Augmenting Aircraft Engine Flight Data with Generative Adversarial Networks for Fault Detection

MSc. Thesis at KLM Engine Services

Daniel Cisneros Acevedo

Augmenting Aircraft Engine Flight Data with Generative Adversarial Networks for Fault Detection

Thesis Report

by

Daniel Cisneros Acevedo

to obtain the degree of Master of Science
at Delft University of Technology

Thesis committee:	
Chair:	Dr. Carmine Varriale
Supervisors:	Dr. Marcia L. Baptista
	Tim Rootliep (KLM)
External examiner:	Dr. Marta J. Ribeiro
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	June, 2023 - April, 2024
Student number:	4657349

Faculty of Aerospace Engineering · Delft University of Technology

Preface

In the second quarter of my master in Aerospace Engineering I knew I wanted to work on data and machine learning projects. I am fortunate enough to have found this graduate internship opportunity at KLM Engine Services, where my skills in data science could be applied to highly complex engineering systems, such as aircraft engines. Grasping a relatively new topic, Generative Adversarial Networks, was every bit as challenging as it was fun to learn.

I want to express my gratitude towards my supervisors, Marcia Baptista (TU Delft) and Tim Rootliep (KLM), whom both have always been very helpful and supportive of my work. Furthermore, I want to thank Walid Brachmi, a graduate intern at KLM within the same team, whom I have shared many conversations with during our travels to Schiphol-Oost. I am also grateful for my other colleagues, Antonis, Leandro, Yaïr, Albert, Juan, and Pieter. Special thanks to José Velazquez who encouraged my curiosity in research and to Dennis van den Berg who helped me several times with sophisticated machine learning concepts. Finally, I am thankful to my family for providing their unconditional love and support during my thesis. I would not be here without them.

Daniel Cisneros Acevedo
April 2024

Contents

Nomenclature

List of Abbreviations

AE	Autoencoders
AI	Artificial Intelligence
ANN	Artificial Neural Network
C-MAPSS	NASA Commercial Modular Aero-Propulsion System Simulation
CBM	Condition Based Maintenance
CC	Combustion Chamber
CNN	Convolutional Neural Network
FCNN	Fully Connected Neural Network
FOD	Foreign Object Damage
GAN	Generative Adversarial Network
GEnx	General Electric Next Generation
GPA	Gas Path Analysis
GSP	Gas turbine Simulation Program
HPC	High Pressure Compressor
HPT	Low Pressure Turbine
KLM	Royal Dutch Airlines
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multilayer Perceptron
MRO	Maintenance, Repair, and Overhaul
N-CMAPSS	New CMAPSS
OEM	Original Equipment Manufacturer
PHM	Prognostics & Health Management
RGAN	Recurrent GAN
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RQ	Research Question
WGAN	Wasserstein GAN
WGAN-GP	WGAN with Gradient Penalty

List of Symbols

M	Mach
N_1	Fan Speed
N_2	Core Speed
P_{amb}	Ambient Pressure
P_{s3}	Static Pressure after HPC
P_{t2}	Total Inlet Temperature
T_{amb}	Ambient Temperature
T_{t3}	Total Temperature after HPC
T_{t49}	Total Exhaust Gas Temperature
TAT	Total Air Temperature
W_f	Fuel Flow

List of Figures

List of Tables

1 Introduction

High expenses toward the upkeep and servicing of aircraft engines are driving the aviation industry to utilize condition monitoring data for optimal decision-making and improving operating efficiency. AI-driven methods are particularly valued for their ability to understand complex patterns, handle uncertainties in measurements, and overcome the challenges posed by the reduced number of sensors on turbofan engines.

Deep learning techniques are becoming the method of choice for their proficiency in autonomously identifying features and deciphering complex relationships within data. However, the reliance on simulated datasets for Prognostics and Health Management (PHM) in aircraft engines raises issues regarding their applicability to real-world conditions. Further complicating this issue is the lack of failure data compared to normal operating data which leads to class imbalance and limits the performance of fault detection models.

This study aims to explore the potential of Generative Adversarial Networks (GANs) to mitigate the class imbalance issue in real-world turbofan engine data, thereby improving the performance of deep learning models in fault detection tasks. Since their introduction in 2014, GANs have demonstrated remarkable proficiency in creating synthetic data that closely mimics the characteristics of original datasets.

Focusing on operational and gas path parameters collected from General Electric Next Generation (GEnx) turbofan engines, this research establishes a baseline model first using Recurrent Neural Networks (RNNs) for fault detection. Then, by generating synthetic time series failure data through a GAN and assessing its impact on the model's classification accuracy, this work aims to demonstrate the value of data augmentation in aerospace maintenance. Additionally, recognizing the ongoing challenge in assessing the realism of synthetic time series data, this study also proposes an innovative validation approach employing a GEnx Gas Path Analysis (GPA) engine performance model to ensure the generated data accurately reflects the engines' physical behaviors.

The report is organized into two main parts. The first part, Part I, introduces the paper that lays the groundwork for this investigation. Subsequently, the second section provides a comprehensive review of the literature and related works that form the basis of this study.

Part I

Scientific Paper

Highlights

Augmenting Aircraft Engine Flight Data with Generative Adversarial Networks for Fault Detection

Daniel Cisneros Acevedo

- To address the industry's challenge of limited failure data compared to normal operational data, we propose a generative adversarial network (GAN) for augmenting real-world GEnx turbofan data to improve deep learning fault detection models.
- This study utilizes a 1D convolutional neural network Wasserstein GAN with Gradient Penalty (WGAN-GP) to generate synthetic time series data.
- A GPA-based engine performance model is used to validate the physical relationship between the operating and gas path parameters of the synthetic samples.
- Introducing GAN synthetic samples to the original dataset improved the F1-score of the baseline fault detection model by an average of 2.8%.

Augmenting Aircraft Engine Flight Data with Generative Adversarial Networks for Fault Detection

Daniel Cisneros Acevedo[a]

[a] *Section of Air Transport and operatingions, Delft University of Technology, Aerospace Engineering Faculty, Delft, The Netherlands*

ARTICLE INFO

Keywords:
Aircraft engine
Predictive maintenance
Condition based maintenance
Data augmentation
Generative models
Generative adversarial networks
Fault detection
Prognostics and health management
GEnx turbofan engine
Real world data

ABSTRACT

Recent advancements in deep learning for aircraft engine fault detection have been predominantly focused on research using simulated datasets. Despite significant progress, the gap between simulated and real-world data underscores a pressing need for models that are more applicable and adaptable to the aerospace industry. This discrepancy stems from factors such as water washes, maintenance activities, noise, and nuanced variations in operating conditions. Further complicating this issue is the lack of failure data leading to class imbalance and limiting the performance of fault classification models. In response to these challenges, this study uses Generative Adversarial Networks (GANs) to augment real-world failure data from General Electric Next Generation (GEnx) aircraft engines. New synthetic data are generated using a Wasserstein GAN with Gradient Penalty (WGAN-GP) and convolutional layers. Evaluation of GAN-generated data remains an active area of research. Accordingly, we also introduce a novel validation method based on a GEnx Gas Path Analysis model. This evaluation step revealed that the GAN could effectively generate gas path response variables that were physically meaningful and consistent with the operating conditions. Furthermore, integrating the GAN-generated data into the original dataset improved the baseline fault detection model's F1-score by an average of 2.8%. This research also highlights the GAN's ability to learn and reproduce degradation patterns applicable across different engine units, emphasizing its potential to overcome the challenges between engine unit-to-unit variations. Additionally, this work can potentially be extended to other engine families that require synthetic data to improve maintenance strategies.

1. Introduction

With the advent of big data and the Internet of Things, the aviation industry is increasingly focused on leveraging condition monitoring data for optimal decision-making and enhancing operating efficiency [35]. This focus is particularly relevant to the expensive maintenance of turbofan engines [28], positioning Prognostics and Health Management (PHM) as a critical area for investment and innovation. In recent years, extensive research has been conducted into a variety of methods aimed at diagnostics and prognostics of industrial assets, ranging from statistical and physics-based models to hybrid and artificial intelligence (AI) techniques [32]. In particular, AI-driven techniques have gained traction for their capacity to understand nonlinear patterns, handle measurement uncertainties, and overcome limitations posed by the reduced number of sensors in turbofan engines [16]. Motivated by the lack of public run-to-failure datasets available for research, the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) software [49] has played a crucial role in the development and benchmarking of machine learning models in turbofan engine PHM.

Deep learning approaches have emerged as a preferred research area due to their ability to automatically extract features and learn complex relationships in the data [55, 56]. This trend continues to persist after the introduction of the new C-MAPSS (N-CMAPSS) dataset [1] which further enhanced the realism of turbofan data simulations by integrating authentic operating conditions and a detailed degradation model. The datasets from the publications enabled the study of sophisticated deep neural network architectures and showed that both recurrent neural networks (RNNs) [38] and convolutional neural networks (CNNs) [50] have been validated as effective tools in detecting and predicting system failures.

While simulated datasets have significantly contributed to deep learning research in PHM, there exist concerns about the differences between simulated and real-world datasets [33, 43, 32]. Simulated datasets are typically generated using zero-dimensional turbofan performance models, which are commonly used for Gas Path Analysis (GPA). These simulators rely on thermodynamic principles to approximate engine parameters across various gas path stations [24]. Besides common input parameters describing operating conditions such as Mach number, thrust, ambient temperature, or pressure, the engine performance models can also incorporate health parameters from various component groups enabling the simulation of deteriorated responses. Hence, to create run-to-failure datasets, mathematical damage propagation models have been used to describe the health parameters' evolution until a predefined failure condition is reached. However, this approach may not capture the complex and nuanced patterns of real-world engine degradation. For instance, engine degradation patterns may change after major maintenance events when certain parts are replaced or repaired. Routine practices like water washes introduce extra complexity as these interventions disrupt the natural evolution of engine degradation patterns [25]. In addition, the simulations do not account for the variability due to a broad spectrum of conditions, influenced by diverse factors like pilot behavior and aircraft weight, or even the swapping of engines between aircraft. Consequently, these disparities underscore a critical gap between academic research and practical industrial application, stressing the urgent need for real-world data to improve the relevance and transferability of data-driven models.

Although turbofan engines generate a vast amount of operating data, the class imbalance remains a challenging issue due to the abundance of normal condition monitoring data relative to failure instances [18, 21, 13]. Given the critical importance of maintaining turbofan engine safety, their design for high reliability and the application of preventive maintenance strategies contribute to the scarcity of engine failures. As a result, training fault detection models on imbalanced data have a higher risk of model bias towards the majority class, whereas they tend to generalize poorly to the minority class. This issue severely affects the performance of machine learning models in classification tasks, which are crucial for fault detection.

This research aims to investigate how data augmentation using Generative Adversarial Networks (GANs) can address the class imbalance issue common in real-world turbofan engine datasets and improve deep learning based fault detection models. Introduced by Goodfellow et al. [22], GANs have shown to possess exceptional capabilities at generating synthetic data indistinguishable from the original dataset. Initially, they found their primary application in computer vision tasks. Yet, their potential in the time-series domain, crucial for the application of turbofan PHM, is still an emerging and active area of research [19, 27]. In the context of turbofan data, our literature review identifies only three studies that explored the utility of GANs on the C-MAPSS and N-CMAPSS dataset [31, 57, 59].

The core of our methodology involves the comprehensive preparation of the dataset, which accounts for maintenance events, water washes, and the labeling process. A baseline model using Recurrent Neural Networks (RNN) is then developed to differentiate between failure and non-failure states. It is trained to identify critical patterns between the operating conditions and the gas path parameters as observed from the in-flight measurements. Moving forward, we propose to design and implement a GAN architecture based on convolutional layers to generate synthetic engine failure data. Within this framework, the generator is tasked to produce synthetic yet realistic representations of time series data, while the discriminator evaluates the authenticity of these representations in comparison to real data samples. It is anticipated that this adversarial process will progressively refine the generator's ability, ultimately enabling it to replicate the critical characteristics inherent in the real data. Lastly, the GAN is utilized to augment augment the original dataset with new synthetic data allowing for an evaluation of the impact on the fault detection model's predictive performance after retraining.

The process of evaluating the quality and utility of the synthetic time series data from GANs continues to be a focus of research efforts [6]. Unconditional GANs, which generate data without specific conditions, pose unique challenges in verifying the realism of the synthetic time series due to the lack of a 'targeted evaluation' mechanism. On the contrary, in conditional GANs, synthetic data can be directly compared against a validation set under specific conditions, facilitating a more straightforward evaluation. Despite these challenges with unconditional GANs, GPA-based simulators present a valuable opportunity to assess the physical interdependence of the synthetically produced sensor variables. By using the domain knowledge encapsulated in these simulators, which relate operating conditions with engine responses, an extra layer of validation is introduced. GPA-based performance models serve as an intermediate reference and enable a comparison method where both real data and synthetic data generated by the GAN can be evaluated against this benchmark. This approach allows for assessing if the generated data holds up to the realistic standards set by actual engine behaviors and conditions.

The principal contributions of this study are outlined as follows:

1. Development of an RNN fault detection model to differentiate between non-failure and failure samples collected from real-world GEnx-1B turbofan data, including a detailed preprocessing methodology for water washes and major maintenance activities.
2. Introduction of a 1D-CNN Wasserstein GAN with Gradient Penalty (WGAN-GP) designed to generate synthetic time series data of failure instances, mitigating the class imbalance problem between healthy and failure data.
3. Utilization of a GPA-based performance model for the GEnx-1B engine to assess that the GAN-generated sensor data maintains realistic interdependence between the operating and gas path parameters.

In summary, this research aims to provide insights into the generation of synthetic GEnx-1B aircraft engine failure data with GANs for improved fault detection. By leveraging the capabilities of generative models, the issue of class imbalance is mitigated using high quality synthetic time series data. Furthermore, this research integrates domain knowledge through engine GPA-based thermodynamic models to assess the physical consistency of GAN-produced samples. To the best of knowledge, no other publication has explored the application of GANs for augmenting real-world aircraft engine data.

The remainder of this paper is organized as follows: section 2 provides the reader with information about DL within the scope of PHM. Furthermore, recent works on the application of GANs in time series data are also briefly summarized. Section 3 outlines the research methodology, detailing the utilization of GEnx-1B engine data, the development of a surrogate model for GSP, classifier training processes, and the proposed GAN architecture. This is followed by the results and discussion in section 4. Finally, conclusions drawn from this study, accompanied by recommendations for further investigation and potential areas for future research, are presented in Section 5 and Section 6, respectively.

2. Related work

The related work section is organized into two main parts. The first part, detailed in subsection 2.1, focuses on the applications of deep learning within Prognostics and Health Management (PHM). The second part, outlined in subsection 2.2, explores the utilization of Generative Adversarial Networks (GANs) for time series data augmentation. Given the limited research of GANs in turbofan condition monitoring data, this review extends to include important studies from other research fields to provide a global overview.

2.1. *Deep learning in prognostics & health management*

Deep learning techniques have emerged as powerful tools for prognostics and health management (PHM) of aircraft engines. This interest is attributed to advancements in sensor technologies and big data analytics, enabling data-driven approaches in predictive maintenance [43, 21, 19, 30]. Specifically, the introduction of the C-MAPSS dataset in 2008 has catalyzed deep learning research to analyze and predict engine health [49, 12].

Central to PHM is the development of health indicators, which provide insights into the system's health state. These indicators are broadly classified into physics-based and virtual categories, with efficiency and corrected mass flow being popular in aircraft engine monitoring [32]. However, deep learning models have proven to be effective in PHM, managing to successfully learn the nonlinear behaviors and degradation patterns of complex systems. As a result, the identification of health indicators is implicitly learned from condition monitoring data without the need to develop them manually [33]. This is particularly useful for deciphering simultaneous faults and degradation patterns in complex multivariate time series data [56, 20].

Extensive research has been conducted on the application of different neural network architectures in PHM. For instance, multilayer perceptrons (MLPs) have been able to learn health state representations of gas turbines, making them suitable for diagnostics [16]. Nonetheless, MLPs fail to capture the temporal characteristics hidden in the data which is crucial for the evolution of degradation patterns. The inclusion of temporal context has been critical as it is recognized that certain faults are sequential by nature [44]. Additionally, physics-based diagnostics can be compromised by the smearing effect, where the absence of sensors in modern turbofan engines may cause faults to appear across multiple modules in the gas turbine. Instead, including temporal context was useful in mitigating the smearing effect [8]. Because of their ability to capture temporal patterns, the exploration into Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has been significant in advancing fault detection from historical sensor data [61, 17]. Liu et al. [37] applied fault diagnostics to motor bearings using RNN-based autoencoders (AE) learning useful characteristics hidden across the time domain. Furthermore, Mansouri et al. [40] proposed an enhanced RNN technique for fault detection and classification in wind energy systems by simplifying the model's training and complexity through hierarchical K-means clustering.

The emphasis on learning sequential features is particularly important in prognostics [30]. Specifically, long-short term memory (LSTM) based architectures have emerged as leading models in the prognostics of aircraft engines, demonstrating superior performance [55]. Applications of LSTMs are found on the C-MAPSS and the N-CMAPSS (new C-MAPSS) datasets where some researchers also extended the LSTM's capabilities with the attention mechanism [36, 38, 10]. In an effort to further motivate collaboration, Darrah et al. [11] established a comprehensive framework for the development of deep learning models aimed at predicting the Remaining Useful Life (RUL). On the other hand, the CNN's utility in RUL estimation has also been validated through their capacity to effectively capture localized features. A notable contribution by Li et al. [34] involves the proposal of a model comprising four stacked convolutional layers, succeeded by a two-dimensional fully connected layer. Contributing to this field, Solís-Martín et al. [50] employing a deep stacked CNN to predict RUL on the N-CMAPSS dataset for the 2021 PHM Conference Data Challenge. Their model achieved third place in the challenge and underscored the potential of CNNs in prognostics.

2.2. *Generative adversarial networks in time series*

In general, research on Generative Adversarial Networks (GANs) for time series data remains relatively limited compared to their use in computer vision tasks. The challenge of training GANs is amplified in the context of time series data [19] with only a handful of studies addressing aircraft condition monitoring data [31, 57, 59]. Consequently, a broader review is required by including insights from diverse applications and domains.

Previous studies have predominantly adopted Recurrent Neural Networks (RNNs) for both the generator and discriminator components of GANs. Mogren [42] was among the first researchers to develop a Recurrent GAN (RGAN) for generating classical music. This approach was expanded by Esteban et al., who tailored RGANs for multivariate medical time series generation [14]. Further, RGANs have been adapted for sensor data synthesis in autonomous driving, demonstrating the versatility of recurrent architectures in GANs [4]. Specific to turbofan engines, Lang et al. [31] demonstrated how GAN-augmented datasets could enhance training and predictive accuracy on the C-MAPSS dataset. Moreover, Xiong et al. introduced a physics-informed GAN for data augmentation focusing on the monotonic degradation patterns of engine health [57]. Additionally, Zhang et al. proposed a hybrid GAN, combining convolutional and recurrent layers, to improve RUL estimation on the C-MAPSS dataset and suggested further exploration into the evaluation of generated time series with domain expertise [59].

Training challenges, such as mode collapse and vanishing gradients, have been mitigated through advancements like Wasserstein GANs (W-GAN) and its enhanced version with gradient penalty (WGAN-GP), improving stability and performance [3, 23]. These advancements have been successfully applied beyond turbofan applications, as seen in the adoption of WGAN-GP for learning from fMRI data by Qiang et al. [46]. Furthermore, other challenges such as high computational load and extended training time associated with RGANs have motivated research into alternative models. Huang and Deng [26] demonstrated the effectiveness of one-dimensional CNNs in generating time series data, highlighting CNNs' capability in synthesizing data. Similarly, Baptista and Henriques [5] utilized 1D CNNs within GAN frameworks to reduce noise and enhance RUL prediction accuracy on the C-MAPSS dataset. Further advancements in CNN architectures aimed at augmenting sensor data for human movement analysis [58].

A drawback of convolution filters is the inability to extract features at different scales due to their fixed-size kernels. Addressing this limitation, Zhao et al. [60] developed a CNN model incorporating various kernel sizes and integrated them through a concatenation layer. This approach significantly improves the model's ability to distinguish short-term and long-term degradation patterns. The concept of a concatenation layer has been further extended to merge time series data from different sources. Choudhary et al. [9] applied this strategy to combine vibration and acoustic data from induction motors, achieving superior fault diagnostic results. These studies, while not exhaustive, underscore the research into CNNs for GAN applications, particularly in the generation and refinement of time series data for diverse applications.

3. Methodology

This section outlines the research paper's methodological approach. First, the principal research questions are revisited in subsection 3.1, followed by an introductory overview in subsection 3.2 of the General Electric Next Generation (GEnx) aircraft engine data. Then, subsection 3.3 discusses the application of Gas Path Analysis (GPA) in turbofan health management and gives an overview of Gas turbine Simulation Program (GSP), the engine performance simulator tool used in this study. After, section 3.4 describes the development of the surrogate model based on the GSP simulated data. Furthermore, data preprocessing and fault classification model training are discussed in section 3.5. Lastly, the GAN architecture and training process are discussed in section 3.6 for data augmentation.

3.1. Research questions

In this study, we employ data augmentation using generative adversarial networks on a real world turbofan data provided by KLM Engine Services. The primary research question of this study is as follows:

R1 How can generative adversarial networks further improve the predictive performance of failure detection models on real-world flight data?

Our hypothesis for the primary research question is as follows

H1 Integrating generative adversarial networks (GANs) into the training process of turbofan failure diagnostics models can improve the model's overall F1-score and generalization capabilities, as GANs introduce a broader range of failure samples.

We further extend the primary research question with the following two subquestions:

R1.1 How can generative adversarial networks be applied to real-world turbofan data to produce high-quality synthetic time series data?

R1.2 In what way can domain-knowledge from Gas Path Analysis contribute to validating the GAN generated turbofan data?

Our hypotheses for these sub research questions are listed below:

H1.1 Generative adversarial networks, based on deep neural networks that capture temporal patterns, can produce high-quality synthetic data for data augmentation.

H1.2 Domain-knowledge from Gas Path Analysis may serve as an indication for quantifying the interrelations between the operating and gas path parameters generated by the GAN.

3.2. Real-world flight data

This study focuses on measurements that are directly related to the operating conditions and the engine gas path parameters of the GEnx-1B aircraft engine. Figure 1 illustrates a total of 9 sensors with the associated description summarized in Table 1. The essential operating parameters include the Mach number (M), the fan speed (N_1), the total pressure at station 2 (P_{t2}), and the total air temperature (TAT). The other sensors, total temperature and static pressure at station 3 (T_{t3} and P_{s3}), the fuel low (W_f), the core speed (N_2), and the exhaust gas temperature (T_{t49}) relate to the internal gas path parameters of the engine. The combinations of these sensors ultimately lead to a multivariate vector describing the state of the engine at each time step.

The dataset from KLM (Royal Dutch Airlines) contains approximately 373,000 in-flight measurements from multiple GEnx-1B engine units. The data capture important engine states during three flight phases: takeoff, climb, and cruise. In Figure 2, a typical example of in-flight measured data from a single engine sensor is plotted along with the shop visits and water washes. In certain cases, the degradation recovery as a result of the water washes can directly be observed on the rolling mean. In others, this change

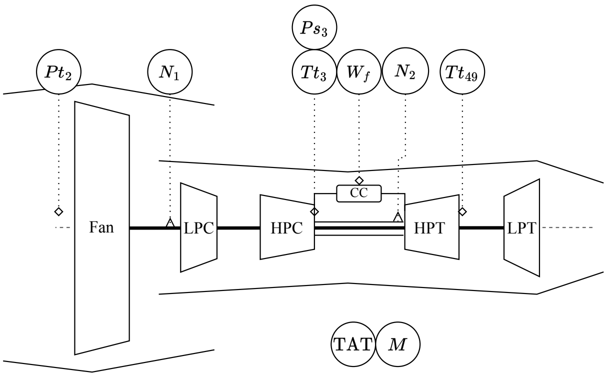


Figure 1: The GEnx-1B turbofan layout of relevant sensors in this research adjusted from [47, 48]. This research uses data from nine distinct sensors. Operating conditions are monitored by four sensors: M, N_1, P_{t2}, TAT. The remaining five sensors relate to performance and gas path parameters.

is less visible due to seasonality trends and the impact of operating conditions. It can also be observed that the in-flight measurements exhibit similar short-term and long-term degradation characteristics as discussed by Hepperle et al. [25] and by Hanachi et al. [24].

Table 1
Description of the sensors corresponding to the GEnx-1B turbofan in Figure 1.

Symbol	Description	Units
M	Mach number	-
N_1	Fan speed	%
P_{t2}	Total pressure at fan inlet	Pa
TAT	Total air temperature	K
T_{t3}	Total temperature HPC outlet	K
P_{s3}	Static pressure at HPC outlet	Pa
W_f	Fuel flow	kg/s
N_2	Core speed	%
T_{49}	Total temperature at HPT outlet	K

The dataset includes records of water washes and maintenance visits for each engine within the fleet. Maintenance logs contain detailed reasons for engine overhauls and which engine modules were serviced. The data in this study only accounts for all unscheduled maintenance events. Hence, this paper defines a "trajectory" as a sequence of flight cycles leading to an unscheduled maintenance event. After maintenance, the engine enters a new trajectory, allowing for multiple trajectories throughout its operating life. Each set of measurements leading up to maintenance is assigned a unique trajectory identification number. Similarly, each measurement is also assigned an identification number associated with the number of water washes that the engine has received.

3.3. Gas path analysis

Gas Path Analysis (GPA) is a popular technique, employed over the operating lifespan of an engine, to identify a variety of potential issues ranging from erosion and corrosion to more complex conditions such as foreign object damage (F.O.D.) and wear [52]. As Urban [52] states in their research, the core objective of GPA is the economical and effective detection of such faults by monitoring parameters that reveal implicit signs of degradation. Although GPA effectively identifies many types of defects, it may not detect certain issues, such as fatigue-induced cracks or subtle blade corrosion, because they do not significantly change the parameters that are monitored in GPA. For these types of defects, additional diagnostic techniques, such as radiography or boroscopy, are essential for fault detection.

The Gas turbine Simulation Program (GSP), developed by Visser et al. [54] at Delft University of Technology and National Aerospace Laboratory (NLR), is a modular zero-dimensional tool for simulating aircraft engine performance. GSP, which can be used for conducting GPA, is versatile enough to model different engines, including the latest turbofan variants. This tool estimates average gas path parameters across multiple engine stations, factoring in variables such as ambient conditions, Mach number, and fan speed [53]. Furthermore, GSP can also simulate a deteriorated response by modifying the health parameters within each engine module. The core of GSP's computational approach in simulating off-design conditions is the Newton-Raphson numerical solver, which tackles the system's non-linear algebraic equations to uphold conservation principles. The goal is to minimize an error vector $\bar{E}$ as a function of $\bar{S}$, where the engine state vector $\bar{S}$ is represented by $[s_1, s_2, ..., s_n]$. Through the Newton-Raphson method, updates to the state variables s_n are made by linear approximation of the gradient of e_n against s_n, with each variable slightly adjusted to calculate its gradient about the error. This process iteratively updates the state vector using the Jacobian matrix, continuing until the error falls beneath a certain threshold, thus solving the system of equations and determining the engine's state.

To accurately simulate off-design conditions within the GSP it is essential to have access to component maps specific to the engine model. However, acquiring these maps is a common issue in aircraft engine modeling since these are often proprietary to OEMs [51]. Previous work by Ramdin et al. [47] introduced a methodical framework for developing turbofan engine models within GSP while addressing the challenge of limited gas path sensors in modern turbofan engines. By tuning and scaling the engine turbo machinery maps with test-cell correlation data, an accurate representation of the GEnx-1B engine was created covering all different operating conditions. Since it is known that the operating and gas path parameters recorded from in-flight measurements are physically interdependent [24], our study utilizes this specific GEnx-1B engine model, optimized in GSP, to evaluate whether the GAN learned a reasonable relationship between these variables. In essence, the model

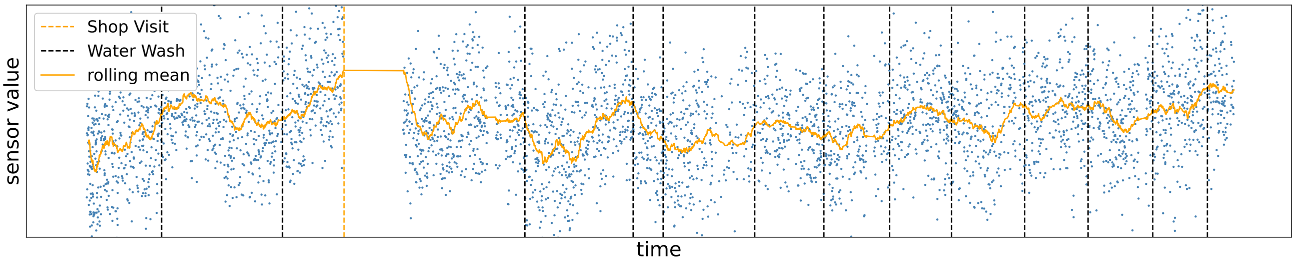


Figure 2: The plot shows a time series of the in-flight measurements captured by one GEnx-1B engine unit for a single sensor during cruise. The rolling mean with a window of 70 is plotted. The shop visits or major maintenance events and the water washes are depicted by the yellow and black striped lines, respectively.

offers a means to apply domain knowledge systematically during the assessment of GAN-generated samples.

3.4. *GSP surrogate model development*

Although GSP is proven to be effective at simulating modern turbofan engines, it falls short in terms of computational speed for heavy iterative tasks. Instead, a surrogate model was pre-trained on GSP simulated data produced by the digital GEnx-1B engine model discussed in subsection 3.3. This simplifies the application of GSP in validating the authenticity of synthetic turbofan engine data.

Figure 3 illustrates the concept of the surrogate model. A fully connected neural network (FCNN) was trained over approximately 1.2 million simulated scenarios covering a wide range of Mach numbers (M), engine fan speeds (N_1), ambient pressures (P_{amb}), and ambient temperatures (T_{amb}). These simulations, generated via GSP, provided the necessary gas path parameters for each set of operating conditions. Thus, the surrogate model effectively predicts gas path parameters based on operational inputs. The ambient pressure and temperature were derived from the total pressure at station 2 and total air temperature (TAT), using the isentropic relations described by Equation 1.

$$T_{amb} = \frac{TAT}{\left(1+\frac{\gamma-1}{2}M^2\right)} \qquad P_{amb} = \frac{P_{t2}}{\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}}} \qquad (1)$$

The architecture of the four layer FCNN begins with an initial layer of 64 units, followed by layers that progressively halve in unit count. The final layer contains the number of units equal to the number of sensors. Notably, this training process did not apply any regularization techniques. Furthermore, the features are scaled using a Min Max scaler since the extent of the operating region can be confidently estimated with the in-flight measurements.

While GSP allows for the adjustment of health parameters across different engine modules, these remained unchanged when producing the simulated dataset. Consequently, the response variables P_{s3}, T_{t3}, Tt_{49}, N_2, and W_f are based on the GEnx-1B turbomachinery maps that were optimized using the test cell correlation data retrieved from a single engine test. Hence, the response surface learned by the surrogate model is biased towards the test cell engine and will not be optimal for each engine in the fleet. The surrogate model was then trained and validated on this dataset, with its performance assessed through the Root Mean Square Error (RMSE) on a validation subset, accounting for 10% of the simulations. After training, the model's performance was further evaluated by comparing its outputs against actual GEnx-1B engine data, specifically analyzing the Mean Absolute Percentage Error (MAPE) for each sensor and flight phase. This benchmark provides a method of comparison when assessing the quality of GAN-generated data.

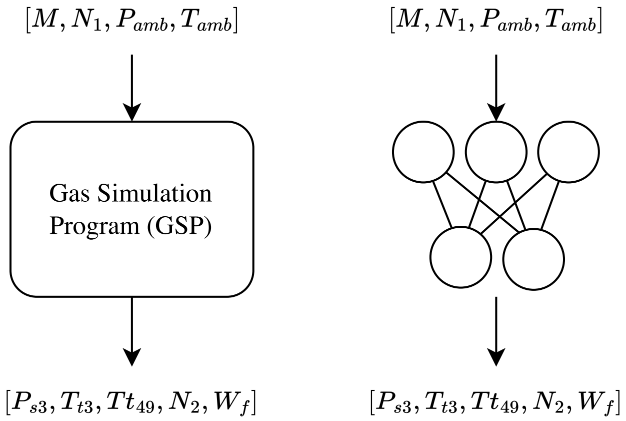

$[M, N_1, P_{amb}, T_{amb}]$
$[M, N_1, P_{amb}, T_{amb}]$
Gas Simulation Program (GSP)
$[P_{s3}, T_{t3}, Tt_{49}, N_2, W_f]$
$[P_{s3}, T_{t3}, Tt_{49}, N_2, W_f]$

Figure 3: A surrogate model, consisting of a four layer fully connected neural network, is pre-trained on 1.2 million scenarios simulated by Gas turbine Simulation Program (GSP) for different operating conditions of the GEnx-1B.

3.5. *Engine fault detection model*

The influence of the GAN-generated data on the failure classification performance, as stated by **R1.2** in subsection 3.1, is evaluated by first establishing the predictive performance on the original dataset before applying the GAN. The failure classification model is designed to differentiate between two conditions: 'non-failure' and 'failure'. Thus, the classifier serves as a tool for fault detection in the fleet.

3.5.1. *Feature & model selection*

The primary purpose of the failure classification model is to distinguish between data from an engine's initial operating phase, where deterioration is minimal, and data before an unscheduled engine overhaul that suggests a failure might be imminent. In light of the challenges presented by manual feature engineering, especially due to the diverse range of potential faults in engine systems [19], we use deep learning techniques to automate the feature engineering process. The deep learning model is specifically guided to explore features within a predefined scope, comprising operating conditions and gas path parameters. This approach builds on the premise that GPA-based simulators can effectively reproduce the degraded states of an engine under various operating conditions by adjusting the health parameters of each module.

To also facilitate learning across time series data, a recurrent neural network (RNN) is utilized containing a single Long-Short Term Memory (LSTM) layer comprising 16 units. This is followed by a fully connected layer with 32 units, activated by a Rectified Linear Unit (ReLU) function, and then a batch normalization layer is added. The architecture is completed with a final layer that employs a sigmoid activation function, which is suitable for addressing binary classification problems. Moreover, we selected the default settings for the Adam optimizer.

3.5.2. *Preparing training samples*

The training samples are prepared in batches that include sequences from both non-failure and failure classes, ensuring each sequence has a fixed length. These sequences are fed into the classification model per batch with dimension size of batch size, window size, and feature count. Adhering to a similar framework described by Fawaz [15]. The training dataset consists of a collection of pairs (X_i, Y_i), where X is a time-ordered two-dimensional array with each sensor, $[x_1, x_2, ..., x_t]$, spanning t time steps.

Given the scarcity of failure data relative to non-failure data, we adopt the strategy of using the smallest stride possible (stride = 1) when sampling within the failure region. This is depicted in Figure 4 where the non-failure region is greater than the failure region. On the contrary, the stride for training samples in the non-failure region is adjusted such to achieve a balanced training dataset. It is important to note that we define the length of the training samples in terms of flight cycles, with one flight cycle equivalent to three measurements (i.e. take-off, climb, and cruise).

To ensure clarity and consistency in the training process of the classification model, we strategically sample the training data from periods between trajectories and water washes. For this approach, we utilize the unique identification numbers assigned to each trajectory and water wash to accurately group the sequences, as explained in subsection 3.2. As a result, this approach prevents abrupt sensor measurement changes within the samples at the cost of a reduced number of training samples. Nonetheless, this strategy considerably enhanced the quality of the training dataset. Experiments indicated that neglecting these preprocessing steps compromised the model's ability to distinguish between the non-failure and failure classes; it was no better than flipping a coin.

3.5.3. *Labeling & cross validation*

Estimating the failure region prior to engine overhaul is challenging as the exact time of failure is generally unknown. This uncertainty makes it hard to label the data accurately for the purpose of training the classification model. In this research, we attempt to classify whether a fixed time period shows signs of failure or non-failure limiting to unscheduled maintenance events only. It is based on the idea that an accurate guess of the failure region will demonstrate good predictive performance on the validation data. Furthermore, the failure class does not distinguish the different faults, removal reasons, and module specific problems that triggered the unscheduled maintenance events. Hence, there exist a variety of faults in the failure data.

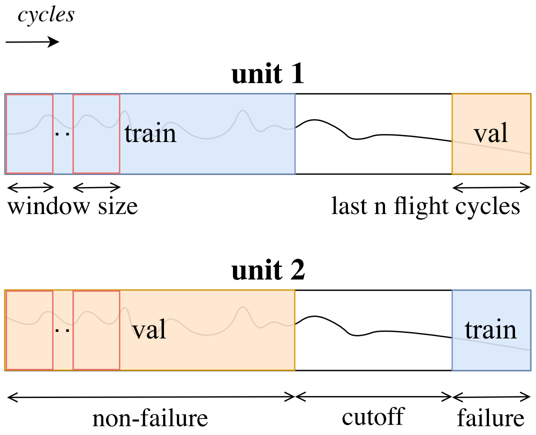


Figure 4: The schematic illustrates the allocation of engine measurement data to training (blue) and validation (orange) sets based on engine unit and class label, ensuring no overlap. The *last N flight cycles* parameter indicates how many cycles before an unscheduled engine removal were labeled to the failure class within a single trajectory.

The extent of the failure region is governed by the *last N flight cycles* parameter, as shown in Figure 4. During model training and validation, the window size and the *last N flight cycles* parameters are varied. For each combination, the model is subjected to 5-fold cross-validation. Engine measurement data are segregated by engine unit and class label, ensuring that training samples from a specific engine in a given class are exclusive to the training set and not duplicated in the validation set. Adhering to this strict separation ensures that performance metrics will generalize across different engine units since differences between individual

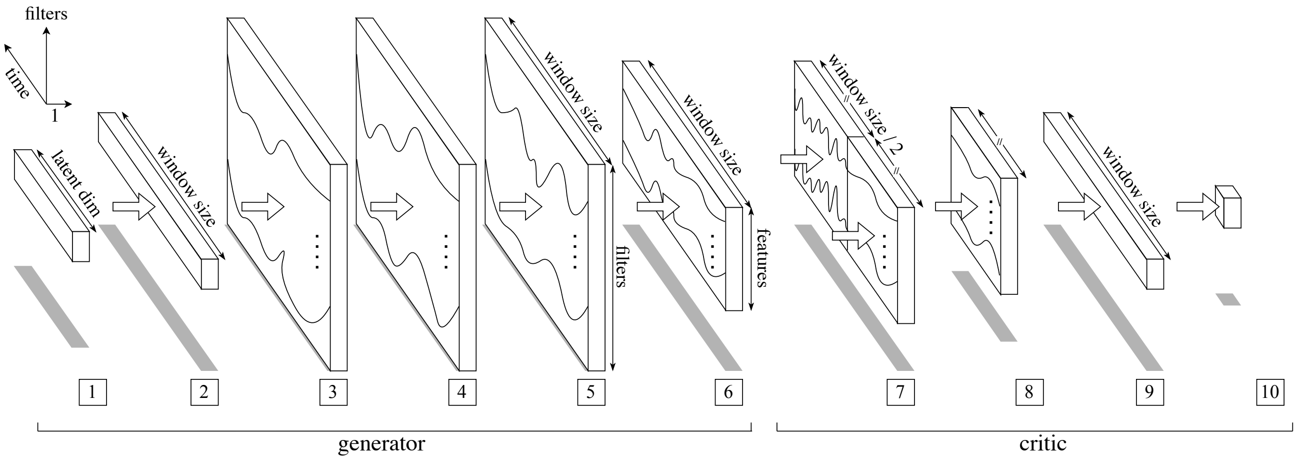


Figure 5: The generative adversarial network architecture used to learn the distribution of the GEnx-1B failure data comprises of two 1D convolutional neural networks. The generator network (1-6) consists of three transposed convolutional layers (3-6) which upscale the latent space representation. The critic (7-10), consisting of a multi-scale convolutional layer (7), down-samples the input time series and assesses the authenticity the generated sequences against real sequences with the final (11) output layer.

engines within the same family have also been observed due to manufacturing issues [49, 19]. Furthermore, to avoid model confusion we take only 60% of the total trajectory as depicted by the cutoff. The rest of the data in the cutoff region is not included in the training or validation set.

The 5-fold cross-validation is executed with four different seeds, and the window size is evaluated at 30, 50, and 70, while the *last N flight cycles* are tested at intervals of 90, 110, 130, and 150. Model training is stopped after 8 epochs at which the F1-score of the model is assessed. Finally, data augmentation with GANs is reserved for instances where the classification model was successful in differentiating between the two classes.

3.6. Data augmentation with GANs

To study the primary research question of this paper, **R1** from section 3.1, a generative adversarial network (GAN) is trained with data from the failure region. The classification model is then cross-validated with the new augmented dataset as described in section 3.5.3. The overall architecture of the GAN is based on 1D Convolutional Neural Networks (CNNs) optimized for a Wasserstein loss with gradient penalty.

3.6.1. GAN Architecture

The generator was structured with four transposed convolutional layers for data upsampling. As Illustrated in Figure 5, the GAN maps the latent space (1) to the first dense layer (2) with a number of units equal to the window size of the training samples reshaping it to a two-dimensional vector of dimensions (window size, 1). Following this are four one-dimensional transposed convolutional layers (3-6) sharing the same parameters: a stride of 1, 32 filters, a kernel size of 10, and equal padding. Between each layer, we use a Leaky ReLU activation function with an alpha of 0.01. The experiments conducted during the research demonstrated that the Leaky ReLU activation function substantially improved convergence results. The activation function in the last layer is omitted as suggested by Huang and Deng [26]. The final layer's filter count (6) is adjusted based on the number of sensors the GAN is expected to simulate, providing the GAN with the flexibility to generate samples of different fixed sizes and sensor counts. Thus, the generator will synthetically produce a sequence of values for multiple operating conditions and gas path parameters as required by the classification model: M, N_2, P_{amb}, T_{amb}, P_{s3}, T_{t3}, Tt_{49}, N_2, and W_f.

The critic, or discriminator, is structured with three layers in total: two convolutional layers (7-8) with 16 filters each, and a fully connected layer (9) with 16 units. The final layer (11) outputs the score of the associated input sequence. Furthermore, the multi-scale layer (7) utilizes kernel sizes of 9 and 27 to capture features across multiple temporal scales. Introducing this feature benefited the GAN's ability to reach good convergence results. Because of the varied kernel sizes, even padding is applied to the convolutions before merging the filters in the concatenation layer. Subsequently, in the second convolutional layer (8), we use a kernel size of 18, after which the output is flattened before entering the fully connected layer (9).

3.6.2. GAN training

The development of a baseline classification model, discussed in subsection 3.5, not only benchmarks its performance but also helps in identifying the extent of the failure region within the dataset. This allows for unsupervised GAN training focusing only on augmenting the data in the failure region. As a result, more failure data samples are synthetically created whereas samples from the "non-failure" region are acquired by changing the stride of the windowing algorithm for that region.

For optimization, we utilize the Wasserstein loss function with gradient penalty (WGAN-GP) [2, 23]. This strategy was found to consistently produce stable and successful results compared to using the conventional binary cross entropy loss function. Both the generator and critic training are governed by the Adam optimization algorithm, adopting a learning rate of 5×10^{-3} and beta values $\beta_1 = 0$ and $\beta_2 = 0.9$ [23]. Furthermore, the critic is updated twice for every single update of the generator over the course of 40 epochs.

The GAN is individually trained for each fold in the cross-validation process. This process begins with the random partitioning of the dataset into two distinct subsets: one for training and the other for validation purposes. In the initial phase, the classification model undergoes training on the training set, followed by a performance evaluation on the validation set. Subsequently, the GAN is trained using the same training set, focusing only on data samples within the failure region. Upon completing this training, the original training dataset is augmented with the failure data synthesized by the GAN. Finally, the classification model is trained again on the augmented dataset, leading to an updated metric of the validation F1-score for fault detection. Since this process is repeated across for multiple folds and seeds, the uncertainty in the evaluation process is mitigated providing more robust and reliable results about the GAN's impact on the baseline fault detection model.

3.6.3. GAN evaluation with the surrogate model

The GAN is designed to generate synthetic data that contains both operating conditions and gas path parameters. These variables are interrelated and follow physical principles. Once the GAN training is complete, the synthetic sensor data it generates is assessed for its physical realism. This assessment uses the pre-trained GPA-based surrogate model, detailed in Section 3.4, to analyze the GAN-produced data. Using the domain knowledge integrated into the surrogate model increases the understanding of the GAN's capability to replicate the complex interdependencies among sensor readings [59]. This approach also facilitates an additional evaluation process for unsupervised GANs which do not generate new samples based on preset conditions.

The differences between the data generated by the GAN and predictions from the surrogate model are compared under varied operational conditions by calculating the percentage error across 300 time series samples for the five gas path parameters ($P_{s3}, T_{t3}, Tt_{49}, N_2$, and W_f). These errors are anticipated due to the surrogate model being trained on simulated data, which naturally differs from the GAN's training on actual in-flight data. To set a reference error, the errors from real flight data are also determined against the surrogate model's predictions. The comparison of these error distributions provides insights into the similarity between real and GAN-generated data based on domain knowledge about turbofan engines.

4. Results & Discussion

This section outlines the results of the research. First, the surrogate model in section 4.1, developed with data from a GPA-based simulation tool, is validated against in-flight measured data collected from a fleet of GEnx-1B turbofan engines. Secondly, section 4.2 analyzes the classification model's performance in differentiating between "failure" versus "non-failure" samples. Thirdly, insights are shared in section 4.3 on the GAN's ability to create synthetic samples by evaluating the real and synthetic distributions and ensuring plausible gas path parameter values that adhere to known physical laws and operating conditions. The latter is carried out using the aforementioned surrogate model which holds domain knowledge about the relationship between operating conditions and gas path parameters. Moreover, the classification model undergoes retraining with the GAN-augmented dataset to determine the method's effectiveness on fault detection, utilizing the F1-score as the evaluation metric.

4.1. Validation of the surrogate model

The surrogate model was initially trained and validated on the simulated dataset achieving an overall Root Mean Square Error (RMSE) of 2×10^{-3}. However, to thoroughly assess the surrogate model's effectiveness, it was applied to real-world operating conditions and sensor data from in-flight engine measurements. Figure 6 illustrates the comparison between the simulated response variables to the values measured during flight allowing for the error analysis. For each sensor, the mean absolute percentage error is plotted per flight condition. The analysis revealed significant discrepancies in simulated sensor values, particularly for the exhaust gas temperature (T_{t49}) and fuel flow (W_f), which displayed the largest errors. Additionally, the error variation for parameters T_{t49}, N_2, and W_f was influenced by the flight conditions, with more pronounced deviations during cruise conditions. This observation aligns with the direct application of the GPA simulation tool also found in [47].

The discrepancies can be attributed to a variety of factors, including oversimplifications within the engine performance model itself, disparities between the surrogate model and the GPA engine performance model, and the inherent deteriorated condition of engines leading to different response surfaces. The latter may be substantial since the in-flight measured snapshot data covers a broad range of the engine life cycle. Additionally, noise in the GEnx-1B flight data and differences among individual engine units also contribute to these discrepancies. Specifically, the GPA model was calibrated using test cell correlation data from an engine later in its life cycle, which may introduce variability. Installation effects, though requiring further verification, may also play a role.

4.2. Baseline failure classification model

As outlined in Section 3.5.3, an RNN-based classification model is established to classify a time series sample as

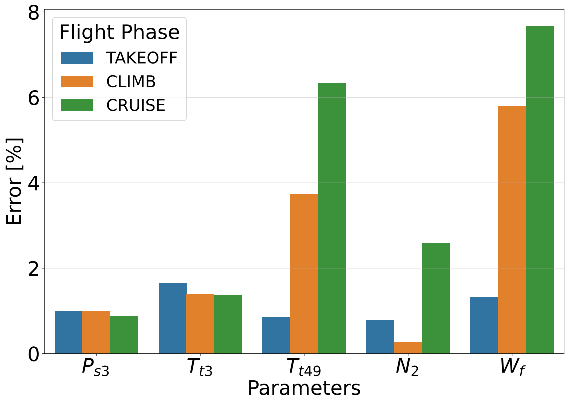


Figure 6: The mean absolute percentage error deviation between in-flight and surrogate model simulated data plotted per flight phase and gas path parameter.

either "failure" or "non-failure". Due to the unavailability of precisely labeled data on failure events, various dataset configurations were evaluated by adjusting the *last number of cycles* parameters, which represents the number of cycles assigned to the "failure" class label before an unscheduled maintenance event. Besides this labeling parameter, the window size is also varied when training the classifier.

Table 2 demonstrates the F1-score of the classifier for all parameter combinations. The model did not appear to have learned correct decision boundaries for seven of the twelve tests. The scores were mostly around 0.55. Nevertheless, five out of the twelve tests resulted in a 0.82-0.83 F1-score on the validation set. The configurations that resulted in a high F1-score (bold) provide evidence of the classifier's ability to detect differences between failure and non-failure samples. These distinctions are critical for defining the failure region since the failure data will be augmented by the GAN. The reason for poor performance on the other seven tests was not determined. Because of this, GAN augmentation was only applied to the five successful tests. The results are discussed in the next section.

Table 2
Average F1-scores for the failure classification model are provided for various combinations of the *last number of cycles* and the window size of the time series samples. The average scores are based on a 5-fold cross validation for 4 different seeds. Successful runs, as highlighted in the table, are further studied for GAN augmentation.

F1-score	*window size (cycles)*		
last N cycles	*30*	*50*	*70*
90	**0.827**	0.539	0.460
110	0.559	0.564	**0.824**
130	0.557	**0.831**	0.551
150	0.588	**0.834**	**0.828**

4.3. GAN evaluation

This section utilizes different methods to assess the quality of the synthetic samples produced by the GAN. Initially, these samples are assessed by analyzing their feature distribution and comparing them to the real dataset. Subsequently, dimensionality reduction techniques, specifically t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP), are employed to compare the real and synthetic samples in a reduced dimensional space. Following this, the surrogate model is utilized to examine how well the GAN has learned the relationship between synthetic operating and gas path parameters. Lastly, the classifier performance is reassessed using the GAN-augmented dataset to determine the effect on the F1-score.

4.3.1. Data distribution

The density plots in Figure 7 indicate a close match between the real and GAN-generated data distributions across most sensor variables which suggests that the GAN was able to effectively capture the statistical properties of the data. The density plot pairs exhibit the same shapes with peaks and troughs occurring around comparable values. A similar conclusion is drawn for the spread and tail distribution. In certain cases, however, such as at the Mach number M, ambient temperature T_{amb}, and core speed, N_2, the peaks are slightly higher compared to the training dataset. Overall the GAN can capture key properties and has obtained a reasonable similarity to the GEnx-1B flight data.

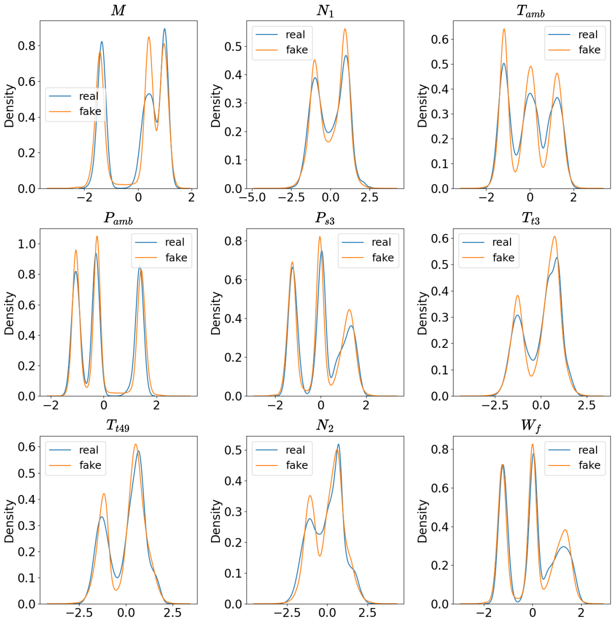


Figure 7: The distribution of the GAN generated samples (fake) is compared to the in-flight measured GEnx-1B data distribution (real) for each variable.

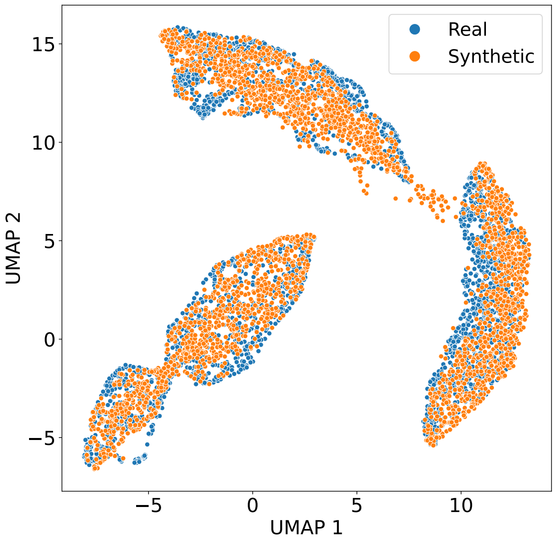


Figure 8: The UMAP algorithm was applied to the GAN generated data (fake) and the GEnx-1B flight data (real). The three clusters of the synthetic and real data overlap and represent the three flight phases recorded by the engine monitoring unit.

The degree of overlap is also studied between the synthetic and real samples using the UMAP algorithm [41] and the t-SNE [39] comparing the distribution of the data on a reduced feature space. The results are presented in Figure 8 and Figure 9, respectively. The UMAP algorithm is configured using 30 neighbors and a minimum distance of 0.2 to characterize the data in two-dimensional space. Furthermore, we utilize default values for the t-SNE algorithm; perplexity 30, early exaggeration 12, automated learning rate, 1000 iterations, and use the Euclidean distance as the metric. Both figures demonstrate three clusters for the in-flight and synthetic data where each cluster represents one of the three flight phases. This similarity indicates that the synthetic data points could potentially be used as a proxy for real data in applications where additional data is beneficial.

4.3.2. *Evaluation with the surrogate model*

The GPA-based surrogate model facilitates the evaluation of synthetic operating conditions alongside corresponding gas path parameters generated by the GAN. The assessment is carried out after training the generator. Figure 10 summarizes the critic loss of the GAN for 100 training runs plotting the 75% percentile interval. It can be observed that the training process for multiple training runs converges close to zero. Additionally, the variance of the loss decreases over time which is a sign of stable learning. Since multiple GANs are trained for different training sets the results imply that the training process is robust for these variations.

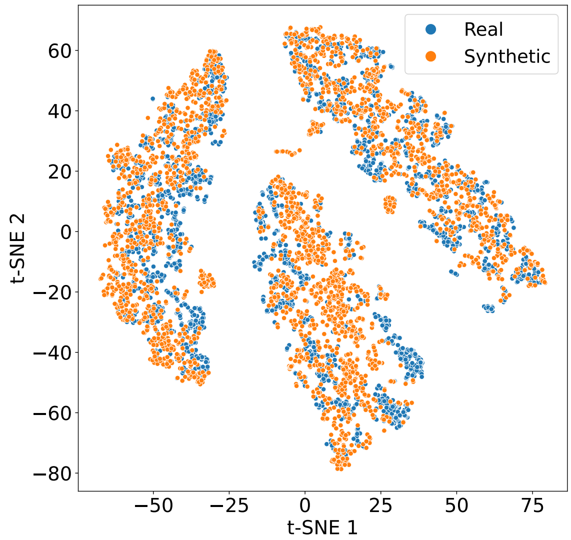


Figure 9: The GAN-generated data (fake) and GEnx-1B flight data (real) were analyzed using the t-SNE algorithm resulting in three clusters, each representing one of the three flight phases.

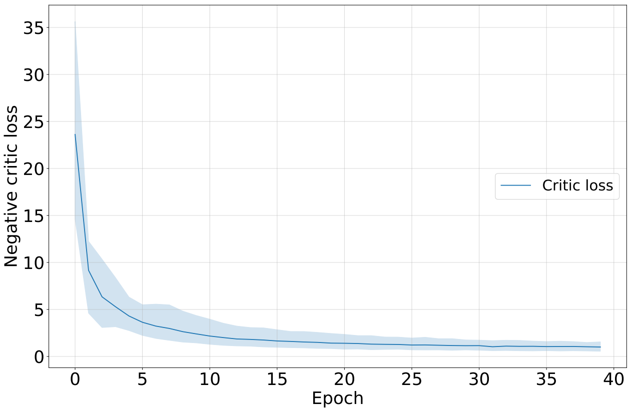


Figure 10: A total of 100 GAN runs are evaluated and their 75% percentile intervals are plotted over the epochs. The negative critic loss exhibits robust convergence, indicating a stable learning curve between the generator network and the critic.

The GAN-generated and in-flight measured gas path parameters were evaluated against the expected values determined by the surrogate model under specific operating conditions. The error distributions displayed in Figure 11 reveal significant alignment between the GAN outputs and actual data, particularly for parameters T_{t3}, T_{t49}, and N_2. This alignment highlights that the discrepancies observed between real and modeled gas path parameters have been effectively captured by the GAN. Notably, the different peaks observed in T_{t49} and N_2 stem from flight phase error biases, as detailed in subsection 4.1. The GAN's ability to mirror these error distributions indicates it has successfully learned physically relevant characteristics from the real flight data. Regarding the P_{s3} parameter, while the GAN-generated data shows greater variance than the actual data, the mean values of both distributions align closely. This difference may be caused due to higher variance already present in the real data resulting in a increased difficulty for GAN training. A similar patterns can also be observed for the fuel flow W_f parameter. However, in this case, the fuel flow data is expected to have high variability since the fuel flow sensor is relatively inaccurate compared to the other sensors. Yet, the majority of GAN-generated data remains within a reasonable range of values. These observations underscore the GAN's capacity to generate data that reflects realistic physical properties.

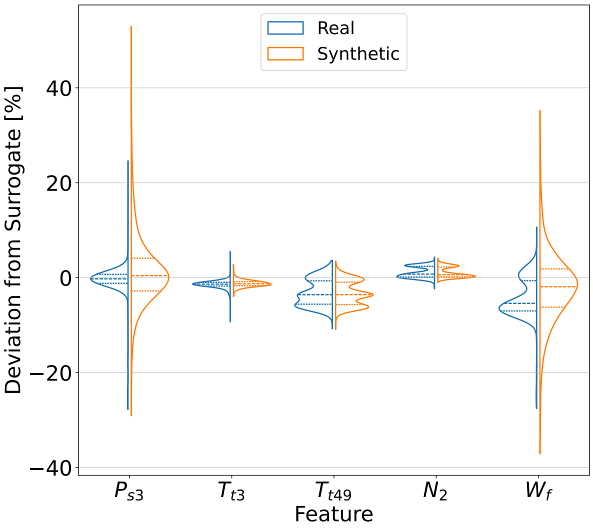


Figure 11: Per operating condition, the real and generated gas path parameters are compared with the expected values from the surrogate model. Having the real data as a reference, the percentage error distributions are plotted for each sensor variable.

4.3.3. *GAN impact on fault classification*

A total of 5 parameter sets were evaluated to study how augmenting the dataset with a GAN affects the classifier's performance. These corresponded to the parameter sets that led to good results in the baseline fault detection model. Each parameter set indicated the final number of cycles labeled as "failure" before an unscheduled maintenance event and the specific window size used to create the time series samples for the classifier. The performance of the classifier trained on the original dataset and the augmented dataset were compared based on the F1-score. Within the pipeline, it was ensured that the same dataset was used to train the baseline classifier and the GAN before augmentation.

Out of the 5 experiments, the augmented dataset achieved good results on 3 experiments, illustrated in Figure 12. For these parameter sets, the GAN-augmented data exhibited a higher median F1-score compared to the original dataset, suggesting an improvement in the classifier's ability to predict failures. The improvement is particularly notable in the (lastn130, window50) and (lastn150, window70), where the augmented data's interquartile range and median surpass the original. The first set, (lastn90, window30), also shows an elevated median F1-score for the augmented data but the effect is less. It must be noted that the experiment testing the (lastn110, window70) parameter combination resulted in a substantially high variance in the distribution of the F1-score raising concerns about the validity of these results specifically. However, the second experiment, which resulted in an overall lower F1-score for all trained models, did have a reasonable variance.

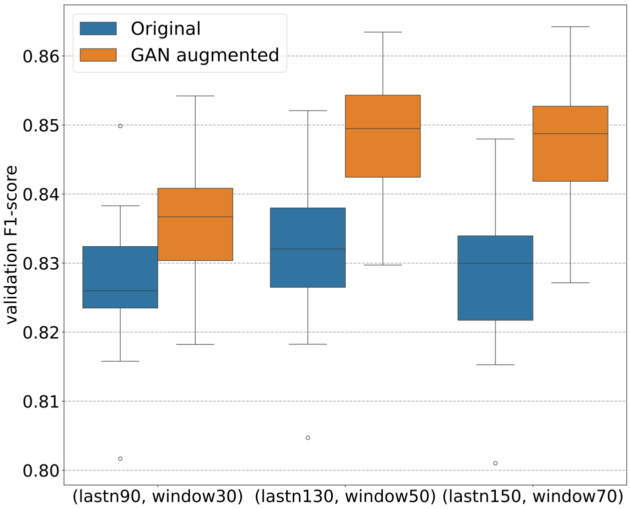


Figure 12: Results after augmenting the original dataset with the GAN are provided for the three successful parameter combinations. Cross validation (5-fold) of the fault detection model with the GAN shows improvements in the F1-score as a result of data augmentation.

These findings demonstrate the effectiveness of data augmentation with GANs and their ability to improve fault detection. Increases of up to 2.8% in the average F1-score are observed with respect to the fault detection model trained on the original dataset. For these cases, the consistent improvement underscores the GAN's ability to generate data samples that are representative of the underlying dataset. Additionally, because of the way the data is preprocessed, the training data contains flight data labels that do not appear in the validation set and, hence, the results also demonstrate how GAN augmentation improves performance on unseen engine class labels. While these are promising results, the impact of the sample window sizes and labeling process must still be considered and further studied.

5. Conclusion

This study explored the potential of Generative Adversarial Networks (GANs) in augmenting operating and gas path parameters collected from real-world GEnx-1B aircraft engine data. Amidst challenges such as data imbalance and the scarcity of failure instances, our research introduced a Wasserstein GAN with Gradient Penalty (WGAN-GP) based on 1D convolutional neural networks (CNNs) to generate synthetic time series of failure data. The predictive performance of a baseline recurrent neural network (RNN) was first established by distinguishing between non-failure and failure time series data. After augmenting the original dataset with the GAN, the performance of the baseline model was reevaluated. Furthermore, to compensate for the unsupervised nature of GAN learning, a surrogate model, pre-trained on a GPA-based simulated dataset, was used to verify the credibility of the GAN-generated samples. This validation focused on analyzing the interrelation between generated operating conditions and response variables to determine their physical plausibility.

The GAN's loss metrics indicated stable and convergent training across multiple experiments, suggesting an effective architecture in this regard. Furthermore, analysis via the surrogate model reveals that the GAN produced plausible gas path response variables for the synthetic operating conditions, confirming its capability to understand the underlying physics between these variables. However, while most synthetic response variables remained within acceptable limits, some synthetic variables exhibited more variance in the error distribution than others. Nevertheless, these findings suggest that the GAN is capable of generating data with significant physical relevance. Importantly, the results demonstrated that integrating GAN-generated data into the original dataset enhanced the fault detection classification model's validation F1-score by as much as 2.8%. Furthermore, given that the training dataset included class labels from various engine units not present in the validation set, these improvements suggested the GAN's proficiency in identifying and replicating critical degradation patterns across different engines.

6. Future work

This research explored the application of data augmentation to improve fault detection of deep learning models in aircraft engine applications. In particular, Generative Adversarial Networks (GANs) were studied to produce synthetic time series data. The lack of labeled data in safety-critical systems, such as aircraft engines, poses considerable challenges for fault detection. Future work should extend towards improving the labeling process of real-world data. Literature indicates that unsupervised or semi-supervised labeling techniques can be a better approach in terms of time and costs for these types of applications [56, 7, 45, 29].

Although this study focused on fault detection, the GANs effect on the predictive performance can also be extended to diagnostics and prognostics models. For instance, the training process of the GAN could be focused towards generating synthetic data of a specific type of failure mode. From a prognostics perspective, new synthetic data may improve the predictions on the future health state of the engine. Furthermore, including traditional machine learning models in the analysis may offer greater transparency compared to "black-box" deep learning frameworks. This could enable more research in explainable artificial intelligence for predictive maintenance. Aligning with suggested research directions by Zhang et al. [59], domain-specific knowledge can be included to guide the optimization process of the GAN which could improve the quality of the generated samples. Finally, introducing the operating conditions in combination with conditional GANs also presents a promising avenue for simulating synthetic data of aircraft engines and may offer a more direct evaluation approach.

Acknowledgements

Thanks to KLM Engine Services for supporting this research with GEnx-1B flight data. Furthermore, the authors are grateful to Walid Brachmi for providing the simulated data from Gas turbine Simulation Program to develop the surrogate model.

CRediT authorship contribution statement

Daniel Cisneros Acevedo: Conceptualization, Methodology, Software, Investigation, Validation, Writing - review & editing.

References

[1] Arias Chao, M., Kulkarni, C., Goebel, K., Fink, O., 2021. Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics. Data 6, 5. doi:10.3390/data6010005.

[2] Arjovsky, M., Bottou, L., 2017. Towards Principled Methods for Training Generative Adversarial Networks .

[3] Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein GAN .

[4] Arnelid, H., Zec, E.L., Mohammadiha, N., 2019. Recurrent Conditional Generative Adversarial Networks for Autonomous Driving Sensor Modelling, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE. pp. 1613–1618. doi:10.1109/ITSC.2019.8916999.

[5] Baptista, M.L., Henriques, E.M., 2022. 1D-DGAN-PHM: A 1-D denoising GAN for Prognostics and Health Management with an application to turbofan. Applied Soft Computing 131, 109785. doi:10.1016/j.asoc.2022.109785.

[6] Brophy, E., Wang, Z., She, Q., Ward, T., 2023. Generative Adversarial Networks in Time Series: A Systematic Literature Review. ACM Computing Surveys 55, 1–31. doi:10.1145/3559540.

[7] Chen, C., Lu, N., Jiang, B., Xing, Y., 2022. A Data-Driven Approach for Assessing Aero-Engine Health Status. IFAC-PapersOnLine 55, 737–742. doi:10.1016/j.ifacol.2022.07.215.

[8] Chen, Y.Z., Zhao, X.D., Xiang, H.C., Tsoutsanis, E., 2021. A sequential model-based approach for gas turbine performance diagnostics. Energy 220, 119657. doi:10.1016/j.energy.2020.119657.

[9] Choudhary, A., Mishra, R.K., Fatima, S., Panigrahi, B., 2023. Multi-input CNN based vibro-acoustic fusion for accurate fault diagnosis of induction motor. Engineering Applications of Artificial Intelligence 120, 105872. doi:10.1016/j.engappai.2023.105872.

[10] Costa, P.R.d.O.d., Akcay, A., Zhang, Y., Kaymak, U., 2023. Attention and Long Short-Term Memory Network for Remaining Useful Lifetime Predictions of Turbofan Engine Degradation. International Journal of Prognostics and Health Management 10. doi:10.36001/ijphm.2019.v10i4.2623.

[11] Darrah, T., Lovberg, A., Frank, J., Biswas, G., Quinones-Gruiero, M., 2022. Developing Deep Learning Models for System Remaining Useful Life Predictions: Application to Aircraft Engines. Annual

Conference of the PHM Society 14. doi:10.36001/phmconf.2022.v14i1.3304.

[12] De Giorgi, M.G., Menga, N., Ficarella, A., 2023a. Exploring Prognostic and Diagnostic Techniques for Jet Engine Health Monitoring: A Review of Degradation Mechanisms and Advanced Prediction Strategies. Energies 16, 2711. doi:10.3390/en16062711.

[13] De Giorgi, M.G., Menga, N., Mothakani, A., Ficarella, A., 2023b. A data-driven approach for health status assessment and remaining useful life prediction of aero-engine. Journal of Physics: Conference Series 2526, 012071. doi:10.1088/1742-6596/2526/1/012071.

[14] Esteban, C., Hyland, S.L., Rätsch, G., 2017. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs .

[15] Fawaz, H.I., 2020. Deep learning for time series classification .

[16] Fentaye, Baheta, Gilani, Kyprianidis, 2019. A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities. Aerospace 6, 83. doi:10.3390/aerospace6070083.

[17] Fentaye, A.D., Zaccaria, V., Kyprianidis, K., 2021. Aircraft Engine Performance Monitoring and Diagnostics Based on Deep Convolutional Neural Networks. Machines 9, 337. doi:10.3390/machines9120337.

[18] Ferreira, C., Gonçalves, G., 2022. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. Journal of Manufacturing Systems 63, 550–562. doi:10.1016/j.jmsy.2022.05.010.

[19] Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.J., Ducoffe, M., 2020. Potential, challenges and future directions for deep learning in prognostics and health management applications. Engineering Applications of Artificial Intelligence 92, 103678. doi:10.1016/j.engappai.2020.103678.

[20] Fontes, C.H., Pereira, O., 2016. Pattern recognition in multivariate time series – A case study applied to fault detection in a gas turbine. Engineering Applications of Artificial Intelligence 49, 10–18. doi:10.1016/j.engappai.2015.11.005.

[21] Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K., Abraham, A., 2023. Multi-fault diagnosis of Industrial Rotating Machines using Data-driven approach : A review of two decades of research. Engineering Applications of Artificial Intelligence 123, 106139. doi:10.1016/j.engappai.2023.106139.

[22] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Networks .

[23] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. Improved Training of Wasserstein GANs .

[24] Hanachi, H., Liu, J., Banerjee, A., Chen, Y., Koul, A., 2015. A Physics-Based Modeling Approach for Performance Monitoring in Gas Turbine Engines. IEEE Transactions on Reliability 64, 197–205. doi:10.1109/TR.2014.2368872.

[25] Hepperle, N., Therkorn, D., Schneider, E., Staudacher, S., 2011. Assessment of Gas Turbine and Combined Cycle Power Plant Performance Degradation, in: Volume 4: Cycle Innovations; Fans and Blowers; Industrial and Cogeneration; Manufacturing Materials and Metallurgy; Marine; Oil and Gas Applications, ASMEDC. pp. 569–577. doi:10.1115/GT2011-45375.

[26] Huang, F., Deng, Y., 2023. TCGAN: Convolutional Generative Adversarial Network for time series classification and clustering. Neural Networks 165, 868–883. doi:10.1016/j.neunet.2023.06.033.

[27] Iglesias, G., Talavera, E., González-Prieto, , Mozo, A., Gómez-Canaval, S., 2022. Data Augmentation techniques in time series domain: A survey and taxonomy doi:10.1007/s00521-023-08459-3.

[28] International Air Transport Association, 2021. Airline Maintenance Cost Executive Commentary. URL: https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2021-mctg-report_public.pdf.

[29] Koutroulis, G., Mutlu, B., Kern, R., 2022. Constructing robust health indicators from complex engineered systems via anticausal learning. Engineering Applications of Artificial Intelligence 113, 104926. doi:10.1016/j.engappai.2022.104926.

[30] Kumar, P., Raouf, I., Kim, H.S., 2023. Review on prognostics and health management in smart factory: From conventional to deep learning perspectives. Engineering Applications of Artificial Intelligence 126, 107126. doi:10.1016/j.engappai.2023.107126.

[31] Lang, P., Peng, K., Cui, J., Yang, J., Guo, Y., 2021. Data augmentation for fault prediction of aircraft engine with generative adversarial networks, in: 2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS), IEEE. pp. 1–5. doi:10.1109/SAFEPROCESS52771.2021.9693711.

[32] Lei, Y., Li, N., Guo, L., Li, N., Yan, T., Lin, J., 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mechanical Systems and Signal Processing 104, 799–834. doi:10.1016/j.ymssp.2017.11.016.

[33] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., Nandi, A.K., 2020. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mechanical Systems and Signal Processing 138, 106587. doi:10.1016/j.ymssp.2019.106587.

[34] Li, X., Ding, Q., Sun, J.Q., 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety 172, 1–11. doi:10.1016/j.ress.2017.11.021.

[35] Li, Y.G., 2010. Gas Turbine Performance and Health Status Estimation Using Adaptive Gas Path Analysis. Journal of Engineering for Gas Turbines and Power 132. doi:10.1115/1.3159378.

[36] Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., Zhang, H., 2019. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering & System Safety 183, 240–251. doi:10.1016/j.ress.2018.11.027.

[37] Liu, H., Zhou, J., Zheng, Y., Jiang, W., Zhang, Y., 2018. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. ISA Transactions 77, 167–178. doi:10.1016/j.isatra.2018.04.005.

[38] Lövberg, A., 2021. Remaining Useful Life Prediction of Aircraft Engines with Variable Length Input Sequences. Annual Conference of the PHM Society 13. doi:10.36001/phmconf.2021.v13i1.3108.

[39] Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research 9.

[40] Mansouri, M., Dhibi, K., Hajji, M., Bouzara, K., Nounou, H., Nounou, M., 2022. Interval-Valued Reduced RNN for Fault Detection and Diagnosis for Wind Energy Conversion Systems. IEEE Sensors Journal 22, 13581–13588. doi:10.1109/JSEN.2022.3175866.

[41] McInnes, L., Healy, J., Melville, J., 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction .

[42] Mogren, O., 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training .

[43] Ochella, S., Shafiee, M., Dinmohammadi, F., 2022. Artificial intelligence in prognostics and health management of engineering systems. Engineering Applications of Artificial Intelligence 108, 104552. doi:10.1016/j.engappai.2021.104552.

[44] Palacios, A., Martínez, A., Sánchez, L., Couso, I., 2015. Sequential pattern mining applied to aeroengine condition monitoring with uncertain health data. Engineering Applications of Artificial Intelligence 44, 10–24. doi:10.1016/j.engappai.2015.05.003.

[45] de Pater, I., Mitici, M., 2023. Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder. Engineering Applications of Artificial Intelligence 117, 105582. doi:10.1016/j.engappai.2022.105582.

[46] Qiang, N., Dong, Q., Liang, H., Li, J., Zhang, S., Zhang, C., Ge, B., Sun, Y., Gao, J., Liu, T., Yue, H., Zhao, S., 2022. Learning brain representation using recurrent Wasserstein generative adversarial net. Computer Methods and Programs in Biomedicine 223, 106979. doi:10.1016/j.cmpb.2022.106979.

[47] Ramdin, S., Visser, W., Regueiro, J., Rootliep, T., 2023. Systematic Approach for Modelling Modern Turbofan Engines, in: Volume 1: Aircraft Engine, American Society of Mechanical Engineers. doi:10.1115/GT2023-103548.

[48] Rootliep, T.O., Visser, W.P.J., Nollet, M., 2021. Evolutionary Algorithm for Enhanced Gas Path Analysis in Turbofan Engines, in: Volume 1: Aircraft Engine; Fans and Blowers; Marine; Wind Energy; Scholar Lecture, American Society of Mechanical Engineers. doi:10.1115/GT2021-59089.

[49] Saxena, A., Goebel, K., Simon, D., Eklund, N., 2008. Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, IEEE. pp. 1–9. doi:10.1109/PHM.2008.4711414.

[50] Solís-Martín, D., Galán-Páez, J., Borrego-Díaz, J., 2021. A stacked deep convolutional neural network to predict the remaining useful life of a turbofan engine. Annual Conference of the PHM Society 13. doi:10.36001/phmconf.2021.v13i1.3110.

[51] Tahan, M., Tsoutsanis, E., Muhammad, M., Abdul Karim, Z., 2017. Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. Applied Energy 198, 122–144. doi:10.1016/j.apenergy.2017.04.048.

[52] Urban, L.A., 1973. Gas Path Analysis Applied to Turbine Engine Condition Monitoring. Journal of Aircraft 10, 400–406. doi:10.2514/3.60240.

[53] Visser, W.P.J., 2015. Generic Analysis Methods for Gas Turbine Engine Performance: The development of the gas turbine simulation program GSP .

[54] Visser, W.P.J., Pieters, H., Oostveen, M., van Dorp, E., 2006. Experience With GSP as a Gas Path Analysis Tool, in: Volume 2: Aircraft Engine; Ceramics; Coal, Biomass and Alternative Fuels; Controls, Diagnostics and Instrumentation; Environmental and Regulatory Affairs, ASMEDC. pp. 175–182. doi:10.1115/GT2006-90904.

[55] Vollert, S., Theissler, A., 2021. Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE. pp. 1–8. doi:10.1109/ETFA45728.2021.9613682.

[56] Xie, J., Sage, M., Zhao, Y.F., 2023. Feature selection and feature learning in machine learning applications for gas turbines: A review. Engineering Applications of Artificial Intelligence 117, 105591. doi:10.1016/j.engappai.2022.105591.

[57] Xiong, J., Fink, O., Zhou, J., Ma, Y., 2023. Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions. Mechanical Systems and Signal Processing 197, 110359. doi:10.1016/j.ymssp.2023.110359.

[58] Yun, J., Kim, D., Kim, D.M., Song, T., Woo, J., 2023. GAN-based sensor data augmentation: Application for counting moving people and detecting directions using PIR sensors. Engineering Applications of Artificial Intelligence 117, 105508. doi:10.1016/j.engappai.2022.105508.

[59] Zhang, X., Qin, Y., Yuen, C., Jayasinghe, L., Liu, X., 2021. Time-Series Regeneration with Convolutional Recurrent Generative Adversarial Network for Remaining Useful Life Estimation .

[60] Zhao, K., Jia, Z., Jia, F., Shao, H., 2023. Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine. Engineering Applications of Artificial Intelligence 120, 105860. doi:10.1016/j.engappai.2023.105860.

[61] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing 115, 213–237. doi:10.1016/j.ymssp.2018.05.050.

Part II

Literature Study

7 Introduction

Aircraft maintenance contributes approximately 11% to the overall operating cost of airliners [1]. A considerable share of these expenses is directed toward the upkeep and servicing of aircraft engines. Given the intense competition in the aviation industry, airliners are investing in inventive strategies to reduce maintenance costs.

Condition Based Maintenance (CBM) emerges as a crucial strategy that utilises machinery sensor data to inform optimal decision-making based on real-time conditions. This capability not only aids in creating cost-effective maintenance strategies but also improves engine reliability [2]. Central to CBM is the ability to quantify the engine's health based on the monitored data. Because engine health indicators are not directly measurable, Maintenance, Repair, and Overhaul (MRO) engineers combine engine data with Gas Path Analysis (GPA) tools to extract vital health parameters such as efficiency and flow [3][4].

Prognostics and Health Management (PHM) further extends CBM with the ability to predict the future health of an engine. Its primary objective is to predict the Remaining Useful Life (RUL), the period at which a system ceases to perform its intended function. Methods to estimate RUL span from statistical and physics-based to hybrid and artificial intelligence models [5]. In recent years, Deep Learning models have proven to be effective tools in detecting and predicting system failures [6]. Their ability to accurately predict RUL was particularly observed in the 2021 PHM Data Challenge where neural networks were the most popular amongst the winners [7].

However, the limited availability of failure data from safety-critical systems poses considerable challenges for training neural networks [5]. In response, researchers have generated synthetic training data through turbofan system models, such as the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [8][9]. Yet, these approaches, reliant on GPA model approximations and pre-determined degradation paths, are not inherently geared towards industrial applications raising concerns about their usefulness in real-world settings [10][11].

This literature review intends to examine the feasibility of augmenting KLM's GEnx-1B lifecycle data using generative adversarial networks. These have shown to possess exceptional capabilities at generating synthetic data indistinguishable from the original dataset [12]. Furthermore, the cross-disciplinary approach between deep learning and physics-based modelling may play a critical role in producing high-quality synthetic data [13]. For this reason, we also explore potential methodologies of integrating KLM's GPA tools during the training process.

8
Research Question(s) and Objective

Training diagnostics models is challenging due to the limited available failure data in safety-critical systems. To solve this issue, augmenting training set may potentially be a solution. Therefore, the question central to this literature review is as follows:

Research Objective

How can generative models effectively augment existing turbofan life cycle data to improve the predictive performance of diagnostics models?

To establish the scope of this literature review, our focus extends to understanding Gas Path Analysis (GPA) and state-of-the-art deep learning models that have proven to be effective in diagnostics models. Additionally, the groundwork is laid for an analysis of different data augmentation techniques, focusing particularly on time series data. Expanding on the primary research question, this literature review is dedicated to the following sub-level questions.

1. How can generative models be implemented to create realistic turbofan deterioration data?
   (a) How can generative models learn the intrinsic nature of damage propagation and model it realistically?
   (b) How do operational settings and ambient conditions play a role in the turbofan life cycle?
   (c) How can prior knowledge on operational settings and ambient conditions aid in the generation of synthetic data?
   (d) What should the frequency of the generated data be?
   (e) Which sensor measurements are essential for training robust diagnostics models?
   (f) How is the realism of the synthetic turbofan degradation data assessed?
   (g) How do we ascertain that the generative model is not simply ”memorising” the original dataset?
2. How can prior knowledge from the Gas Path Analysis tool, Gas Simulation Program, be included in the training process?
   (a) How should existing architectures of generative models be adapted to include physics-based knowledge?
   (b) How should reference engine models in the Gas Simulation Program be modified to derive health indicators?

9
Gas Turbine Maintenance

Traditional maintenance strategies rely on historical failure event data and are based on periodical and corrective measures [14]. Such an approach may lead to early replacement of healthy engines or, in more dangerous situations, allow malfunctions to occur prior to scheduled maintenance. This approach could potentially compromise engine reliability and increase overall maintenance costs of airliners.

This chapter focuses on three critical areas: (Section 9.1) an exploration of deterioration patterns in gas turbines, (Section 9.2) an examination of the commonly used maintenance strategies by airline operators, and (Section 9.3) a discussion on Gas Simulation Program (GSP) to extract key state variables that describe the health of the engine and methods to quantify deterioration.

9.1. Gas Turbine Deterioration

In general, any faulty component within the engine can lead to machine degradation [15]. Tahan et al. [16] identifies these faults into four different groups: gas path faults, faults in auxiliary subsystems, mechanical errors, and sensor uncertainties (Figure 9.1). Non-performance-based techniques exist to monitor malfunctions in subsystems or mechanical faults such as thermography, acoustics, load, vibration and temperature analysis. In contrast, gas path faults often occur because of aerodynamic or performance related problems. Examples of these issues include fouling, erosion and corrosion of blades, and improper combustion.

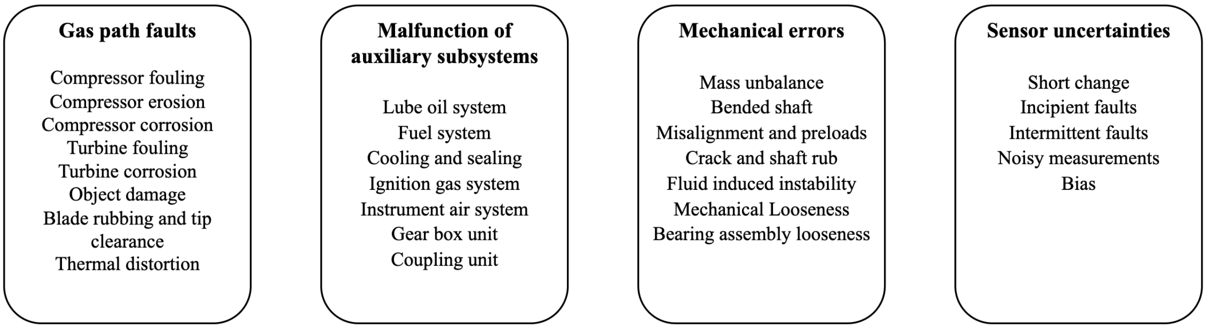


Figure 9.1: Four categories of gas turbine faults as specified by Tahan et al. [16].

The compressors and turbines are the most expensive and important components in aircraft engines and, hence, are often seen as the most critical ones [16]. Kurz, Brun, and Wollie [15] discusses three main degradation effects that influence the performance of compressors. Over its lifetime, the compressor will typically be subjected to increased tip clearances, changes in airfoil geometry, and increased airfoil surface roughness. Furthermore, these effects are augmented since deteriorated stages lead to alteration in the flow exit conditions. Consequently, the downstream compressor stages also operate at non-optimum design conditions.

Besides affecting other compressor stages, a degraded compressor will also affect other components such as the turbine as they start to mismatch. The changes in physical aspects of the component will inherently affect the response surface of the compressor map as shown in in Figure 9.2 and Figure 9.3 which has a compounding effect. In literature [15, 17], the degradation phenomena per component are studied separately. However, researchers also note that in reality these effects interact with each other and do not occur in isolation.

At constant load, loss of compressor efficiency results in reduced discharge pressure and flow, whereas it increases the TIT and heat rate. Simultaneously, compressor fouling or increased tip clearance reduces flow capacity which limits available power at lower ambient temperatures [15]. The relationship between pressure ratio and flow does not alter as it depends on the turbine. However, the engine must rotate faster and, consequently, the compressor power consumption increases as shown in Figure 9.2 and Figure 9.3.

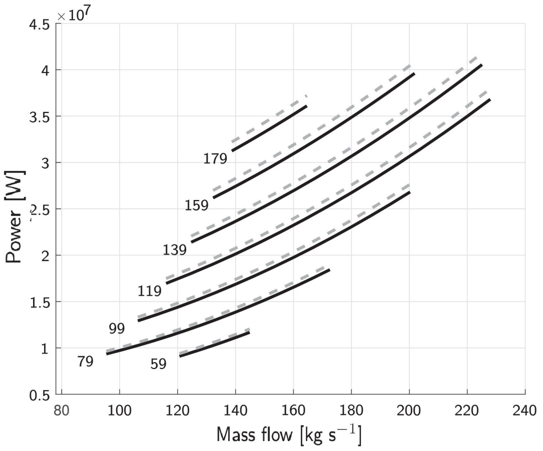


Figure 9.2: Increased power consumption as a result of fouling [18].

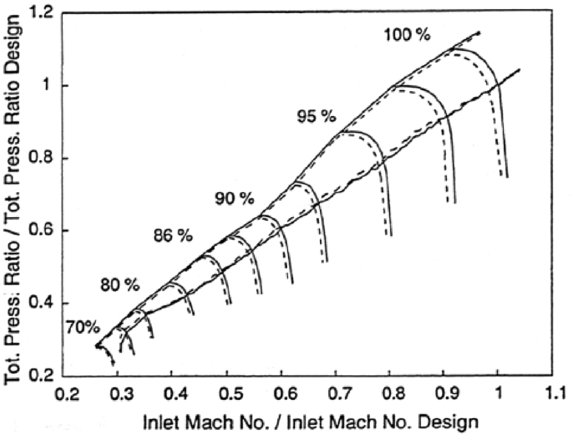


Figure 9.3: Degraded response surface [17].

9.2. Maintenance Strategies

MROs maintain their engines on fixed basis or when certain thresholds are reached in terms of performance degradation. Therefore, it is crucial to extract important data describing the state of the engine to effectively transition the fleet towards condition based maintenance. In the engine life cycle, different reasons exist that may trigger its removal. MROs categorise these reasons based on causes related to the engine (basic) and unrelated to the engine (non basic) such as human error and the environment. The synthesis of this categorisation is depicted in Figure 9.4, where the primary engine removal categories are illustrated [1].

The definitions of relevant engine removal reasons from Figure 9.4 are described in Table 9.1. Within the scope of unplanned reasons, not all removals are due to deteriorated engine performance. For example, *maintenance condition* and *foreign object damage* are reasons unrelated to engine performance. Other reasons are more applicable to a deteriorated engine state.

Table 9.1: unplanned engine removal definitions

Unplanned Reasons	**Definitions**
Item Part Problem	removal for specific part malfunction
Performance	unexpected performance issues
Maintenance Condition	removal because of human error during inspection
Operation Condition	removal due to operational problem
Foregin Object Damage	ingestion of an object unrelated to engine

Planned engine removals are instances where the engine removal process is scheduled. When an engine is maintained before it has failed, the collected data throughout the engine life cycle gets prematurely

[1] KLM document: ERR causes GE definitions R-01-CF6-002

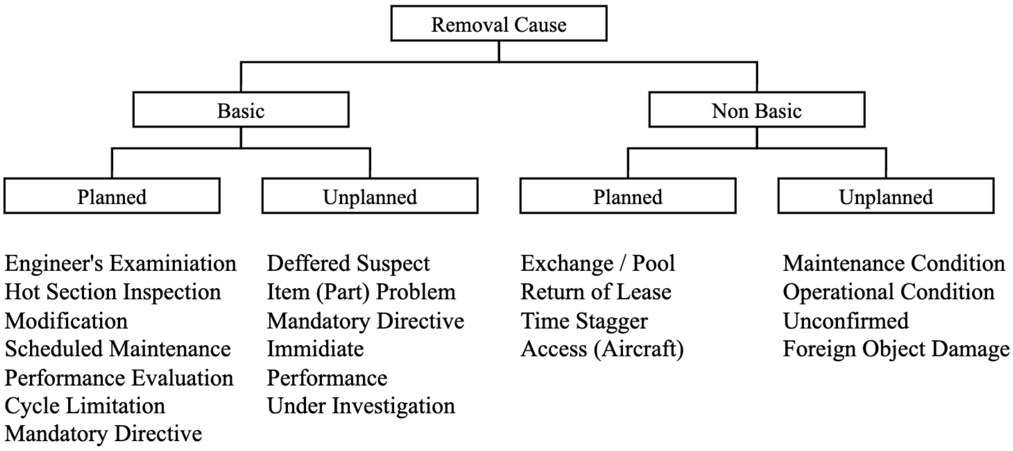


Figure 9.4: Primary removal reasons for engine overhaul. The planned and unplanned causes impact the recorded data for run-to-failure trajectory.

truncated. This may pose challenges to the accumulation of complete run-to-failure datasets which are required for RUL prediction. It is also the main reason why complete run-to-failure datasets of safety-critical systems are rare.

In contrast, unscheduled engine removals are instances where removals are not planned and may, therefore, contain complete run-to-failure datasets of an engine. However, identifying when the engine exactly failed remains difficult since in some cases the observation of a failure during a scheduled inspection does not directly indicate the exact time of the failure. In other words, the engine may have been flying at a failed state the last couple of flights. This could ultimately affect the labelled RUL dataset. Unless the failure is detectable in the recorded data, it is not immediately clear when it happened. Therefore, besides awareness of different engine removal reasons, it is critical to establish clear definitions of failure when collecting representative run-to-failure datasets. In addition, the life cycle of each individual engine must be analysed and considered carefully for the training set.

9.3. Gas Path Analysis

Monitoring gas path faults requires apriori knowledge about the aircraft engine and appropriate placements of sensors along the gas path [16]. Gas Path Analysis (GPA) is a popular performance-based method to analyse engine conditions and detect the gas path faults discussed in Section 9.1.

9.3.1. Gas Simulation Program

A specific example of a GPA tool is the Gas Simulation Program (GSP), a collaborative effort by the Dutch National Aerospace Laboratory (NLR) and Delft University of Technology. GSP allows for modelling and simulation of gas turbines which allows for performance diagnostics and quantification of engine health [19]. The application has the option to adjust the flow and efficiency parameters, which are indicative of engine health. This feature is especially useful when trying to match the simulated gas path variables with in-flight measurements. As a result, state variables related to specific health conditions of the engine can be quantified. With data like ambient pressure, temperature, Mach number, and N1 spool speed, depicted in Figure 9.5, GSP can estimate the sensor variables at multiple gas path stations.

GSP utilises Newton-Raphson numerical solver to find a solution of the system of non-linear algebraic equations that satisfy the conservation laws. To explain the convergence process, Visser [3] considers the vector $\bar{S} = [s_1, s_2, ..., s_n]$ describing the state of the engine. Simple engine models use as few as 4 independent engine states, while more complex models require up to 50 states. In other words, an error vector $\bar{E}$ exist as a function of $\bar{S}$ which has to be reduced to near-zero:

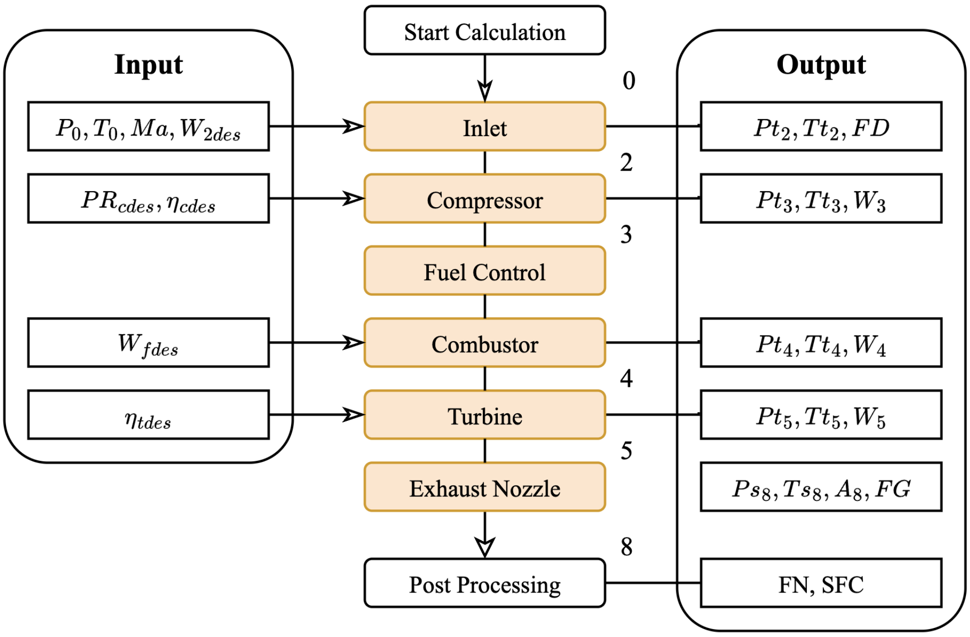


Figure 9.5: Overview of input and output variables from GSP [3]. The numbers correspond to the different engine stations along the gas path.

$$\bar{E}(\bar{S}) = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \tag{9.1}$$

The Newton-Raphson method aims to update the state variables s_n towards error reduction by linearly estimating the gradient of e_n relative to s_n. Each state variable is slightly perturbed to find its gradient with respect to the error. Consequently, the state vector is updated with the Jacobian matrix until the error is reduced below a specified threshold. Hence, the resulting state represents a solution to the system of equations.

$$J_{i,j} = \frac{\Delta e_i}{\Delta s_j} \qquad \bar{S}_{i+1} = \bar{S}_i - f \cdot J_i^{-1} \cdot \bar{E}_i \tag{9.2}$$

$$\tag{9.3}$$

9.3.2. Quantifying Degradation with Adaptive Modelling

Extracting the current health of an operational engine is essential for diagnostics. Visser, Kogenhop, and Oostveen [19] implemented Adaptive Modelling (AM) within GSP to estimate engine deterioration parameters. The concept compares the measurements of a deteriorated engine with the baseline engine model, where efficiency and mass flow are unknowns, and adapts the healthy component maps with so-called map modifiers.

In GSP, the measured variables are included during the Newton-Raphson optimisation process (Equation 9.1) by extending the core set of equations using the *AM control module* [3]. The model equations are extended by adding one equation per measured variable. An equal number of unknown component health parameters such as efficiency and flow must also be passed to maintain a square matrix. This constraint is needed to calculate the inverse Jacobian matrix when executing the Newton-Raphson method.

The number of measured gas path variables differs per engine type. Newer engine architectures, such as the GEnx-1B and the LEAP-1A used by KLM, have fewer installed sensors along the gas path. Consequently, the system of equations in Adaptive Modelling becomes underdetermined when the parameters of all components have to be estimated as noted by Rootliep, Visser, and Nollet [20]. To solve this problem, the authors proposed a differential evolution optimisation scheme applied to adaptive modelling using Multiple Operating Point Analysis (MOPA). However, introducing an additional optimisation scheme during the synthetic data generation process may lead to convergence instabilities.

10

Prognostics & Health Management

The academic exploration of machinery health prognostics has gained increasing interests in recent years. It is an interdisciplinary research field that integrates sensor analytics and engineering practices to monitor and predict the health of a system. Prognostics and Health Management (PHM) aims at predicting the Remaining Useful Life (RUL) at which the system no longer performs its intended function. Using this information, companies managing and operating their assets are able to create effective maintenance strategies that ultimately lead to the reduction of operating costs and increased reliability.

10.1. History

Because of the safety-critical nature and high-maintenance cost, professionals within the aerospace industry pioneered early on with PHM. While various condition-based maintenance strategies have been in practice since as early as the 1940s, the specific terminology of PHM was first introduced in 2009 by the US Air Force within their F-35 Joint Strike Fighter program [21].

Over the last decade, PHM has been implemented in various fields such as Aerospace, Energy, Transportation, and Manufacturing, and is widely regarded as an essential technology for effective system maintenance and operational reliability [22]. Various technical societies were established to promote and facilitate collaboration within prognostics: PHM society (2009), Intelligent Maintenance Systems (IMS) center (2001), Center for advanced Life Cycle Engineering (CALCE)(1986), Prognostics Center of Excellence (PCoE)(2016), and Integrated Vehicle Health Management (IVHM) center (2008) [22]. The creation of these societies have catalysed research in failure physics, sensor technology, feature extraction,

10.2. Diagnostics & Prognostics

The words diagnosis and prognostics originate from the Greek language: *diagnōstikos* ”able to distinguish” and *prognōstikos* ”come to know beforehand”. Diagnostics focuses on identifying the root cause of a failure or abnormality in a system. This is generally based on the analysis of real-time sensor data, historical data, and expert knowledge about the system. Diagnostics dedicates itself towards detecting and isolating system faults and answering questions to why it is not performing optimally.

Prognostics pertains to the process of predicting the future performance of a system or its Remaining Useful Life (RUL) given its current state, past performance, and anticipated usage conditions. This process uses data-driven, model-based, or hybrid approaches to forecast the degradation of the system, thereby facilitating condition based maintenance and system optimisation. Prognostics is concerned with questions regarding the time at which a system will fail or how much longer it can perform its intended function.

Figure 10.1 presents a schematic showing where diagnostics and prognostics fit within the overall framework of PHM. The ability to predict a system's health relies on using models that can tell us about its degradation state and the Remaining Useful Life (RUL). Monitoring emphasises real-time fault detection, commonly using anomaly detection techniques. Diagnostics, meanwhile, identifies the nature of the fault and measures its impact on the system. Finally, the insights from these various components are integrated and analysed to create cost-effective strategies for maintenance and logistics.

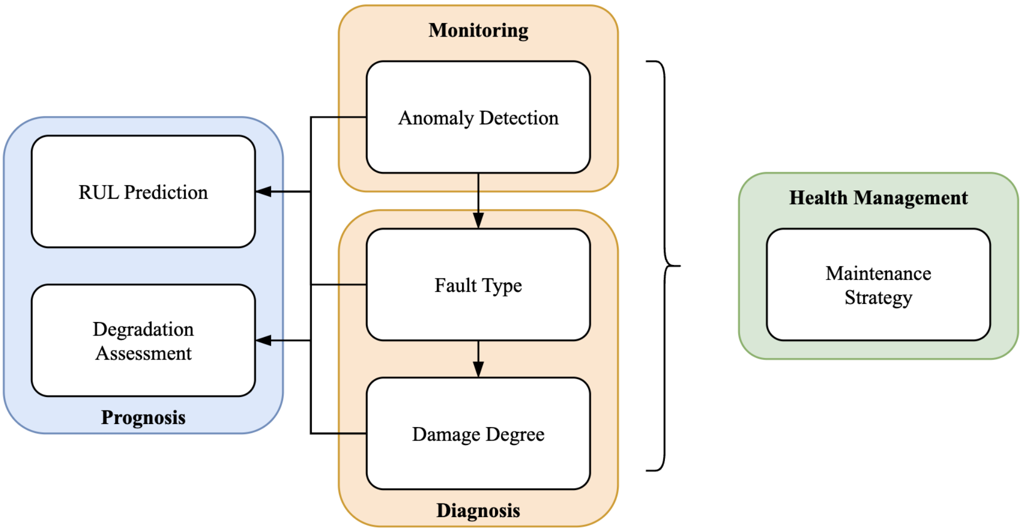


Figure 10.1: Overview of Prognostics & Health Management and its relationship towards other sub modules [14].

10.3. Health Indicators

The development of health indicators play a critical role in forecasting the future state of a system. Several health indicators have been proposed as key performance parameters that describe the health of an engine. Efficiency and flow are typical indicators of gas turbines. However, Hanachi et al. [23] suggest the use of heat loss and power deficit index for industrial gas turbines where internal gas path parameters are hard to access. In essence, multiple indicators can be used in RUL prediction which makes it important to select the most suitable indicator(s).

Lei et al. [5] propose two questions related to the development of health indicators: (a) In what way are effective health indicators derived from the measured data? (b) How is their effectiveness assessed? The authors classify health indicators into physics-based health indicators and so-called virtual indicators. Furthermore, De Giorgi, Menga, and Ficarella [7] present two techniques to extract health indicators from the data: a physics-based and data-driven approach.

10.3.1. Physics-Based Health Indicators

Physics-Based health indicators, customary to any machinery, represent the data with physical significance. These are often extracted from statistical approaches in the time and frequency domain. For example, in vibration signals, the root mean square (RMS) is used an a key performance parameter as it describes the system's energy level. In gas turbine applications, physics-based health indicators could include flow, efficiency, heat loss, and the power deficit index [23]. However, variables such as flow and efficiency are not directly measurable and have to be derived from physics-based models.

10.3.2. Virtual Health Indicators

Unlike physics-based health indicators, virtual health indicators do not inherently represent any physical meaning and are often made up of multiple physical health indicators. For example, the combination of the four physical health indicators of gas turbines mentioned earlier creates a new virtual health indicator. Common techniques focus on reducing feature space dimensionality. Principal component analysis (PCA) is a technique often used in data-driven methods. Newer deep learning methods can learn the underlying representation of the data during training and automatically construct the most important virtual indicators to predict the RUL as discussed in Section 10.4.

10.3.3. Health Indicators Evaluation

This section quantifies the utility and effectiveness of the physical and virtual health indicators. Lei et al. [5] discusses several methods to quantify their effectiveness: monotonicity, robustness, and trendability.

Monotonicity assesses the health indicator against the irreversible nature of degradation. It does so by counting the number of occurrences where the health indicator difference is either positive or negative.

$$\text{Monotonicty}(X) = \frac{1}{K-1} \cdot \mid \left(\frac{d}{dx} > 0\right)_{count} - \left(\frac{d}{dx} < 0\right)_{count} \mid$$

Health indicators should aim for smooth degradation trends and, therefore, robustness aims to quantify the noise of a health indicator. The robustness of the health indicator can be evaluated by comparing the the value x_k at time t_k with x_k^T which represents the value from a smoothing method.

$$\text{Robustness}(X) = \frac{1}{K} \sum_{k=1}^{K} \exp\left(-\left|\frac{x_k - x_k^T}{x_k}\right|\right)$$

Finally, trendability reflects a correlation with time since the system's degradation likelihood increases with operating time. Due to nonlinear degradation trends, researchers tend to use the Spearman's ranks coefficient to determine the trendability. Here, $\{\tilde{x}_k\}_{k=1:K}$ and $\{\tilde{t}_k\}_{k=1:K}$ represent the health indicator's rank sequence.

$$\text{Trendability}(\tilde{X}, \tilde{T}) = \frac{K\left(\sum_{k=1}^{K} \tilde{x}_k \tilde{t}_k\right) - \left(\sum_{k=1}^{K} \tilde{x}_k\right)\left(\sum_{k=1}^{K} \tilde{t}_k\right)}{\sqrt{\left[K\sum_{k=1}^{K} \tilde{x}_k^2 - \left(\sum_{k=1}^{K} \tilde{x}_k\right)^2\right]\left[K\sum_{k=1}^{K} \tilde{t}_k^2 - \left(\sum_{k=1}^{K} \tilde{t}_k\right)^2\right]}}$$

10.4. Predicting Remaining Useful Life (RUL)

Researchers and professionals define the RUL of machinery either as the remaining time left until the machine stops performing its intended function or until its health state drops below a predefined failure threshold [5]. The industry benefits from estimating the RUL by avoiding downtime, costs, improve operations, and preventing system failures. The two key issues in RUL prediction revolve around the method used to estimate the RUL from monitored sensor data and how to evaluate its accuracy.

Figure 10.2 presents the distribution of publications that follow one of the four RUL prediction methods: statistical models, hybrid approaches, physics-based models, and artificial intelligence [5]. With the advent of artificial intelligence methods this distribution has most likely changed. In the next sections, several common techniques are described.

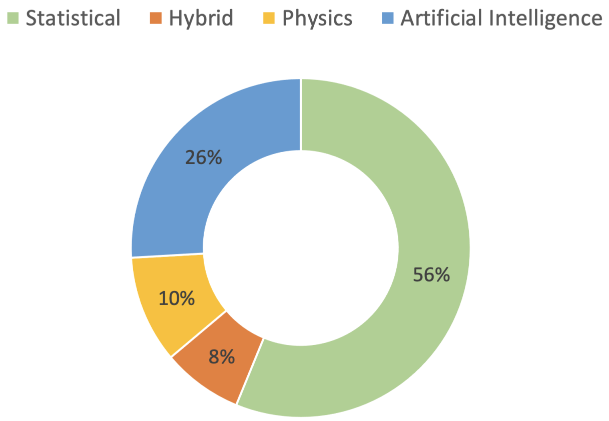


Figure 10.2: Methodology distribution from several publications in estimating RUL up to 2015 [5].

10.4.1. Physics-based Models

A physics-based model seeks to derive the health parameters from mathematical and thermodynamical principles [7]. These focus on describing the evolution of deterioration through mathematical damage propagation models

Physics-Based health indicators represent the data with physical significance [5]. For example, in vibration signals, the root mean square (RMS) is used an a key performance parameter as it describes the system's energy level. For gas turbines, physics-based health indicators could include flow, efficiency, heat loss, and the power deficit index. However, variables such as flow and efficiency are not directly measurable and have to be derived from physics-based models.

Gas Path Analysis (GPA) is a common tool for gas turbine applications and is also used at KLM. Its objective is to quantify health indicators using derived variables from the measured sensor data. By understanding the physics behind the engine its health indicators can be related to the recorded data at different stations along the gas path using Adaptive Modelling (AM), as discussed in Section 9.3.2.

10.4.2. Statistical Models

Autoregressive (AR) and Markov models are two examples of statistical models that have been utilised for RUL prediction. AR models fundamentally rely on the assumption that the future state of a system depends linearly on historic data and errors. These models are simple but may lead to large forecasting errors as they depend on trends and history.

Markov models are based on the assumption that a system changes its state among a finite set of states. This property asserts that the future state of the system depends only on its current state. These models are often applied to the health states which are often observable [5]. Another notable limitation of Markov models is their difficulty in capturing complex scenarios, as the number of potential states can increase exponentially, leading to substantial computational overhead [24].

10.4.3. Data-Driven Models

Data-Driven techniques to predict RUL span from Bayesian belief networks and artificial neural networks to traditional machine learning techniques such as Support Vector Machines. Since the release of the C-MAPSS dataset for the 2008 PHM challenge [8], numerous researchers have utilised data-driven methods to forecast RUL. Amongst the winners of the more recent 2021 PHM Data Challenge, deep learning approaches were the most popular [7]. The deep learning trend was also observed by Vollert and Theissler [6], especially in recurrent neural networks (RNNs).

RNNs excel at learning sequential dependencies intrinsic to data [25]. Given the time-series nature of turbofan sensor data, they are well-suited for this task. Long-Short Term Memory (LSTM) architectures are particularly beneficial, thanks to their utilisation of the hidden memory state concept. The strength of LSTMs lies in their capability to remember long-term dependencies by mitigating the vanishing gradient problem often encountered in traditional RNNs. This makes them very proficient in handling complex time-series prediction tasks.

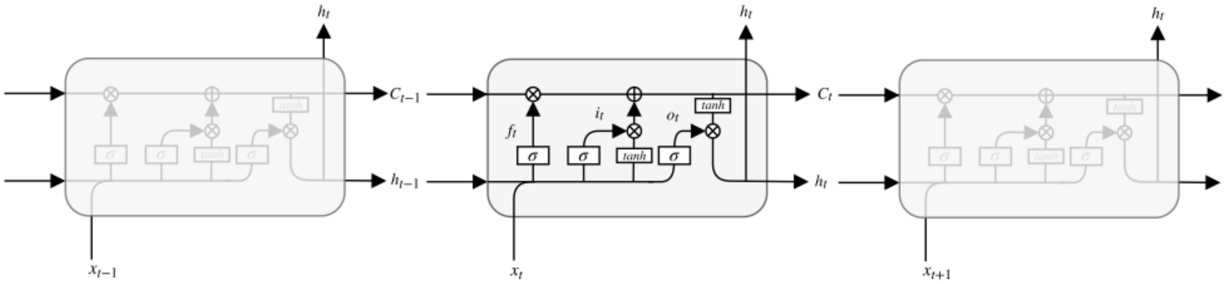


Figure 10.3: Long Short-term Memory (LSTM) cell depicting the cell and hidden state [26]. The variable x_t represents a vector of n features at time step t.

The flow of information is summarised in Figure 10.3 where the LSTM cell is governed by the three non-linear gating units in Equation 10.1: the forget gate, f_t, the input gate i_t, and the output gate, o_t. The matrices and biases corresponding to these gates serve as the trainable parameters of the neural network. The candidate for the new cell state, denoted as $\widetilde{C}_t$, is computed via the formula $\widetilde{C}_t = \tanh\left(W_C x_t^i + R_C h_{t-1} + b_C\right)$. This candidate cell state is then combined with the previous cell state to determine the new cell state, $C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t$. Finally, the new hidden state h_t is determined by the output gate, which effectively decides which components of the cell state should be retained for future operations.

$$\begin{aligned} f_t &= \sigma\left(W_f x_t^i + R_f h_{t-1} + b_f\right) \\ i_t &= \sigma\left(W_i x_t^i + R_i h_{t-1} + b_i\right) \\ o_t &= \sigma\left(W_o x_t^i + R_o h_{t-1} + b_o\right) \end{aligned} \tag{10.1}$$

The architecture of the full LSTM model may consists of several layers and different neurons. An example of a general architecture is illustrated in Figure 10.4. These can further be enhanced by using bidirectional layers [27], adding attention mechanisms to the output of the LSTM cells [26], or introducing a convolutional layer in the architecture [28]. In addition to adjusting the LSTM architecture, other research propose integrating physics-based simulators in the training process [29]. However, despite the demonstrated effectiveness of RNNs in various applications, their utility in safety-critical engineering systems is often hampered by their significant training data requirements. The scarcity of run-to-failure data, a consequence of systems being routinely replaced before failure, makes training challenging.

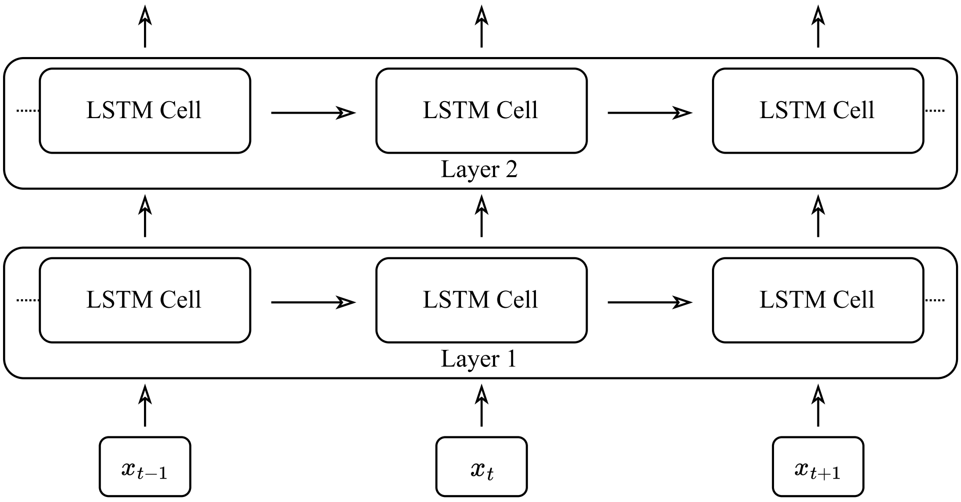


Figure 10.4: Example of an LSTM neural network architecture with two layers [26].

10.5. Evaluation

Evaluating the performance of different models involves assessing whether the requirements of the prognosis are met. In safety-critical systems, a late failure prediction could compromise safety and impose considerable economic strain on the company. Therefore, it is essential to establish evaluations reflecting these factors. This section aims to discuss the evaluation of both predictive models and health parameters.

10.5.1. Model Evaluation

A simple method of establishing a model's performance is the root mean squared error (RMSE). Equation 10.2 defines as the root mean square of RUL predictions starting from the first predicted time until the end of life (EoL).

$$RMSE = \sqrt{\frac{1}{EOL - FPT} \sum_{k=FPT}^{EoL} \left(l_{t_k} - l_{t_k}^*\right)^2} \tag{10.2}$$

In contrast to the RMSE, Saxena et al. [8] proposes an asymmetric scoring function, Equation 10.3, to evaluate model performance. Predicting the estimated RUL before the true RUL is preferred to preserve safety whereas predicting RUL after the system fails imposes a hazard. Equation 10.3 shows how the scoring function depends when deviating from the true value with $d = \bar{t}_{RUL} - t_{RUL}$. The corresponding values are plotted in Figure 10.5.

$$penalty = \begin{cases} \sum_{i=1}^{n} e^{-\left(\frac{d}{a_1}\right)} - 1 \text{ for } d < 0 \\ \sum_{i=1}^{n} e^{\left(\frac{d}{a_2}\right)} - 1 \text{ for } d \geq 0 \end{cases} \tag{10.3}$$

In addition to the symmetric scoring function, Saxena et al. [8] expands on the idea that predicting on a larger time horizon is more difficult compared to a smaller window. In contrast, if the prediction is made while the true RUL is closer to the operational system, the accuracy of the forecast is more valuable. This idea introduces another asymmetric dimension when evaluating models and improves the comparison of different models.

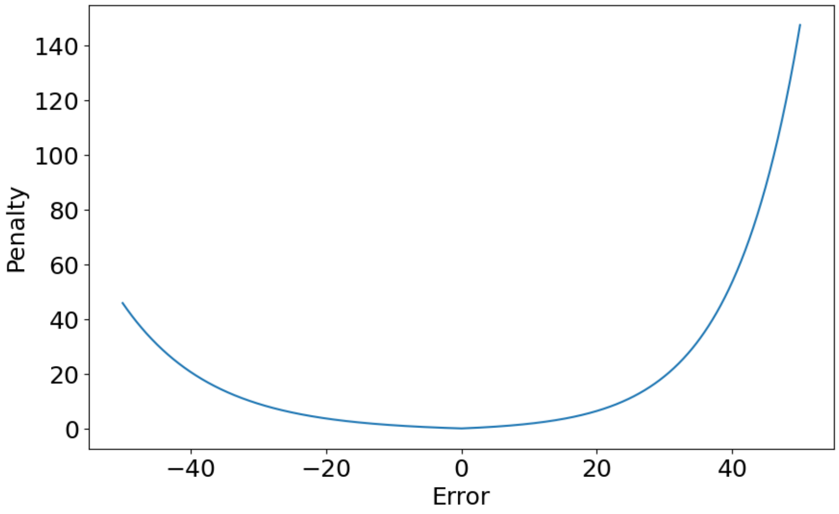


Figure 10.5: Asymmetric scoring function (Equation 10.3) proposed by Saxena et al. [8]. Predicting RUL after the failure has happened should have a more severe penalty to maintain reliability of safety-critical systems.

11

Turbofan Data Augmentation

Data augmentation techniques hold significant value as they enable the artificial expansion of a dataset. This expansion is achieved through the application of subtle transformations to the original data or by learning the underlying distribution to generate entirely new, yet representative samples. In turn, the increased diversity within the training set reduces overfitting and contributes to more robust machine learning models, particularly within deep learning, where large quantities of data are often essential for training [30].

A major issue in the development of prognostics models within the aviation industry is the lack of turbofan run-to-failure data. To address this challenge, researchers have proposed various strategies. These approaches span from employing basic data augmentation techniques to more complex methods, such as synthetically simulating turbofan degradation cycles using Gas Path Analysis (GPA) models [31][28][8]. Regardless of the specific technique, the common goal of these methods is to create new synthetic data. While the synthetic data are intentionally designed to be different from the original data, it must continue to preserve the fundamental characteristics of the data and produce artificial patterns that can be learned and are relevant.

11.1. Time Series Data Augmentation

Traditional statistical methods often rely on the assumption that observations are independent and identically distributed. However, this assumption does not hold when data is recorded at consecutive points in time [32]. For this reason, augmenting existing run-to-failure trajectories requires understanding of the nature of time series data.

Within the space of time series data augmentation, different techniques exists ranging from basic approaches to more advanced methods. Maintaining temporal trends is crucial in time series analysis and, consequently, not all augmentation techniques are applicable. In Figure 11.1, Wen et al. [33] propose a taxonomy of 6 groups dedicated towards data augmentation in time series related tasks: time series transformations, frequency based augmentation, a hybrid combination of time and frequency, decomposition of trends, statistical methods, and learning methods.

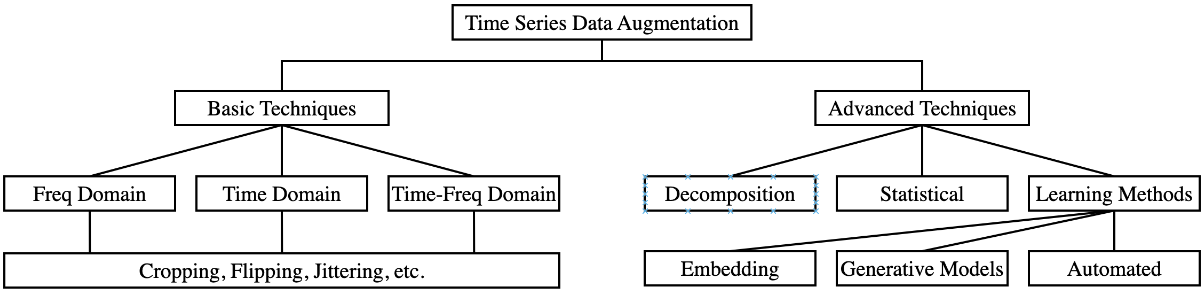


Figure 11.1: Overview of time series data augmentation techniques proposed by Wen et al. [33].

Basic techniques apply transformations to the data directly. They involve slicing, jittering, scaling, rotation, and permutation. Figure 11.2 presents a few examples of traditional augmentation methods in the time domain. These techniques generally apply minor transformations to the data. Moreover, the methods can be combined and super-imposed to produce additional variations of the original data. In contrast, advanced techniques learn the probability distribution of the training data to create new synthetic data [34]. These are discussed in Section 11.3.

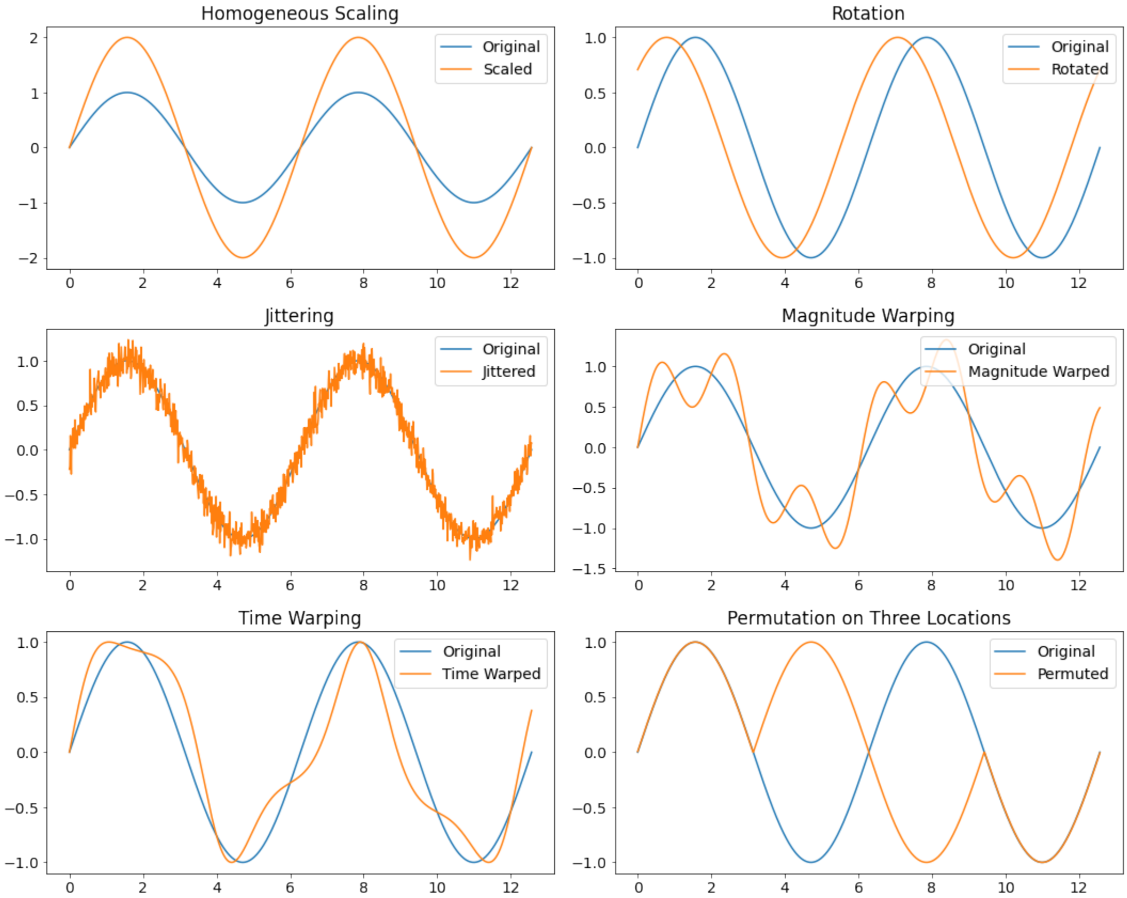


Figure 11.2: Examples of basic time series data augmentation techniques [34].

The study by Gay et al. [31] assessed various standard data augmentation techniques using the C-MAPSS dataset. Their results indicate that methods like time warping, time slicing, and interpolation positively impacted the Root Mean Square Error (RMSE). In contrast, magnitude warping was found to adversely affect RMSE in the resulting prognostic models. Besides using the simulated C-MAPPS dataset, the authors also demonstrated that these elementary augmentation methods can substantially increase the prediction confidence in industrial applications.

11.2. Simulating Turbofan Degradation

To address the lack of data, several researchers have aimed to generate synthetic run-to-failure data by using GPA tools to model turbofan degradation cycles [8][9]. The degradation trends are produced synthetically by imposing mathematical damage propagation models which often originate from literature. Modelling the dynamics of damage propagation between the cycles presents an additional challenge on top of developing a representative turbofan model that can accommodate health indicators. Both Saxena et al. [8] and Arias Chao et al. [9] have contributed significantly to the PHM community by providing synthetic turbofan run-to-failure datasets for the RUL prediction PHM data challenges. Their work has opened new pathways for ongoing research and is discussed in the following two sections.

11.2.1. C-MAPSS

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), developed by NASA, serves as a simulation tool for modern turbofan engines and operates within MATLAB and Simulink [35]. It is similar to GSP, allowing users to manipulate input parameters and model specific conditions.

In 2008, Saxena et al. [8] employed C-MAPSS as the engine model, integrating a damage propagation model to simulate the trajectories of various engine units. This damage propagation model, derived from Goebel et al. [36], assumes exponential evolution of faults. The equation for wear, denoted by $w = Ae^{B(t)}$, encompasses overall degradation characteristics. The health indicators, efficiency e and flow f, are modelled as:

$$h(t) = 1 - d - e^{a(t) \cdot t^{b(t)}} \tag{11.1}$$

Here, d symbolises initial wear that may arise from manufacturing issues. The overall health index is expressed as $H(t) = g(e(t), f(t))$. The complete data generation process with C-MAPSS is described below:

1. The health parameter's initial condition for deterioration is selected.
2. The degradation model is imposed to each health indicator of the HPC where the change in f and e are constrained by 1%, $a \in [0.001, 0.003]$ and $b \in [1.4, 1.6]$.
3. The C-MAPSS simulation is executed until the overall health indicator reaches zero.
4. Additional measurement noise is added to the generated data to simulate sensor noise.

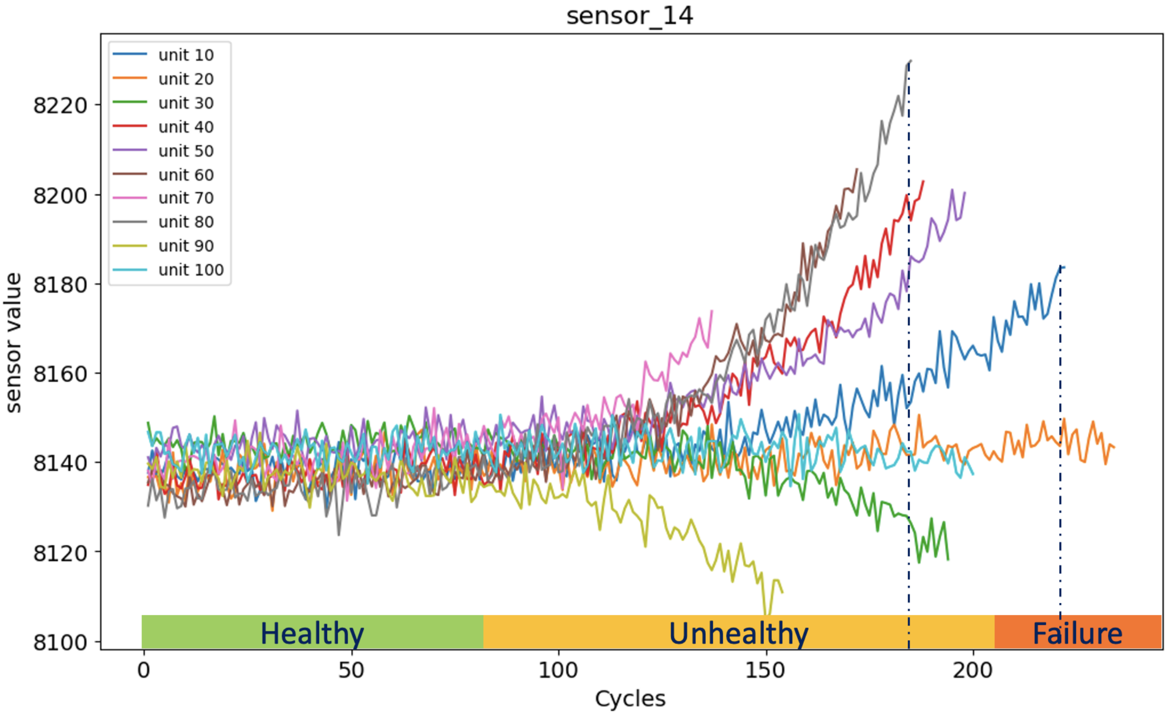


Figure 11.3: An example of a generated run-to-failure trajectory for one sensor from Saxena et al. [8]. The deviating trends become more prominent as the turbofan enters the unhealthy stage. The stage names are indications on how the data points can be divided over the cycles.

Figure 11.3 shows an example of the evolution of deterioration for several engine units where sensor 14 represents any arbitrary sensor that monitors a specific variable along the gas path. The figure shows how the units follow different trajectories and may fail at different moments in time.

11.2.2. N-CMAPSS

Saxena et al. [8] did not implement real operating and ambient data from real. Besides imposing initial deterioration and implementing randomness within the damage propagation model, the following aspects to why a specific engines may experience unique degradation trends should also be considered: (1) is the engine from the same type? (2) how far did the engine fly? (3) what were the weather conditions during take-off and cruise? (4) which aircraft body was the engine mounted on? (5) what was the maximum take-off weight of the aircraft? (6) how many times has the engine been removed from the aircraft?

In 2021, Arias Chao et al. [9] contributed to new improvements to the original C-MAPSS dataset by not only introducing operational and ambient data from real flights, but also increasing the complexity of the degradation model, linking it to historic data. When considering the C-MAPSS system model, the inputs are categorised by operating conditions, w, and health parameters, θ. Similar to GSP, the C-MAPSS equation is summarised as

$$\left[x_s^{(t)}, x_v^{(t)}\right] = F\left(w^{(t)}, \theta^{(t)}\right) \tag{11.2}$$

Unlike Saxena et al. [8], the N-CMAPSS is generated using real sensor data, recorded during multiple flights. The NASA DASHlink-Flight Data For Tail 687 is used as input for the operating conditions w and results in more realistic generated samples [37].

The authors propose to model the overall degradation with an initial degradation, a normal degradation, and an abnormal degradation. Furthermore, the degradation model is not only applied to the HPC, but also any rotating sub-component such as fan, LPC, HPT, and LPT leading to more complex degradation patterns.

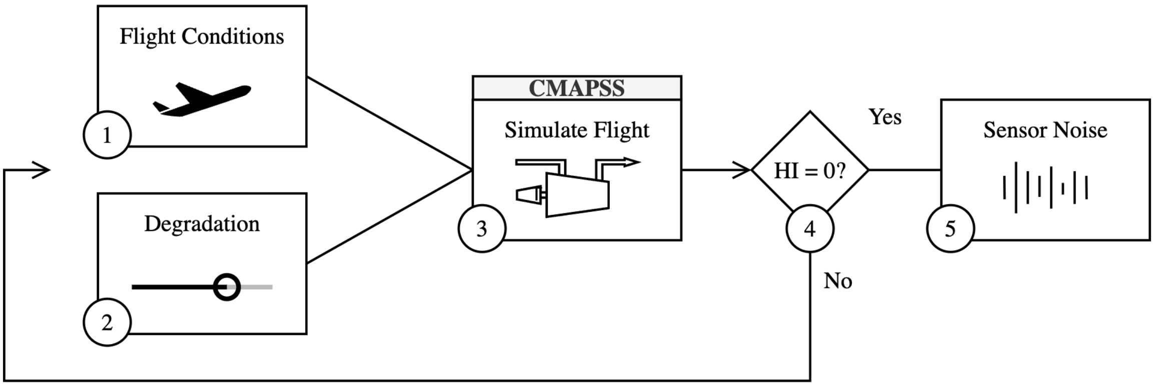


Figure 11.4: The data generation process for the N-CMAPSS dataset as proposed by Arias Chao et al. [9].

A step-by-step summary of the data generation process for the N-CMAPSS dataset is shown in Figure 11.4 with its corresponding description below:

1. The real flight conditions are used as input for the C-MAPSS simulation.
2. The degradation of the different engine components are imposed after each flight.
3. The engine performance is simulated throughout each flight with C-MAPSS.
4. The first three steps are repeated until the engine reaches the failure criterion: $H(t) = 0$
5. In addition to the measurement noise from the real flight data, extra noise is added to the generated data.

Although the presented datasets have enabled numerous advances in prognostics and aircraft engine health management, the question remains whether this data generation method provides enough realistic data for prognostics models to be generalisable when deployed in the real world. The synthetic datasets are subjected to the assumptions of the 0D models and the assumed mathematical degradation models which may not always capture the reality of industrial processes [31].

11.3. Generative Models

Section 11.1 provided an overview of the different time series data augmentation techniques where they were divided based on their complexity. This section aims to discuss more advanced techniques of augmenting time series data. In recent years, generative models have experienced significant progress. They operate by learning to match the model's parameters to generate samples that are indistinguishable from the original distribution [38].

Variational Autoencoders

Variational Autoencoders (VAEs) create compressed representations of the input data from which new data is sampled. The lower dimensional encoded space, or latent space, is learned by comparing the reconstructed input values with the original samples shown in Figure 11.5. VAEs were first proposed by Kingma and Welling [39].

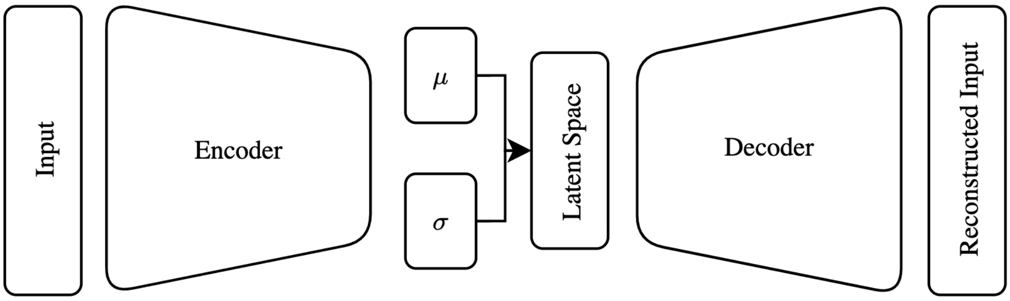


Figure 11.5: Schematic of the variational autoencoder architecture.

The two main components, the encoder and the decoder, aim to learn the reduced feature space using self-supervised learning [34]. Unlike the original autoencoders, VAEs learn from a probability distribution, μ and σ, to avoid overfitting. Upon producing new examples, the VAEs samples synthetic data from this distribution.

Generative Adversarial Networks

The fundamental learning process of GANs are based on an adversarial approach. Two neural networks, the generator and the discriminator, compete against each other. The generator tries to fool the discriminator by forging realistic samples. On the other hand, the discriminator seeks to differentiate between real and synthetic samples. Over time, the generator improves its ability to create realistic data, while the discriminator enhances its ability to distinguish between real and fake data. This data augmentation technique is further discussed in Chapter 12.

11.4. Advantages and Disadvantages

This section evaluates the data augmentation techniques covered in prior sections. Table 11.1 presents a scoring metric ranging from 1 to 5 for each technique, based on criteria that include ease of implementation, theoretical effectiveness, scientific relevance, and computational resource demands. These criteria are weighted at 30%, 20%, 35%, and 15%, respectively. While the scientific relevance of each method is important for the PHM community, the practicality of the method is also crucial. Furthermore, theoretical effectiveness is evaluated to include the prospective impact of each method. Lastly, considering the constraints often imposed by available computational resources, each technique is also scored on this criterion.

The process of generating synthetic data with physics-based models comes with certain limitations, as the data is subject to simplifications in turbofan simulation and degradation assumptions. Consequently, machine learning models trained on this data may not perform adequately when exposed to real flight data [10][11]. However, comparing the output of physics-based models with real world data can strengthen the understanding of gas turbine deterioration.

Table 11.1: Data augmentation techniques trade-off

	Ease (30%)	**Effect (20%)**	**Relevance (35%)**	**Resources (15%)**	
Basic	5	1	1	5	**2.80**
VAEs	3	3	2	3	**2.65**
GANs	1	5	5	2	**3.35**
Simulation	2	2	3	3	**2.50**

Instead, KLM could leverage the limited amount of data to learn deteriorated turbofan performance and enhance model predictions during operational use. Basic data augmentation techniques are relatively straightforward to implement, supported by robust research, and do not require model training, thereby reducing both time and complexity. However, techniques such as slicing, jittering and other transformations might risk producing lower quality, or even invalid, examples when modifying specific data points from the original datasets [34].

VAEs offer greater control over parameters and flexibility in diversifying the original dataset. However, compared to GANs they generate less data due to the nature of the algorithm. While GANs have demonstrated impressive results in different industries and areas of research, they are inherently challenging to train due to mode collapse, instability, and convergence evaluation. In addition, evaluating the performance of GANs, particularly in quantitative metrics, remains an active area of research [40]. The next chapter, Chapter 12, focuses on these issues and discusses the possible GAN architectures in time series related tasks.

12

Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning. Goodfellow et al. [12] introduced the concept of adversarial training by letting two neural networks compete against each other: a generator and a discriminator. The generator produces artificial examples, while the discriminator evaluates them for their realism. The generator improves through the adversarial process, aiming to generate data that the discriminator cannot differentiate from real instances. GANs have found extensive applications in image synthesis, semantic image editing, style transfer, image super-resolution and classification.

12.1. Adversarial Training

The process of training happens in two steps where the discriminator starts with learning the distribution of real and fake data. For the discriminator, the objective is to correctly classify the real instances from the training set and the generated instances from the generator. The loss function for the discriminator is typically formulated as a binary cross-entropy loss, which is minimised when the discriminator accurately distinguishes real instances from fake ones. During updating the parameters of the discriminator the generator's parameters are held constant as shown in Figure 12.1.

The discriminator seeks to correctly classify real instances from the training set and generated instances from the generator.

The generator seeks to fool the discriminator into believing that its generated instances are real.

The second step involves in updating the generator's parameters. The objective of this step is to update the generator's parameters to create even more realistic examples using the information from the discriminator. To achieve this, we use the discriminator's predictions on the generator's outputs but with the labels flipped as if they were real instances. The reason for this lies in back propagation. When we pass the generator's fake instances through the discriminator and label them as real, the resulting gradient will indicate how to change the generator's parameters to make its outputs more like real instances. After passing a batch, the generator's parameters are updated in isolation as shown in Figure 12.2.

At core, the losses of the generator and discriminator are at odds with each other which results in the adversarial part of training. In terms of mathematical formulation, the generator and discriminator try to both minimise and maximise the following the objective function:

$$\min_{G} \max \ DV(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}\ (\boldsymbol{x})}[\log\ D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log\ (1 - D(G(\boldsymbol{z})))](12.1)$$

12.1.1. Mode Collapse

Modes represent areas of data concentration within a space, and a dataset can exhibit multiple such modes. Consider, for instance, a dataset of handwritten digits ranging from 1-10. Here, the discriminator would aim to categorise each handwritten digit into one of the 10 classes. During the training phase, if the discriminator struggles to distinguish between ones and sevens, this information is conveyed back to the generator. Consequently, the generator improves its capability to create more convincing examples of these

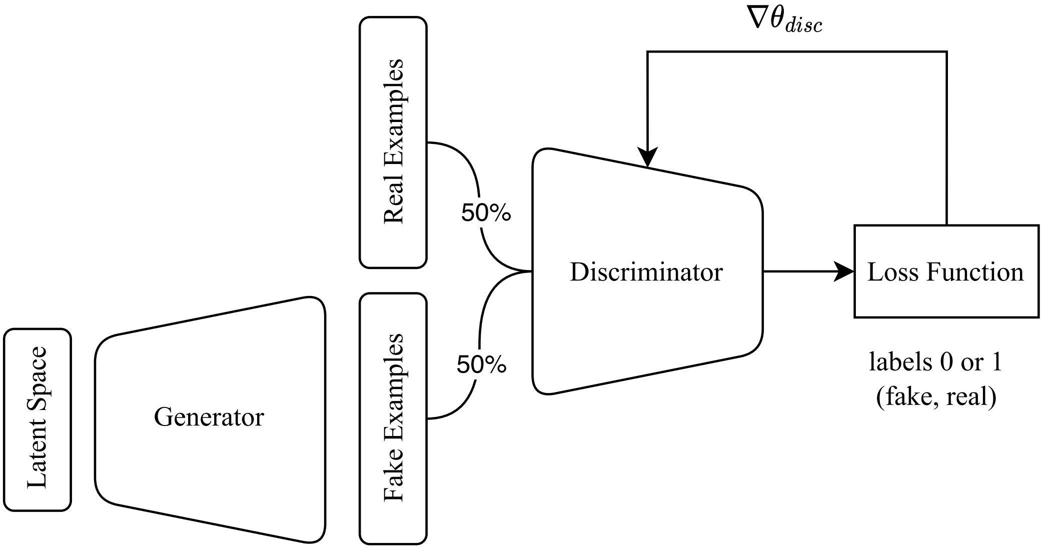


Figure 12.1: The discriminator starts first with learning the distribution of the fake and the real data. It does so by computing the loss and updating its parameters with $\nabla\theta_{disc}$ to better classify real and fake examples.

specific classes. However, this selective enhancement may lead to a phenomenon called 'mode collapse', where the generator produces only a certain subset of data. When relating this issue to the generation of turbofan life cycle data, it would mean that the generator starts producing examples of one failure mode specifically. Consequently, the generator is locked and generates examples of a specific type predominantly, thereby not representing the entirety of the training set.

12.1.2. Vanishing Gradients

In the original GAN paper Goodfellow et al. [12], the authors used the binary cross entropy (BCE) loss function to run the optimisation. The problem with BCE loss is the declined effect of gradients as the discriminator starts becoming better. Consequently, the information used by the generator to update its parameters is more weak and less effective because of very different distributions. In other words, the generator fails at improving its generated examples.

To address this issue, a Wasserstein metric based on Earth Mover's Distance (EMD) was introduced by Arjovsky, Chintala, and Bottou [41]. This metric quantifies the effort required to transform the fake data distribution to match the real data distribution. Unlike traditional discriminators that classify data as either 0 or 1, the implementation here involves a critic that assigns an arbitrary positive number. For effective training, it was proposed that the critic be Lipschitz continuous with a constraint of 1, implying that the gradients should not exceed this limit. This was initially achieved through gradient clipping. However, a subsequent enhancement proposed by Gulrajani et al. [42] replaced gradient clipping with a regularisation parameter that penalises gradients based on their magnitude, thereby enforcing Lipschitz continuity more effectively.

12.2. Recurrent Generative Adversarial Networks

The understanding of time series data, discussed in Section 11.1, has led various researchers to train Generative Adversarial Networks with Recurrent Neural Networks (RNNs). RNNs, particularly long short-term memory (LSTMs) networks, are able to capture inherent temporal dependencies which make them a popular architecture in time series applications [25].

Mogren [43] was among the first researchers to develop a Recurrent Generative Adversarial Networks (RGAN) to generate classical classical music as shown in Figure 12.3. In their model, both the generator and discriminator are LSTM networks to suppress the vanishing gradient problem. During training the authors often noticed the discriminator overpowering the generator which is why they froze updates to the

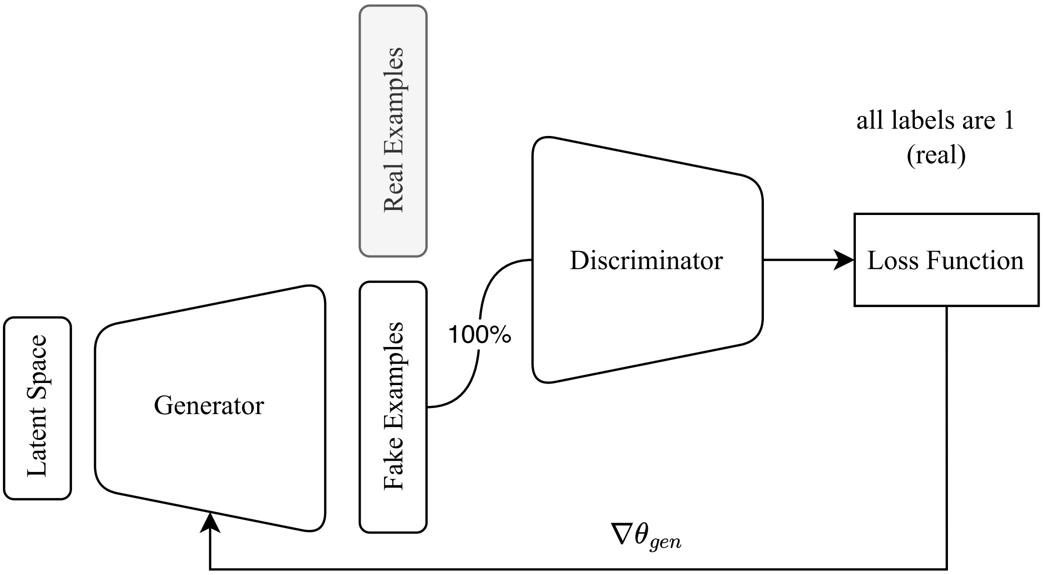


Figure 12.2: The generator updates its weights with $\nabla\theta_{gen}$ after the discriminator has been updated. In this step only fake examples are passed to the discriminator and are labelled as real.

discriminator's weights for multiple batches until the generator caught up. The authors also implemented other techniques such as feature matching to reduce overfitting the discriminator.

In their study, Esteban, Hyland, and Rätsch [44] developed an RGAN to generate multidimensional medical time-series data. Medical machine learning often confronts challenges due to the relatively small datasets available, a consequence of the sensitivity surrounding such data, making it difficult to train deep learning models. Their work presents various methodologies to evaluate RGAN output, including a method termed "Train on Synthetic, Test on Real" (TSTR), highlighting the potential for using synthetic data in real-world applications. However, they acknowledged difficulties in implementing Wasserstein GANs due to the insufficient research on enforcing the Lipschitz constraint on RNNs at the time. The subsequent introduction of gradient penalty has since mitigated this issue.

With the advent of new sophisticated transformer based techniques, Li et al. [45] introduced the Transformer-Based Time-Series GAN (TTS-GAN) in 2022. The difficulty faced by RNN-based GANs in modelling long sequences and parallel training led to the advent of transformers, which exclusively use the attention mechanism, as detailed by Vaswani et al. [46]. A notable technique in this context involves processing time-series data similar to images, a method also acknowledged in the recent literature review by Brophy et al. [40].

Esteban, Hyland, and Rätsch [44] also demonstrated the utility of RGAN by conditioning its output to enable the generation of labeled medical time series data. The architecture of a Recurrent Conditional GAN (RCGAN) is similar to the one proposed in Figure 12.3 by Mogren [43]. However, slight modifications are added to the input and training to include labeled data as seen in Figure 12.4. Similarly, Arnelid, Zec, and Mohammadiha [47] utilise the RCGAN to generate synthetic sensor data for autonomous vehicles. In this paper, the data for the output, input, and latent space is represented by $\boldsymbol{x}_t \in \mathbb{R}^{T_t \times k}$, $\boldsymbol{y}_t \in \mathbb{R}^{T_t \times \ell}$, and $\boldsymbol{z}_t \in \mathbb{R}^{T_t \times m}$. Here, the conditional input is related to the raw sensor data captured by the vehicle which is used to direct the GAN.

Other researchers attempt at improving the RCGAN training process by including the Wasserstein loss function. As mentioned in Section 12.1.2, Wasserstein GAN can improve training stability of a GAN and limit mode collapse. Qiang et al. [48] implement the Wasserstein distance metric in their Recurrent GAN to learn brain representations from the fMRI data. They include the gradient penalty term, $\lambda \mathrm{E}_{x\sim\Omega} \left[|\nabla D(x)|_p - 1 \right]^2$, in the discriminator's loss function to enforce the 1-Lipschitz constraint.

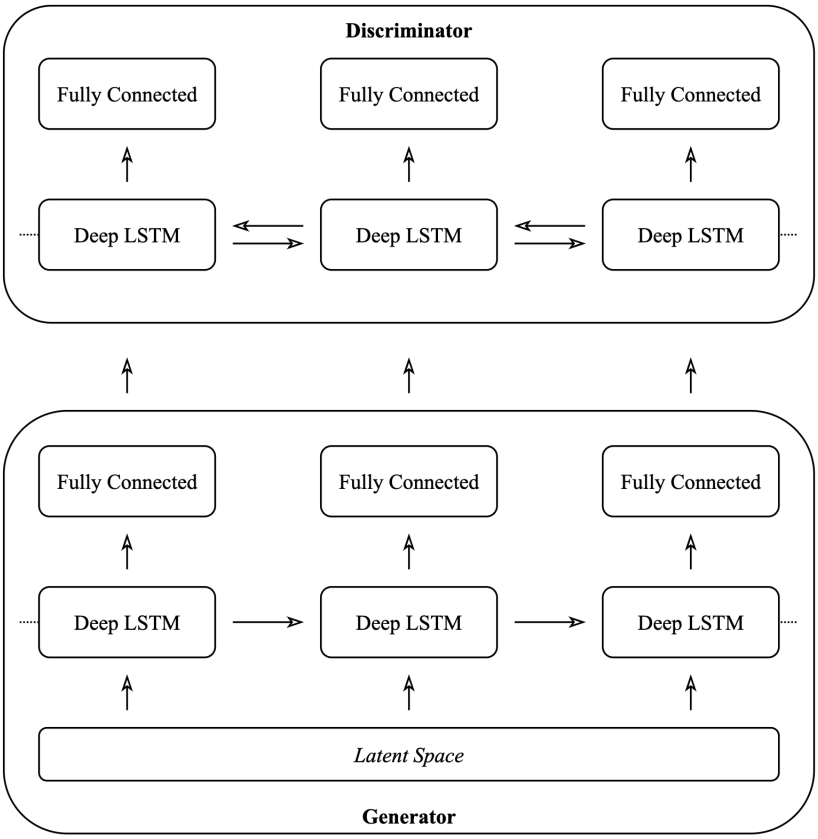


Figure 12.3: The Continuous Recurrent Generative Adversarial Network (C-RNN-GAN) was developed by Mogren [43] for classical music generation. Both Generator and Discriminator are LSTM based neural networks competing in an adversarial setting.

12.3. Augmenting the C-MAPSS Dataset with GAN

The original C-MAPSS dataset was created with the objective to stimulate research in Prognostics and Health Management by modelling the degradation cycles in aircraft engines [8]. Much of the research focuses on RUL prediction as explained in Section 10.4. In practice, deep learning models require vast amounts of real data for training which is often unavailable in the industry.

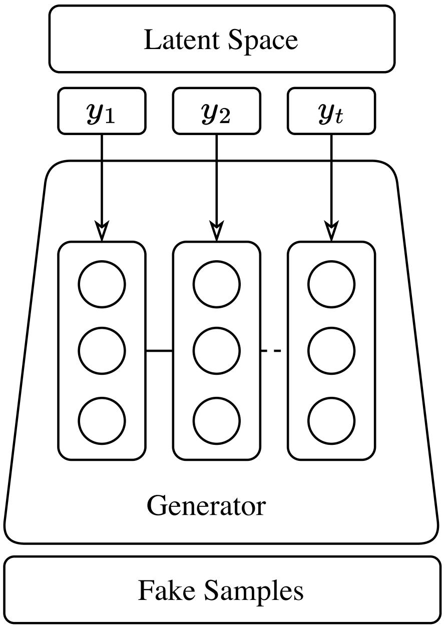
12.3.1. Time Window

Lang et al. [28] proposed augmenting the C-MAPSS dataset with a GAN leading to a greater training dataset and higher predictive performance on the test set. Considering that the C-MAPSS dataset is based on time series data, the authors used a time window, presented in Figure 12.5, to train the GAN and the subsequent RUL prediction model. As a result, the input shape of the model is (n_windows, window_length, n_sensors) where the number of samples is determined by the stride and the window length.

The authors solely verified their implementation of the GAN be contesting different models with their proposed model. The prediction model would only be trained on the real data, but later also with the real and generated data to observe any differences in prediction model. They do not show any other evaluation techniques or compare other data augmentation techniques, such as more simpler methods like introducing noise, with the GAN approach.

12.3.2. Variable Length Time Series

In the 2021 PHM data challenge, Lövberg [49] used CNNs with variable length window sizes to predict the RUL on the N-CMAPSS dataset. They argued that a fixed window size may lead to the loss of valuable context, as the smallest window becomes the limiting factor for longer predictions. To address the challenge



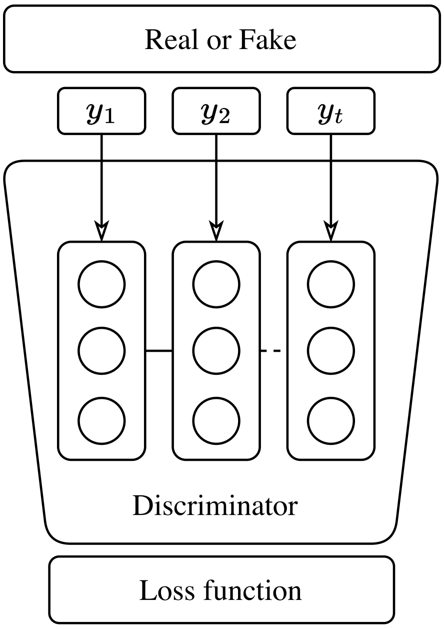


Figure 12.4: The Recurrent Conditional Generative Adversarial Network (RCGAN) proposed by Esteban, Hyland, and Rätsch [44]. At each time step the condition, y_t, is used as an input in the generator and discriminator.

of varying time-series lengths, is to apply padding to the training data. However, this approach becomes infeasible when dealing high variety in length as the padding impacts the distribution of generated samples significantly [50]. Consequently, Demetriou et al. [50] condition the GAN to control the output length of the trajectory.

12.4. Physics-Informed Machine Learning

In addition to requiring extensive datasets, machine learning models generally lack understanding of scientific principles and often are inconsistent with established physical laws [51]. These data-driven approaches also present challenges in terms of interpretability, robustness, and alignment with physical constraints [52]. In contrast, exclusively relying on physics-based models, which attempt to approximate reality, carries the risk of oversimplifying complex processes. The interplay between these modelling paradigms is depicted in Figure 12.6.

Hybrid models are subjected to different methodologies of integrating physics in machine learning models: (1) Physics-Guided Loss Function, (2) physics-guided initialisation, (3) physics-guided architecture design, (4) residual modelling, and (5) hybrid models. However, after compiling the different methods and their corresponding applications, Willard et al. [13] found that only the architectures and loss functions have been adjusted for the application of data generation. Thus, a hybrid combination of a generative model and turbofan simulation model may be a potential gap in this field.

12.4.1. Loss Function

Leveraging prior physical knowledge of the target data could mitigate the issue of high sample complexity in GANs. By constraining the loss function with physical insights, the GAN can generate more realistic samples. Data-driven models often struggle to learn the interrelation between variables, particularly when confronted with data scarcity. Lack of run-to-failure data, as described in Chapter 10, is a common issue in aerospace related applications. The loss function would be updated as follows [13]:

$$\text{Loss} = \text{Loss}_{\text{ML}}\left(Y_{\text{true}}, Y_{\text{pred}}\right) + \gamma\,\text{Loss}_{\text{phy}}\left(Y_{\text{pred}}\right) \tag{12.2}$$

Willard et al. [13] mentions three benefits of physically constraining the loss function.

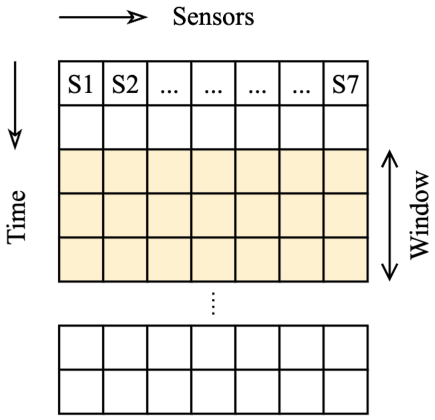


Figure 12.5: Windowing technique used by [28]. Each window with a specified window length generates a new sample.

1. Extra labelled observation data is not required.
2. Limiting the search space ultimately leads to less training data required.
3. Physics informed machine learning models tend to be more generalisable.

Xiong et al. [53] introduced a physics-informed loss function designed for N-CMAPSS data augmentation that is based on the monotonic degradation of engine health parameters. Assuming that engines are unable to self-repair, the loss function penalises trajectories that contradict the expected decline in health indicators. Additional penalty terms, such as trendability and robustness (Section 10.3.3), could also be included.

12.4.2. Architecture Design

Apart from modifying the loss functions, recent research suggests designing novel machine learning structures that facilitate the integration of domain knowledge at specific neural network nodes [13]. Employing intermediate physical variables has found usage in diverse applications, such as lake temperature modelling [51]. Moreover, this approach enhances interpretability, addressing a common shortcoming of deep learning models. Other studies propose to relate a specified set of weights to meaningful physical values.

12.4.3. Hybrid Physics-based Models

Willard et al. [13] uses the example from Daw et al.[51] where the authors combine a physics-based model, f_{PHY}, and a neural network, f_{NN}, such that they complement each other and suppress their disadvantages. An approach is to use merge a physics-based model with a machine learning model by combining its inputs and outputs. Even though the concept was utilised for a Neural Network, the concept can also be adapted to GANs as shown in Figure 12.7.

12.5. Evaluation

Unlike the established evaluation metrics of GANs in computer vision related tasks, evaluation of time series based networks remains a challenging and open topic [40]. Evaluation metrics may also differ per field; medical, financial, audio, and other applications. This section aims to discuss the different approaches found within the time series generation literature.

El Emam [54] argues that synthetic data should be backed by strong evidence of utility. In other words, data augmentation makes sense if downstream prediction models can benefit from it. The authors examine different evaluation methods to determine the utility of the generated data. In this section, three different methods are discussed: assessment by experts, replication, and general metrics.

12.5.1. Assessment by Experts

Evaluating the quality of generated time series is generally more complicated for humans, making quantitative metrics the preferred choice for assessing time series-based GANs. Nonetheless, given that the synthetic

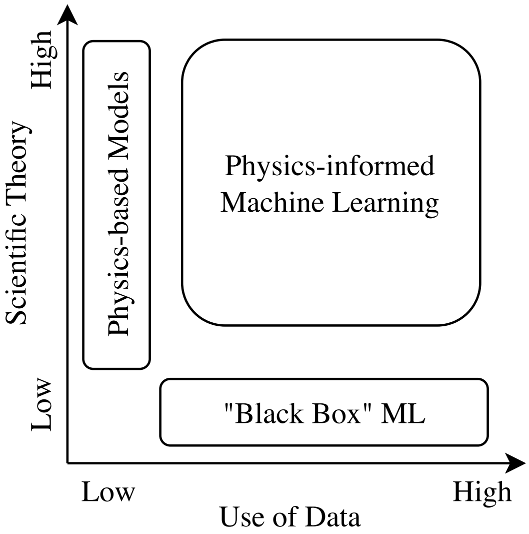


Figure 12.6: Physics-informed Machine Learning combines scientific knowledge with the power of machine learning (taken from Daw et al. [51]). This framework may either improve the generalisability of data-driven models or reduce the amount of labelled data required.

data aims to realistically represent turbofan engine degradation, visual inspection aligned with existing literature is a possible evaluation method. Domain experts are tasked to distinguish both original and artificially generated datasets. The success rate of the expert's classification could serve as an indicator of the generator's ability to produce real data.

12.5.2. Latent Space Interpolation

Latent space interpolation seeks to proof whether smooth variations of the training data can be created from the latent space. Esteban, Hyland, and Rätsch [44] implemented this test to assess whether the generator has learned the underlying distribution instead of replicating the training data.

For a more comprehensive understanding, consider Figure 12.8 where two real time series are traced back to their respective coordinates in the latent space, Z_1 and Z_5. The selected training examples must be similar, but not identical. By generating synthetic data at each point in the latent space along the interpolation of these two training samples, we can study the progression of the synthetic data. A gradual change of the generated time series indicates that the generator has learned the underlying distribution instead of memorising the data.

12.5.3. Replication

A general method for utility assessment involves reproducing results using synthetic data, as discussed by El Emam [54]. The essence of this approach lies in the following hypothesis: the synthetic data can be considered useful if similar conclusions can be drawn from both the original and synthetic datasets.

This concept is also utilised in the study by Esteban, Hyland, and Rätsch [44], where they introduce a method known as ”Train on Synthetic, Test on Real” (TSTR) presented in algorithm algorithm 1. This method requires labelled data as the performance of downstream prediction models are compared. The validity of the synthetic data is determined by whether a prediction model trained on the synthetic data yields a predictive performance equal or similar to that achieved when trained on the original data. This process can also be carried out the other way around (e.g. TRTS).

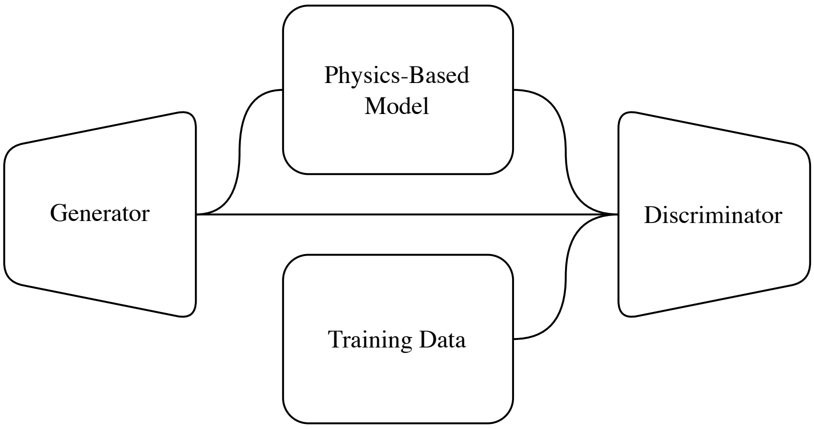


Figure 12.7: A hybrid GAN architecture inspired by Daw et al.[51]. The generator output is used to guide a set of parameters that run a physics-based model. Then, the discriminator is tasked to distinguish between real and fake data.

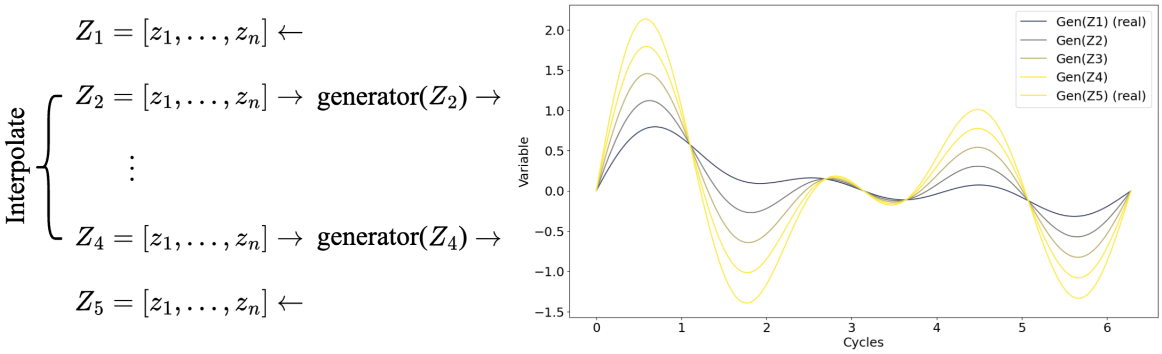


Figure 12.8: Smooth variations produced by the generator ($G(Z)$) between two real trajectories indicates the model has not memorised the data [44].

Algorithm 1: Train on Synthetic, Test on Real (TSTR) [44]

```
# Train on synthetic data
synthetic_train_data = generator(random_vector)
synthetic_train_RUL = get_labels_from_series(synthetic_train_data)
predictive_model.fit(synthetic_train_data, synthetic_train_RUL)

# Test on real data
source_test_data, source_test_RUL = get_source_data()
predictions = predictive_model.predict(source_test_data)
TSTR_score = evaluate(predictions, source_test_RUL)
```

12.5.4. General Metrics

Several general metrics to evaluate a GAN are summarised in this section. These are primarily based on visualising and computing the distributions of the real and fake datasets.

Maximum Mean Discrepancy (MMD)

The maximum mean discrepancy (MMD) calculates the statistical distribution of two sets and assesses whether they belong to the same distribution [55]. Minimising this metric corresponds to a better performing generator. The MMD is based on a kernel, K, which calculates the distance between two points in higher dimensional space. A commonly used kernel is the radial basis function (RBF): $K(x, y) =$

$\exp\left(-\|x-y\|^2/\left(2\sigma^2\right)\right)$. Equation 12.3 compares the two sets after expanding the inner dot product.

$$\widehat{\mathrm{MMD}}_u^2 = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} K\left(x_i, x_j\right) - \frac{2}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} K\left(x_i, y_j\right) + \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} K\left(y_i, y_j\right) \tag{12.3}$$

PCA & t-SNE

Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are both statistical methods for dimensionality reduction and data visualisation. They enable visual inspection of the data by comparing the distribution in lower dimensional space. Yoon, Jarrett, and Schaar [56] applied these techniques to show the overlap between the generated and original data. An example of a t-SNE plot is provided in Figure 12.9. In this particular example, the fake data appears to have a poor resemblance of the real distribution.

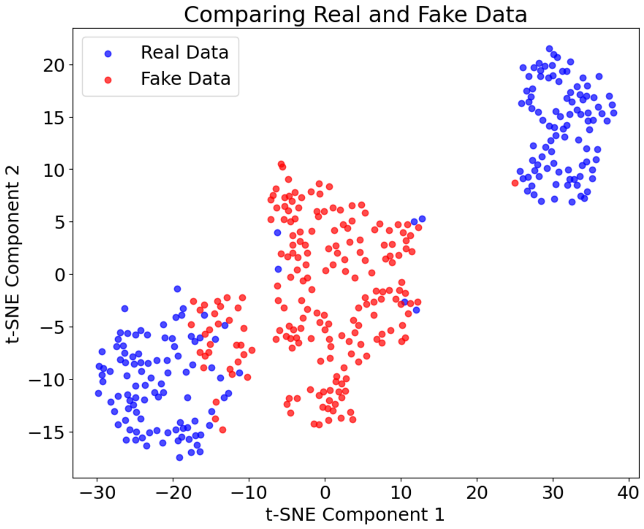


Figure 12.9: An example of a t-distributed Stochastic Neighbor Embedding (t-SNE) graph visualising the distribution of a real and fake dataset. Generation of the fake data should resemble the real distribution as close as possible

13 Conclusion

A substantial part of aircraft maintenance expenses results from turbofan engines [1]. Maintenance, Repair, and Overhaul (MRO) engineers could see gain using Condition Based Maintenance to design cost-effective maintenance strategies by monitoring the engine's current health [2]. For this, MRO engineers resort to sophisticated simulators, often based on Gas Path Analysis (GPA), to derive important health indicators with the measured engine data [3].

Prognostics and Health Management (PHM) further extends this framework by predicting the future health of the engine. Advanced machine learning techniques, such as Neural Networks, have demonstrated strong capabilities in forecasting the Remaining Useful Life (RUL) of complex systems [7]. However, due to the limited turbofan lifecycle data, a consequence of safety-critical systems, training these networks poses a significant challenge [5].

This review offers a comprehensive analysis of the various data augmentation techniques designed to mitigate the data scarcity issue in turbofan engines [33]. In an attempt to solve this issue, researchers have simulated run-to-failure trajectories using GPA based models [8][9]. While these simulated datasets have enabled more research in RUL prediction, especially for machine learning [6], their underlying assumptions and limitations raise questions about their generalisability to real-world scenarios [15, 10, 11].

Generic augmentation techniques have proven to be useful in improving downstream prognostics models for both simulated and industrial datasets [31]. However, these methods have inherent limitations and may sometimes lead to lower quality samples [34]. Instead, generative models offer the potential for creating new, high-quality samples coherent with the original data distribution [12]. In recent years, Generative Adversarial Networks (GAN) have demonstrated promising results in the computer vision domain [34]. However, their role in sequential or time-series datasets remains relatively unexplored. This literature review draws inspiration from other fields where Recurrent GANs have been utilized to generate classical music [43], multi-variate medical time series [44], synthetic scenarios for autonomous vehicle datasets [47]. Current literature has limited focus on the specific application of GANs in turbofan datasets [28, 53].

Besides problems like mode collapse and vanishing gradients, a critical issue in training GANs is their requirement for extensive data. Physics-informed Machine Learning (PIML) offers a potential solution by narrowing down the search space using prior knowledge on turbofan physics and deterioration patterns [13]. Moreover, researchers propose hybrid architectures or adapt the loss function to penalise the GAN when health parameters deviate from from physical laws [51]. While a standardised evaluation technique in time-series applications is yet to be established, existing methods such as expert assessment, latent space interpolation, result replication, and other generalised metrics offer crucial insights into GAN performance [40]. To increase confidence in the results, researchers often use multiple assessment methods to substantiate their findings [54].

To conclude, an opportunity is presented to further narrow the research-industry gap by utilizing KLM's engine fleet data for GAN-based augmentation of turbofan lifecycle data. Furthermore, hybrid GAN architectures that merge physics-based GPA models with data-driven techniques remain an unexplored opportunity for research. Finally, we also identify an opportunity to contribute towards evaluating time-series GANs as it remains an active area of research.

References

[1] International Air Transport Association. *Airline Maintenance Cost Executive Commentary.* 2021. URL: https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2021-mctg-report_public.pdf.

[2] Y. G. Li. "Gas Turbine Performance and Health Status Estimation Using Adaptive Gas Path Analysis". In: *Journal of Engineering for Gas Turbines and Power* 132.4 (Apr. 2010). DOI: 10.1115/1.3159378.

[3] Wilfired P J Visser. "Generic Analysis Methods for Gas Turbine Engine Performance: The development of the gas turbine simulation program GSP". In: (2015).

[4] Michel L. Verbist et al. "Experience With Gas Path Analysis for On-Wing Turbofan Condition Monitoring". In: *Journal of Engineering for Gas Turbines and Power* 136.1 (Jan. 2014). DOI: 10.1115/1.4025347.

[5] Yaguo Lei et al. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical Systems and Signal Processing* 104 (May 2018), pp. 799–834. DOI: 10.1016/j.ymssp.2017.11.016.

[6] Simon Vollert et al. "Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set". In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).* IEEE, Sept. 2021, pp. 1–8. DOI: 10.1109/ETFA45728.2021.9613682.

[7] Maria Grazia De Giorgi et al. "Exploring Prognostic and Diagnostic Techniques for Jet Engine Health Monitoring: A Review of Degradation Mechanisms and Advanced Prediction Strategies". In: *Energies* 16.6 (Mar. 2023), p. 2711. DOI: 10.3390/en16062711.

[8] Abhinav Saxena et al. "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *2008 International Conference on Prognostics and Health Management.* IEEE, Oct. 2008, pp. 1–9. DOI: 10.1109/PHM.2008.4711414.

[9] Manuel Arias Chao et al. "Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics". In: *Data* 6.1 (Jan. 2021), p. 5. DOI: 10.3390/data6010005.

[10] Simon Zhai et al. "Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning". In: *Journal of Manufacturing Systems* 61 (Oct. 2021), pp. 830–855. DOI: 10.1016/j.jmsy.2021.02.006.

[11] Carlos Ferreira et al. "Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods". In: *Journal of Manufacturing Systems* 63 (Apr. 2022), pp. 550–562. DOI: 10.1016/j.jmsy.2022.05.010.

[12] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: (June 2014).

[13] Jared Willard et al. "Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems". In: (Mar. 2020). URL: https://arxiv.org/abs/2003.04919v4.

[14] Zhibin Zhao et al. "Challenges and Opportunities of AI-Enabled Monitoring, Diagnosis &amp; Prognosis: A Review". In: *Chinese Journal of Mechanical Engineering* 34.1 (Dec. 2021), p. 56. DOI: 10.1186/s10033-021-00570-7.

[15] Rainer Kurz et al. "Degradation Effects on Industrial Gas Turbines". In: *Journal of Engineering for Gas Turbines and Power* 131.6 (Nov. 2009). DOI: 10.1115/1.3097135.

[16] Mohammadreza Tahan et al. "Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review". In: *Applied Energy* 198 (July 2017), pp. 122–144. DOI: 10.1016/j.apenergy.2017.04.048.

[17] R. Kurz et al. "Gas Turbine Tutorial - Maintenance And Operating Practices Effects On Degradation And Life". In: *Proceedings of the 36th Turbomachinery Symposium* (2007).

[18] Marta Zagorowska et al. "Influence of compressor degradation on optimal operation of a compressor station". In: *Computers & Chemical Engineering* 143 (Dec. 2020), p. 107104. DOI: 10.1016/J.COMPCHEMENG.2020.107104.

[19] W. P. J. Visser et al. "A Generic Approach for Gas Turbine Adaptive Modeling". In: *Journal of Engineering for Gas Turbines and Power* 128.1 (Jan. 2006), pp. 13–19. DOI: 10.1115/1.1995770.

[20] T. O. Rootliep et al. "Evolutionary Algorithm for Enhanced Gas Path Analysis in Turbofan Engines". In: *Volume 1: Aircraft Engine; Fans and Blowers; Marine; Wind Energy; Scholar Lecture*. American Society of Mechanical Engineers, June 2021. DOI: 10.1115/GT2021-59089.

[21] Yang Hu et al. "Prognostics and health management: A review from the perspectives of design, development and decision". In: *Reliability Engineering & System Safety* 217 (Jan. 2022), p. 108063. DOI: 10.1016/j.ress.2021.108063.

[22] Nam-Ho Kim et al. "Prognostics and health management of engineering systems". In: *Switzerland: Springer International Publishing* (2017).

[23] Houman Hanachi et al. "A Physics-Based Modeling Approach for Performance Monitoring in Gas Turbine Engines". In: *IEEE Transactions on Reliability* 64.1 (Mar. 2015), pp. 197–205. DOI: 10.1109/TR.2014.2368872.

[24] Stuart Russell et al. "Artificial Intelligence: a modern approach, 4th US ed". In: *University of California, Berkeley* (2021).

[25] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network". In: (Aug. 2018). DOI: 10.1016/j.physd.2019.132306.

[26] Paulo Roberto de Oliveira da Costa et al. "Attention and Long Short-Term Memory Network for Remaining Useful Lifetime Predictions of Turbofan Engine Degradation". In: *International Journal of Prognostics and Health Management* 10.4 (June 2023). DOI: 10.36001/ijphm.2019.v10i4.2623.

[27] Timothy Darrah et al. "Developing Deep Learning Models for System Remaining Useful Life Predictions: Application to Aircraft Engines". In: *Annual Conference of the PHM Society* 14.1 (Oct. 2022). DOI: 10.36001/phmconf.2022.v14i1.3304.

[28] Pengxue Lang et al. "Data augmentation for fault prediction of aircraft engine with generative adversarial networks". In: *2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS)*. IEEE, Dec. 2021, pp. 1–5. DOI: 10.1109/SAFEPROCESS52771.2021.9693711.

[29] Manuel Arias Chao et al. "Fusing physics-based and deep learning models for prognostics". In: *Reliability Engineering & System Safety* 217 (Jan. 2022), p. 107961. DOI: 10.1016/j.ress.2021.107961.

[30] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", 2022.

[31] Antonin Gay et al. "Data Augmentation-based Prognostics for Predictive Maintenance of Industrial System". In: *CIRP Annals* 71.1 (2022), pp. 409–412. DOI: 10.1016/j.cirp.2022.04.005.

[32] Robert H Shumway et al. *Time series analysis and its applications.* Vol. 4. Springer, 2017.

[33] Qingsong Wen et al. "Time Series Data Augmentation for Deep Learning: A Survey". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 4653–4660. DOI: 10.24963/ijcai.2021/631.

[34] Guillermo Iglesias et al. "Data Augmentation techniques in time series domain: A survey and taxonomy". In: (June 2022). DOI: 10.1007/s00521-023-08459-3.

[35] Dean Frederick et al. "User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)". In: *NASA Technical Manuscript* 2007–215026 (Jan. 2007).

[36] Kai Goebel et al. "Modeling Propagation of Gas Path Damage". In: *2007 IEEE Aerospace Conference*. IEEE, 2007, pp. 1–8. DOI: 10.1109/AERO.2007.352835.

[37] B Matthews. *NASA DASHlink Flight Data For Tail 687*. 2012. URL: https://c3.nasa.gov/dashlink/resources/664/.

[38] David Foster. *Generative deep learning*. 2nd. " O'Reilly Media, Inc.", 2022.

[39] Diederik P Kingma et al. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[40] Eoin Brophy et al. "Generative Adversarial Networks in Time Series: A Systematic Literature Review". In: *ACM Computing Surveys* 55.10 (Oct. 2023), pp. 1–31. DOI: 10.1145/3559540.

[41] Martin Arjovsky et al. "Wasserstein GAN". In: (Jan. 2017).

[42] Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: (Mar. 2017).

[43] Olof Mogren. "C-RNN-GAN: Continuous recurrent neural networks with adversarial training". In: (Nov. 2016).

[44] Cristóbal Esteban et al. "Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs". In: (June 2017).

[45] Xiaomin Li et al. "TTS-GAN - A Transformer-Based Time-Series Generative Adversarial Network". In: 2022, pp. 133–143. DOI: 10.1007/978-3-031-09342-5{_}13.

[46] Ashish Vaswani et al. "Attention Is All You Need". In: (June 2017).

[47] Henrik Arnelid et al. "Recurrent Conditional Generative Adversarial Networks for Autonomous Driving Sensor Modelling". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, Oct. 2019, pp. 1613–1618. DOI: 10.1109/ITSC.2019.8916999.

[48] Ning Qiang et al. "Learning brain representation using recurrent Wasserstein generative adversarial net". In: *Computer Methods and Programs in Biomedicine* 223 (Aug. 2022), p. 106979. DOI: 10.1016/j.cmpb.2022.106979.

[49] Andreas Lövberg. "Remaining Useful Life Prediction of Aircraft Engines with Variable Length Input Sequences". In: *Annual Conference of the PHM Society* 13.1 (Dec. 2021). DOI: 10.36001/phmconf.2021.v13i1.3108.

[50] Andreas Demetriou et al. "Generation of Driving Scenario Trajectories with Generative Adversarial Networks". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Sept. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294362.

[51] Arka Daw et al. "Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling". In: (Oct. 2017).

[52] Zhongkai Hao et al. "Physics-Informed Machine Learning: A Survey on Problems, Methods and Applications". In: (Nov. 2022).

[53] Jiawei Xiong et al. "Controlled physics-informed data generation for deep learning-based remaining useful life prediction under unseen operation conditions". In: *Mechanical Systems and Signal Processing* 197 (Aug. 2023), p. 110359. DOI: 10.1016/j.ymssp.2023.110359.

[54] Khaled El Emam. "Seven Ways to Evaluate the Utility of Synthetic Data". In: *IEEE Security & Privacy* 18.4 (July 2020), pp. 56–59. DOI: 10.1109/MSEC.2020.2992821.

[55] Arthur Gretton et al. "A Kernel Method for the Two-Sample Problem". In: *CoRR* abs/0805.2368 (2008).

[56] Jinsung Yoon et al. "Time-series Generative Adversarial Networks". In: *Neural Information Processing Systems (NeurIPS)* (2019).