is indeed able to distinguish between different failure modes it is useful to visualise the embedding of degradation trajectories in the feature space of the last layer of the GAN discriminator. This visualisation is given below in Fig. 15 using the first two principal components to project the high dimensional space into 2D. By inspection it is clear that the GAN learns a representation able to automatically differentiate between the failure modes. Hence the Discriminator architecture once trained can easily be used for diagnostics if labels are available by simply using the weights of the DDT discriminator and replacing the 1 dimensional output layer with a K dimensional output layer which can then be fine tuned (where K corresponds to the number of failure modes).



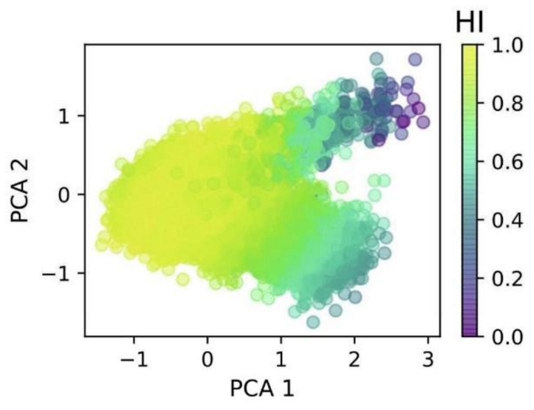
(a) True RUL in discriminator feature space for fleet #1 (1 fault mode present)



(b) HI value in discriminator feature space for fleet #1 (1 fault mode present)



(c) True RUL in discriminator feature space for fleet #2 (2 fault modes present)



(d) HI value in discriminator embedding space for fleet #2 (2 fault modes present)

Fig. 15. 2D PCA plot of the embedding learned by the last layer of the GAN discriminator (8 hidden units).

4.3. Discussion

The CMAPSS dataset was primarily selected to investigate the ability of the DDT to represent the state of damage of an asset over its operational lifespan. From this investigation it is clear that the DDT is able to produce a health indicator which is consistent with the expected degradation profiles of the dataset without using entire run-to-failure examples nor assuming a functional form for the degradation. Moreover the DDT is able to generate sensor data which is indistinguishable from healthy sensor data for a number of different discrete operating modes. This investigation has also demonstrated that the implicit DDT learns a representation of health which is sensitive to the nature of the deviation, allowing it to be used as a diagnostic tool.

Capability Assessment:

- Detection—From Fig. 14 it is clear that when trained less than 5% of the initial life of the asset the DDT is able to maintain its healthy state until the initial inception of the fault at which the health indicator begins to decrease rapidly.

- Prognostics—From both Fig. 13, Fig. 14 it is clear that the DDT is able to output a health indicator that decreases proportionally to the level of degradation present.

- Diagnostics— From Fig. 15 it is demonstrated that the embedding learned by the DDT is able to map different between degradation trajectories into different regions of the learned embedding space.

- Robustness—From Fig. 14 it is apparent that the DDT's health indicator is invariant to the varying operating conditions and allows for health to be compared over various operating conditions.

- System Identification—From Fig. 12 it is clear that the DDT is able to learn the distribution of healthy data under both stationary and non-stationary operating conditions. In addition it is apparent that mode collapse has not occurred as all operational modes are learnt by the DDT.

5. Real experimental data: IMS bearing dataset

The IMS dataset [34] was selected to evaluate the performance of the DDT on a real asset over its operational life. In addition the IMS data was selected to investigate the behaviour of the DDT when producing multiple independent health indicators from different sensors on the same asset. The IMS dataset contains the vibration signals of a shaft with four bearings on it over its entire lifespan. Of the four bearings, B3, B4 failed at the end of the test and the other two bearings (B1, B2) remained healthy until the experiment was stopped.

The experimental set up consists of a rotating shaft which is connected to four Rexnord ZA-2115 bearings [34] corresponding to B1, B2, B3 and B4 as shown in Fig. 16. Each of the four bearings were monitored by two PCB 353B33 High Sensitivity Quartz ICP accelerometers [34]. Each accelerometer took 1 s acceleration recordings every 20 min, at a sampling rate of 20480 Hz. The accelerometer S1 observed vibrations in the x direction and S2 observed vibrations in the y direction. During the experiment the shaft was driven at a constant angular velocity of 2000 RPM and was loaded by 26.7 kN distributed over B2 and B3. According to Qiu et al. [34] the experiment was carried out for 35 days in total and was stopped when 'a significant amount of metal debris was found on the magnetic plug of the test bearing.' The state of each bearing at the end of the trial is tabulated in Table 3.

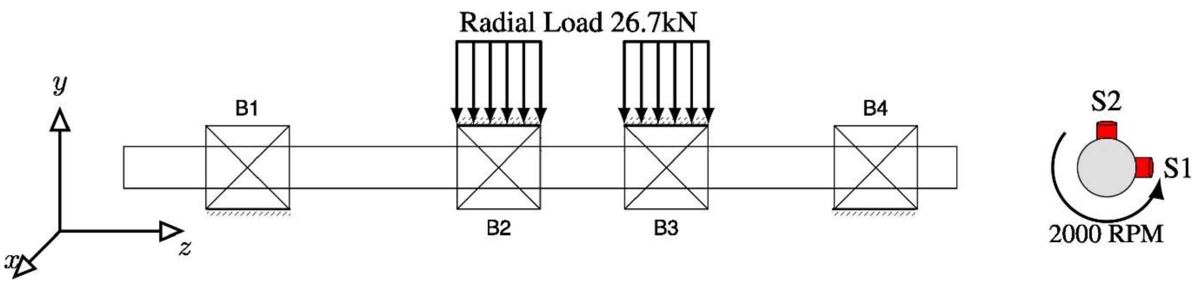


Fig. 16. Diagram of IMS Bearing Dataset experimental set up.

Table 3. IMS Bearings' health at end of test.

Component	Health	Nature of Failure
B1	Healthy	–
B2	Healthy	–
B3	Failed	Inner race defect
B4	Failed	Roller element and outer race defects

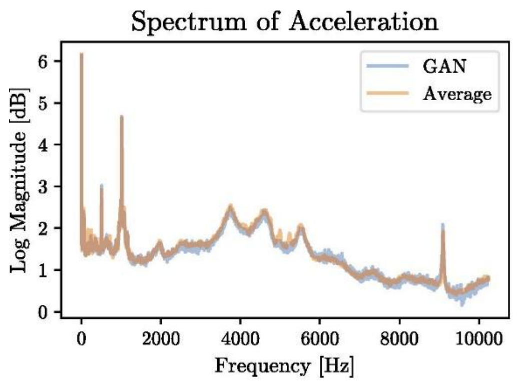
5.0.1. Data preprocessing

The bearing dataset consists of 2156 observations of length 20480 per sensor per bearing over 35 days. This was considered in [11] to be insufficient for data driven prognostics, however it is proposed that this dataset should be sufficient as it corresponds to 71867 observed revolutions per sensor per bearing—as the revolutions represent the inherent periodicity which the GAN needs to learn.

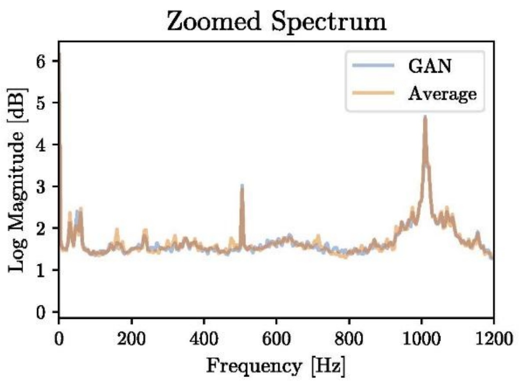
In order to present the data to the GAN, the one second signals of 20480 observations are randomly cropped to length 4080 using a randomly positioned window over the entire 20480 signal. This is to reduce computational time and to improve the diversity of the training samples. Once again, the GAN was only trained on healthy data (first 5 days of asset life).

5.1. IMS dataset results

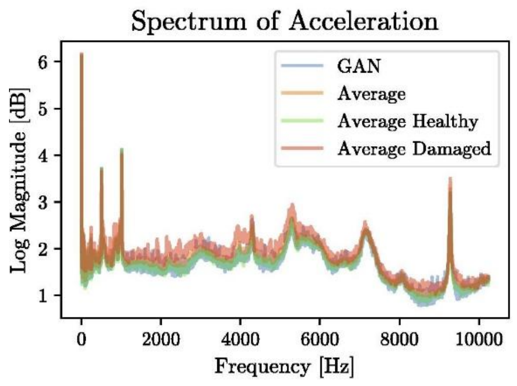
For the IMS dataset each bearing was coupled with 2 DDTs, one for each sensor (in total 8 DDTs). This was done to investigate the level of agreement between two independent DDT's trained on the same component but with different sensors. In addition, as GAN training is path dependent, training two DDTs on the same underlying asset with different sensors gives an indication of the sensitivity of the produced HI to the training process.



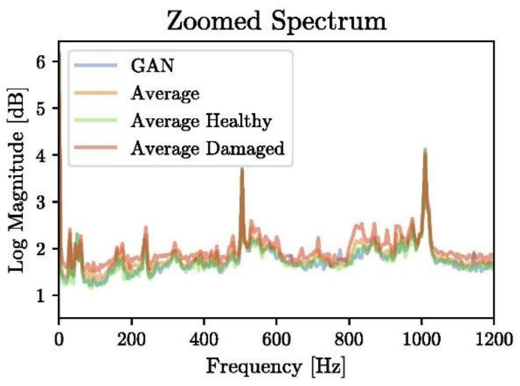
(a) Spectrum for bearing 1 sensor 1



(b) Zoomed spectrum for bearing 1 sensor 1



(c) Spectrum for bearing 3 sensor 3



(d) Zoomed spectrum for bearing 3 sensor 3

Fig. 17. PSD for (a)-(b) bearing 1 sensor 1 and (c)-(d) bearing 3 sensor 3 depicting the GAN predicted spectrum, average, healthy and unhealthy PSD.

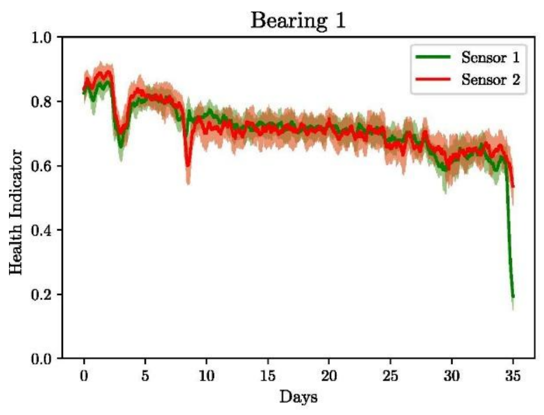
5.1.1. Convergence evaluation

To demonstrate that the GAN is able to capture the manifold of healthy data, consider the PSD given in Fig. 17 for bearing 1 sensor 1 and bearing 3 sensor 3. It is clear, from Fig. 17, that the spectrum of the vibration signals generated by the GAN are essentially indistinguishable from the spectrum of the true healthy vibrations. It is also evident from Fig. 17 that the spectrum for unhealthy data deviates from both the healthy real and generated signals. Evidently, the GAN is a representative DDT instance of the asset as captured by the information of the respective sensors. To conclude the study we investigate and demonstrate that the GAN is able to track continuous degradation of assets, which are the four bearings in this case. This is in contrast to the fixed levels of damage investigated previously on the simulated gearbox dataset.

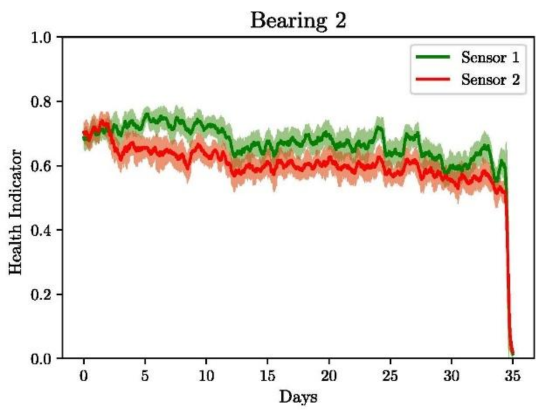
5.1.2. Degradation tracking

5.1.2.1. Bearing 1

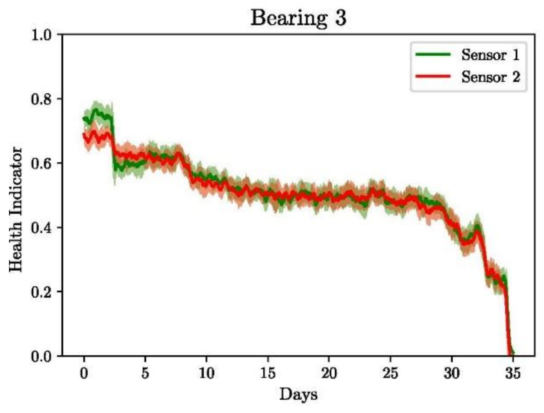
Bearing 1 did not fail during the run till failure experiment, this is consistent with the HI in Fig. 18a of both GANs. In Fig. 18a it is clear that the HI decreases from 0.85 to around 0.6, over the course of the experiment until near total system failure. In the time directly preceding total system failure the HI on sensor 1 drops significantly to around 0.2, this is most probably due to the system approaching failure which results in unhealthy behaviour elsewhere in the system leaking over into the other bearings.



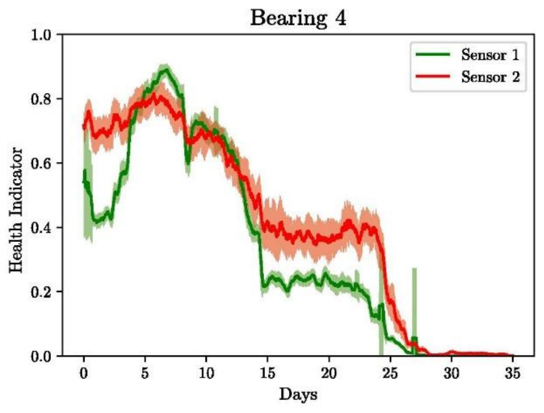
(a) Plot of bearing 1 HI over experiment



(b) Plot of bearing 2 HI over experiment



(c) Plot of bearing 3 HI over experiment



(d) Plot of bearing 4 HI over experiment

Fig. 18. Health indicator over the duration of the entire experiment.

5.1.2.2. Bearing 2

Bearing 2 displays similar behaviour to bearing 1 as both bearings did not fail during the run to failure experiment. It is clear in Fig. 18b that the HI decreases from around 0.75 near the beginning of life to around 0.6, near total system failure. In the time directly preceding total system failure the HI drops to near zero. As bearing 2 is closer to the failed bearings (3,4) than bearing 1, it is expected that the unhealthy response before total system failure is significantly more apparent in the HI of bearing 2 than bearing 1.

5.1.2.3. Bearing 3

Bearing 3 experienced an inner race defect which is clearly represented in both GANs as the HI degrades from around 0.75 to 0.2 near the end of life and falls to zero immediately before total system failure. Both GANs display a similar output over the entire bearing life, with both HIs decreasing monotonously to zero over the bearing life.

5.1.2.4. Bearing 4

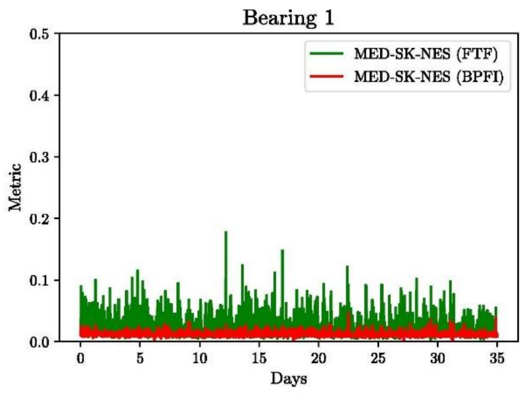
Bearing 4 failed due to both an outer race and roller element defect, the degradation of the bearing is well represented in the HI for the GANs for both sensors. Although the correlation between the output of the GANs for both sensors is the weakest for bearing 4 compared to the other bearings. There is still a clear pattern to the degradation where the HI initially falls sharply then recovers and degrades again. In the initial investigation by the researchers who compiled the IMS dataset the researchers also found a “healing” phenomenon which they attributed to the damage propagation procedure the fault [34].

Evidently, the GAN trained on healthy data can be interrogated continuously to estimate the condition of an asset and hence track asset degradation, delivering a DDT instance that can be deployed for PHM online monitoring or at in-field on the Edge Deployment.

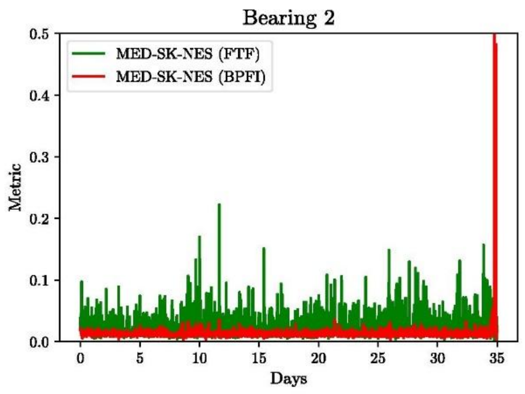
5.2. Comparison

In order to verify the performance of the GAN based DDT on the bearing dataset, the performance of the model is compared to two state of the art signal processing techniques. Namely, cyclostationary improved envelope spectrum (IES) and minimum entropy deconvolution – spectral Kurtosis – normalized squared magnitude of the squared envelope spectrum (MED-SK-NES). These two techniques are explored in detail and benchmarked on the same dataset in [1].

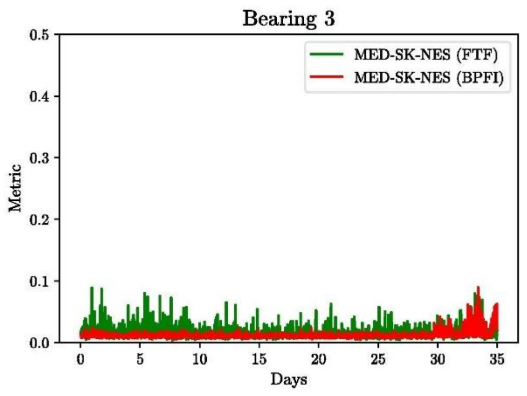
For comparative purposes the two techniques are implemented for both the bearing fundamental train frequency (FTF) and the inner ball pass frequency (BPFI). It should be noted that the implementation of these two techniques requires prior knowledge regarding the geometry of the bearing as well as a pre-emption of the nature of the fault which will occur.



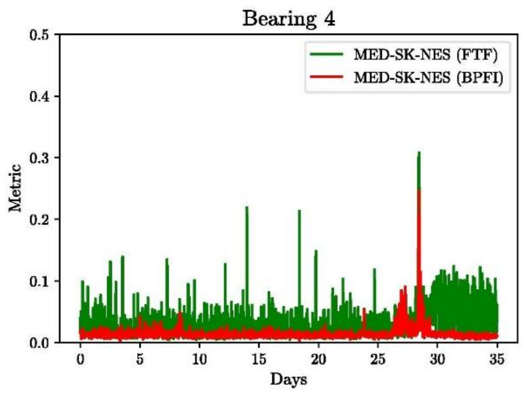
(a) Plot of bearing 1 MED-SK-NES over experiment



(b) Plot of bearing 2 MED-SK-NES over experiment



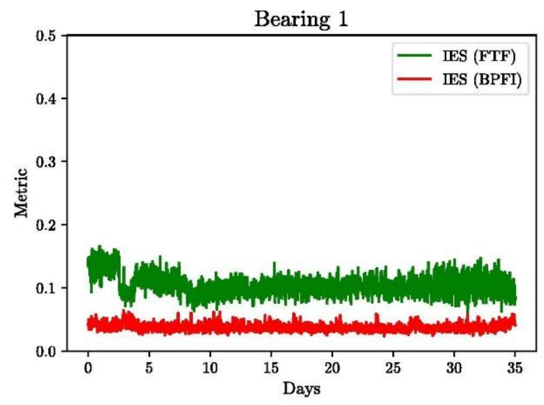
(c) Plot of bearing 3 MED-SK-NES over experiment

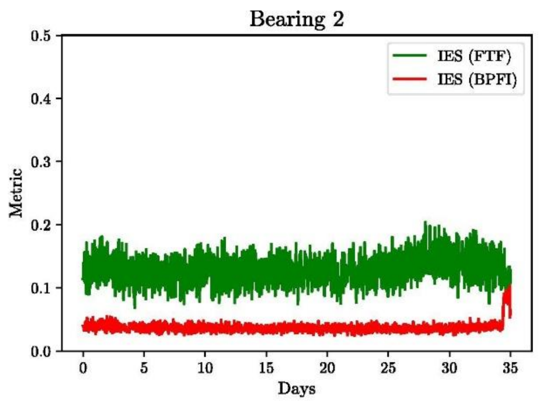


(d) Plot of bearing 4 MED-SK-NES over experiment

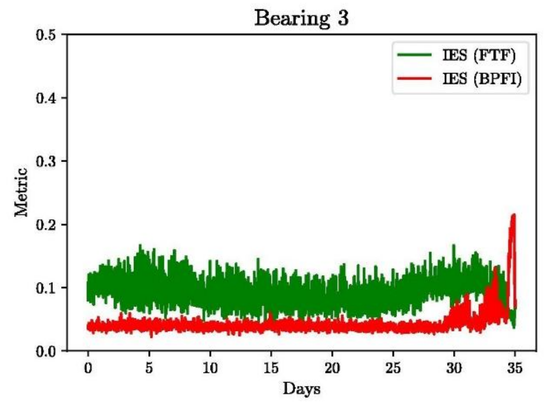
Fig. 19. MED-SK-NES degradation indicator over bearing experiment for both the fundamental train frequency (FTF) and the inner ball pass frequency (BPFI).

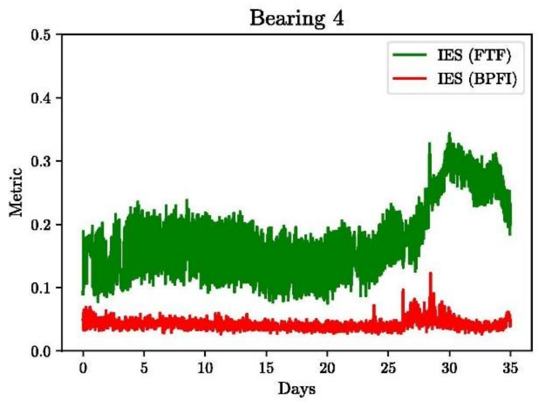
From Fig. 19, Fig. 20 it is observed that the signal processing based methods are indeed able to detect the faults. However, it is clear that the indicators are significantly less sensitive to the initial degradation than the proposed DDT based approach. Moreover, the nature of the fault dictates which of the fundamental frequencies (BPFI, FTF) will carry the fault signal. For bearing 3 the BPFI based metric is more sensitive for both methods and for bearing 4 the FTF is more sensitive for both methods.





(a) Plot of bearing 1 IES over experiment (b) Plot of bearing 2 IES over experiment





(c) Plot of bearing 3 IES over experiment (d) Plot of bearing 4 IES over experiment

Fig. 20. Cyclostationary IES degradation indicator over bearing experiment for both the fundamental train frequency (FTF) and the inner ball pass frequency (BPFI).

Although both methods clearly demonstrate anomalous behaviour near the end of life they do not demonstrate the early degradation behaviour observed in Fig. 18c and d.

5.3. Discussion

It is clear that the DDT is able to produce a health indicator which is consistent over numerous sensors and demonstrates a clear degradation trend. This behaviour is most apparent in Bearing 3 where the outputs for the health indicator for the different sensors is almost identical. For Bearing 4 although the HIs do not overlap entirely the overall form of the trend is consistent. From the bearing investigation it is clear that a DDT can be implemented 'on-line' with no historical data from similar assets or fleets or prior information regarding the asset geometry.

The DDT based indicator was compared to advanced signal processing techniques and demonstrated higher sensitivity to the faults. Moreover the DDT provides a monotonically decreasing degradation metric near the end of life as opposed to the signal processing based approaches which do not provide degradation trends which monotonically decrease from near end of life to end of life. These characteristics of the DDT make it a powerful choice for prognostics applications as the prediction of remaining useful life requires a metric which reaches a minimum immediately before failure.

The ability of the DDT to embed different failure modes into different regions of its final layer embedding is also demonstrated in detail in Baggeröhr et al. [2].

6. Discussion

This study demonstrated the role deep learning can play in PHM to construct Deep Digital Twin (DDT) instances that are representative of the information manifold of healthy data embedded in the data captured by a sensor network of the asset for two synthetic datasets and one real dataset. In particular, we identified the generative adversarial network (GAN), as a well suited deep learning framework to construct the Deep Digital Twin. For the synthetic gearbox dataset, we demonstrated that the generative adversarial network (GAN) is better suited to a variational auto-encoder (VAE) to instantiate DDT instances.

In addition, the discriminator used during GAN training has learned to differentiate probability densities from the healthy probability density by considering density ratios. This allows for the discriminator to be used as a Health Indicator (HI), to track and monitor the degradation of an asset over its life-cycle as well as identify various failure modes within a fleet. This was demonstrated on a synthetic gearbox dataset for discrete levels of damage as well as the IMS bearing dataset for actual continuous degradation of an asset.

The DDT is a low-cost alternative to expensive physics-based digital twin instances with the additional benefit of providing a HI. Due to the implicit nature of the DDT and its provision of a HI, the methodology enables asset users to implement their own digital twin based asset management and maintenance strategies without OEM support over a heterogeneous fleet of assets.

In Fig. 21 the role of the DDT in PHM is demonstrated in the situation where only nominal operational data is available and illustrates how the method enables additional PHM steps such as time to failure and semi-supervised diagnostics when sparse failure and run-to failure data is available.

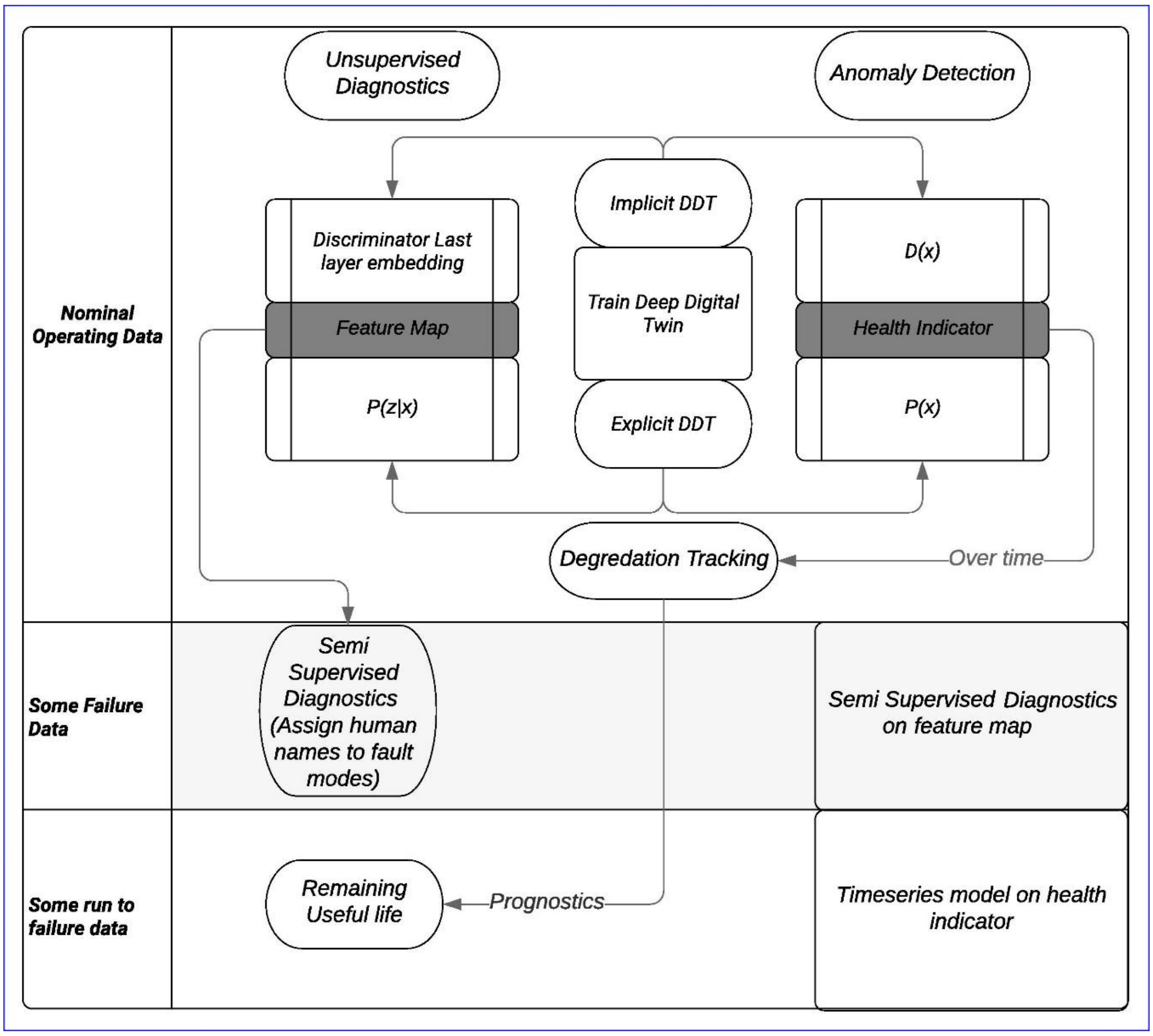


Fig. 21. Flow diagram illustrating the role of Deep Digital Twins in the Prognostics and Health Management space, while directly enabling unsupervised diagnostics, fault detection and degradation tracking from only nominal operational data. When additional failure data is available it becomes possible to perform semi-supervised diagnostics as well as full time-to-failure predictions using the health indicator and feature map as the input features to the semi-supervised learning and time-series models.

6.1. Future work

Although the DDT framework has been introduced and evaluated in terms of Variational Autoencoders for the case of explicit probabilistic models and Generative Adversarial networks for implicit probabilistic models. There do exist other deep generative models which are suitable for DDT implementations whose performance has not been investigated in the DDT setting. Some examples which are:

Explicit deep generative models:

- Masked Autoregressive Flows (MAF) [33].
- Non-linear Independent Components Estimation (NICE) [8].
- Real-valued Non-volume Preserving transformations (real NVP) [9]
- The Neural Autoregressive Distribution Estimator (NADE) [23].
- Pixel Recurrent Neural Networks (Pixel RNN) [32].
- Wavenet [31].

Implicit deep generative models:

- Deep Generative Stochastic Networks (GSN) [3].
- Info-GAN [7].
- BIGAN [10].
- Noise-contrastive Estimation [17].

Even though the DDT is able to produce an estimate of the health of the asset being monitored, it is imperative that the use of the health indicator is extended into a prognostics framework for predicting time till failure. This may be done by using the health indicator as a change-point detection algorithm in order to correctly train supervised time till failure models without a piecewise linear heuristic as used in Heimes et al. [18].

In addition when using a DDT in practise it may be desirable to interrogate the source of the degradation, to this end it may be useful to investigate the counter-factual reasoning abilities of a DDT.

It has been shown that the DDT is able to discern between different failure modes. In order to use these learned representations for diagnostic purposes it is necessary to explore semi-supervised learning approaches which may be used for fault diagnosis. These approaches would use the learned embeddings of the DDT and sparse labelled failure data to 'annotate' the failure classes within the learned embedding.

References

[1] D. Abboud, M. Elbadaoui, W.A. Smith, R.B. Randall, Advanced bearing diagnostics: a comparative study of two powerful approaches, Mech. Syst. Signal Process. 114 (2019) 604–627.

[2] S. Baggeröhr, W. Booyse, P.S. Heyns, D.N. Wilke, Novel bearing fault detection using generative adversarial networks, in: Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management, Curran Associates, Red Hook, NY, 2018, pp. 243–251.

[3] Yoshua Bengio, Eric Laufer, Guillaume Alain, Jason Yosinski, Deep generative stochastic networks trainable by backprop, in: International Conference on Machine Learning, 2014, pp. 226–234.

[4] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, arXiv:1809.11096 (2018).

[5] Fakher Chaari, Walid Baccar, Mohamed Slim Abbes, Mohamed Haddar, Effect of spalling or tooth breakage on gearmesh stiffness and dynamic response of a one-stage spur gear transmission, Eur. J. Mech.-A/Solids 27 (4) (2008) 691–705.

[6] Fakher Chaari, Walter Bartelmus, Radoslaw Zimroz, Tahar Fakhfakh, Mohamed Haddar, Gearbox vibration signal amplitude and frequency modulation, Shock Vib. 19 (4) (2012) 635–652.

[7] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, Inc., 2016, pp.2172–2180.

[8] L. Dinh, D. Krueger, Y. Bengio, NICE: non-linear independent components estimation, arXiv:1410.8516, (2014).

[9] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, arXiv:1605.08803, (2016).

[10] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning. arXiv:1605.09782, (2016).

[11] O.F. Eker, F. Camci, I.K. Jennions, Major challenges in prognostics: study on benchmarking prognostics datasets, in: European Conference of Prognosticsand Health Management Society, 2012, pp. 1–8, https://doi.org/10.1002/qre.1393.

[12] W. Fedus, M. Rosca, B. Lakshminarayanan, A.M. Dai, S. Mohamed, I. Goodfellow, Many paths to equilibrium: GANs do not need to decrease a divergence at every step, arXiv:1710.08446, (2017).

[13] Tryphon T. Georgiou, What is a natural notion of distance between power spectral density functions?, in: Control Conference (ECC), 2007 European, IEEE, 2007, pp 358–361.

[14] Kai Goebel, Hai Qiu, Neil Eklund, Weizhong Yan, Modeling propagation of gas path damage, in: Aerospace Conference, 2007 IEEE, IEEE, 2007, pp. 1–8.
[15] I. Goodfellow, Efficient per-example gradient computations, arXiv:1510.01799, (2015).

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates Inc, 2014, pp.2672–2680.

[17] Michael Gutmann, Aapo Hyvärinen, Noise-contrastive estimation: a new estimation principle for unnormalized statistical models, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 297–304.

[18] O.H. Felix, Recurrent neural networks for remaining useful life estimation recurrent neural networks for remaining useful life estimation, International Conference on Prognostics and Health Management, 2008, pp. 1–6.

[19] A.K.S. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510.

[20] Samir Khan, Takehisa Yairi, A review on the application of deep learning in system health management, Mech. Syst. Signal Process. 107 (2018) 241–265.

[21] D.P. Kingma, L.J. Ba, Adam a method for stochastic optimization, in: International Conference on Learning Representations, Ithaca, 2014, pp. 1–13.
[22] D.P. Kingma, M. Welling, Auto-encoding variational bayes. arXiv:1312.6114, (2013).

[23] Hugo Larochelle, Iain Murray, The neural autoregressive distribution estimator, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 29–37.

[24] L. Mescheder, S. Nowozin, A. Geiger, The numerics of GANs, arXiv:1705.10461, (2017).

[25] Lars Mescheder, Andreas Geiger, Sebastian Nowozin, Which training methods for GANs do actually converge?, in: International Conference on Machine Learning, 2018, pp 3478–3487.

[26] L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein, Unrolled generative adversarial networks, arXiv:1611.02163, (2016).

[27] T. Miyato, M. Koyama, cGANs with projection discriminator, arXiv:1802.05637, (2018).

[28] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, arXiv:1802.05957, (2018).

[29] S. Mohamed, B. Lakshminarayanan, Learning in implicit generative models, arXiv:1610.03483, (2016).

[30] John F. Nash et al, Equilibrium points in n-person games, Proc. Natl. Acad. Sci. 36 (1) (1950) 48–49.

[31] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, Senior A., Kavukcuoglu K., Wavenet: a generative model forraw audio, arXiv:1609.03499, (2016).

[32] A. van den Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, arXiv:1601.06759, (2016).

[33] George Papamakarios, Theo Pavlakou, Iain Murray, Masked autoregressive flow for density estimation, in: Advances in Neural Information Processing Systems, 2017, pp. 2338–2347.

[34] Hai Qiu, Jay Lee, Jing Lin, Gang Yu, Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics, J.Sound Vib. 289 (4–5) (2006) 1066–1090.

[35] E. Ramasso, Investigating computational geometry for failure prognostics, Int. J. Prognostics Health Manage. 5 (1) (2014) 1–18.

[36] E. Ramasso, A. Saxena, Performance benchmarking and analysis of prognostic methods for CMAPSS datasets, Int. J. Prognostics Health Manage. 5 (2) (2014) 1–15.

[37] R.B. Randall, Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications, Wiley, 2011.

[38] Kenneth Reifsnider, Prasun Majumdar, Multiphysics stimulated simulation digital twin methods for fleet management, in: 54th AIAA/ASME/ASCE/AHS/ ASC Structures, Structural Dynamics, and Materials Conference, 2013, p. 1578.

[39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, arXiv:1504.01391, (2016).

[40] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–9.

[41] S. Schmidt, A Cost-Effective Diagnostic Methodology Using Probabilistic Approaches for Gearboxes Operating under Non-Stationary Conditions, University of Pretoria, Masters Degree, 2016, URL:https://repository.up.ac.za/bitstream/handle/2263/61332/Schmidt_Cost_2017.pdf?.

[42] J.E. Shigley, Applied Mechanics of Materials, International student edition., McGraw-Hill, 1976.

[43] C.K. Sønderby, J. Caballero, L. Theis, W. Shi, F. Huszár, Amortised MAP inference for image super-resolution, arXiv:1610.04490, (2016).

[44] Eric Tuegel, The airframe digital twin: some challenges to realization, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012, p. 1812.
[45] J. Zhao, M. Mathieu, Y. LeCun Energy-based generative adversarial network, arXiv:1609.03126, (2016).

Appendix A. GAN Training for DDT instances

GAN training is still a developing and active field in Deep Learning research since GANs are notoriously hard to train. For example, it is well known that the standard GAN training method may not converge as expected [15]. As the performance of the GAN is directly dependent on the convergence characteristics of the GAN training, it is necessary to modify the standard GAN training, in order to successfully train the GAN to be useful as a DDT instance. These modifications are:

1. Modification to the Generator loss function [29], [43].
2. Addition of a zero-centred gradient penalty [25].
3. Use of simultaneous gradient descent [24].
4. Use of Spectral Normalisation [28].
5. Use of Projection Discriminator for conditional data [27].
6. ReLU Latent space [4].

All modifications except 1 and 2 have been widely reported as modifications to standard GAN training when considering modern GAN architectures [4], [25]. The first non-standard modification used is to change the Generator loss function to directly minimise the KL divergence $D_{\mathrm{KL}}(q|p)$, during training, as given by:

$$\mathscr{L}_G(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\log \frac{D_\theta(G_\phi(\mathbf{z}))}{1-D_\theta(G_\phi(\mathbf{z}))} \tag{15}$$

The second non-standard modification used is a zero centred gradient penalty of a similar form to the R2 gradient penalty from Mescheder et al. [25]. However, unlike the R2 gradient penalty in Mescheder et al. [25], the gradient penalty used for the GAN aims to encourage the discriminator output to be invariant to small changes within the latent space. The R2 penalty for the GAN is given by:

$$R2_{GAN} = ||\nabla_{\mathbf{z}} D_\theta(G_\phi(\mathbf{z}))|| \tag{16}$$

This R2 penalty cannot be applied to the Generator as this would result in it producing samples which are invariant to $\mathbf{z}$. To this end the R2 penalty is applied as a one-sided gradient penalty on the discriminator [25], resulting in the modified GAN training loss, which is given by Eq. (17) for the discriminator and by Eq. (18) for the generator. Note λ is a regularisation constant that needs to be found by hyper-parameter search.

$$\mathscr{L}_D(\boldsymbol{\theta},\boldsymbol{\phi}) = -\tfrac{1}{2}\log(D_\theta(\mathbf{x})) - \tfrac{1}{2}\log(1 - D_\theta(G_\phi(\mathbf{z}))) + \lambda||\nabla_{\mathbf{z}} D_\theta(G_\phi(\mathbf{z}))|| \tag{17}$$

$$\mathscr{L}_G(\boldsymbol{\theta},\boldsymbol{\phi}) = -\log\frac{D_\theta(G_\phi(\mathbf{z})))}{1-D_\theta(G_\phi(\mathbf{z}))} \tag{18}$$

Although these modifications significantly enhance the probability of successful GAN training it does not guarantee the absence of mode collapse, which would be catastrophic for our application of the GAN. It is therefore pertinent to confirm the proper training of the GAN. See Brock et al. [4] for a full investigation into the instabilities of GAN training.

A.1. Detecting mode collapse

As pointed out, GAN training is susceptible to mode collapse, which is reflected by a generator, $G(\mathbf{z},\boldsymbol{\phi})$, failing to produce diverse samples regardless of the value of the random latent variable $\mathbf{z}$. $G(\mathbf{z},\boldsymbol{\phi})$ may even return the same sample irrespective of the value of the random latent variable $\mathbf{z}$.

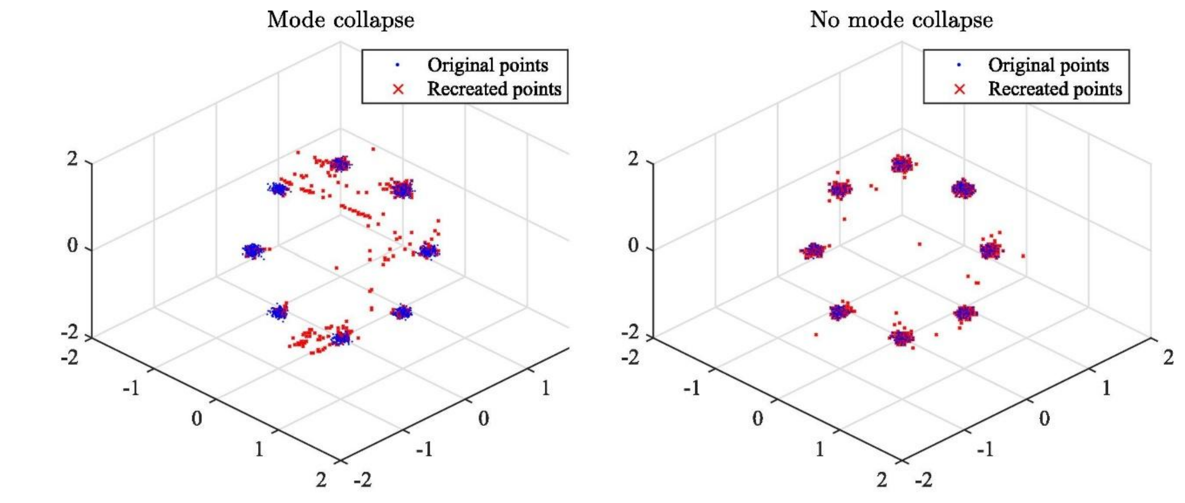
One metric to quantify mode collapse introduced by Metz et al. [26] is known as Inference via Optimization (IvO). IvO quantifies the ability of the generative model to generate a sample from the training set by finding the location $\mathbf{z}_{opt}$ in the latent variable space $\mathbf{z}$ which minimizes the recreation error

$$\mathscr{L}_{IvO} = ||\mathbf{x}_{opt} - \mathbf{x}_{real}||_2^2, \tag{19}$$

where

$$\mathbf{z}_{opt} = \text{argmin}_{\mathbf{z}}||\boldsymbol{G}(\mathbf{z},\boldsymbol{\phi}) - \mathbf{x}_{\boldsymbol{real}}||_2^2.$$

IvO is able to detect mode collapse as it is expected that the generator will not be able to create samples from regions in the training data where mode collapse has occurred. To illustrate mode-collapse detection using IvO, consider the problem of fitting a disconnected Gaussian mixture distribution. Two GANs were trained on the Gaussian mixture, one which experienced mode collapse, and another which did not. When the optimisation is run on the GAN which experienced mode collapse it is clear that there are domains where the GAN is unable to recreate the real data (Fig. 22a). On the other hand for a GAN which did not suffer from mode collapse it is clear that the Generator is able to accurately recreate the real data over the entire data manifold, which indicates that the Generator is able to produce samples from the entire distribution of healthy data (Fig. 22b).

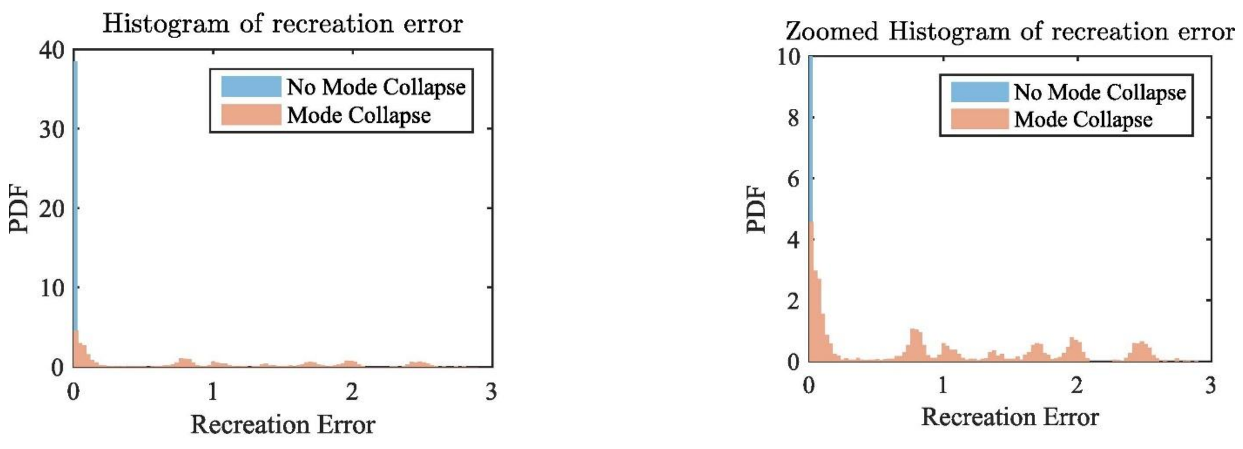


(a) IvO optimisation result for GAN experiencing mode collapse

(b) IvO optimisation result for GAN without mode collapse

Fig. 22. GAN trained on Gaussian mixture comparing effects of mode collapse.

Although it is still possible to visually observe the GAN's coverage of the data domain in three-dimensional space, this will not be possible in higher dimensional spaces. Therefore in order to determine whether mode collapse has occurred or not it is useful to plot the histogram of recreation errors. The intuition is that each additional mode of the histogram of recreation errors should correspond to a mode which has not been captured by the generative distribution of the Generator, this behaviour is visible in Fig. 23 for the GANs trained on the mixture distribution.



(a) Normal scale

(b) Zoomed

Fig. 23. Effect of mode collapse on histogram of IvO recreation error.

Although IvO should be sufficient to detect mode collapse on conventional problems—for vibration data, the RMSE between a real signal and a recreated signal is not necessarily a good measure of how well the GAN approximates the training data. The reason for this is the GAN may be able to recreate the large scale components of the vibration signal very well, resulting in a small RMSE between the real and recreated signals. However, this does not guarantee that other components of the signal—which may have smaller magnitudes—are recreated.

In order to compare recreated and observed vibration signals, it is suggested that instead of using the RMSE between the two signals in the time domain, the IvO metric should rather be based on the distance between the power spectral densities (PSD) given in Eq. (9).

Appendix B. Network architectures and hyper-parameter choices

B.1. Gearbox dataset

Both the VAE and the GAN used for this investigation used three convolutional layers. Both networks used the ReLU activation function for all layers except the output. The encoder of the VAE was constrained to represent the latent variable space $\mathbf{z}$ as a 200 dimensional conditionally independent Gaussian distribution. Both networks were trained using Adam [21] with a learning rate of 0.0005 exponentially decayed over training. The GAN was trained using simultaneous gradient descent coupled with the R2 gradient penalty ($\lambda = 3.5$ halved every 2500 updates). Both networks were trained for a total of 2000 epochs. The noisy rotational speed was included into the networks as a conditional variable $\mathbf{y}$ (See Table 4, Table 5).

Table 4. GAN model architecture for the gearbox dataset.

Layer	GAN Generator	GAN Discriminator
Input	$\mathbf{z} = [200 \times 1], \mathbf{y} = [1 \times 1]$	$\mathbf{x} = [2048 \times 1], rvy = [1 \times 1]$
1	$Dense_{8192}$(ReLU)	$Conv_{32\times 25\|\|\ Stride\|2\|}$(ReLU)
2	$Deconv_{128\times 25\|\|Stride=[1]}$(ReLU)	$Conv_{64\times 25\|\|\ Stride\|2\|}$(ReLU)
3	$Deconv_{64\times 25\|\|Stride=[4]}$(ReLU)	$Conv_{128\times 25\|\|\ Stride\|5\|}$(ReLU)
4	$Deconv_{32\times 25\|\|Stride=[4]}$(ReLU)	$Conv_{256\times 25\|\|\ Stride\|5\|}$(ReLU)
5	$Deconv_{1\times 25\|\|Stride=[4]}$(Tanh)	$Dense_{256}$(ReLU)
6		$Dense_{1}$(Sigmoid)

Table 5. VAE model architecture for the gearbox problem.

Layer	VAE Decoder	VAE Encoder
Input	$\mathbf{z} = [200 \times 1], \mathbf{y} = [1 \times 1]$	$\mathbf{x} = [2048 \times 1]$
1	$Dense_{8192}$(ReLU)	$Conv_{32\times 25\|\|\ Stride\|2\|}$(ReLU)
2	$Deconv_{128\times 25\|\|Stride=[1]}$(ReLU)	$Conv_{64\times 25\|\|\ Stride\|5\|}$(ReLU)
3	$Deconv_{64\times 25\|\|Stride=[4]}$(ReLU)	$Conv_{128\times 25\|\|\ Stride\|5\|}$(ReLU)
4	$Deconv_{32\times 25\|\|Stride=[4]}$(ReLU)	$Conv_{256\times 25\|\|\ Stride\|5\|}$(ReLU)
5	$Deconv_{1\times 25\|\|Stride=[4]}$(Tanh)	$\mu = Dense_{1}$(Tanh):$\sigma = Dense_{1}$(Softplus)
6		$Dense_{1}$(Sigmoid)

B.2. CMAPSS dataset

For the CMAPSS dataset the GAN was trained using simultaneous gradient descent coupled with the R2 gradient penalty ($\lambda = 3.5$ halved every 2500 updates). The network was trained for a total of 2000 epochs using Adam [21] with a learning rate of 0.0005 exponentially decayed over training. Z was sampled from a 4 dimensional conditionally independent Gaussian latent variable space. The 3 flight conditions are included into the network as conditional variables $\mathbf{y}$ (See Table 6).

Table 6. GAN model architecture for the C-MAPSS.

Layer	GAN Generator	GAN Discriminator
Input	$\mathbf{z} = [4 \times 1], \mathbf{y} = [3 \times 1]$	$\mathbf{x} = [21 \times 1], \mathbf{y} = [3 \times 1]$
1	$Dense_{8}$(ReLU)	$Dense_{64}$(ReLU)
2	$Dense_{16}$(ReLU)	$Dense_{32}$(ReLU)
3	$Dense_{32}$(ReLU)	$Dense_{16}$(ReLU)
4	$Dense_{64}$(ReLU)	$Dense_{8}$(ReLU)
5	$Dense_{21}$(Linear)	$Dense_{1}$(Sigmoid)

B.3. IMS bearing dataset

The GAN used for the bearing investigation had four convolutional layers with ReLU non-linearities in all layers bar the last. The GAN samples from a 100 dimensional conditionally independent Gaussian latent variable space. The network was trained using simultaneous gradient descent coupled with the R2 gradient penalty ($\lambda = 3.5$ halved every 2500 updates). The GAN was trained for a total of 2000 epochs using Adam with a learning rate of 0.0005 exponentially decayed over training (See Table 7).

Table 7. GAN model architecture for the IMS Bearing problem.

Layer	GAN Generator	GAN Discriminator
Input	$\mathbf{z} = [100 \times 1]$	$\mathbf{x} = [4096 \times 1]$
1	$Dense_{4096}$(ReLU)	$Conv_{32 \times 25 \|\| \; Stride[2]}$(ReLU)
2	$Deconv_{128 \times 25 \|\| Stride=[4]}$(ReLU)	$Conv_{64 \times 25 \|\| \; Stride[2]}$(ReLU)
3	$Deconv_{64 \times 25 \|\| Stride=[4]}$(ReLU)	$Conv_{128 \times 25 \|\| \; Stride[5]}$(ReLU)
4	$Deconv_{32 \times 25 \|\| Stride=[4]}$(ReLU)	$Conv_{256 \times 25 \|\| \; Stride[5]}$(ReLU)
5	$Deconv_{1 \times 25 \|\| Stride=[4]}$(Tanh)	$Dense_{256}$(ReLU)