University of Central Florida
STARS

Electronic Theses and Dissertations, 2020-

2021

Model-Form Uncertainty Quantification in Prognosis and Fleet Management with Physics-Informed Neural Networks

Arinan De Piemonte Dourado
University of Central Florida

Part of the Mechanical Engineering Commons
Find similar works at: https://stars.library.ucf.edu/etd2020
University of Central Florida Libraries http://library.ucf.edu

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

De Piemonte Dourado, Arinan, "Model-Form Uncertainty Quantification in Prognosis and Fleet Management with Physics-Informed Neural Networks" (2021). *Electronic Theses and Dissertations, 2020-*. 491.
https://stars.library.ucf.edu/etd2020/491

MODEL-FORM UNCERTAINTY QUANTIFICATION IN PROGNOSIS AND FLEET MANAGEMENT WITH PHYSICS-INFORMED NEURAL NETWORKS

by

ARINAN DE PIEMONTE DOURADO
M.Sc. Universidade Federal de Uberlandia, 2013
B.S. Universidade Federal de Uberlandia, 2011

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando - Florida

Spring Term
2021

Major Professor: Felipe A. C. Viana

© 2021 Arinan Dourado

ABSTRACT

Prognosis and health management play an important role in the control of costs associated with the operation of large industrial equipment. By properly comprehending hardware degradation and accurately predicting the remaining useful life of such equipment, we can significantly lower operational costs by reducing asset downtime and maintenance interventions. However, complex interactions between operational conditions and component capability make accurately modeling damage accumulation for large fleets a daunting task. Unforeseen factors such as aggressive missions introduced by operators, exposure to harsh environments, manufacturing issues, amongst many others, can lead to large discrepancies between predicted and observed useful life. Motivated by the growing availability of data and computational power as well as the advances in hybrid modeling frameworks, capable of merging elements of physics, machine learning, and statistical learning, in this dissertation, we focus on the development of novel approaches to minimize the impact of unforeseen factors in fleet management.

In this dissertation, we focus on the challenges of accounting for the impacts of such unforeseen factors on two specific stages of a component service life; early-file and end-life. Two numerical case studies are derived to emulate two common issues in fleet life management; manufacturing issues leading to an infant mortality problem, and unexpected exposure to harsher environments by operators, accelerating wear-out and significantly reducing component's useful life.

In the first analysis, two key aspects in a prognosis and health management perspective are addressed; detecting the emerging issue (i.e., the infant mortality problem), and the evaluation of risk mitigation procedures to minimize/mitigate its effects on the overall fleet reliability. Bayesian networks implementing physics-based models are used to model the fleet unreliability and assist in the quantification of the infant mortality impact on the fleet useful life. Additionally, steps to adapted the derived Bayesian networks to assist in the evaluation of possible mitigation approaches to minimize the impacts of fleet-wide early life problems are presented.

Concerning the wear-out analysis, a civil aviation case study is derived, in which an aircraft fleet mainly operates in coastal routes, significantly increasing its exposure to saline corrosion. These conditions lead to accelerated degradation of the aircraft wing panels due to the combined effects of corrosion and mechanical fatigue. Such corrosive conditions are not accounted for by the fleet prognosis model generating a significant epistemic uncertainty (i.e., a missing physics issue). To address this issue, we proposed hybrid recurrent neural network modules to compensate for the model-form uncertainty. In the formulated neural network cell, well-understood aspects of the degradation mechanism are addressed by a physics-based model, while data-driven models are trained to account for the missing physics effects. After proper training, the hybrid neural network can compensate for the unaccounted effects in the model damage forecast and generates accurate predictions to assist in the fleet prognosis analysis.

Obtained results illustrate the capabilities of the proposed frameworks in compensating for the considered unforeseen factors impacts in fleet management. Additionally, the obtained results have prominently shown the significance and importance of properly account for such factors on fleet prognosis and how these factors can drastically hinder engineers' ability to properly perform prognosis and health management analysis.

I dedicate this dissertation to my lovely wife Mirella De Piemonte Borges Dourado. Thank you for the love and support you gave me during this endeavor. This is our accomplishment and another significant step in making our dreams come true. With all my love.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my family, in special to my mother for all the sacrifices you made to provide me with the solid education basis that led me to all of my accomplishments.

I am also extremely grateful to have a true angel in my life, my wife Mirella, thank you for sharing this dream, for embracing this enormous challenge of immigrating to a different country, and for all the support and help you continuously give me to make our dreams come true.

I thank also my colleges from the Probabilistic Mechanics Laboratory (PML), Andre, Kajetan, Renato and Yigit in addition to Douglas, George and Wayne, for the support and friendship that you all provided during this journey.

I would also like to express my sincere thanks to my advisor Dr. Felipe Viana, who I learned to admire not only for its professional achievements but also for his life history.

A special thanks to the faculty and staff of the MAE department, not only for all the support during this graduate research, but also for providing me with the unique opportunity to teach undergraduate courses alongside my doctoral research. A special thanks also to Dr. Ali Gordon for trusting in my research allowing us the opportunity to collaborate.

Lastly, but by no means least, my sincere thanks to the members of my advisory committee, Dr. Ali Gordon, Dr. Dazhong Wu, and Dr. Necati Catbas, for their willingness on evaluating this research effort and contributing to its improvement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Large industrial equipment, such as aircraft, are unique in which their maintenance represents a significant portion of the total costs of the particular industry segment. For example, Tab. 1.1 illustrates the breakdown of the total airline cost [5]. Even though percentages might fluctuate year over year, maintenance, repair, and overhaul are likely to keep being one of the main contributors to cost. Therefore, the ability to perform diagnosis (identifying the problem) and prognosis (forecasting or predicting what will probably occur in the future) is critical for companies operating these machines.

Most industrial segments have long moved away from reacting to field issues and failures. Instead, operations and maintenance are highly guided by predictive models, and as such, successful implementation of prognosis and health management programs dependents on the understanding of the complex stochastic interactions between operating conditions and component capability. As a matter of fact, there exist a very active market targeting services of industrial assets. Typically,

Table 1.1: Total airline cost in 2013 (adapted from [2]).

Item	Contribution (%)
Fuel and oil	33.4
Aircraft ownership	10.6
Maintenance and overhaul	9.4
General and administrative	7.3
Flight deck crew	6.8
Reservation, ticketing, sales, and promotion	6.5
Cabin attendants	6.5
Station and ground	5.1
Airport charges	4.9
Passenger service	4.2
Air navigation charges	4.1
Other	1.2

original equipment manufacturers, operators, independent service providers, and software companies compete with one another through offers covering everything from inspection, to digitization, to monitoring and day-to-day services, as well as major maintenance and repairs [6, 7, 8, 9, 10].

With the exciting and vivid market for prognosis and health management in mind, few questions might puzzle the uninitiated. Firstly, why do not companies perform tests to detect early life problems before shipping every machine to customers? Unfortunately, this would be a prohibitively expensive approach. While few components (or at least samples out of a manufacturing batch) might be tested, companies can rarely afford exhaustive testing. Secondly, why do not engineers and scientists design components in a way to avoid or radically mitigate hardware degradation and failure in the first place? In a way, the design of industrial equipment does take hardware degradation and failure into consideration. For the most part, companies adopt different strategies, but it is fair to say that machines are designed to mitigate problems with known failure modes. Then, a level of conservatism is added to account for unknowns. Designing for maintenance and sustainment could easily get even more complicated if we consider the influence of manufacturing variability, quality control, maintenance, and service levels over time.

Under the previously discussed limitations, in almost all industrial segments, one would, unfortunately, find discrepancies between predicted and observed useful lives. Therefore, in prognosis and health management, much of the work is spent on building credible models for *unexpected* failure modes and/or machine degradation rates. These unexpected failures arise due to the complex interactions between operational conditions and component capability combined with unforeseen factors such as aggressive missions introduced by operators, exposure to harsh environments, etc, leading to large discrepancies between predicted and observed useful life. Often, our ability to accurately model damage accumulation hinges on our understanding of the physics of degradation as well as our ingenuity in overcoming limitations introduced by the aforementioned factors.

Motivated by the growing availability of data and computational power as well as the advances in hybrid modeling frameworks, capable of merging elements of physics, machine learning, and statistical learning, in this dissertation, we focus on the development of novel approaches to minimize the impact of such factors in fleet management. In the scope of this dissertation, fleet management can be understood as the set of procedures to oversee entire fleets of industrial assets performance to increase productivity and reduce costs. A significant way to lower operational costs in most fleet management applications is to reduce asset downtime and maintenance interventions. To this end, proper comprehension of hardware degradation and accurate predictions of remaining useful life is crucial. Despite the advances in uncertainty propagation in prognosis and health management, uncertainty estimations on a fleet level are still a challenging task. Multiple sources of uncertainties can act asynchronously throughout the fleet usage history and must be properly handled by the fleet model.

In this dissertation, we focus on the challenges associated with accounting for the impacts of such unforeseen factors on two specific stages of a component service life; early-file and end-life. Two numerical case studies are derived to emulate two common issues in fleet management; manufacturing issues leading to an infant mortality problem, and unexpected exposure to harsher environments by operators, accelerating wear-out and significantly reducing component's useful life. It is worth mentioning that, our goal is not to address all possible sources of uncertainties in these scenarios but to tackle the impacts of the epistemic uncertainties introduced in the damage prediction models by these factors. Specifically, we focus on compensating for the partial knowledge of the physics of degradation and its effects on component damage prediction.

In the first analysis, two key aspects in a prognosis and health management perspective are addressed; detecting and characterizing the emerging issue (i.e., the infant mortality problem), and the evaluation of risk mitigation procedures to minimize/mitigate its effects on the overall fleet reliability. For this analysis, a probabilistic framework able to identify and characterize early life problems in fleets models is proposed. The proposed framework can also provide an assessment of mitigation measures on a fleet level. We will present the analysis of an infant mortality problem using a numerical experiment, focusing on answering the following fundamental question: *how do fleet size and the number of failures interact with each other when characterizing an infant mortality problem?* We use a physics-based prognosis model as a way to forecast the remaining useful life through the progression of hardware distress by fusing design, manufacturing, and services information. The proposed framework is composed of Bayesian networks used to model fleet unreliability considering the effects of manufacturing problems (due to bad batch of materials).

The proposed Bayesian models use two model parameters: material capability degradation, model as a shift in the material nominal stress-life curve (S-N curve); and fleet penetration level, i.e., the percentage of units in the fleet plagued by the material capability degradation. One advantage of the approach is that we can leverage loads and environment information without the need for direct damage measurement. Although we point out that our approach offers only a damage estimate (which can only be confirmed with inspections).

Concerning the wear-out analysis, a civil aviation case study is derived, in which an aircraft fleet mainly operates in coastal routes, significantly increasing its exposure to saline corrosion. These conditions lead to accelerated degradation of the aircraft wing panels due to the combined effects of corrosion and mechanical fatigue. Such corrosive conditions are not accounted for by the fleet prognosis model generating a significant epistemic uncertainty (i.e., a missing physics issue). To address this issue, we proposed hybrid recurrent neural network modules to compensate for the model-form uncertainty. In the formulated neural network cell, well-understood aspects of the degradation mechanism are addressed by a physics-based model, while data-driven models are trained to account for the missing physics effects. The final result is a hybrid damage model, where physics-based models provide the baseline for damage accumulation, while data-driven models implement correction terms to compensate for the unaccounted and unexpected uncertainties (see

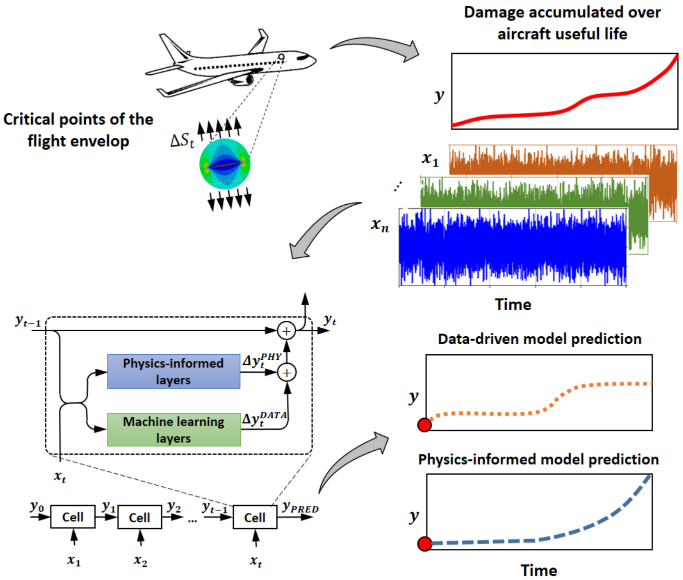


Figure 1.1: Illustration of the proposed formulation. Fleet usage history is fed into a hybrid recurrent neural network that is trained to accurately represent asset degradation.

Fig. 1.1). After proper training, the hybrid neural network can compensate for the unaccounted effects in the model damage forecast and generates accurate predictions to assist in the fleet prognosis analysis.

In summary, the main problem addressed in this research effort concerns *epistemic uncertainties effects on prognosis and health management models predictions for fleet-level analysis*. In this regard, the main challenges when undertaking this problem can be encapsulated by the following fundamental questions; in a modeling perspective, *is our proposed framework able to account for the impacts of model form uncertainties in the fleet damage forecast?* From an engineering perspective, *are the proposed hybrid frameworks capable to reliably assist with fleet management tasks (e.g., design of inspection campaigns, risk mitigation assessment, etc)?*

The remainder of this dissertation is organized as follows. Chapter 2 presents a survey of modeling and analytics used in prognosis and health management. Discussions concerning the general aspects of maintenance, contributing factors for the implementation of prognosis and health management, as well as supporting technology are also presented. Chapter 3 focuses on the discussion concerning the early-life failure analysis and its related case study as well as the proposed probabilistic framework and most significant results obtained. Chapter 4 illustrates a modeling approach that combines physics-informed and machine learning models into a single hybrid model concerning a wear-out analysis focused on a corrosion-fatigue case study. This strategy is growing in popularity since it combines the understanding of machine operation through reduced-order models with the potential that machine learning has in quantifying uncertainty. The corrosion-fatigue case study is then expanded in Chapter 5 to consider seasonality effects in the corrosion damage. Additionally, different functional forms for the data-driven portion of the hybrid neural network cell derived from varying physics-based considerations and constraints are presented and properly discussed. In Chapter 6, the use of ensembles of hybrid modular neural networks for model-form uncertainty compensation is discussed considering the expanded corrosion-fatigue case study, and the previously defined hybrid neural network cells. A brief discussion concerning the use of the ensemble uncertainty to evaluate the ‘‘credibility’’ of predicted damage forecasts is also presented. Finally, chapter 7 closes the dissertation highlighting salient points and presenting few suggestions for future research.

CHAPTER 2: LITERATURE REVIEW - A SURVEY OF MODELING FOR PROGNOSIS AND HEALTH MANAGEMENT

In this chapter, we present a survey of the literature and discuss important aspects of prognosis and health management with focus on modeling approaches. In the presented survey, we confirmed that maintenance approaches for industrial equipment have evolved over the years. Part of the reason is the shift in paradigm. Most operators have moved away from run-to-failure and corrective maintenance and have been adopting predictive maintenance, where models assist decision regarding operation and maintenance levels. In addition, the growth in computational power and access to data have enable sophisticated models in line with recent initiatives towards digitization and the industrial internet of things [11, 12, 13]. In the presented review, we also confirm that prognosis models rarely rely on a single approach. Instead, most applications tend to lead to elaborated models with elements of statistical and machine learning often intertwined with reduced-order physics-based models. In addition, we present specific discussion of diagnosis and prognosis for civil aviation.

2.1 General Considerations and Implementation of Prognosis and Health Management

In this section, we briefly discuss what goes into implementing prognosis and health management. Fig. 2.1 illustrates some of the key aspects. Even though interconnections between the circles are not shown, one can imagine how complex they are in real life. For example, the business model of a particular organization definitely influences the maintenance approach. For some companies, it might make sense to implement proactive maintenance; for others, reactive maintenance might be just fine. In addition, we will also discuss how modeling and analytical methods for prognosis and health management have evolved with the operations and maintenance strategies that different

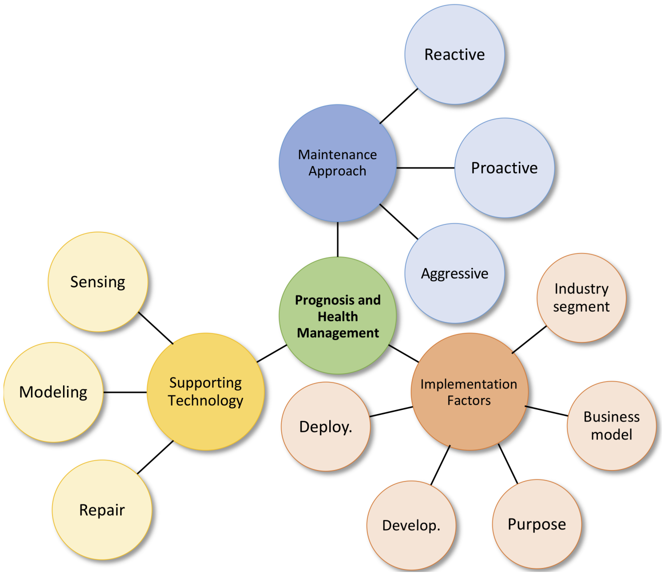


Figure 2.1: Prognosis and health management and peripheral web of approaches, factors, and technologies.

industry segments have in place. Nevertheless, as a way to put modeling in context, we first discuss maintenance approaches and supporting technology (taking the opportunity to also introduce some of the jargon).

2.1.1 Maintenance Approaches

Maintenance is at the heart of operation of complex industrial equipment as it constitutes a large portion of the costs involved in many industries. Semantics aside, Swanson [14] presents an interesting discussion of maintenance strategies and their impact in the performance of production plants. The author starts by defining overarching terminology such as:

- *Reactive maintenance*: where equipment is operated until failure occurs; and only then repairs are performed to bring equipment back to operation.

- *Proactive maintenance*: where equipment deterioration is monitored so that maintenance/repairs are performed to restore equipment to satisfactory condition.
- *Aggressive maintenance*: where equipment is maintained with the aim of improving plant performance as opposed to the individual equipment performance or reliability.

In Swanson's view, preventive and predictive maintenance are included in proactive maintenance and target reducing the probability of unexpected equipment failures. The author also mentioned that predictive maintenance is sometimes referred to as *condition-based maintenance*.

One might argue that the specific nomenclature and taxonomy vary from one industry to another, or even from one scientific community to another. Nevertheless, two points are worth considering. Firstly, maintenance is related directly to cost; which can come in the tangible form of parts, labor, etc., but also in terms of downtime, loss of production, warranties, etc.. Secondly, proper maintenance improves equipment reliability and can restore its performance. For some industries, such as civil aviation, reliability is as important as performance and therefore, companies tend to be very careful while implementing their maintenance strategies. Other industries might weight performance in contrast to strict reliability margins. For example, production plants with redundant production lines (redundancy acts as a parallel system and accommodates reduction in reliability of individual lines).

Naturally, maintenance approaches also evolved over time. Advances in the ability to run complex computational models, improvements in material and manufacturing technologies, quality control, as well as sensing and monitoring allowed the systematic reduction of conservatism in design and operation of industrial equipment . With that, operators have been moving away from reactive maintenance and have been able to adopt different forms of proactive maintenance.

2.1.2 Implementation Factors

We now recognize the different approaches to maintenance and we can imagine that the adopted approach can influence the level of sophistication of models. *The more proactive the maintenance approach, the more predictive the models need to be*. Before we discuss modeling though, we need to elaborate on important factors that shape the implementation of prognosis and health management (as they influence how one perform modeling and implement prognosis analytics):

- *Industry segment:* one can argue that understanding the industry segment is potentially the most important aspect before even considering what prognosis means for that industry. Hardware degradation is usually associated with loss of performance and decreasing reliability. Some industry segments might be very averse to loss in performance. For example, consider machines in a production line. Deteriorating performance of critical machines can create a bottleneck in the production and dramatically decrease yielding. Other industries might be very sensitive to reliability. For example, in civil aviation reliability is related to safety and we do not need to discuss how serious the consequences of degrading reliability could be.
- *Business model:* modeling is supporting technology to a business. Therefore, the business model of a company is an important aspect to consider when building models that will support decision making. For example, if an equipment has parts that are very expensive and dominate the maintenance costs (e.g., hot gas path blades of an aircraft engine), prognosis models tend to focus on accurate prediction and monitoring of hardware degradation. On the other hand, if maintenance cost is dominated by downtime, maintenance equipment and labor; then, analytics tend to focus on optimizing maintenance schedule. Therefore, it is unlikely that one will find ‘‘one size fits all’’ solutions, although patterns might exist.
- *Purpose:* once the business model is understood, the next immediate point to consider might

be the purpose of the analytic. The purpose of a prognosis model is to aid decision regarding a specific aspect of asset operation and/or fleet management. For example, an analytic can help deciding how many spare parts are needed next month or year. Alternatively, analytics can also help with decision of what are the duty cycles (missions) that lead to the safest operation in the next few days. Obviously, these two models are very different in terms of implementation, data required, acceptable uncertainty, etc.

- *Development:* analytics are developed, tested, and validated with care and rigour before final deployment. Most of the time, a multidisciplinary team of highly skilled scientists and engineers work together to tackle challenges associated with several aspects of model development. Activities in the model development include, but not limited to, understanding of hardware operation and degradation, data acquisition and curation, analysis of modeling methods and impact of assumptions, verification and validation, quality assurance and robustness test of resulting computer models, etc.
- *Deployment:* in other words, once the analytics are developed and validated, what is the computational environment the models will run and what is the support level they will need throughout their use? The growth in computational power and access to data opens up the opportunity for sophisticated high-performance and cloud computing solutions [15, 16]. While some organizations might be willing to take on the task of deploying and sustaining analytics, others might prefer outsourcing at least part of the solutions.

2.1.3 Supporting Technologies

Once maintenance approaches and implementation factors are understood, we are ready to discuss the supporting technologies for prognosis and health management:

- *Sensing:* there are no questions that data is at the center of most recent developments in prognosis. However, to the surprise of the uninitiated, in industrial applications, data still tends to be highly unstructured. At least part of this data comes from the machine control units, which is *optimized for machine control and performance*, as opposed to diagnosis and prognosis. Although they are still a luxury, it is true that structural health monitoring systems with sensors and processing units dedicated to diagnosis and prognosis have been increasingly deployed in many applications. Besides thermocouples and accelerometers, structural health monitoring sensors include, but are not limited to, comparative vacuum monitoring [17], fiber Bragg grating sensors [18], Eddy current [19], ultrasound sensors [20], and even dye penetrant inspection [21].

  Unfortunately, in many cases, and certainly for legacy equipment, such systems might not be present (cost analysis might even rule them out as a retrofit). Practitioners can also face problems beyond the ones previously mentioned. While data obtained through sensors used for controlling the equipment tends to be plentiful (which does not mean this data is problem-free), degradation and failure data can impose significant challenges due to to be poor sensing and characterization methods.

  For example, Fig. 2.2 shows two examples of high pressure turbine vanes after they accumulated a number of cycles. While it is easy to recognize that both present different levels of damage, one can imagine the challenges involved in quantifying the damage (degradation) level. For example, should the damage quantification segment the vane per region of interest (leading edge, trailing edge, concave side, and convex side)? What are the metrics (crack length, percent loss of coating, etc.) that could be used to quantify damage? Does it pay to develop an automated technique that quantifies damage (however it is measured)? Or alternatively, are visual inspection and expert judgement sufficient information for building predictive models?

- *Modeling:* in industrial applications, more often than not, multiple models are used to simultaneously diagnose, track and forecast the state of machine health. With these multiple models, engineers and practitioners can take informed decisions about what to do with the specific machine. With that in mind, one might feel tempted to start categorizing the types or families of algorithms that can be used to build prognosis models, such as Fig. 2.3 does for machine learning. The cold reality in prognosis is that, most of the time, models are built to understand machine deviation from design. After all, if failure could be avoided in design, prognosis would not be needed in the first place. Given how challenging this task is, it is unlikely that one single modeling approach is used for building prognosis models.
- *Repair:* in this dissertation, we consider ‘‘repair’’ to be all that goes into fully or partially restoring the functionality of a component (or system) that has already accumulated service life while also preserving acceptable reliability levels. This way, repair technologies have to balance cost of repair with the regained useful life and are highly industry- and application-dependent. Although further discussion on repair technologies is outside the scope of this dissertation, the interested reader is referred to the following literature for some examples in gas turbines [22, 23, 24] and aircraft structures [25, 26].

It is expected that the relative importance of the previously discussed factors will vary on a case-by-case basis. Nevertheless, once this is realized, prognosis and health management becomes much easier to propose, develop, and implement. In the next section, we discuss important elements from an analytics perspective.

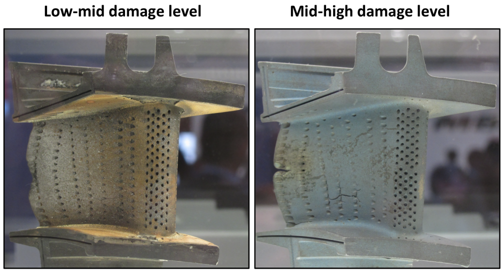


Figure 2.2: Condition of high-pressure turbine guide vanes after many cycles of usage. Pictures adapted from `https://commons.wikimedia.org/wiki/File:Repair_process_for_a_V2500_high-pressure_turbine_guide_vane_(1).jpg` and `https://commons.wikimedia.org/wiki/File:Repair_process_for_a_V2500_high-pressure_turbine_guide_vane_(3).jpg`.

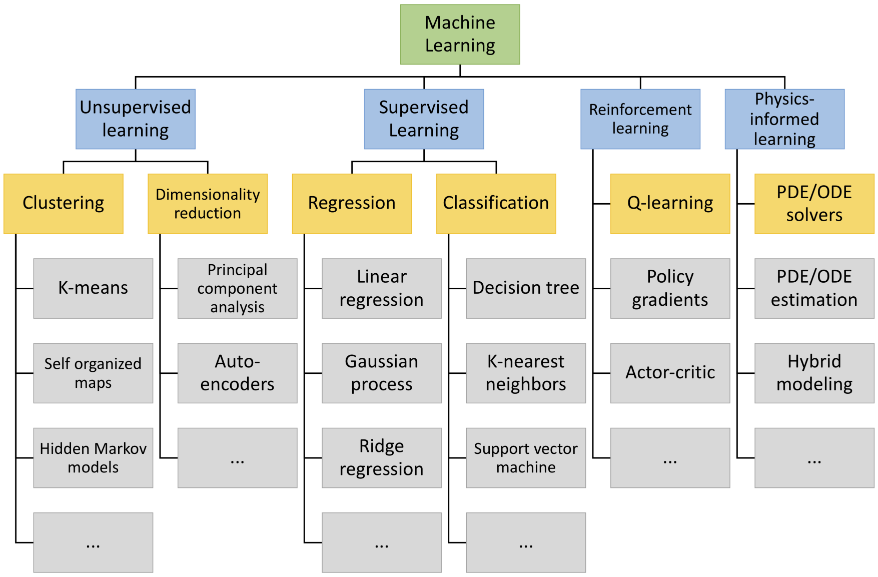
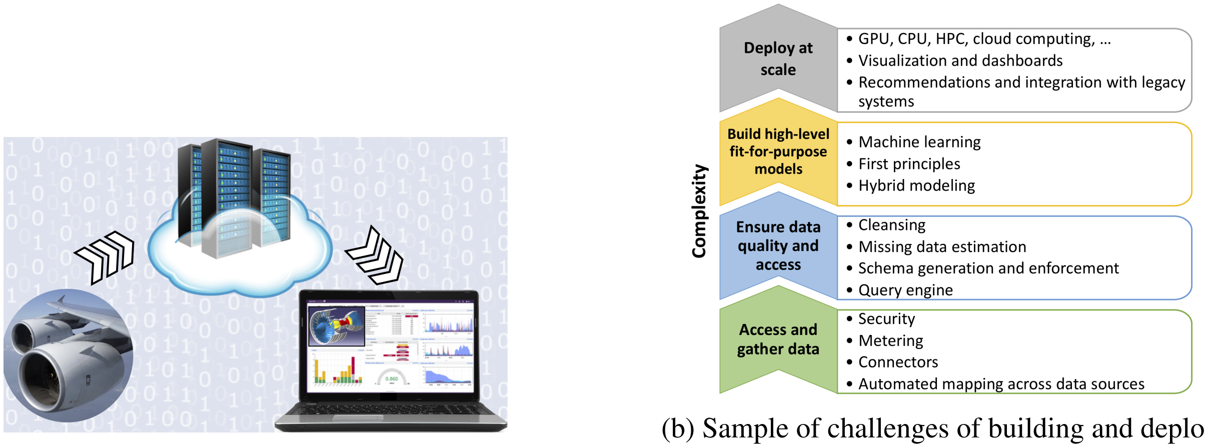


Figure 2.3: Sample of machine learning methods that can be used to build prognosis models.

2.2 Predictive Modeling and Analytics for Prognosis and Health Management

As we have been discussing, prognosis and health management encompass a complex web of approaches, factors and technologies (Fig. 2.1). On the supporting technology side, we would


Deploy at scale
• GPU, CPU, HPC, cloud computing, …
• Visualization and dashboards
• Recommendations and integration with legacy systems
Build high-level fit-for-purpose models
• Machine learning
• First principles
• Hybrid modeling
Ensure data quality and access
• Cleansing
• Missing data estimation
• Schema generation and enforcement
• Query engine
Access and gather data
• Security
• Metering
• Connectors
• Automated mapping across data sources
Complexity

(a) What analytics for prognosis could look like.

(b) Sample of challenges of building and deploying prognosis models.

Figure 2.4: Idealization and technical challenges of building analytics for prognosis of industrial equipment.

like to extend the discussion about modeling approaches, historical perspective, implementation, and deployment of analytics. Fig. 2.4a illustrates an idealized final goal of a prognosis and health management analytics. Assets, such as a jet engines, are monitored on a serial number basis. Sensor data is broadcast directly to the ‘‘cloud.’’ This valuable information run against a pipeline of analytics. At the end, powerful interfaces, designed with user-experience in mind, display the results for decision-makers. In order to achieve this vision though, there are many challenges an organization needs to overcome. As briefly depicted in Fig. 2.4b, from accessing and gathering the data to deploying the analytics at scale, there is a long list of technologies that need to be put in place. Some of these, such as cyber security (obviously the connectivity should not expose industrial equipment to cyber attacks) and metering (data is usually billed by volume) exemplify how information technology plays an important role in prognosis and health management. Further discussion on how some of the challenges shown in Fig. 2.4b relate to prognosis and health management can be found at [27, 28, 29, 30].

Next, we will focus our attention on modeling for prognosis and health management. We start by reviewing some of the commonly used modeling approaches. We will cover different modeling methods but also offer a historical perspective for how analytics evolved over time.

2.2.1 Modeling Approaches

By now, one should realize that much of the challenge in building prognosis analytics comes from dealing with data issues including, but not limited to, noisy sensors, outliers, missing and corrupted data, data forecast, and partial information. Relevant literature is drawn from fields such as statistical and machine learning and popular methods include probabilistic principal component analysis [31, 32, 33], spatial dynamic panel data model [34, 35], Bayesian temporal tensor factorization [36], auto-encoder, variational auto-encoder [37, 38], generative adversarial networks [39, 40], among others.

In many applications, departures and anomalies in the data are used as diagnosis, while drifts in certain trends can be used as rudimentary prognosis. Obviously, simplified models that feed from anomaly detection and monitoring of trends can be very convenient and easy to implement. However, the use of such models for prognosis is often limited by their lack of explainability and their inability to extrapolate. On top of that, industrial equipment is monitored by remote control and operation centers. This way, an excessive number of false alarms and near misses produced by a poorly performing analytic could reduce the credibility of the approach and, as a result, the prognosis model becomes invalid.

Therefore, credible models are of utterly importance in prognosis and health management. Unarguably, a very desirable situation happens when physics is well understood to the point that degradation and failure can be modeled accurately and computational cost of such models are compatible with the application requirements. Even under these circumstances, one should be aware that purely physics-based models are rarely used. In general, there is at least one element of uncertainty quantification and model updating that is almost inevitable due to cumulative uncertainties coming from loads and boundary conditions as well as variations in material properties.

For example, Daigle and Goebel [41] formulated physics-based prognostics as joint state-parameter estimation problem, in which the state of a system along with parameters describing the damage progression are estimated. This is followed by a prediction problem, in which the joint state-parameter estimate (and associated uncertainty) is propagated forward in time to predict end of life and remaining useful life. They demonstrate their methodology in the estimation of remaining useful life of centrifugal pump used for liquid oxygen loading located at the Kennedy Space Center. Literature reporting successful implementation of physics-based prognosis and health management is very rich. Examples include, but are not limited to, bearings failure [42], Lithium-ion batteries [43, 44], structures subjected to corrosion-fatigue [45], fatigue of turbine disks [46], powertrain of unmanned aerial vehicles [47].

One of the benefits of using theory-guided models is that they are based on understanding of degradation mechanisms (using laws of physics, chemistry, etc.). Unfortunately, real-life applications are inevitably subjected to challenges associated with identification of failure modes, sensitivity of failure and damage accumulation to operating conditions, robust isolation of hardware degradation during operation, data problems (noisy sensor readings and missing data, etc.), accuracy and computational cost of physics-based models, interdependence of large system processes, blunt lack of domain knowledge. Therefore, it is understandable that machine learning methods find home in prognosis and health management.

With that said, we would like to open the discussion about machine learning methods by re-stating that they are subjected to the *no free lunch theorems* [48, 49]. These theorems (established at the end of the dark ages of machine learning) state that all learning and optimization algorithms will perform equally well if their performances are checked against all possible learning problems. Alternatively, we also can state that no single learning algorithm will always outperform all others. As a consequence, practitioners are never bounded by one particular algorithm.

Si et al. [50] reviewed statistical data driven approaches for prognosis that rely only on available past observed data and statistical models (regression, Brownian motion with drift, gamma processes, Markovian-based models, stochastic filtering-based models, hazard models, and hidden Markov models).

Khan and Yairi [51] reviewed the application of deep learning in structural health management (simple autoencoders, denoising autoencoder, variational autoencoders, deep belief networks, restricted and deep Boltzmann machines, convolutional neural networks, and purely data-driven versions of recurrent neural networks, including the long short-term memory and gated recurrent units). They found that most approaches are still application specific (unfortunately, they did not find a clear way to select, design, or implement a deep learning architecture for structural health management). They also advise that a trade-off study should be performed when considering complexity and computational cost. Instead, most acknowledge that algorithms should be seen as tools.

A very short discussion of the different modalities of machine learning in prognosis and health management is shown in Tab. 2.1. Besides the modalities discussed, we recognize the substantial work merging physics and statistical learning methods such as Bayesian networks and Kalman filter [42, 52, 53].

Table 2.1: Modalities of machine learning and application to prognosis modeling.

Modality	**Applicability**	**Examples**
Unsupervised learning	When the data set lacks detected patterns and/or clearly defined input-output labeling, algorithms are used to separate and start identifying hidden patterns in the data.	Prognosis and health management literature reports successful applications of clustering in electrical motors [54] and milling machines [55]; self-organizing maps in monitoring of mechanical structures [56] and pneumatic actuator fault detection [57]; and hidden Markov models in fault identification of turbofan engines [58] and bearings [59].
Supervised learning	When the data allows distinction between inputs and outputs, models are built to explain the relationship that maps inputs into outputs. This is probably the machine learning modality with most reported examples in prognosis and health management.	Examples include, but are far from being limited to, ensembles of classifiers in remaining useful life estimation of tungsten filaments [60]; recurrent neural networks for gear prognosis [61]; deep learning for degradation of aircraft engines [62]; and support vector machines for failure of reactor coolant pumps [63].
Reinforcement learning	When machine learning is modeled as a multi-agent problem where data is used to train agents seeking to maximize a reward function [64, 65].	This is a relatively unexplored modality, but examples from the prognosis and health management literature include design of fault-adaptive control strategies [66] and bridge health monitoring [67].
Physics-informed learning	When knowledge about partial differential equations is used to design and/or train the deep learning models [68, 69, 70].	Chao et al. [71] discuss how hybrid physics-informed and machine learning models can be used for prognostics of complex safety critical systems. Viana et al. [72] discuss how to build hybrid recurrent neural networks for cumulative damage modeling by merging physics-of-failure kernels with machine learning for estimation of model inadequacy.

2.2.2 Summary of the past 40 years

Researchers have been very active on the field of modeling and analytics for prognosis and health management. Here, we provide few samples of papers published over the past 40 years on a decade basis.

Sample of papers published in the 1980's:

- 1980 - Nelson [73] discussed maximum likelihood methods to estimate a model for life as a function of constant stress. Initially, the author provided an illustrative data from a step-stress test of cable insulation, and then used a Weibull model that describes the stress-life relation for the case. Then, the author estimated the parameters of the model through maximum likelihood, and validated the estimations over a different cable's test data. Inferring from the validation results, the paper is concluded with the remarks on how to interpret the statistical findings and the effect of number of samples used to fit the models.

- 1984 - Chow and Willsky [74] published a work on designing robust failure detection systems. Their proposed method utilizes parity functions in order to generate residuals, which are exposed to several statistical tests for fault diagnosis and identification. The authors tested their approach on a numerical example, where they considered a four-dimensional system operating at a set-point with two actuators and three sensors. The goal was to successfully detect a sensor failure of a system. The paper pointed out the importance of residual generation process, which yields to accurate fault identification.

- 1985 - Renwick and Babson [75] studied the benefit of using vibration signals as a predictive maintenance tool. The authors combined discrete frequency vibration data with the demodulated signal-conditioning technique, which they refer as ‘‘quality information’’. Their proposed method is demonstrated on several case studies, such as a loose pedestal bearing on the pinion drive of finish mill, shearing of coupling bolts on a cooler vent fan motor shaft, and a bearing fault in a gear reducer. Finally, the authors layout three different level of instruments to acquire vibration signals on different qualities, and discussed the usage of each instrument for each case presented in terms of reliability and cost trade-off perspective.

- 1988 - Gertler [76] investigated failure detection, isolation, and identification algorithms based on reduced order models, residual generation, statistical testing, and sensitivity analysis. The author discussed the applicability of these methods in problems such as sensor biases, actuator malfunctions, leaks, and equipment deterioration.

Sample of papers published in the 1990's

- 1993 - Lu and Meeker [77] studied how degradation measures obtained from life cycle tests (which have either no failures or only a few) can be used to create a time-to-failure distributions. The authors evaluated approaches such as general path and multivariate normal models for nonlinear regression of observed degradation. They also suggested using Monte Carlo simulations in the case where the degradation model does not have a closed-form expression for time-to-failure distribution. The authors demonstrated the effectiveness of their approach on a fatigue crack growth example.

- 1996 - Lee [78] implemented a pattern discrimination model based on cerebellar model articulation component for fault detection and preventive maintenance. The neural network model is initially trained with desired (or normal) behavior of the machine. Then, the model is used to detect any deviation from the normal behavior and depending on the magnitude of deviation, a new confidence level is assigned for reliability of the machine.

- 1996 - Ray and Tangirala [79] presented a stochastic model of fatigue damage, online sensing, failure prognosis, decision-making, and life extending control in complex dynamical systems.

The authors described their method in two parts. The first part is the deterministic model which follows a short crack growth model, and the second part is the stochastic model based on the extended Kalman filter. They closed the paper with a verification study against actual fatigue crack data collected from 2024-T3 aluminum alloy panels. With these results, the authors showed that their method is computationally efficient and suitable for online application.

- 1999 - DePold and Gass [80] presented a framework based on artificial neural networks to build gas turbine prognostics and diagnostics. The authors combined neural networks, Kalman filters, Bayesian and evidence based decision making, along with an expert system to perform value analysis in making condition-based maintenance recommendations. Their framework was tested on a case where a crack occurred on a gas turbine engine combustor with notable prediction accuracy.
- 1999 - Li et al. [81] proposed an adaptive approach to estimate remaining life of rolling element bearings using a reduced-order model tuned with vibration data. Their methodology is based on fine-tuning a model inspired in the Paris' law using recursive least square. The authors provide both simulation and experimental investigation of the performance of their approach.

Sample of papers published in the 2000's

- 2001 - Wang and Vachtsevanos [82] used dynamic wavelet neural networks for prognosis. Their proposed model was able to take the information from diagnosis, predict the remaining useful life of the component dynamically, and also output a maintenance window for condition-based-maintenance. The authors illustrated their method on an example where crack propagation on a bearing is tracked through vibration measurements.

- 2003 - Qiu et al. [83] investigated rolling element bearing prognosis using wavelet filters. They addressed the problem of weak anomaly signals, which are challenging to identify in onset of failure. In addition, the authors propose a self organizing map based method to construct feature space, detect degradation, and assess performance. An experimental validation is performed with run-to-failure test and vibration data from four bearings.

- 2006 - Jardine et al. [84] reviewed diagnostic and prognostic approaches such as statistical models, artificial intelligence methods, and model-based techniques. The authors divided and discussed the condition-based maintenance process in three distinct steps: data acquisition, data processing, and maintenance decision support. Encouraged by the increasing number of complex systems with multiple sensors, they also examined the techniques to fuse sensor data for advanced diagnosis and prognosis of machinery.

- 2008 - Wu and Liu [85] considered using discrete wavelet transform and neural networks for the problem of internal combustion engine fault diagnosis. While the discrete wavelet transform is used to extract features from acoustic emission signals, a multi-layer perceptron is utilized for classification of the fault. The authors verified their framework on an experimental rig, where a gasoline direct injection engine's sound emission signals are analyzed and classified with high accuracy.

- 2009 - Zhang et al. [86] outlined a framework in order to merge physics-of-failure models with data-driven approaches for prognostics and health management. In their framework, physics-of-failure model is used to define failure criteria and thresholds for the data-driven portion, and data-driven part is used for the calibration of physics-of-failure model. The authors close the paper with a futuristic vision where the hybrid approaches will be a cost-effective option for the prognostics and health management of complex industrial systems.

Sample of papers published between 2010's

- 2011 - Sikorska et al. [87] presents an in depth overview for prognostics model selection for industrial application. The discussion begins with the dissection of basic definitions of prognostics. Then, the authors underline some key aspects from the implementation point of view, and present a brief classification to the methods of the time. The paper provides a comparison of modeling approaches with discussion of advantages, disadvantages, and niche of applications.

- 2016 - Mosallam et al. [88] presented a data-driven prognostic method based on Bayesian approaches. The authors described their two-phase framework. While the offline phase constructs custom health indicators that contain information on the degradation of the system and builds reference models; the online phase attempts to classify real-time health indicators to the ones in the reference database to evaluate remaining useful life. Finally, a discrete Bayesian filter is employed to estimate the state of the degradation. Battery and turbofan engine degradation data obtained from a NASA repository are utilized to test the introduced framework, and low mean absolute percentage error is observed in both applications.

- 2018 - Li et al. [89] used deep convolutional neural networks for prognostics of industrial assets. The method adopted a time window approach for improved feature extraction, avoiding the need for prior expertise knowledge or signal processing. The authors illustrated the accuracy of the approach with the remaining useful life estimation of aero-engine units. Authors also compared their proposed framework against the other state-of-the-art data-driven approaches, such as long short-term memory, convolutional neural networks, and deep belief networks.

- 2019 - Kordestani et al. [90] reviewed recent advancements in failure prognosis. Initially, the authors laid out some fundamental concepts regarding fault diagnosis and prognosis, and categorized current available and widely used prognosis methods based on their characteristics, advantages, and disadvantages. Then, authors go over the model building task with approaches such as wavelet transform, extended Kalman filter, fuzzy logic, Gaussian process, hidden Markov modeling, deep learning, and many others. Later in the paper, the authors built a connection in between these methods and the two of the most popular application fields, batteries and rotating machinery systems. Authors conclude the manuscript with a rich discussion on promising research directions.

- 2020 - Yucesan and Viana [91], Dourado and Viana [92], and Nascimento and Viana [93] introduced a hybrid implementation that combines together physics-based reduced order models and neural networks for cumulative damage modeling. In their approach, the physics-informed kernels constrain the neural network outputs while the data-driven kernels quantify model-form uncertainty. Authors demonstrated their approach with applications in wind turbine main bearing fatigue as well as corrosion-fatigue and fatigue crack growth of aircraft fuselage panels. The Python packages that implements their hybrid physics-informed neural networks and application-specific data are freely available at [94, 95, 96].

Prognosis and health monitoring have found a broad spectrum of applications over the years. Besides many of the papers already mentioned, we illustrate in Tab. 2.2 few other publications to show the rich variety of applications.

2.3 Prognosis and Health Management in Civil Aviation

In this section, we will further the discussion about modeling for prognosis and health management applied specifically to civil aviation. Based on this industry own unique features, we will explore how they affect its modeling implementations.

The safety requirements associated with civil aviation have driven implementation of prognosis and health management for this industry. As much of the industry, operations and maintenance

Table 2.2: Additional examples of applications.

Field	Area	References
Physical	Aerospace/Space	[97, 98]
Sciences	Chemical engineering	[99, 100]
& Engineering	Civil/Infrastructure	[101, 102, 103]
	Energy/Propulsion	[104, 105, 106, 107]
	Materials/Manufacturing	[108, 109]
Life & Health	Agriculture	[110, 111]
Sciences	Environmental sciences	[112]
	Pharmaceutical industry	[113]
	Medicine	[114, 115]
Social Sciences	Business	[116, 117]
& Humanities	Decision making	[118, 119]

are heavily regulated, both at the national and international levels. A well-known time-based maintenance set of checks in aviation is the ''ABCD'' check system shown in Tab. 2.3. The guidelines cover everything from daily checks, such as broad visual inspection grounding the aircraft for up to 8 hours, to major inspection involving structural components and grounding the aircraft for up to 20 days. The frequency of each type of check is usually defined in accordance between original equipment manufacturers, operators and airworthiness authorities.

Unfortunately, despite all the regulations and efforts to increase safety, improper maintenance still has a significant impact on reported accidents and incidents, as described in a study published by Marais and Robichaud [120]. As discussed by the authors, despite the downward trend, maintenance-related reports still accounts for roughly 5% of the overall accident and incident reports[1]. Their analysis also indicates that maintenance-related accidents are 6.5 times more likely to be fatal in general.

Prognosis and health management in civil aviation reflects the complexity of the aircraft as a system.

[1] In the USA, the National Transportation Safety Board maintains a repository with aviation accident reports published at `https://www.ntsb.gov/investigations/AccidentReports/Pages/aviation.aspx`

Table 2.3: Aircraft maintenance: the ABC check system (adapted from [3]).

Check	**Average frequency**
Transit	Aircraft grounded for more than 4 hours.
A	500 flight hours or 2 months.
B	1100 flight hours or 6 months.
C	4000 flight hours or 20 months.
D	25000 flight hours or 6 years.
Check	**Description**
Transit	Visual inspection and search for obvious damage and deterioration.
A	- Performed at specialized maintenance station. - Special tooling required.
B	- Slightly more detailed check of components and systems. - It does not involve detailed disassembly.
C	- Include A and B checks. - It requires a thorough visual inspection of specified areas, components and systems as well as operational or functional checks. - This is an extensive check of individual systems and components for serviceability and function. - It removes aircraft from service for 3-5 days.
D	- Include A,B and C checks. - It is an intense inspection of the structure for evidence of corrosion, structural deformation, cracking, and other signs of deterioration or distress. - It includes detailed visual and other non-destructive test inspections of the aircraft structure, alongside extensive disassembly to gain access for inspection. - It removes the airplane from service for at least 20 days.

Nevertheless, here, we will give examples from the literature on modeling of critical sub-systems such as aircraft engines, airframe, actuators, landing gear, and avionics.

2.3.1 Engines

Several factors can lead to engine degradation, with the most commonly reported mechanisms being associated to blade erosion, fouling, and debris deposit [121]. Saxena et al. [122] described how damage propagation could be modeled within the modules of aircraft gas turbine engines. Response surfaces of engine sensors are generated using a thermo-dynamical simulation model for the engine. Then, a health index is defined as the minimum of several superimposed operational margins at any given time instant and the failure criterion is reached when health index reaches zero. Xu et al. [123] presented a fusion prognostics framework based on sensor data, aiming to increase aircraft engine's condition forecasting accuracy. The proposed framework was employed to predict the remaining useful life of an aircraft gas turbine engine based on sensor data providing more accurate and robust estimates than any of the considered single prognostics methods.

Other interesting applications involving the use of machine learning techniques such as, support vector machine frameworks for engine health state classification can be found in [124, 125, 121, 126], while applications based on neural networks are presented in [127, 128].

2.3.2 Airframe

Although there might be many reasons associated with failure different airframe components (including fatigue, corrosion, overload, wear, abrasion, erosion, buckling, flutter), in terms of prognosis, the two most important forms are fatigue and corrosion [129, 130]. The combined effect of corrosion and fatigue is also very important [131, 132, 133].

Duquesnay et al. [134] investigated the growth behaviour of fatigue cracks initiated at corrosion pits of 7075-T6511 aluminium alloy coupons subjected to aircraft loading spectrum. The authors founded that the depth of the corrosion pit was a suitable parameter for characterizing the corrosion damage and fatigue life prediction. Additionally, they suggested that the size of the deepest corrosion pit in the area of corrosion damage on an aircraft, to be used as the metric for predicting fatigue life.

Iyyer et al. [135] presented an extensive discussion on crack initiation and crack growth models to aircraft life predictions and fleet management. Authors describe methods used by the US Navy for fleet fatigue life assessment and individual aircraft fatigue tracking and discuss the history of life estimation methods of P-3C aircraft, with brief descriptions of conducted full-scale fatigue test. In addition, limitations of safe-life methods for aging aircraft are discussed, alongside a deterministic total life approach for continued operations beyond crack initiation up to a safe limit using full-scale test as a basis.

In Li et al. [136], a probabilistic model for diagnosis and prognosis based on dynamic Bayesian networks was proposed and evaluated on a fatigue crack growth on the leading edge of an aircraft wing problem. In diagnosis, the dynamic Bayesian network was used to track the evolution of the time-dependent variables and calibrate the time-independent variables; while in prognosis, it was used for probabilistic prediction of crack growth in the future.

Other important modeling aspects to consider when addressing fatigue in civil aviation involves loads estimation [133, 137, 138] and materials (e.g. metal alloys [139, 140] or composite materials [141, 142, 143]).

2.3.3 Actuators and landing gear

Berri et al. [144] proposed a framework for prognosis of aircraft electromechanical actuator for secondary flight controls. Their approach included signal acquisition, fault detection and isolation, and remaining useful life estimation. In order to keep computational cost manageable, they proposed using strategies for signal processing combined with physical models of different fidelity and machine learning techniques.

Byington et al. [145] presented a study on the use of neural networks for prognosis of aircraft actuator components. The framework covered tasks such as feature extraction, data cleaning, classification, information fusion, and prognosis. They successfully demonstrated their approach on F-18 stabilator electro-hydraulic servo valves.

Jacazio and Sorli [146] presented an enhanced particle filter framework for prognosis of electro-mechanical flight controls actuators. They achieved promising results and showed the benefits of their approach as compared to other published methods.

With regards to landing gear, published literature cover issues such as load estimation [147, 148, 149], hard landing [150], sub-components [145, 146] and landing gear design optimization and health monitoring [151, 152, 148, 153, 154].

2.3.4 Avionics

Wilkinson et al. [155] presented an approach to anticipate the onset of failures in electronic equipment in real-time. A model-based technique utilizing embedded life models and environmental information obtained from aircraft mounted sensors is proposed with promising benefits for reducing the number of unexpected failures.

Kirkland et al. [156] proposed an optimum set of metrics through which the performance of avionics assemblies can be monitored. Considerable insight into the relative performance of a wide range of avionics assemblies has been gained through analysis of test parameters and failure information from automated test equipment. The insights gained from the technique has led to cost avoidance by reducing no faults found occurrences on USAF F-16 fleet.

Batzel and Swanson [157] developed prognostic tools to detect the onset of electrical failures in an aircraft power generator, and to predict the generator’s remaining useful life. Focus is given to the rotor circuit since failure mode, effects, and criticality analysis studies indicated as a high priority candidate for condition monitoring. The interested reader is referred to [158, 159, 144, 160] for other applications and sub-components.

There are few important takeaways from this survey. As carefully pointed out, learning methods (statistical learning, classical machine learning, deep learning, etc.) are inevitable present in most applications. However, they are all subject to the *no free lunch theorems*. Therefore, practitioners are likely to still use a host of approaches depending on the application. With regards to the civil aviation industry, we could clearly see the degree of specialization involved in modeling. We illustrated how regulations and safety play a major role in the widespread adoption of prognosis and health management. While the focus is aircraft airworthiness, modeling seems to follow the cost associated with maintaining the specific sub-system (in which case, the aircraft engine dominates). In the following chapters this dissertation will focus on the challenges associated with epistemic uncertainty propagation in the prognosis and health management analysis of airframe components, specifically associated with early-life and wear-out stages of such components.

CHAPTER 3: EARLY-LIFE FAILURES AND SERVICES OF INDUSTRIAL FLEETS

On April 1, 2011, a Boeing 737-300 operating in the Southwest Airlines flight 812 suffered rapid depressurization while cruising, leading to an emergency landing. The National Transportation Safety Board (NTSB) investigation pointed out that the incident was caused by structural failure of the fuselage skin. Evidence of manufacturing errors in the joining fuselage crown skin panels was found (see [161]) accelerating metal fatigue on the panels. This was the second incident with a Boeing 737-300 in a period of two years in which a fuselage skin manufacturing issue lead to an unexpected decompression during a flight (the first was Southwest Airlines flight 2294 in 2009). These incidents are an example of how manufacturing problems can significantly reduce useful life and lead to unexpected failures in large fleets of engineering assets (e.g. airplanes, jet engines, wind turbines, etc.).

In general, unexpected premature asset failures is an issue referred to as infant mortality failures [162]. This issue is usually identified by higher than expected failure rates in early life (short period after deployment). It is a major concern among original equipment manufacturers (OEM) and operators of industrial assets. Early-life failures *(a)* increase the total cost of ownership due to an increase in costs of maintenance, warranty, services, etc., and *(b)* can reduce asset performance and availability. In addition, they can impose difficulties in meeting compliance and regulations standards (as hardware degradation can be a lead cause of safety standard infringements, elevation of noise and emission levels, etc.).

Understanding how the complex interactions between operating conditions and component capability define useful life is a key issue in the services market (ability to comprehend hardware degradation and predict remaining useful life). This gives operators the chance to make decisions that directly impact their financial outcomes. In this chapter we will focus on a numerical experiment in which a probabilistic framework is used to identify and characterize early-life problems in the fleet management scope. In the proposed framework remaining useful life is forecast through a prognosis model composed of two dynamic Bayesian networks: an asset level network that implements component stress-life curves employing a tunable lognormal distribution, and a fleet-level network that accounts for the integration of Palmgren-Miner's rule. Since the implementation is based on component stress-life curve we considered our methodology as a physics-based prognosis model.

We also provide an assessment of mitigation measures on a fleet level. We will present the analysis of the infant mortality problem focusing on answering the following fundamental question: **how do fleet size and the number of failures interact with each other when characterizing an infant mortality problem?** We answer this question using prognosis, uncertainty quantification, reliability and fleet management. The proposed framework is composed of dynamic Bayesian networks used to model fleet unreliability considering the effects of manufacturing problems (due to bad batch of materials). The proposed Bayesian models use two model parameters: material capability degradation, model as a shift in the material nominal stress-life curve (SN curve); and fleet penetration level, i.e. the percentage of units in the fleet plagued by the material capability degradation. One advantage of the approach is that we can leverage loads and environment information without the need of direct damage measurement. Although we point out that our approach offers only a damage estimate (which is only confirmed with inspection).

3.1 Case Study - Infant Mortality in a Fleet of Assets Due to a Bad Batch of Materials

We use a numerical experiment to study how fleet size and number of failures impact the characterization of infant mortality in fleets of assets. This numerical experiment aim at emulating scenarios in which fleets of industrial assets are plagued with a bad batch of materials. In other words, the material is contaminated during manufacturing yielding in a significant loss in its capability. We consider a hypothetical component to be made out of the Al 2024-T3 alloy[1] subjected to alternating loads and assume that initiation cycles dominate fatigue life. This hypothetical component is mission-critical (its failure does not affect directly the safety of asset operation). Due to the nature of the considered manufacturing problem, components made with pristine material and components made with degraded material would be indistinguishable in operation other than by the unexpected failures. Hence, repairment of plagued components is not possible, although the components could be replaced. Therefore, the numerical experiment focuses on fleets of different sizes plagued with contaminated components that can not be detected based solely on operational performance (i.e., yielding unexpected early life failures).

We use readily available S-N curves commonly found in material handbooks [1] to model low cycle fatigue life at different average and alternating stress levels. Then, we mimic problems with manufacturing (bad batch of materials) by degrading the S-N curves. We designed two missions and two mission mixes to emulate variation due to customer behavior. Finally, we simulate different fleet sizes to understand how failure observations affect overall fleet reliability and detection of emerging issues.

[1] Al 2024-T3 is commonly used in aircraft fuselage, flaps, trim tabs, servo tabs and control surfaces.

3.1.1 Damage Accumulation at the Component Level

We used the readily available S-N curves illustrated in Figure 3.1 and the following suggested fatigue life model as a function of equivalent stress [1]:

$$
\begin{gathered}
N_f \sim logNormal\left(\mu_{Nf},\ \sigma_{Nf}\right) \\
\mu_{Nf} = \theta_1 \log\left(S_{eq} + \theta_2\right) + \theta_3 \\
S_{eq} = \left(S_{max}(1 - (2\,S_{mean} - S_{max}))^{\theta_4}\right), \textit{ and} \\
\theta_1 = -3.33,\ \theta_2 = -12.3,\ \theta_3 = 9.2,\ \theta_4 = 0.68,\ \theta_5 = \sigma_{Nf} = 0.89, \textit{ for } S_{eq} + \theta_2 > 0
\end{gathered}
\tag{3.1}
$$

where:

- N_f is the fatigue life,
- μ_{Nf} and σ_{Nf} are the parameters of the fatigue life lognormal distribution,
- S_{eq} is the equivalent stress of a given load cycle,
- S_{mean} and S_{max} are the mean and maximum stress of a given load cycle, and
- θ_1 to θ_5 are model parameters.

Damage is accumulated following Palmgren-Miner's rule:

$$
D = \sum \Delta d_i = \sum \frac{n_i}{N_f^{(i)}} \tag{3.2}
$$

where:

- D is the damage accumulated throughout the life of the component

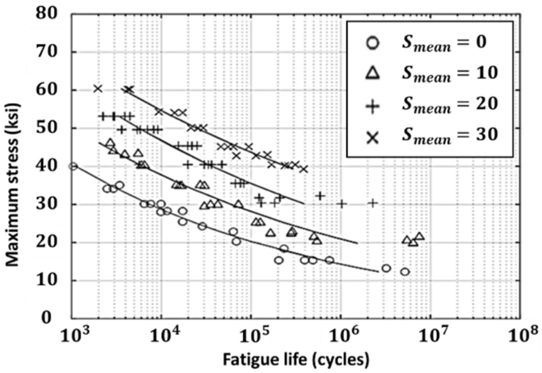


Figure 3.1: S-N curves for the Al 2024-T3 alloy (adapted from [1]).

- Δd_i is the damage accumulated by running n_i cycles at the ith load level
- n_i is the number of cycles run at the ith load level (uniquely defined by mean and maximum stress).
- $N_f^{(i)}$ is the fatigue life at the ith load level, and
- the threshold for end of life is $d_{TH} = 1$.

In many studies, fatigue lives of metal materials are generally assumed to follow either lognormal or Weibull distributions [163, 164]. In this chapter we assume fatigue life follows a lognormal distribution. Since fatigue life $N_f^{(i)}$ follows a lognormal distribution, D is a random variable with no closed-form expression for its probability density function. However, considering that all Δd_i

have the same variance, then, D can be approximated by:

$$
\begin{aligned}
D \;\sim\; logNormal\left(\mu_D,\;\sigma_D\right) \\
\sigma_D^2 = \ln\left[(e^{\sigma_{N_f}^2} - 1)\frac{\sum e^{2\mu_{\Delta d_i}}}{(\sum e^{\mu_{\Delta d_i}})^2} + 1\right] \\
\mu_D = \ln\left(\sum e^{\mu_{\Delta d_i}}\right) + \frac{\sigma_{N_f}^2}{2} - \frac{\sigma_D^2}{2}, \textit{ and} \\
\Delta d_i \;\sim\; logNormal\left(-\mu_{N_f^{(i)}} + \ln(n_i),\;\sigma_{Nf}\right)
\end{aligned}
\tag{3.3}
$$

Since damage is accumulated after each mission, for a given component, the number of missions to failure (MTF) is a random variable with cumulative density function defined by $F_{MTF}\left(m\right) = \Pr\left[MTF \leq m\right] \; = \Pr\left[D_i^{(m)} \geq 1\right]$, where $D_i^{(m)}$is the damage accumulated up to m missions of component *i*. This implies that *ith* component reliability $R_i(m)$ (probability to perform below the threshold) and unreliability $Q_i(m)$ (probability to perform above the threshold) at mission m are given by:

$$
\begin{aligned}
R_i\left(m\right) = 1 - \Pr\left[D_i^{(m)} \geq 1\right], \textit{ and} \\
Q_i\left(m\right) = 1 - R_i\left(m\right) = \Pr\left[D_i^{(m)} \geq 1\right]
\end{aligned}
\tag{3.4}
$$

We designed the two missions shown in Figure 3.2. At the end of each mission, the accumulated damage is distributed around 2.63×10^{-4} and around 6.55×10^{-5} for mission #1 and #2, respectively. The $50^t h$ percentile of fatigue life is approximately 3,800 and 15,260 missions for missions #1 and #2, respectively. At any given mission, the load history can be modeled with the mission index M_{IDX}, which follows a Bernoulli distribution with probability θ_6:

$$
\begin{aligned}
LH^{(m)} = \begin{cases} Mission\;1, & if\; M_{IDX} = 1 \\ Mission\;2, & if\; M_{IDX} = 2 \end{cases}, \textit{ and} \\
M_{IDX} \sim Bernoulli(k = M_{IDX},\; p = \theta_6)
\end{aligned}
\tag{3.5}
$$

where:

- $LH^{(m)}$ is the load history for mission m,
- M_{IDX} is an index that defines which mission to assign, and
- θ_6 is a calibration parameter.

In this contribution, we considered two mission mixes as detailed in Table 3.1, in which mission #1 and mission #2 bound the life distributions for any mission mix. This way, when variations due to both loads in the form of mission mixes and material capability (spread in S-N curves) are considered, the distribution of fatigue life is as illustrated by Figure 3.3.

We emulate degradation in material capability by shifting S-N curves to the left (i.e., for the same stress level, the material has a shorter fatigue life as compared to the nominal material). For simplicity, we can consider that the shift on the S-N curves is only affecting its mean prediction and not acting on the curve variance. Thus, material degradation can be formulated as a shift on the mean value of the damage *D* lognormal distribution given in Eq. 3.3. In this contribution, the degradation shift is modeled through the introduction of the model parameter θ_7 as illustrated by Eq. 3.6. As formulated θ_7 represents percentage-wise the level of knockdown due to the material capability loss. This way, we model relatively the large deviation caused by problems during manufacturing (such as problems in surface treatment and/or microstructure).

$$D_{\mathit{Debit}} \sim logNormal\left((1{-}\theta_7)\mu_D,\ \sigma_D\right) \tag{3.6}$$

where θ_7 is a calibration parameter (defining the degradation in material capability).

Figure 3.4 illustrates the effects of considered material capability degradation on the fatigue life distribution for the aggressive mission mix. At the highest degradation considered (20%), the median of missions to failure can be reduced from 9230 to 860 missions.

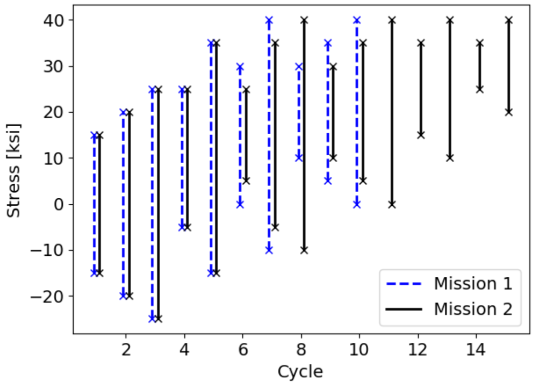


Figure 3.2: Alternating stress levels (S_{Min} to S_{Max}) for the two designed missions.

Table 3.1: Mission mix formulation. Every asset in the fleet is expected to operate at or between, or even alternating between, aggressive and mild mission mixes.

	Aggressive	**Mild**
Mission 1	50%	15%
Mission 2	50%	85%
50^{th} *prctl* N_f	9230	13360

3.1.2 *Fleet Commissioning, Reliability, and Failure Observations*

Large fleets of assets are usually commissioned over a period of time (as production follows a backlog of orders, commissioning ramps up for a while before it starts to decline). Commissioning schedule determines the number of units running (and as a consequence, it impacts the number of failure observations). In this study, we arbitrarily model commissioning time through a truncated Gaussian distribution (as illustrated by Figure 3.5), in which assumed commissioning time after product launch $T_C \sim N(\mu = 4.5,\ \sigma = 2.625)$ and $0 \leq T_C \leq 10$. With the considered assumptions

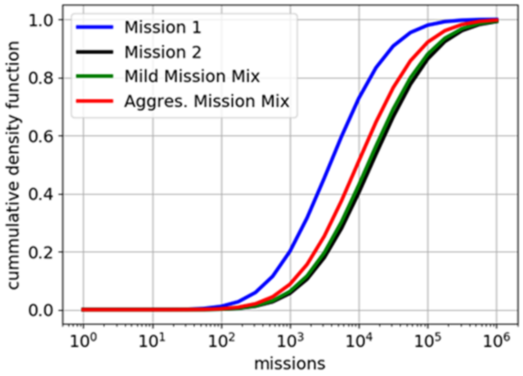


Figure 3.3: Fatigue life distribution in terms of missions to failure considering both load (mission mix) and material capability variations (spread in S-N curve).

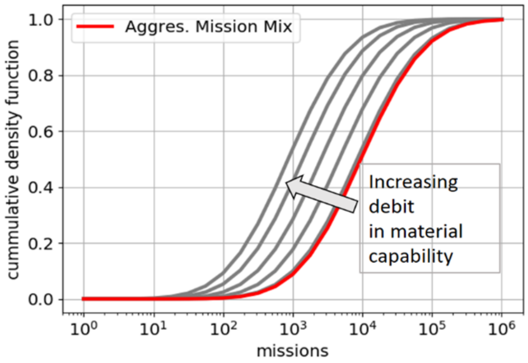


Figure 3.4: Fatigue life distribution for aggressive mission mix and different levels of material capability.

the expected fleet size in the 10^{th} year is 10,000 units. In real life, this distribution is first estimated based on market analysis and can be updated as units are sold and commissioned. Different commissioning time makes the units across the fleet to have different accumulated service lives (and damage, as a consequence).

After commissioning, we assume that each unit runs one mission per day. Integration of asset unreliability, for pristine and material with degradation in capability, up to fleet reliability (i.e. the

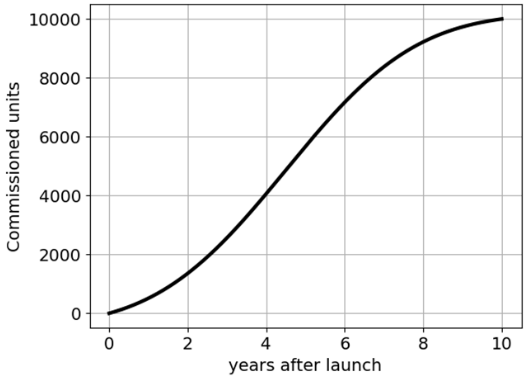


Figure 3.5: Commissioned units over time.

complementary set of the accumulated number of failures in the fleet $(1 - \frac{N_{fail}}{N_{Fleet}})$) is straightforward:

$$\begin{aligned} Q^{(t)}_{Pristine} &= \frac{\sum_{i=1}^{N_{Fleet}} Q_i^{(t-t_i^{comm.})}\Big|_{Pristine}}{N_{Fleet}}, \textit{ and} \\ Q^{(t)}_{Debit} &= \frac{\sum_{i=1}^{N_{Fleet}} Q_i^{(t-t_i^{comm.})}\Big|_{Debit}}{N_{Fleet}} \end{aligned} \tag{3.7}$$

where:

- $Q^{(t)}_{Pristine}$ and $Q^{(t)}_{Debit}$ are the fleet unreliability due to material with the nominal capability and material with certain capability degradation, respectively, both at time t. $Q^{(t)}_{Debit}$ is not related to failures yielded by materials with a specific capability degradation (i.e. to a specific θ_7), but rather accounts for all observed failures due to the material with a non-nominal capability (i.e. degraded material).
- $Q_i^{(t-t_i^{comm.})}\Big|_{Pristine}$ and $Q_i^{(t-t_i^{comm.})}\Big|_{Debit}$ are the unreliability of component i assuming it is made of pristine material and material with certain capability degradation, respectively, both at time t greater than the component commissioning time ($t_i^{comm.}$). $Q_i^{(t-t_i^{comm.})}\Big|_{Pristine}$ and $Q_i^{(t-t_i^{comm.})}\Big|_{Debit}$ only start being computed if the unit is already commissioned (otherwise

both are null).

- N_{Fleet} is the final fleet size.

As a consequence, at time t, fleet unreliability $Q^{(t)}_{Fleet}$ is:

$$Q^{(t)}_{Fleet} = \theta_8 \times Q^{(t)}_{Debit} + (1 - \theta_8) \times Q^{(t)}_{Pristine} \tag{3.8}$$

where θ_8 is a model parameter that defines the penetration of units in the fleet (in terms of a fraction of the fleet) made of a material with a certain degradation level in capability.

With fleet unreliability we can predict the number of failures at any year after product launch by using the binomial distribution to model the number of failures:

$$\begin{gathered} f_{N_{Fail}} \left(n = N_{Fail};\ p = Q^{(t)}_{Fleet},\ \ N = N_{Fleet} \right) = \\ \begin{pmatrix} N_{Fleet} \\ N_{Fail} \end{pmatrix} \times {Q^{(t)}_{Fleet}}^{N_{Fail}} \times \left(1 - Q^{(t)}_{Fleet} \right)^{N_{Fleet} - N_{Fail}} \end{gathered} \tag{3.9}$$

where:

- $f_{N_{Fail}}\left(.\right)$is the probability density function of the binomial distribution,
- N_{Fail} is the accumulated number of observed failures up to year y,
- $Q^{(t)}_{Fleet}$ is the fleet unreliability at year y, and
- N_{Fleet} is the fleet size at year y.

Conversely, given a number of observed failures, we can estimate the model parameters through the Bayes rule[2]:

$$f_{Post}\left(\boldsymbol{\theta} \mid N_{Fail},\ \ N_{Fleet}\right) \propto f_{N_{Fail}}\left(N_{Fail} \mid Q_{Fleet}^{(t)}\ ,\ N_{Fleet}\right)\ f_{Prior}(\boldsymbol{\theta}) \tag{3.10}$$

where $Q_{Fleet}^{(t)}$ is a function of $\boldsymbol{\theta}$.

3.1.3 Fleet Management

Our physics-based prognosis model for fleet management is composed of two Bayesian networks, one for the asset reliability and another one for the fleet unreliability. Figure 3.6 shows the asset-specific dynamic Bayesian network that relates material properties and loads with damage accumulation, where:

- LH corresponds to load history,
- S_{eq} is the equivalent stress of a given load cycle,
- μ_{Nf} and σ_{Nf} are parameters of the fatigue life lognormal distribution,
- $\Delta d_i^{(t)}$ is damage accumulated after running through $LH^{(t)}$,
- $D_i^{(t)}$is the damage accumulated by component *i* up to t,
- $\theta_{1...4}$ and θ_5 are parameters defining material properties,
- θ_6 is the parameter defining the mission mix,

[2] We write the posterior in its proportional form as it is the way it is implemented in most numerical integration methods (such as Markov chain Monte Carlo). Also, computing the binomial coefficient $\begin{pmatrix} N_{Fleet} \\ N_{Fail} \end{pmatrix}$ is not necessary when estimating fleet unreliability given a number of failures for a fleet. This is usually cumbersome and can cause numerical ill-conditioning depending on N_{Fleet} and N_{Fail} .

- θ_7 is the parameter defining degradation in material capability.

Superscripts $(t-2)$, $(t-1)$, and (t) indicate the timestamps in which inference/estimation is performed. Load histories for missions 1 and 2 are defined in Fig. 3.2.

As formulated, the general case for the asset-specific model yields in a Bayesian network with seven model parameters, θ_1 to θ_7, as illustrated by Fig. 3.6. Albeit, the problem treatment may significantly vary from the general case to a simplified model, depending on the application some of the model parameters can be freeze to a given value yielding in a reduced number of model parameters to be estimated. Here, we will consider material properties to be known values, hence parameters θ_1 to θ_5 are freeze to the values given by [1], as shown in Eq. 3.1. In the mentioned numerical application we also consider the mission mixes to be fixed throughout asset life, thus, θ_6 will also be fixed at the values shown in Table 3.1. Therefore, with respect to the asset-specific model, for the considered application, only θ_7, which defines the degradation in material capability, will be calibrated with failure observations. Again, it worth highlighting that the problem treatment and uncertainty predictions presented for this simplified model, may not hold for the general case.

Figure 3.7 shows the fleet dynamic Bayesian networks, where $\boldsymbol{D}^{(t)}_{1 \ldots N_{Fleet}}\Big|_{Debit}$ and $\boldsymbol{D}^{(t)}_{1 \ldots N_{Fleet}}\Big|_{Pristine}$ are vectors of damage accumulated up to t for the N_{Fleet} assets in the fleet with and without degradation in material capabilities, respectively, $\boldsymbol{Q}^{(t)}_{1 \ldots N_{Fleet}}\Big|_{Debit}$ and $\boldsymbol{Q}^{(t)}_{1 \ldots N_{Fleet}}\Big|_{Pristine}$ are vectors of unreliability at time t for the N_{Fleet} assets in the fleet with and without degradation in material capabilities, respectively. $Q^{(t)}_{Debit}$ and $Q^{(t)}_{Pristine}$ are the fleet unreliability at time t with and without degradation in material capabilities, respectively. θ_8 is one more calibration parameter related to the fraction of the fleet made of material with a certain degradation level in capability, i.e., the fleet penetration of material with low capability, that will also be calibrated with failure observations. In the numerical application discussed in this chapter, both the simplified asset-specific model and the fleet dynamic Bayesian network model will be used to make inference about θ_7 and θ_8, as well as estimate and forecast fleet unreliability $Q^{(t)}_{Fleet}$.

The number of failures at time t, $N^{(t)}_{Fail}$, can be modeled through a binomial distribution where N_{Fleet} (fleet size) is the number of trials and $Q^{(t)}_{Fleet}$ is the probability associated with each trial:

$$N^{(t)}_{Fail} \sim Bin(Q^{(t)}_{Fleet},\ \ N_{Fleet}) \tag{3.11}$$

This model can be applied in simple estimation at current time or even in forecast (when $Q^{(t)}_{Fleet}$ and N_{Fleet} can be forecasted).

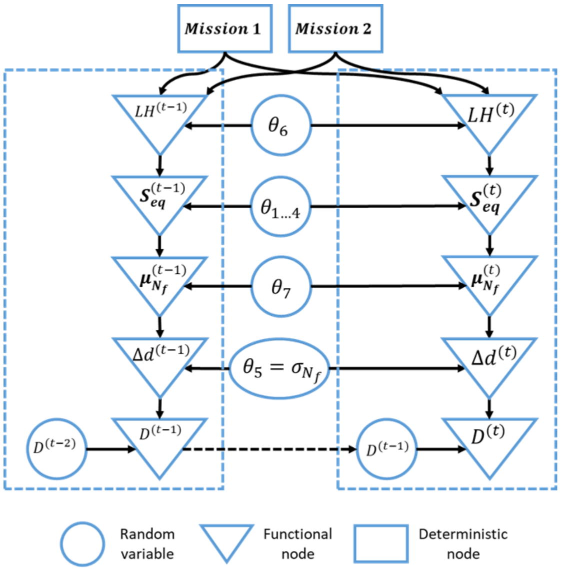


Figure 3.6: Asset-specific dynamic Bayesian network.

In this contribution, we study the effects of fleet sizes in the ability to forecast the number of failures and its implication to fleet management. From Eqs. 3.10 and (3.11), it is expected that inference performed with data from small fleets will result in large uncertainty about the model parameters. This is a direct consequence of the reduced number of observations (in the binomial distribution, estimation of small probabilities is accompanied by large uncertainties when the number of observations is small, see [165] for further details). This is problematic as the model parameters

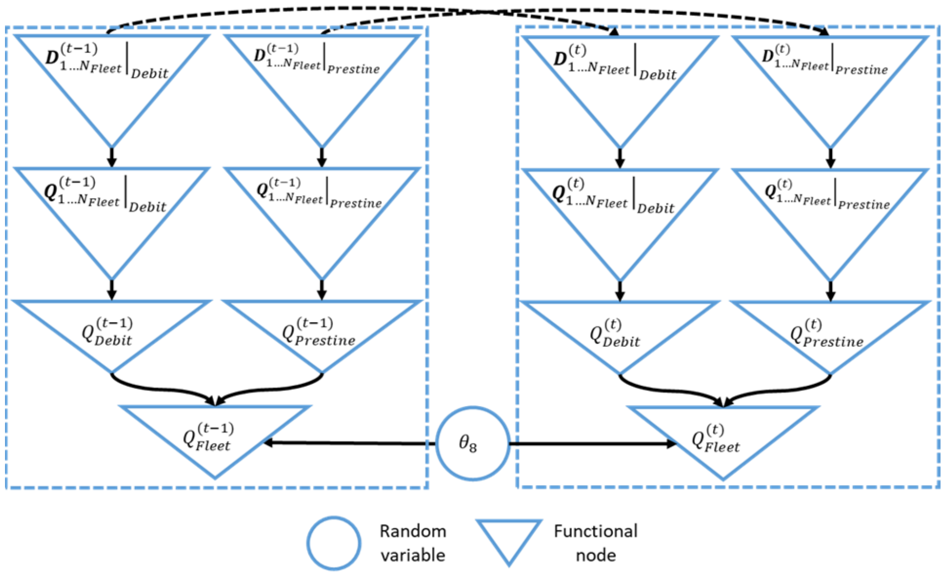

$D^{(t-1)}_{1\ldots N_{Fleet}}\big|_{Debit}$
$D^{(t-1)}_{1\ldots N_{Fleet}}\big|_{Prestine}$
$D^{(t)}_{1\ldots N_{Fleet}}\big|_{Debit}$
$D^{(t)}_{1\ldots N_{Fleet}}\big|_{Prestine}$
$Q^{(t-1)}_{1\ldots N_{Fleet}}\big|_{Debit}$
$Q^{(t-1)}_{1\ldots N_{Fleet}}\big|_{Prestine}$
$Q^{(t)}_{1\ldots N_{Fleet}}\big|_{Debit}$
$Q^{(t)}_{1\ldots N_{Fleet}}\big|_{Prestine}$
$Q^{(t-1)}_{Debit}$
$Q^{(t-1)}_{Prestine}$
$Q^{(t)}_{Debit}$
$Q^{(t)}_{Prestine}$
$Q^{(t-1)}_{Fleet}$
θ_8
$Q^{(t)}_{Fleet}$
Random variable
Functional node

Figure 3.7: Fleet dynamic Bayesian networks. Superscripts *(t-1)*, and *(t)* indicate the time stamps in which inference/estimation is performed.

are then used to estimate and forecast the number of future failures. Large uncertainty in the number of future failures drives conservativeness in the way operators manage their fleets. On the other hand, operators of large fleets, large services and maintenance companies, and original equipment manufacturers tend to observe a large number of failures and should be able to benefit from it in terms of uncertainty quantification. Regardless of the fleet size, effective fleet management asks for a continuous model update as new information is made available throughout service lives (including revisiting the assumptions about model form, failure modes, etc.).

The estimated number of failures can be used to assess the ‘‘risk’’ associated with the forecast. One very straightforward measure is the uncertainty about the forecasted number of failures (i.e., companies have to be prepared to absorb that variation from a financial perspective). There are a number of ways to quantify variation in the number of failures. One can simply use the standard deviation, which might not be convenient given the asymmetric nature of the N_{Fail} estimator.

Alternatively, the difference between 97.5 and 2.5 percentiles of the forecasted number of failures can also be used. Operators of small fleets can use this range to support the decision to either self-perform or buy a contractual service agreement from a third party company. Operator of small fleets tend to have difficulties in absorbing large variations in the forecasted number of failures due to liability associated with it (both in terms of inventory, labor, etc., as well as in terms of loss of revenue). This can make operator of small fleets be overzealous and perform excessive inspection and services in the hope to prevent costly maintenance or catch serious problems when units are still under manufacturer warranty (minimizing the impact of unscheduled removals and cost of repairs/replacements). For large fleet operators, the problem shifts from unexpected downtime to an excessive number of costly maintenance and contractual obligations regarding availability and reliability.

We compare results from a small and a large fleet to mimic the operator of a small fleet vs large service provider dynamic. For this portion of the case study, we assume that the large service provider is more likely to provide an unbiased and accurate estimation of the number of failures. If that is the case, the difference in the forecasted number of failures can help us judge whether self-performing is a good decision or not. Mathematically:

$$FExp = N_{Fail@Small}^{Small} - N_{Fail@Small}^{Large} \tag{3.12}$$

where $N_{Fail@Small}^{Large}$ and $N_{Fail@Small}^{Small}$ are the estimated/forecasted number of failures on the small fleet coming from the large service provider and the small fleet operator models, respectively.

The fleet exposure index ($FExp$) becomes an indicator of whether the small fleet operator is likely to save or lose money by self-performing services and maintenance:

- $FExp > 0$: the operator of a small fleet over predict failures, which drives the allocation of more resources than needed. In other words, the behavior is conservative and it translates in savings due to avoided unscheduled maintenance, reduced downtime, etc.

- $FExp < 0$: the operator of a small fleet under predicts failures, which drive allocation of fewer resources than needed (operator loses money due to unscheduled maintenance, downtime, etc).

Again, this assumes that $N_{Fail@Small}^{Large}$ is a more accurate estimator of N_{Fail}^{True}, i.e. the service provider estimates for the failure observations in the small fleet is more likely to be closer to the true number of failures, since as previously mentioned, it is expected that inference performed with data from small fleets will result in large uncertainty about the model parameters (again see [165], for further details). Although N_{Fail}^{True} can be obtained in this numerical example (through Eq. 3.11 since fleet reliability for known loads can be obtained at any point in time), we avoid using it as it is not available in real life though.

Due to the nature of the considered numerical experiment and for simplicity we opt to focus our analysis on asset derating (i.e. fleet recommissioning). It is important to point out that this is by no means the only possible mitigation measure and depending on the nature of the addressed problem other measures could be considered (e.g burn-in, as usually applied in the electronics industry). Also, the proposed numerical experiment is an academic simplification, in a real-life application many other aspects (e.g. asset decommissioning) would influence observed failures and the performance of the proposed mitigation measures.

3.2 Results and Discussions

3.2.1 Numerical Example

In order to evaluate the effect of fleet size in the number of failure observations, we defined two distinct fleets:

- a large fleet of 10,000 units: emulating an original equipment manufacturer or a large service provider, and
- a small fleet of 1,000 units: emulating a small fleet operator. These units come from the larger 10,000 unit fleet, which also means that the large fleet operator has visibility into what happens with this small fleet.

Both fleets are plagued with a material degradation level of 15%. However, to make things more interesting, we distributed the failures across the fleet such that the small fleet operator has a penetration of 20% of units plagued with material of inferior capability (i.e., 200 out of 1,000 units are plagued), while the larger fleet has an overall 10% penetration (i.e., 1,000 out of 10,000 units are plagued). Thus, in this example we considered that model parameter θ_7 and θ_8 are constant in time (as they represent debit and fraction of the fleet plagued). In real life this assumption may not hold if there are changes in manufacturing process over time (affecting the level of material degradation, θ_7, and the proportion of plagued assets in the fleet, θ_8).

The implications in fatigue life distribution are shown in Figure 3.8, in which load history is exclusively coming from the aggressive mission mix. There is a considerable shift in fatigue life if the entire fleet is plagued with a material of poor capability. Nevertheless, given the 10% and 20% penetration levels for the large and small fleets, respectively, the effect mostly manifested in the lower tail of fatigue life distribution.

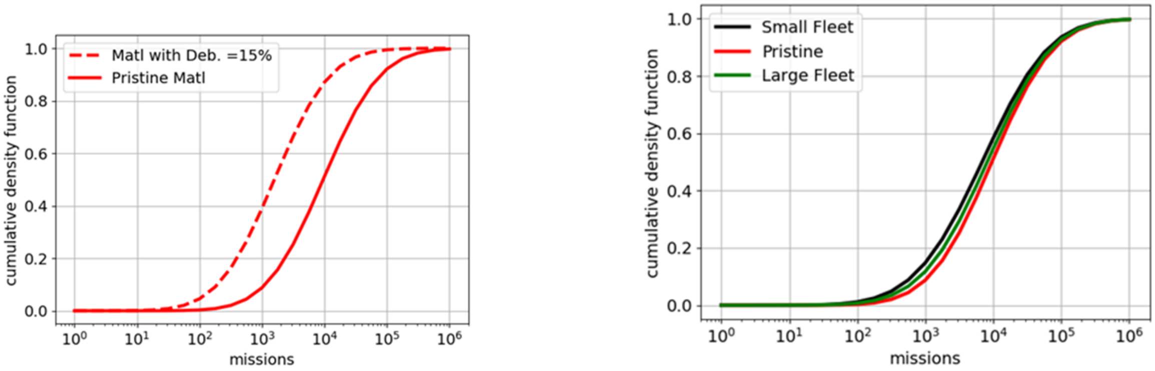
As previously discussed, commissioning has an effect on fleet unreliability as the fleet grows bigger with asynchronous aging. As an illustration, Figure 3.9 shows a comparison between fleet unreliability over time with and without the effect of commissioning. The drastic reduction in unreliability values results in a delay in rising failures observations. Most industrial engineering assets (the focus of this paper) are commissioned over a period of time. In the remainder of this section, we will discuss the results following the commissioning previously detailed.

With the fleet unreliability over time, we can forecast the number of failures. Figure 3.10 highlights the contribution of each subpopulation by material type (pristine and with degradation in capability) in the resulting failure observations. Besides the obvious penetration of material with poor capability (10% versus 20% for the large and small fleet, respectively), commissioning also affects the relative contribution of each material to the number of failures. Early on, most failures come from components plagued by material with poor capability. Over time, the unreliability for pristine material increases (see Figure 3.9), and the relative contribution of each population starts to change. Around the 3^{rd} year after deployment, at least for the large fleet, failures are dominantly coming from components made out of pristine material (although the contribution from the subpopulation with plagued material is still substantial).

3.2.2 Bayesian Networks Calibration

At the 3^{rd} year after deployment, we assume the following number of failure observations:

- Small fleet: 17 failures (lower tail of the predicted number of failures).
- Large fleet: 127 failures (roughly 50^{th} percentile of the predicted number of failures). Obviously, the small fleet failures are contained in this set.



(a) Impact of material degradation in fatigue life when the entire fleet is plagued.

(b) Pristine (expected) and actual (large and small) fleet unreliability at aggressive mission mix.

Figure 3.8: Fleet characterization of fatigue life distribution (without the effects of commissioning and load history is exclusively coming from the aggressive mission mix).

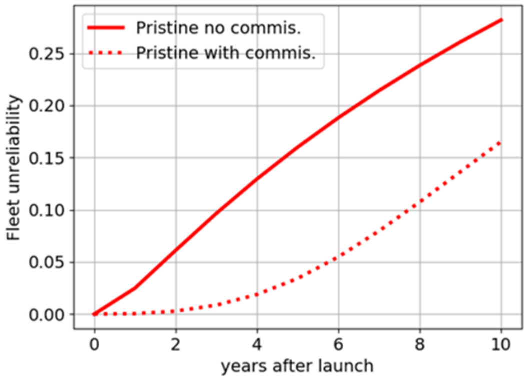
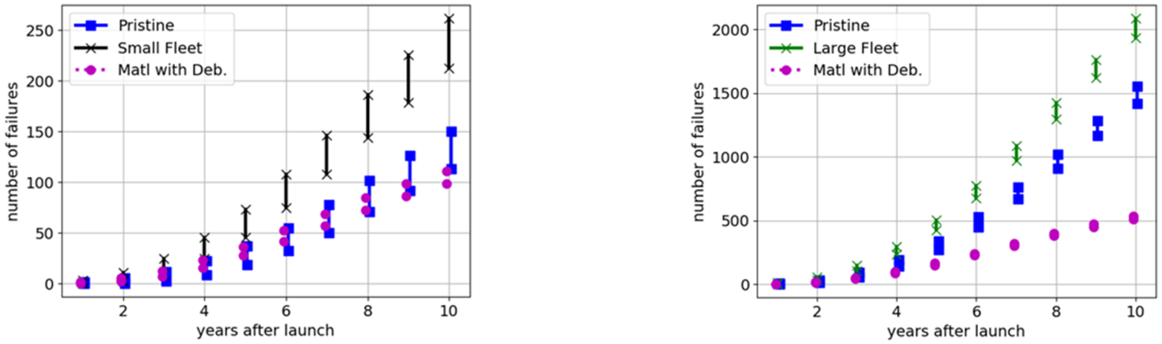


Figure 3.9: Commissioning effect in overall fleet unreliability.

We use these failure observations and fleet unreliability (from known load histories) to calibrate:

- θ_7: degradation in material capability with a uniform prior between 1% to 30%, and
- θ_8: penetration of units with poor material capability in the fleet with a uniform prior between 0.01% and 20%.



(a) Small fleet.

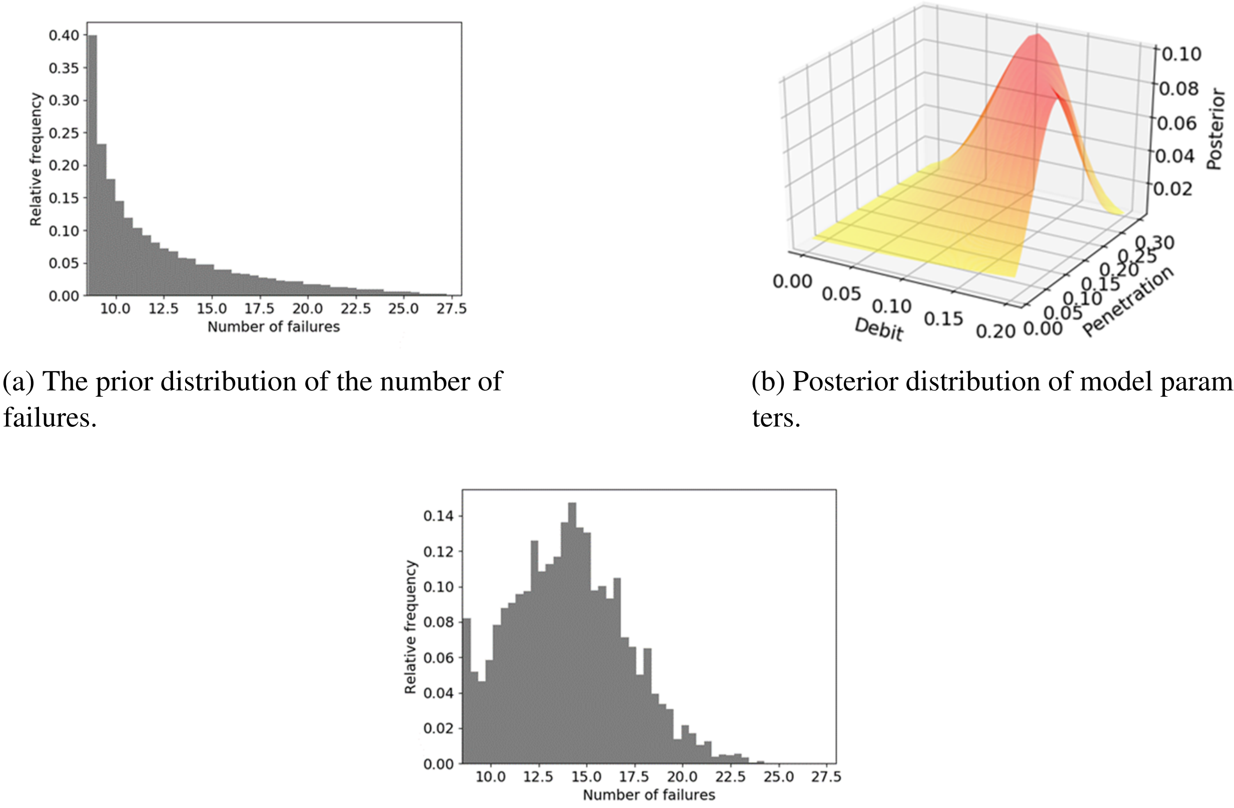
(b) Large fleet.

Figure 3.10: Estimated number of failures in each fleet (error bars represent the 95% prediction intervals).

Figures 3.11 and 3.12 detail the calibration results with regards to both model parameters and the estimated number of failures for the small and large fleets, respectively. Even for the small fleet operator, there is considerable uncertainty reduction and failure estimates are much improved as compared with the uniform priors (see Table 3.2, in which for instance we can observe the reduction in the 95% prediction interval, from an interval range of $(8, 22)$ in the small fleet prior, to $(9, 20)$ in its updated posterior).

As we mentioned before, the large fleet operator has full visibility into what happens with the small fleet. In this numerical example, the relative number of failures with respect to the fleet size can be used to map the posterior distribution of the number of failures at the large fleet into the small fleet, as illustrated by Figure 3.13 and summarized in Table 3.2. When compared to Figure 3.11c, the uncertainty in Figure 3.13 is much smaller and clearly attributed to the much richer information available at the large fleet level.

The models with updated model parameters can be used to forecast the number of failures over time. Figure 3.14 shows how these forecasted values look like for the small fleet. In the 3^{rd} year, both models are unbiased. The uncertainty in the posterior distribution of model parameters for the small fleet model, Figure 3.11b, is larger than the one for the large fleet model, 3.12b. The result is the larger and ever-increasing uncertainty that the small fleet model exhibits when compared to the large fleet model.

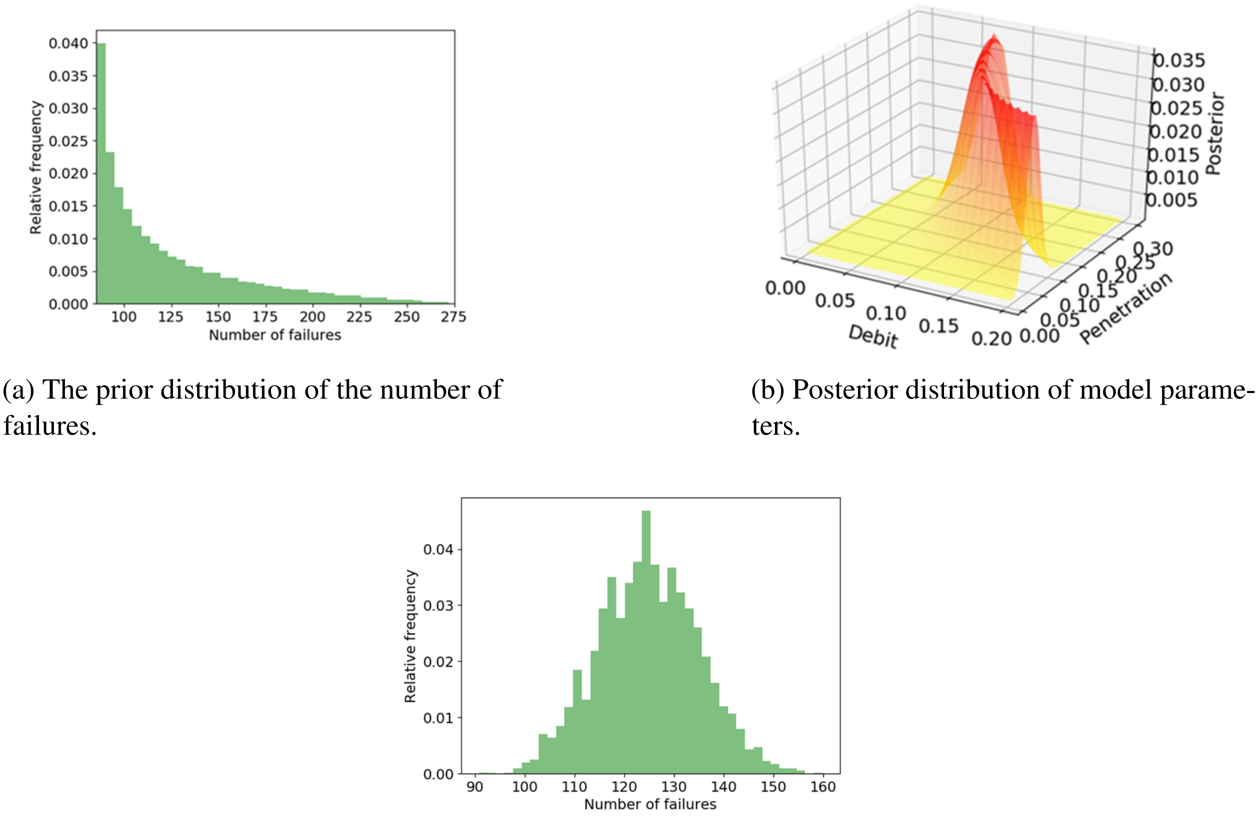


(a) The prior distribution of the number of failures.

(b) Posterior distribution of model parameters.

(c) Posterior distribution of the number of failures.

Figure 3.11: Calibration results for the small fleet. Failure observations in the 3^{rd} year after deployment and uniform priors feed the Bayesian update.



(a) The prior distribution of the number of failures.

(b) Posterior distribution of model parameters.

(c) Posterior distribution of the number of failures.

Figure 3.12: Calibration results for the large fleet. Failure observations in the 3^{rd} year after deployment and uniform priors feed the Bayesian update.

3.2.3 Mitigation Assessment

Once the infant mortality issue is quantified, operators undergo a number of mitigation actions to reduce costs associated with unscheduled maintenance, asset unavailability, etc. In this section We opt to focus on fleet recommissioning (changing the mission mix from aggressive to mild). This approach can be costly, as mild mission mixes are usually associated with some loss in performance or productivity. Figure 3.15 illustrates the estimated/forecasted fleet unreliability and forecasted

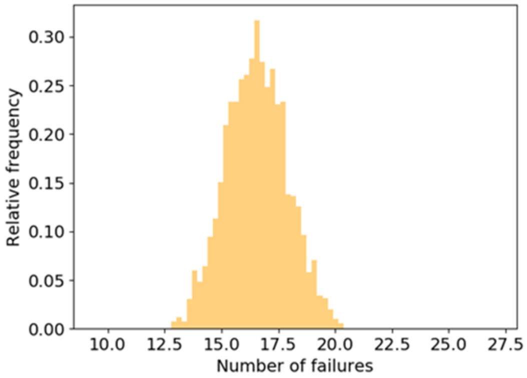


Figure 3.13: Posterior distribution of the number of failures in the 3^{rd} year after deployment for the small fleet as estimated by the large fleet operator.

Table 3.2: Estimates for the number of failures in the 3^{rd} year after deployment for small fleet. The small fleet observed 17 failures.

Model	**Percentile**		
	2.5	50	97.5
Design intent *	3	8	15
Small fleet prior	8	10	22
Small fleet posterior	9	14	20
Large fleet posterior	14	17	20

* Number of failures considering only pristine material.

number of failures over time for the small fleet. Figure 3.15a shows that after the entire small fleet is recommissioned from the aggressive to the mild mission mix, the estimated fleet unreliability with the updated model falls between the estimates for the entirely pristine fleet and the actual fleet composition both operating at the aggressive mission mix. This means that although there is a significant improvement in unreliability, the levels are still above design intent. Interestingly, the distribution in forecasted fleet unreliability might still be useful for estimating the number of failures. Figure 3.15b shows the forecasted number failures coming out of the unreliability estimates of 3.15a. The aggressive mission mix and pristine material represent the design intent. The estimated penetration and material degradation represent the forecasts if the fleet keeps operating at the aggressive mission mix. Visual comparison between the two cases makes it clear that the number of failures could be potentially much larger than what was intended. Recommissioning the fleet knocks down the number of failures and make the prediction interval overlap with the one from the design intent.

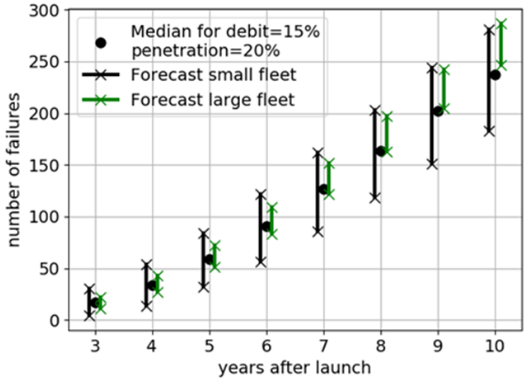
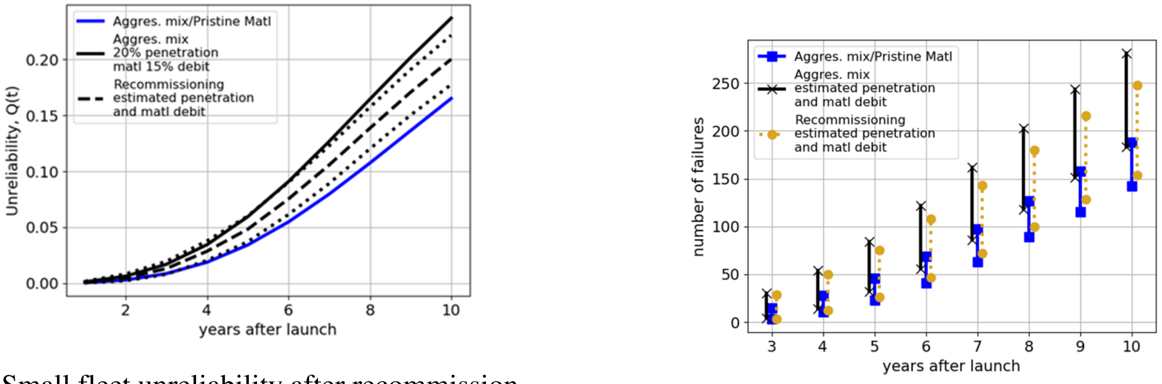


Figure 3.14: The forecasted number of failures for the small fleet as estimated by both small and large fleet models (error bars represent the 95% prediction intervals).

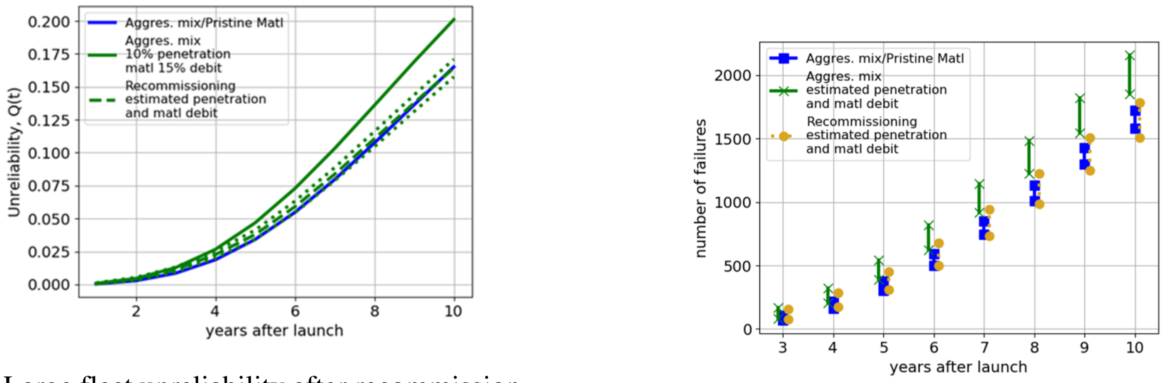


(a) Small fleet unreliability after recommissioning.

(b) Small fleet forecasted number of failures.

Figure 3.15: Small fleet recommissioning. Recommissioning curves show the 50^{th} percentile and the 95% prediction interval, and error bars represent the 95% prediction intervals.

As expected, Figure 3.16 shows that recommissioning as a mitigation measure is much more effective at the large fleet level. Coincidently, the fleet unreliability after recommissioning converges to design intent. Obviously, it comes at the cost of a mild mission mix. In real life, even though the reliability levels would be back to design intent, it might not be practical to recommission the fleet due to performance losses. The final compromise between reliability and performance is a decision that is highly industry dependent (and discussion is outside the scope of this dissertation). Figure 3.16a shows uncertainty levels in fleet unreliability after recommissioning are much smaller than those shown in Figure 3.15a. This has direct implications in the forecasted number of failures, as illustrated by Figure 3.16b, to the point that there is a good overlap between estimated and intended error bars.



(a) Large fleet unreliability after recommissioning.

(b) Large fleet forecasted number of failures.

Figure 3.16: Large fleet recommissioning. Recommissioning curves show the 50^{th} percentile and the 95% prediction interval, and error bars represent the 95% prediction intervals.

Besides recommissioning, the small fleet operator can also consider contracting out services and maintenance from a large service provider as a way to reduce financial exposure due to the upcoming high number of failures. In real life, it is difficult to forecast the costs associated with such an option, as the operator of the small fleet does not know the outcomes of the large fleet operator model (and model form, assumptions, etc. also tend to be unknown). Nevertheless, we can study that in this synthetic example. Figure 3.17 shows the forecasted $FExp$ index as defined in Eq. 3.12 before and after fleet recommissioning for the small fleet. Continuous and dotted lines represent the median and 95% prediction intervals, respectively. The operator of the small fleet is likely to lose money by self-performing services and maintenance when $FExp$ is negative (since unbiased predictions from the operator of the large fleet tend to be larger than the ones from the operator of the small fleet). Conversely, the operator saves money by self-performing when $FExp$ is positive. Although the $FExp$ median is relatively small up to the 5^{th} or 8^{th} year, depending on recommissioning, the uncertainty about it tends to be large and continuously increasing.

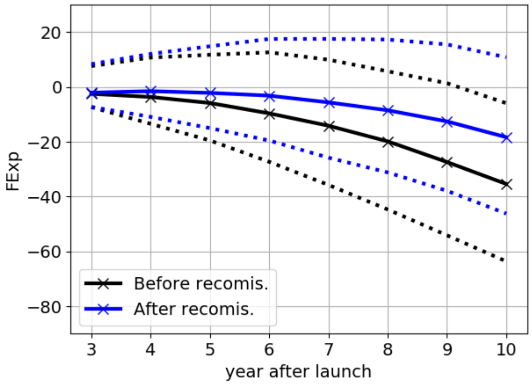


Figure 3.17: Small fleet $FExp$ index (defined by Eq. 3.12) forecast before and after recommissioning.

Another way of looking at the risk associated with self-performing maintenance is through the probability of reward and loss. With the $FExp$ index defined by Eq.3.12, there are three things to keep in mind:

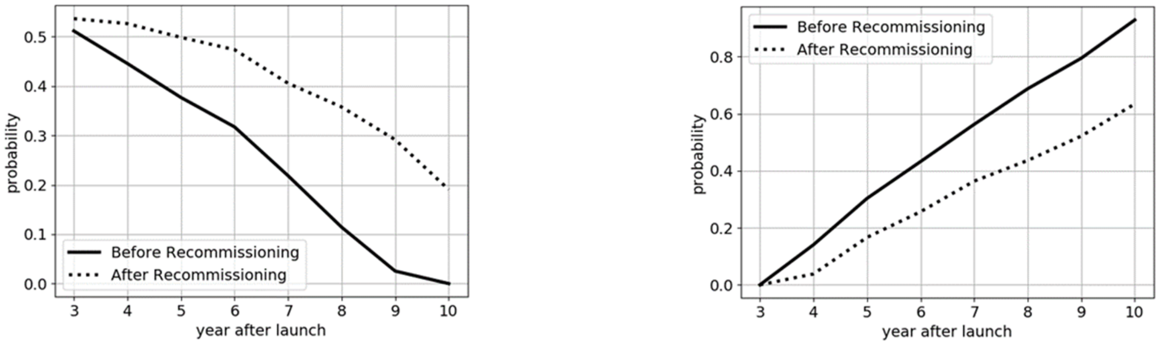
1. when $FExp$ is positive, the operator of the small fleet saves money by self-performing services and maintenance,

2. conversely, when $FExp$ is negative loss of money is more likely; and finally,

3. unbiased number of failure estimates imply that the expected value of $FExp$ is zero.

With that in mind, Figure 3.18a shows that self-performing is reasonable in the short term. For how long it is a reasonable option really depends on the operator attitude towards unaccounted failures. Just to illustrate this concept consider an arbitrary threshold of $Prob\left[FExp \geq 0\right] \geq 0.4$ is imposed (in pratical applications this threshold would be imposed by the small fleet operator depending in its wiliness to handle unexpected failures), then the operator of the small fleet could sustain the aggressive mission mix until almost the end of the 4^{th} year (without having to buy a services and maintenance contract). If the operator decides to recommission the fleet at the third year; then, with the $Prob\left[FExp \geq 0\right] \geq 0.4$ threshold, self-performing is reasonable until the 7^{th} year.

Now, let us assume that the operator of the small fleet is willing to accept underestimating failure of 10 units. Then, Figure 3.18b shows the probability that the operator will have to pay for the extra 10 units (unplanned failures). If again an arbitrary threshold of $Prob\left[FExp \leq -10\right] \leq 0.2$ is imposed, then the operator of the small fleet could sustain the aggressive mission mix until the middle of the 4^{th} year (without having to buy a services and maintenance contract). Switching to a mild mission mix early on, extends that window to the middle of the 5^{th} year.



(a) *Prob[FExp* $\geq$ *0].*

(b) *Prob[FExp* $\leq$ *-10].*

Figure 3.18: Self-performing reward and loss probabilities for the small fleet.

3.3 Summary

In this chapter, we studied early life failures as applied to fleet management. Depending on the scale of the problem, early failures can have a significant impact on the safety, availability, and operational profit of industrial equipment. We used Bayesian networks to model fleet reliability as well as to characterize and quantify a manufacturing-related problem (material capability degradation). We designed a simple numerical experiment where:

- degradation in material capability is used to characterize infant mortality, and
- fleet commissioning is a function of time.

We have studied:

- The effect of degradation in material capability: results confirmed that it dramatically reduces fleet reliability.
- The effect of fleet commissioning over time: results highlighted how it masks the increase in fleet unreliability.
- Fleet size and number of failures interaction and its effects on the infant mortality characterization: in the considered numerical experiment, the proposed framework was able to characterize the percent of the fleet plagued by poor material capability. Nevertheless, results illustrated how the fleet size can hinder the characterization of the problem, especially for small fleet operators.
- Bayesian models for assessing possible mitigation approaches: the proposed framework was able to evaluate fleet recommissioning as a risk mitigation procedure (i.e. derating assets to reduced load levels). Presented results indicate that the proposed model could be used as a

tool for fleet management and aid operators to decide on a proper action course. For example, recommissioning mitigates the loads and therefore slows the failure rate down (at the cost of performance loss).

CHAPTER 4: WEAR-OUT FAILURES AND SERVICES OF INDUSTRIAL FLEETS - A CASE STUDY ON CORROSION-FATIGUE

In this chapter we will focus on the effects of epistemic uncertainties due to partial knowledge of the physics of degradation on the damage accumulation modeling in the wear-out stages of a component service life. Namely, we will present a numerical experiment concerning corrosion-fatigue of an Al 2024-T3 alloy used on panels of aircraft wings.

We present a physics-informed neural network modeling approach for missing physics estimation in cumulative damage models. This hybrid approach is designed to merge physics-informed and data-driven layers within deep neural networks. The result is a cumulative damage model in which physics-informed layers are used to model relatively well understood phenomena and data-driven layers account for hard-to-model physics. A numerical experiment is used to present the main features of the proposed framework. The test problem consists of predicting corrosion-fatigue of an Al 2024-T3 alloy used on panels of aircraft wings. Due to unforeseen variations in operational conditions introduced by the air-carrier, an aircraft fleet mainly operates in coastal routes, significantly increasing its exposure to saline corrosion. These conditions lead to accelerated degradation of the aircraft wing panels due to the combined effects of corrosion and mechanical fatigue. Such corrosive conditions are not accounted for by the fleet prognosis model generating a significant epistemic uncertainty (i.e., a missing physics issue). In this case, we adapted the Euler integration cell discussed in [72] into a hybrid recurrent neural network in which the physics-informed layers implement the well known Walker model for crack propagation, while data-driven layers are trained to compensate the bias in damage accumulation due to the corrosion effects. The physics-informed recurrent neural network is trained using full observation of inputs (far-field loads, stress ratio and a corrosivity index defined per airport) and very limited observation of outputs (crack length at inspection for only a small portion of the fleet). Results show that the proposed hybrid cell is able to learn how to compensate the missing physics of corrosion in the original fatigue model. Predictions from the hybrid model can be used in fleet management, for example, to prioritize inspection across the fleet or forecast ahead of time the number of planes with damage above a given threshold.

4.1 Case Study - The Synthetic Aircraft Fleet Data

Consider a hypothetical control point on the underside of an aircraft wing, as illustrated in Figure 4.1. For simplicity, we consider the aircraft fleet can fly the 10 different flight types detailed in Table 4.1. Each flight type is characterized by a load frequency distribution. Therefore, these flights present different levels of severity in terms of mechanical loads. In order to balance out the exposure of the fleet to the severe flights while attending their demands, airline companies rotate their aircraft through different mission mixes (combinations of route structures mixing different flight types). Here, we considered the 10 arbitrarily designed mission mixes. In our study, each one of the 150 aircraft of the fleet is assigned to only one specific mission mix. Within a given mission mix, flights types are assigned to the aircraft randomly following the probability distribution detailed in Table 4.2. For example, aircraft belonging to mission mix $\#5$ will never fly flight types I and II while they will fly flight types VII and VIII 4.5% and 10.5% of the time, respectively. In other words, when we assign a flight type to an aircraft, we are imposing a loading frequency as detailed in Table 4.1. Based on the load magnitude each loading cycle of the flight can be translated into far-field stress ($\Delta S = S_{max} - S_{min}$) and stress ratio ($R = S_{min}/S_{max}$) values.

Original equipment manufacturers, airline companies, and service providers usually maintain large aircraft fleets (in the range of hundreds to thousands of aircraft). Here, we consider a fleet of 150 aircraft, which are scheduled to fly 8 flights per day. The control points of interest, illustrated by Figure 4.1, is considered to be on panels made of Al 2024-T3 alloy (coefficients for crack propagation in pure air known and shown in Figure 4.2a) with $a_0 = 0.5$ (mm) and $a_{max} = 20$ (mm). Finally, we assume that inspection of this control point is part of the scheduled maintenance. Here, we arbitrarily consider that the first inspection is available for only part of the fleet after 5 years of operation (i.e., nearly after 15,000 flights).

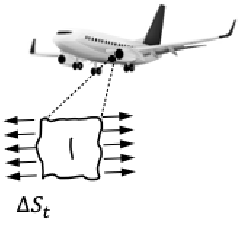


Figure 4.1: Control point on the underside of the aircraft wing.

Table 4.1: Flight type load distribution and related minimum and maximum stresses (MPa) (adapted from [4]).

	Load case		**Flight type**									
	Smin	Smax	I	II	III	IV	V	VI	VII	VIII	IX	X
A	0.07	30.5	2	2	1	1	1	0	0	0	0	0
B	2.45	28	2	2	2	1	1	1	0	0	0	0
C	4.8	26	3	2	2	2	1	1	1	0	0	0
D	7.15	23	5	2	2	2	2	2	2	2	2	0
E	9.5	21	13	12	12	12	12	12	12	12	1	0
F	11.85	19	30	29	29	29	29	29	29	29	7	3

In this dissertation, we consider that corrosive cycles represent, on average, 5% of the total number of cycles in one mission (loads of type E and F are related to take-off and landing, respectively, and thus are subjected to corrosion). We then proceed to penalize this cycles by accumulating damage using curves that are between air and NaCl at 3.5% (see Figure 4.2b). The exact curve will depend on a corrosivity index C^{IDX} (which essentially controls the interpolation between the two curves). We use the curve for air when $C^{IDX} = 0$, . We use the curve for NaCl at 3.5% when $C^{IDX} = 1$.

Table 4.2: Mission mix configuration (flight type probability per mission).

			Flight type		
Mix	I	II	III	IV	V
1	5.10^{-5}	$7.5\ 10^{-5}$	1.10^{-3}	$2.4\ 10^{-3}$	$1.1\ 10^{-2}$
2	$2.5\ 10^{-5}$	$2.5\ 10^{-5}$	$1.2\ 10^{-3}$	$1.5\ 10^{-3}$	$6.75\ 10^{-3}$
3	0	0	$1.3\ 10^{-3}$	$2.3\ 10^{-3}$	$1.6\ 10^{-2}$
4	0	5.10^{-5}	$9.5\ 10^{-4}$	$2.5\ 10^{-3}$	$8.5\ 10^{-3}$
5	0	0	$1.2\ 10^{-3}$	$2.3\ 10^{-3}$	6.10^{-3}
6	1.10^{-4}	$2.5\ 10^{-5}$	$7.5\ 10^{-4}$	$2.3\ 10^{-3}$	$6.13\ 10^{-3}$
7	0	0	0	$2.3\ 10^{-3}$	$7.3\ 10^{-3}$
8	$2.5\ 10^{-5}$	0	1.10^{-3}	$2.5\ 10^{-3}$	1.10^{-3}
9	0	$7.5\ 10^{-5}$	$7.5\ 10^{-4}$	$7.7\ 10^{-3}$	$1.1\ 10^{-2}$
10	$7.5\ 10^{-5}$	0	$7.5\ 10^{-4}$	$2.4\ 10^{-3}$	$6.25\ 10^{-3}$
Mix	VI	VII	VIII	IX	X
1	0.01	0.095	0.15	0.248	0.4824
2	0.025	0.035	0.155	0.273	0.5025
3	0.025	0.025	0.105	0.273	0.5530
4	0.013	0.095	0.055	0.323	0.5030
5	0.015	0.045	0.105	0.2975	0.5278
6	0.015	0.045	0.08	0.3975	0.4528
7	0.015	0.070	0.055	0.3225	0.5278
8	0.02	0.02	0.12	0.285	0.5528
9	0.025	0.025	0.055	0.2225	0.6528
10	0.005	0.05	0.08	0.2725	0.5828

To emulate different corrosive environments in the proposed numerical experiment we consider 10 distinct airports. On each flight, we know from which airports the aircraft took off and landed. Thus, we can proceed to use the respective corrosivity indexes presented in Table 4.3 to penalize the corrosive loading cycles in the flight. While the actual relationship between the corrosivity index and the C and m constants is unknown; here, we arbitrarily used the following equations:

$$\begin{gathered} C = -2.275 \times 10^{-8}(C^{IDX})^2 + 4.505 \times 10^{-8}C^{IDX} + 1.132 \times 10^{-10}, \text{ and} \\ m = 2.046(C^{IDX})^2 - 4.052C^{IDX} + 3.859. \end{gathered} \tag{4.1}$$

Table 4.3: Airport corrosion index values.

	Airport									
	A	B	C	D	E	F	G	H	I	J
C^{IDX}	0.517	0.389	0.361	0.545	0.781	1	0.125	0.58	0.608	0.844

As we discussed, Eq. 4.1 is arbitrarily used in this study as a way to generate synthetic data. In reality, the mapping between corrosivity agents (such as NaCl) and damage accumulation rate (for example, through coefficients of Paris law) is unknown (or only partially understood). The main contribution discussed in this chapter, is the use our hybrid physics-informed neural network to learn that behavior (and adjust the physics-informed portion of the model) with data that could be observed in real life.

Even the few cycles that have the panels exposed to corrosion (average of 5$\%$) can drastically accelerate damage accumulation. When corrosion is not considered, the 5 years inspection is expected to return crack lengths smaller than 1 (mm). However, in the presence of these few corrosive cycles, the fleet crack envelope is above that value, with few aircraft having crack lengths above the 20 mm mark. We consider that, for every aircraft in the fleet, the far-field cyclic stresses and airport corrosivity index of every flight are available. In real life, the far-field cyclic stresses would be obtained by using engineering models (likely based on finite element analysis) that relate specific maneuvers with cyclic loads (here, we rely on Tables 4.1 and 4.2). Corrosivity index could be obtained by analyzing meteorological and environmental data. This information is readily available for most airports, although proprietary at times (here we rely on the arbitrarily chosen values shown in Table 4.3). In terms of crack length data, we assume that for 10% of the fleet (i.e., 15 aircraft) inspection is performed at the end of the 5^{th} year of operation. Hence, the training set can be summarized as:

1. Observed outputs: crack length at 15 inspected aircraft.

2. Observed inputs: time series for a_t, ΔS_t, R_t, and C_t^{IDX}. The number of cycles depends on the mission mix each aircraft is subject to. By the end of the 5^{th} year, the average number of cycles is around 174,000. Since we used 15 aircraft for training, we actually observed around 2,625,000 input conditions

4.2 Physics-informed Neural Network for Corrosion-fatigue Damage Accumulation

Applying artificial neural networks to solve differential equations in engineering is a well-developed methodology [166, 167, 168]. Nevertheless, the computational power available these days contributed to the popularization of machine learning in engineering applications. The scientific community has been studying and proposing deep learning architectures that leverage mathematical models based in physics and engineering principles [169, 170, 69, 171, 172]. Physics laws (in the form of differential equations) help handling the reduced number of data points, and constrain the hyper-parameter space. Raissi [173] approximated the unknown solution of partial differential equations by two deep neural networks. The first network acts as a prior on the unknown solution (enabling it to avoid ill-conditioned and unstable numerical differentiations). The second network works as a fine approximation to the spatiotemporal solution. The methodology was tested on a variety of equations used in fluid mechanics, nonlinear acoustics, gas dynamics, and other fields. Yu et al. [174] developed a physics based learning methodology to simulate aircraft dynamics based on a deep residual recurrent neural network. The authors illustrated their methodology by means of a six degrees-of-freedom Boeing 747-100 aircraft model. The model was trained with simulated data of the aircraft model and then used to predict aircraft response under arbitrary control inputs and disturbances.

In the next two subsections, we present cumulative damage models (and the specialized case of corrosion-fatigue). These models will motivate our approach to physics-informed neural networks.

Then, we detail how to compensate for bias in the physics-informed layers of a recurrent neural network.

4.2.1 Cumulative Damage Models and Corrosion-fatigue

Cumulative damage models [175, 176] describe the irreversible accumulation of damage throughout the useful life of components (or systems) such that the damage at time t, a_t, is the sum of a damage increment Δa_t on top of damage a_{t-1} at previous time step $t-1$

$$a_t = a_{t-1} + \Delta a_t \tag{4.2}$$

where Δa_t is the damage increment, which is often a function of a_{t-1} and other inputs $\mathbf{x}_t$ at time t.

The characterization of the damage a_t and the inputs $\mathbf{x}_t$ is highly problem dependent. The damage a_t is usually associated with a failure mechanism and a_t is ideally an observable quantity[1]. For example, if fatigue is the failure mechanism, crack length is the observable quantity. The inputs $\mathbf{x}_t$ usually express time-dependent loading and boundary conditions (e.g., pressures, temperatures, torques, mechanical and thermal stresses, etc.) or even operating points (e.g., altitude, thrust, angle of attack, etc.). For that matter, fatigue crack propagation is usually modeled through Paris law [179]

$$da \Big/ dt = C(\Delta K_t)^m \tag{4.3}$$

where ΔK is the stress intensity range and C and m are material properties determined through coupon data and many engineering materials have constants documented in handbooks such as [1].

The Paris law can be modified to take into account the effect of mean stress, expressed in terms

[1] Although this is not a requirement, it significantly facilitates the modeling task [177, 178, 176].

of the stress ratio between the minimum and maximum stress levels $R = S_{min}/S_{max}$. We use the Walker model [180] to incorporate the stress ratio:

$$\frac{da}{dt} = \frac{C_0}{(1-R)^{m(1-\gamma)}}(\Delta K_t)^m \tag{4.4}$$

where C_0 and m depend on the environmental conditions and γ depends on the material and loading conditions. When cycle-by-cycle information is available, the discrete form of Eq. 4.4 resembles Eq. 4.2 and becomes:

$$a_t = a_{t-1} + \frac{C_0}{(1-R_t)^{m(1-\gamma)}}(\Delta K_t)^m, \tag{4.5}$$

where stress intensity range, ΔK comes from engineering models that use information about geometry, crack length, and internal loads (stress amplitude). For example, assuming that fatigue damage accumulates under mode I loading condition, the stress intensity range ΔK_t can be expressed as:

$$\Delta K_t = F \Delta S_t \sqrt{\pi a_{t-1}} \tag{4.6}$$

where F is a geometry factor and ΔS_t is the far-field stress.

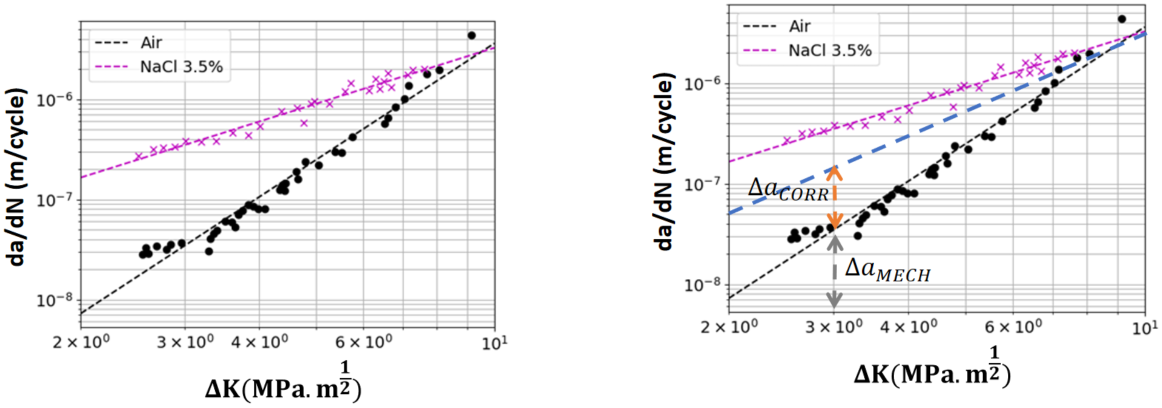
As we mentioned before, corrosion-fatigue is a very complex phenomenon involving pit nucleation, pit growth, fatigue crack nucleation, short crack growth, transition from short crack to long crack, and long crack growth. Modeling corrosion-fatigue has proven to be a daunting task [131, 181] and it is not our objective to propose a new method (nor defend any preferred method) for it. Instead, we will focus on how a **deep neural network** can be designed and trained to act as a **bias estimator**, compensating the corrosion effects in crack growth.

In this dissertation, we assume that the corrosivity of an environment can be associated with an index C^{IDX} (such as the concentration of a particular corrosive agent in the air). Empirical studies, such as the one found in [182], are usually necessary to determine how a particular corrosive agent accelerates the damage accumulation. They showed that corrosivity, that in this case is given by the concentration of sodium chloride (NaCl) can drastically increase the damage accumulation rate (shifting the da/dN vs ΔK curve up, when compared to the rates in pure air), as illustrated in Figure 4.2a.

We assume that, at varying NaCl concentrations, the damage increment per cycle Δa as a function of ΔK can be modeled as a baseline contribution, Δa_{MECH}, plus a shift due to corrosion, Δa_{CORR} (see Figure 4.2b), and Eq. 4.5 can be re-written as

$$\begin{aligned} a_t &= a_{t-1} + \Delta a_{MECH} + \Delta a_{CORR} \\ a_t &= a_{t-1} + \frac{C_0}{(1-R_t)^{m(1-\gamma)}}(\Delta K_t)^m + g(\Delta S_t, R_t, C^{IDX}, a_{t-1}) \end{aligned} \tag{4.7}$$

where $g(.)$ is an unknown function. We discuss how we model $g(.)$ in the next section.



(a) Coupon data (adapted from [182]).

(b) Corrosion effect on cyclic damage.

Figure 4.2: Corrosion-fatigue crack propagation rate. Paris law coefficients when $R = 0$: for pure air $C = 1.132 \times 10^{-10}$ and $m = 3.859$; for NaCl (sodium chloride) at 3.5 % solution $C = 2.241 \times 10^{-8}$ and $m = 1.853$.

4.2.2 Physics-informed Neural Networks for Cumulative Damage Modeling

A recurrent neural network [183] repeatedly apply transformations to given states in a sequence, as shown in Eq. 4.8 and illustrated in Figure 4.3a.

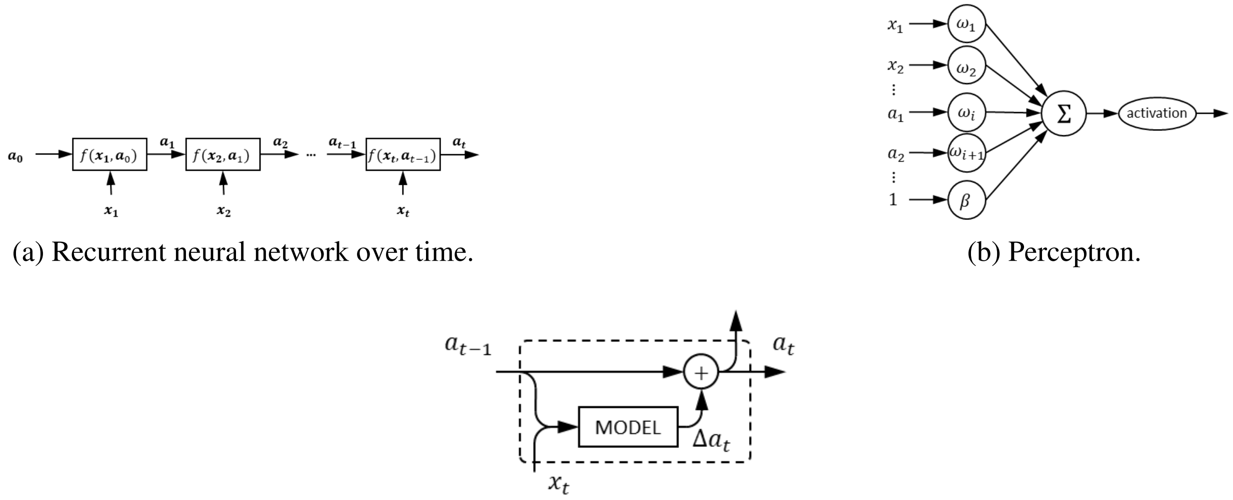
$$\mathbf{a}_t = f(\mathbf{x}_t, \mathbf{a}_{t-1}) \tag{4.8}$$

where $t \in [0, \ldots, T]$ represent the time discretization, $\mathbf{a} \in R^{n_a}$ are the states representing the sequence, $\mathbf{x} \in R^{n_x}$ are input variables, and $f(.)$ defines the transition between time steps (function of input variables and previous states). In the recurrent neural network terminology, different implementations of $f(.)$ are referred to as cells.

As illustrated in Figure 4.3b, a cell implementing $f(.)$ can be as simple as a single-layer perceptron. However, it can also assume sophisticated forms such as the long short-term memory [184] and the gated recurrent unit [185]. Besides improving generalization, these architectures improve training by mitigating the vanishing/exploding gradient problem [183].

Recurrent neural networks have been successfully used in many sequence modeling applications [186, 187, 188, 189, 190]. Nascimento and Viana [93] proposed the cell illustrated in Figure 4.3c specifically for the integration of cumulative damage. In such design, the state represents the cumulative damage at time t and ‘‘MODEL’’ maps the inputs x_t and previously accumulated damage a_{t-1} into a damage increment Δa_t (equivalent re-writing Eq. 4.8 as Eq. 4.2). In a purely physics-based approach, ‘‘MODEL’’ is the computational implementation of the physics of failure (which, again, is highly application dependent). Nevertheless, in the design cell, nothing prevents ‘‘MODEL’’ to be a data-driven model, such as a multi-layer perceptron, or a hybrid model, where some parts are physics-based and others are data-driven.

One might be tempted to ask why use the cell shown in Fig. 4.3c and why use physics-informed



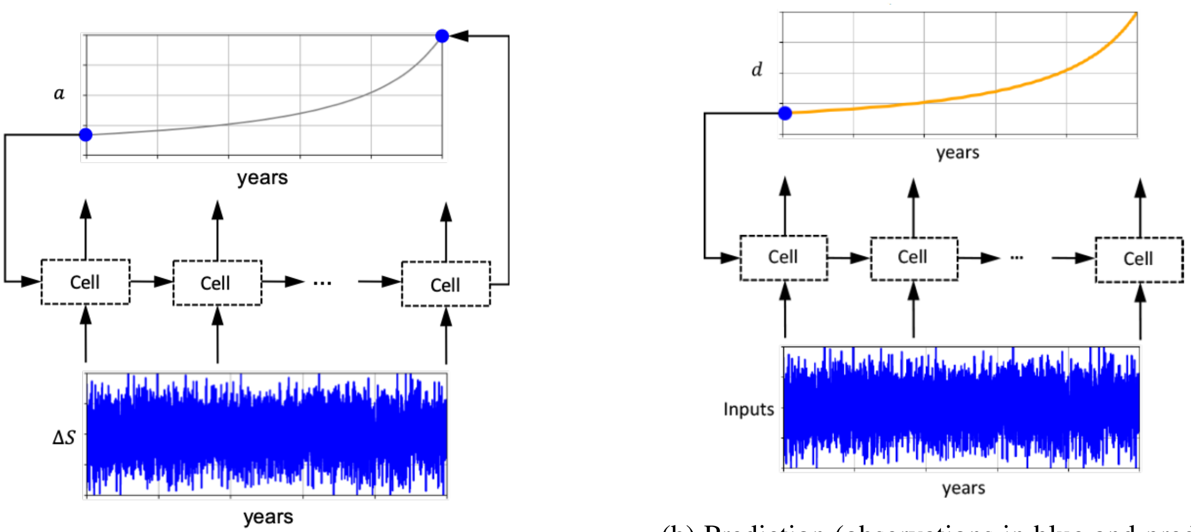
(a) Recurrent neural network over time.

(b) Perceptron.

(c) Cumulative damage cell.

Figure 4.3: Recurrent neural network, perceptron as the simplest cell, and cumulative damage cell.

‘‘MODELS’’ within that cell? The answer to both questions is in the niche of applications we target in this dissertation. **Our application of cumulative damage models involves full observation of input conditions (i.e., loads), and partial observation of damage (e.g., crack length).** This is the case for example when through detailed flight tracking in conjunction with engineering models the loads can be estimated cycle-by-cycle, and a control point in an aircraft fuselage panel is inspected in regular intervals through visual inspection or nondestructive evaluation approaches (e.g., Eddy current [19], ultrasound [20], dye penetrant inspection [21], etc.). Figure 4.4a shows the typical data collected for training the cumulative damage model. In fairness, this represents one specific aircraft, and likely, there would be load history and inspection data available for few aircraft. Figure 4.4b illustrates the typical data collected for prediction using the trained cumulative damage model. As we mentioned, load history is available throughout the useful life and the initial value for the states is either known or assumed. The cumulative damage is then used to estimate the damage growth over time. Given the few observations of damage, we argue that as illustrated in Nascimento and Viana [93], by using purely data-driven architectures (such as long short-term memory and the gated recurrent unit) is unlikely to obtain accurate models. Nevertheless, the long short-term memory and the gated recurrent unit architectures can be useful in cases where there is a full observation of the states. This happens when control points are continuously monitored with dedicated sensors (e.g., comparative vacuum monitoring [17], fiber Bragg grating sensors [18], etc.). However, this is not the focus of this dissertation.



(a) Training (observations in blue).

(b) Prediction (observations in blue and prediction in orange).

Figure 4.4: Typical use-case of recurrent neural network for cumulative damage model.

As previously discussed, the damage accumulation model presented in Eq. 4.7 is composed of two main terms. The base damage increment is set at Δa_{MECH} and can be described by Eq. 4.4. Added to the base damage increment, there is a contribution due to corrosion (Δa_{CORR}) (with an unknown functional form). In this chapter, we propose the repeating recurrent neural network cell illustrated in Figure 4.5 to implement the corrosion-fatigue crack growth model shown in Eq. 4.7. We implemented the Walker model to account for the base Δa_{MECH}. This portion of the model (highlighted in blue) is considered physics-informed. Then, we use a multi-layer perceptron to model the dependence between the previous state value (a_{t-1}) and cycle-by-cycle inputs such as the far-field stress (ΔS_t), stress ratio (R_t) and corrosivity index (C^{IDX}). The multi-layer perceptron is adjusting the damage increment output by the Walker model so that it compensates for the effect of corrosion. Therefore, it works as a discrepancy-correction term.

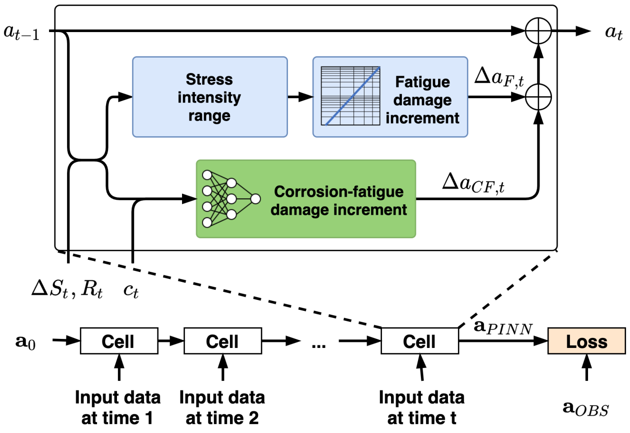


Figure 4.5: Proposed recurrent neural network cell for corrosion-fatigue crack propagation.

4.2.3 Physics-informed Neural Network Design

With the information discussed in Section 4.1, we proceed to built our hybrid physics-informed neural network model for corrosion-fatigue (illustrated in Figure 4.5). Table 4.4 details all the multi-layer perceptron architectures evaluated in this chapter. We initially evaluated all the described architectures and, based on their training performance (i.e. best loss function value, and cross-validation error), we selected the best architecture for the damage forecast analysis. We opted for using these architectures to illustrate the ability to fit a neural network with varying number of trainable parameters. No attempt was made to further refine the multi-layer perceptron architecture and its accuracy.

The constructed multi-layer perceptrons take four inputs (crack length, far-field stress, stress ratio and corrosivity index) and provide one output (bias in damage increment due to corrosion). It

Table 4.4: Multi-layer perceptron (MLP) configurations used to model the bias estimator term.

Layer #	**neurons/activation function**						
	MLP #1	MLP #2	MLP #3	MLP #4	MLP #5	MLP #6	MLP #7
0	5 / tanh	10 / elu	10 / elu	10 / tanh	20 / tanh	20 / elu	40 / elu
1	1 / linear	5 / elu	5 / sig [*]	5 / tanh	10 / elu	10 / sig	20 / sig
2		1 / linear	1 / elu	1 / elu	5 / sig	5 / sig	10 / sig
3					1 / linear	1 / elu	1 / elu
Parameters	31	111	111	111	371	371	1241

[*] here ''sig'' is referring to the sigmoid activation function.

worth mentioning that the multi-layer perceptron output (Δa_{CORR}) is hidden, meaning that we do not observe its value directly. We only observe the recurrent neural network output, a_t, i.e. the crack length after a given number of flights. We can still build this model since the recurrent neural network performs the damage integration (Figures 4.3c and 4.5) adding a_{t-1} to the damage increment contribution due to fatigue Δa_{MECH} and the damage increment contribution due to corrosion Δa_{CORR}. Interestingly, since Δa_{CORR} is hidden, it leverages the roughly 2,625,000 observed input conditions during its training. Although models described in Table 4.4 can have as many as 1,241 trainable parameters, there is no risk of model overfitting.

The optimization of recurrent neural network hyperparameters can be challenging and it is conducted through a gradient-descent algorithm. An initial guess for the hyperparameters that is far away from the optimum might cause divergence of the optimization (or at least a very long time of training). In this chapter, we also propose a way to initialize the multi-layer perceptron parameters. We propose initializing the multi-layer perceptron hyperparameters by using auxiliary planes derived from a simple linear representation of the input-output relationship:

$$\Delta a_{CORR} = \beta_0 + \beta_1 \Delta S + \beta_2 R + \beta_3 C^{IDX} + \beta_4 a \tag{4.9}$$

where β_i are carefully chosen to be physically reasonable.

In other words, we reckon that practitioners would be able to define rough order of magnitude of damage increment as well as its directionality with respect to the input variables. The coefficients are initialized using engineering judgment. For instance, we know that when the corrosion index is zero the bias is also zero, and it is safe to assume that the bias value increases with far-field stress, corrosivity index and crack length (see Figure 4.2b). Using these constraints we randomly generated 20 auxiliary planes. This procedure is used only to yield an initial point for the recurrent neural network training and it does not interfere with its overall performance.

The recurrent neural network is fitted with observations for inputs throughout the time series and observations for crack length only at inspection (besides the initial crack length). In other words, the multi-layer perceptron outputs (Δa_{CORR}) are latent variables of the model. In the training of the recurrent neural network, we use the mean absolute percentage error (MAPE) as loss function:

$$Loss = MAPE = \frac{100}{n} \times \sum_{i=1}^{n} \frac{|a_{PRED,i} - a_{OBS,i}|}{a_{OBS,i}} \tag{4.10}$$

where n is the number of observations (inspected aircraft), $a_{PRED,i}$ and $a_{OBS,i}$ are the predicted and observed crack lengths for the i^{th} inspection, respectively. All training sessions were performed using RMSProp[2] as an optimizer, setup with a learning rate of 10^{-12} for 25 epochs.

The interested reader is referred to [191] for a discussion about $MAPE$. Errors of identical magnitude but opposite sign can have different contributions to the $MAPE$. Further, when actual values are small or zero, then contributions to the $MAPE$ are very large. In our application, we do not have problems with the sign of the errors (since $a_{OBS,i} > 0$ and $|a_{PRED,i} - a_{OBS,i}| \geq 0$). Nevertheless, $MAPE$ accentuates the contribution of small crack lengths. Again, in this application,

[2] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop

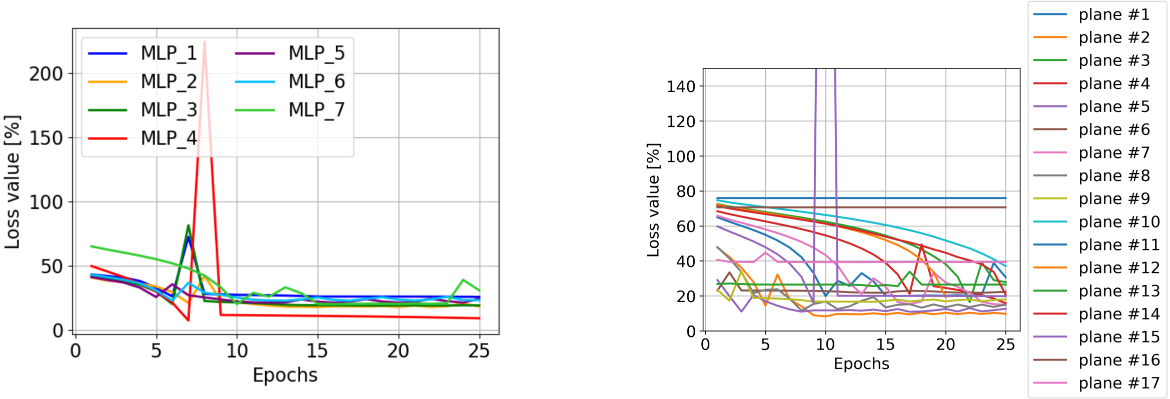
we do not find this to be problematic given the cumulative nature of damage. With slightly overestimation of the small cracks in training, we avoid underestimation in damage forecast (as we integrate damage over time).

4.3 Results and Discussion

4.3.1 Recurrent Neural Network Training

As detailed in section 4.1, the proposed physics-informed neural network was trained considering crack length data for 10% of the fleet at the end of the 5^{th} year of operation. Also, as described in section 4.2.3, different multi-layer perceptron architectures (see Table 4.4) were evaluated along with distinct auxiliary planes for the multi-layer perceptron hyperparameters initialization. Figure 4.6a illustrates the loss function history of all multi-layer perceptron architectures, considering the same initial guess (i.e. auxiliary plane). The number of epochs that yields convergence for each multi-layer perceptron varies, but a rapid convergence rate is noticeable with most architectures achieving stagnation in 10 epochs or less. On the other hand, Figure 4.6b shows the loss function history of the multi-layer perceptron architecture $\#7$ across all the 20 auxiliary planes. Even with the engineered multi-layer perceptron weights initialization, convergence to same value of loss function is not guaranteed. Moreover, 3 out of the 20 considered auxiliary planes eventually led to ‘‘not a number’’ loss function values (caused by consecutive very large Δa_{CORR}), which essentially stops the optimization task. As expected, the success in optimizing the hyperparameters depends on the initial guess. However, we propose different initial guesses following the approach described in section 4.2.3. With respect to computational cost, the average wall-clock time for training was of 168 min.

Table 4.5 summarizes the optimization performance (for all multi-layer perceptron architectures



(a) All multi-layer perceptron architectures starting from same initial auxiliary plan.

(b) Multi-layer perceptron #7 starting from different auxiliary planes.

Figure 4.6: Loss function histories. Multi-layer perceptron architectures are detailed in Table 4.4.

across the 20 planes used for weight initialization). The conversion ratio is the number of times that the optimization managed to reduce the loss function without leading to ‘‘not a number’’ within the considered number of epochs. All proposed multi-layer perceptron architectures exhibited convergence ratios of at least 75% for the considered initial guesses. This results indicates that the recurrent neural network training is more sensitive to the initial guess than the multi-layer architecture itself. In general, the training sessions in which the recurrent neural network converged were based on the same engineered planes used for initializing the trainable parameters. As expected, Table 4.5 the more complex architectures (MLP #6 and MLP #7) lead to the lowest loss function values (tending to better represent the input-output relationship in the considered case study when compared to the other architectures).

Next, we study whether cross-validation analysis as presented in [192] can be used to help choosing a multi-layer perceptron architecture amongst the ones considered for this problem. We considered only the best initial guess for each architecture and; to keep the computational cost low, we used 5-fold cross validation. We randomly split our training set into five subsets; and on each iteration

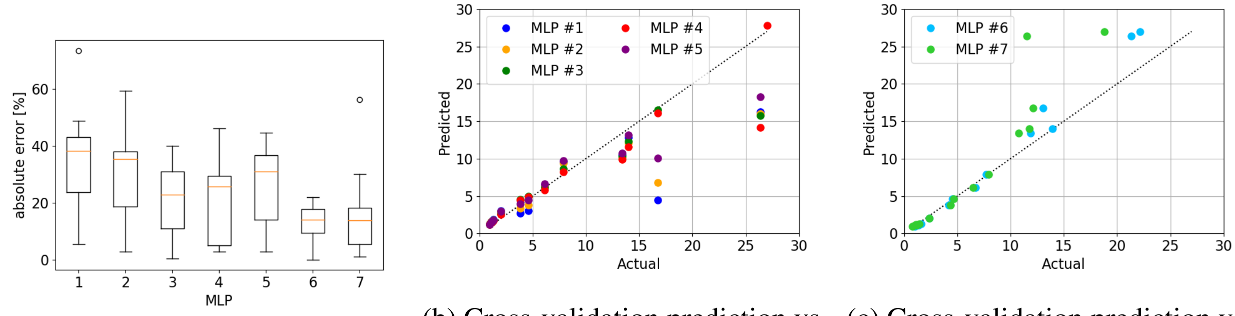
Table 4.5: Multi-layer perceptron (MLP) architectures convergence analysis. These results evaluate the training performance of the multi-layer perceptron architectures proposed in Table 4.4 considering the 20 initial guesses provided by the auxiliary planes. The conversion ratio is the number of times that the optimization successfully finished given the considered number of epochs. The best obtained value of the loss function (Eq. 4.10) is also presented.

MLP	#1	#2	#3	#4	#5	#6	#7
Conversion ratio (out of 20)	15	16	17	16	16	18	17
Best loss value (%)	24	21	19	21	22	12	14

of the cross-validation, three aircraft were removed from the training process. After the training is finished, the predictions are checked against the three aircraft left out[3]. Cross-validation errors are defined as $\mathbf{e}_{XV} = \mathbf{a}_{XV} - \mathbf{a}_{OBS}$, where $\mathbf{a}_{XV}$ are the cross-validation predicted crack lengths and $\mathbf{a}_{OBS}$ are the actual observed crack lengths (at the end of the 5^{th} year of operation). Percent cross-validation errors are defined as $\%e_{XV,i} = 100 \times \frac{|e_{XV,i}|}{a_{OBS,i}}$. Figure 4.7a presents a box plot of the cross-validation absolute errors. The ranking of the models by cross-validation agree with the ranking given by ''best loss values'' presented in Table 4.5 (indicating that the more complex architectures in set, MLP #6 and MLP #7, are the most suitable for the considered case study). Figures 4.7b and 4.7c illustrate the cross-validation predictions against the actual crack length values. The simpler architectures (MLP #1 to MLP #5) tend to underestimate the crack length values higher than 10 mm, while the complex architectures (MLP $\#6$ to MLP $\#7$) tend to overestimate them. Underestimating the large crack lengths is highly undesirable as it would undermine the safe operation of the fleet. With the cross-validation analysis results we can choose the model that not only leads to the best prediction, but it is also slightly conservative. Assuming that a practitioner wants to avoid underestimation of large crack length values, we opt to select architecture $\#6$ for the

[3] In sequence to sequence modeling, it is also common to find strategies for splitting the data along the time or sequence axis. In our application, for the sake of training our hybrid physics-informed neural networks, only inputs are observed throughout time. The output is known at $t = 0$ and observed at the time of inspection (not the entire history, only the value at end of 5^{th} year). Therefore, any splits along the time or sequence axis are not viable.

remainder of this chapter.



(a) Percentage absolute cross-validation errors.

(b) Cross-validation prediction vs. actual crack length (MLPs #1 to #5).

(c) Cross-validation prediction vs. actual crack length (MLPs #6 and #7).

Figure 4.7: Recurrent neural network cross-validation analysis.

Finally, we also studied the contribution of the number of inspected aircraft to the recurrent neural network training. For this analysis we considered MLP architecture #6 with its best initial guess trained with three distinct training set sizes: 5, 10 and 15 inspected aircraft in the 5^{th} year of operation. Figure 4.8 presents the overall crack length prediction for each training set size after training. One immediate observation, as expected, is that the more aircraft are inspected, the better the recurrent neural network is (predictions are closer to actual values). However, another observation is that with small number of inspections, it is naturally harder to sample the entire range of observed crack lengths, 0 to 25 (mm) As a matter of fact, when 5 aircraft are inspected, the crack length observations are limited to less than 5 (mm). With 10 inspected aircraft there is only one aircraft with crack length between 15 and 20 (mm). The final conclusion is that as much as the number of inspected aircraft contributes to the quality of the resulting model, the coverage of output observation is equally important.

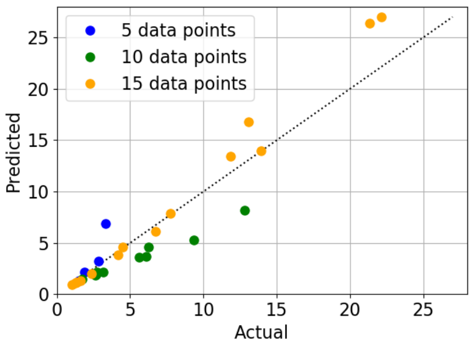


Figure 4.8: Recurrent neural network training results considering architecture #6 best initial guess and varying number of available inspection data during training.

4.3.2 Corrosion-fatigue Diagnosis and Prognosis

After the recurrent neural network was trained, the resulting model was used to perform crack length estimation at the end of the 5^{th} year of operation (diagnosis) and damage forecast at the end of 6^{th} year of operation (prognosis) across the entire fleet (150 aircraft). Figure 4.9a presents the results of the crack length estimation at the 5^{th} year of operation for the entire fleet. We can compare the crack lengths predictions of the proposed hybrid recurrent neural network with the ones coming from the fleet (i.e. full corrosion-fatigue model) as well as the purely mechanical fatigue model (i.e. Walker's equation). The difference between the last two is the bias estimated by the hybrid model. It is noticeable in Figure 4.9a how the recurrent neural network predictions captures the overall trend of the corrosion-fatigue damage, indicating that the proposed hybrid model is acting as a bias estimator (missing physics). In terms of fleet management, Figure 4.9a shows that the hybrid recurrent neural network can be used to prioritize which aircraft should be inspected next (the ones with highest predicted crack lengths).

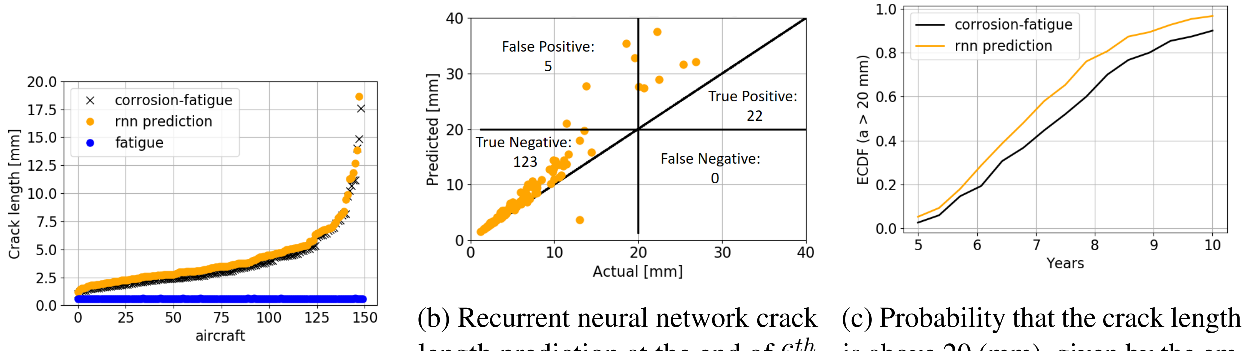
With regards to prognosis, we split the analysis into two time frames: a short-term analysis in which we are forecasting a single year of fleet operation (simulating into the end of the 6^{th} year of operation); and a long-term analysis in which we forecast damage accumulation until the end of the 10^{th} year of operation. Figure 4.9b presents the results of the short-term analysis showing the predicted versus actual crack lenghts at the end of the 6^{th} year of operation. As we discussed before, the model is known to be conservative; and therefore; the predicted values are expected to be higher than the actual ones. In terms of fleet management, we could use this predictions to evaluate how many aircraft in the fleet would be above an arbitrary repair/replacement threshold. Figure 4.9b shows that, if this threshold is 20 (mm), we correctly flagged all aircraft with actual crack length above it (22 true positives and no false negatives). In safety critical applications, having no false negatives is extremmely important (an aircraft being wrongly cleared to flight can have serious implications). We also see a small number of false positives. These 5 aircraft would be flagged for repair/replacement while in reality at least 2 of those would be very near the threshold of 20 (mm).

Finally, we also evaluate long-term damage forecast (meaning, performing damage estimation up to the 10^{th} year of operation). The main findings of this analysis are represented by the empirical cumulative density function (ECDF) illustrated in Figure 4.9c. Obtained results confirms that the selected model is conservative. However, in terms of fleet management, it is useful as it warns operators of the rapid increase over time in the probability that the crack will exceed the threshold of 20 (mm). The number grows from around 5% by end of the 5^{th} year to almost 60% by the end of the 7^{th}. This information can guide how fast mitigation strategies have to be put in place.

4.3.3 Replication of results

Our implementation is all done in TensorFlow[4] (version 2.0.0-beta1) using the Python application programming interface. In order to replicate the results presented here, the interested reader can download the codes and data. First, install the ‘‘PINN’’ python package (base for physics-informed

[4] `https://www.tensorflow.org`



(a) Recurrent neural network crack length prediction across entire fleet at the end of 5^{th} year of operation.

(b) Recurrent neural network crack length prediction at the end of 6^{th} year of operation. Plot is clipped at 40 (mm) even though predictions can be higher than that.

(c) Probability that the crack length is above 20 (mm), given by the empirical cumulative density function (ECDF), from the 5^{th} to the 10^{th} year of operation.

Figure 4.9: Corrosion-fatigue diagnosis and prognosis.

neural networks used in this work) available at [94]. Links for the required data sets and Python scripts demonstrating the proposed framework can be found in [95]. All simulations were conducted using a laptop configured with an Intel Core i7-7820HQ CPU at 2.90GHz, 32GB of RAM, and NVIDIA Quadro M620 graphical processing unit running Windows 10.

4.4 Summary

In this chapter, we proposed compensating for epistemic uncertainties in damage accumulation modeling by simultaneously using physics-informed and data-driven layers within a recurrent neural network. The major achievement of such a hybrid model is the ability to use the data-driven layers to compensate for the missing physics in reduced-order models and accurately estimate damage accumulation. In our numerical experiment, this reduced the need for observing the output of interest at all times. In other words, the physics-informed layers reduce the need for large datasets found in purely data-driven approaches.

In order to evaluate the performance of the proposed framework, we presented a numerical case study focusing on corrosion-fatigue crack propagation. In the considered case, the physics-informed layer only accounts for purely mechanical fatigue. The data-driven layer, on the other hand, adjusts the damage accumulation rate and compensate for the corrosion effects (i.e., working as a bias corrector).

In the following chapter we will focus our discussion on different functional forms for the data-driven portion of the hybrid cell and how it can be used to embed distinct physics-based knowledge into the hybrid model.

CHAPTER 5: EXPANDING THE CORROSION-FATIGUE CASE STUDY BY CONSIDERING OTHER FUNCTIONAL FORMS FOR THE MISSING PHYSICS TERM

In this chapter and in Chapter 6, we will expand the corrosion-fatigue case study discussed in Chapter 4 to consider different functional forms for the data-driven portion of the hybrid neural network cell tasked to compensate for the missing physics effects. In the previous chapter we derived the proposed numerical experiment assuming that once the unexpectedly larger crack lengths were detected, operators would be able to come to the conclusion that the culprit for the higher crack propagation rates was the unaccounted corrosion effects, and act to properly take such effects into consideration in prognosis analysis. Despite not being a farfetched assumption, given that in the case study the epistemic uncertainty is due to corrosion, and that in inspection campaigns, visual cues could significantly help to differentiate a purely mechanical fatigue crack of a corrosion-fatigue crack, it is a little of a stretch to assume that at this point operators would unequivocally assume corrosion-fatigue to be the sole cause of the unexpected observations. Hence, in the next two chapters we will revise the numerical experiment to implement different functional forms for the data-driven layers, derived from different physics-based considerations. The key idea here is to derive ‘‘pools’’ of hybrid models, and use uncertainty propagation tools to segregate the models based on their accuracy to predict the observed behavior of the system. This is an attempt to emulate a scenario in which after the unexpected behavior of the system is detected by operators, they proceed to use the hybrid models to assist in the task of identifying the its ‘‘root cause.’’

Besides detailing the proposed modifications to the corrosion-fatigue case study, in this chapter three novel hybrid recurrent neural network cells will be presented and properly discussed. The test problem still consists of predicting corrosion-fatigue of an Al 2024-T3 alloy used on panels of aircraft wings and the unforeseen high exposure to corrosive environments by the air-carrier it still is the cause of the unexpected larger crack lengths. However, loads and mission mixes were revised to increase variability in the fleet and accelerate crack propagation, in addition to the consideration of a meteorological model to stipulate corrosivity indexes and add seasonality effects to its values. The goal behind the proposed modifications is to hinder the task of distinguishing the main cause of the unexpected observations, e.g., by increasing the load effects in the crack propagation, a hybrid model could erroneously favor load compensation instead of compensating for the corrosion effects.

5.1 Revised Synthetic Aircraft Fleet Data

For the discussions presented in this chapter and in Chapter 6, we considered a synthetic fleet of 100 aircraft, scheduled to fly an average of 8 flights per day. The number of cycles depends on the mission mix each aircraft is subject to (in average at 10,000 flights it revolves around 50,000 cycles).

Wing panels are once again subjected to cyclic mechanical loading alongside saline corrosion. In any given flight, the vast majority of fatigue cycles are purely mechanical. However, take-off, landing, and taxing cycles are subjected to a mix of mechanical loading and saline corrosion. The level of saline corrosion-fatigue depends on the airport and day of the year (seasonality). The control points of interest are still located over aircraft wing panels made of Al 2024-T3 alloy (same coefficients as shown in Figure 4.2) with an initial crack length (a_0) of 0.5 (mm). Here, we arbitrarily considered that the crack length information derived from inspection is initially available for only part of the fleet after 9,000 flight hours (this value was selected to approximated what would be the second ‘‘C’’ check (see Table 2.3) in the fleet).

For simplicity, we consider that an aircraft in the fleet can fly the 10 different flight types detailed in Table 5.1. Each flight type is characterized by a load frequency distribution. Therefore, flights present different severity levels in terms of mechanical loads. In order to balance out the exposure of the fleet to the severe flights, as in the previous case, here, we also considered 10 arbitrarily designed mission mixes. In our study, each aircraft in the fleet is assigned to only one specific mission mix. Within a given mission mix, flights types are assigned to the aircraft randomly following the probability distribution detailed in Table 5.2.

Table 5.1: Flight type load matrix distribution and related normalized minimum and maximum stresses (MPa) (adapted from [4]).

	Load magnitude					
	A	B	C	D	E	F
S_{min}	0.07	2.26	4.46	6.65	8.84	11.0
S_{max}	28.3	26.0	23.8	21.7	19.5	17.3
	Load frequency					
Flight type	A	B	C	D	E	F
I	2	2	3	5	13	30
II	2	2	2	2	12	29
III	1	2	2	2	12	29
IV	1	1	2	2	12	29
V	1	1	1	2	12	29
VI	0	1	1	2	12	29
VII	0	0	1	2	12	29
VIII	0	0	0	2	12	29
IX	0	0	0	0	1	7
X	0	0	0	0	0	3

Several factors need to be considered while determining the corrosivity index C^{IDX}, including meteorological data (namely temperature, humidity, and saline concentration) along with geographical location (e.g., distance from airport from the coast, altitude, etc.) Here, we use a modified version of the PACER LIME model [193, 194], illustrated in Figure 5.1. In this case study we considered corrosion index values coming from a truncated normal distribution, (i.e., $0 < C^{IDX} < 1$), with

Table 5.2: Mission mix configuration (flight type probability per mission).

			Flight type		
Mix	I	II	III	IV	V
1	5.0 10^{-5}	7.5 10^{-5}	0.001	2.4 10^{-3}	0.011
2	2.5 10^{-5}	2.5 10^{-5}	1.2 10^{-3}	1.5 10^{-3}	6.75 10^{-3}
3	0	0	1.3 10^{-3}	2.3 10^{-3}	0.016
4	0	5.0 10^{-5}	9.5 10^{-4}	2.5 10^{-3}	8.5 10^{-3}
5	0	0	1.2 10^{-3}	2.3 10^{-3}	0.006
6	1. 10^{-4}	2.5 10^{-5}	7.5 10^{-4}	2.3 10^{-3}	6.13 10^{-3}
7	0	0	0	2.3 10^{-3}	7.3 10^{-3}
8	2.5 10^{-5}	0	0.001	2.5 10^{-3}	0.001
9	0	7.5 10^{-5}	7.5 10^{-4}	7.7 10^{-3}	0.001
10	7.5 10^{-5}	0	7.5 10^{-4}	2.4 10^{-3}	6.25 10^{-3}
Mix	VI	VII	VIII	IX	X
1	0.010	0.095	0.105	0.2475	0.5278
2	0.025	0.035	0.155	0.273	0.5025
3	0.025	0.025	0.105	0.273	0.5530
4	0.013	0.095	0.055	0.323	0.5030
5	0.015	0.045	0.105	0.2975	0.5278
6	0.015	0.045	0.08	0.3975	0.4528
7	0.015	0.070	0.055	0.3225	0.5278
8	0.02	0.02	0.12	0.285	0.5528
9	0.025	0.025	0.055	0.2225	0.6528
10	0.005	0.05	0.08	0.2725	0.5828

scale $\sigma = 0.05$, and location ($\mu_{C^{IDX}}$) defined by the modified PACER LIME model severity levels (see Figure 5.1). The corrosive severity in the PACER LIME model is defined based on factors such as sulfur dioxide (SO_2) and ozone (O_3) levels, relative humidity, yearly precipitation rate, and airport distance from the coast. In our modified model, this levels are:

- ‘‘AA’’: very severe corrosivity level, and $C^{IDX} = 1$,

- ‘‘A’’: severe corrosivity level, $\mu_{C^{IDX}} = 0.844$,

- ‘‘B’’: moderate corrosivity level, $\mu_{C^{IDX}} = 0.608$, and

- ‘‘C’’: mild corrosivity level, $\mu_{C^{IDX}} = 0.361$.

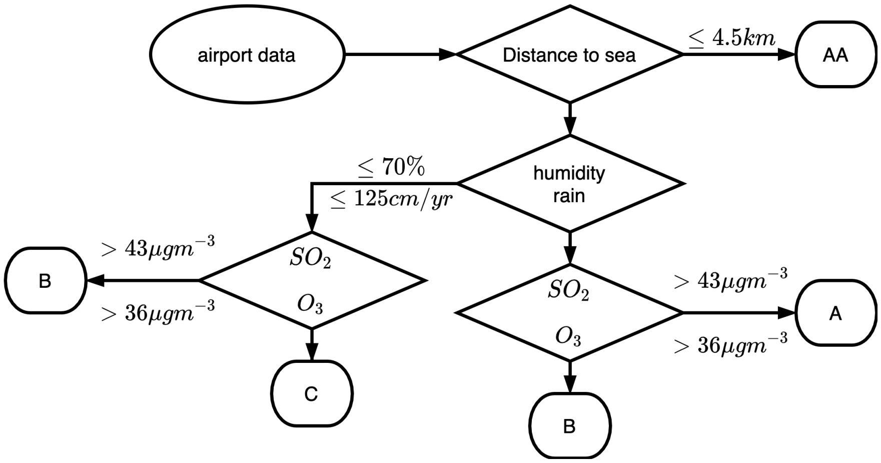


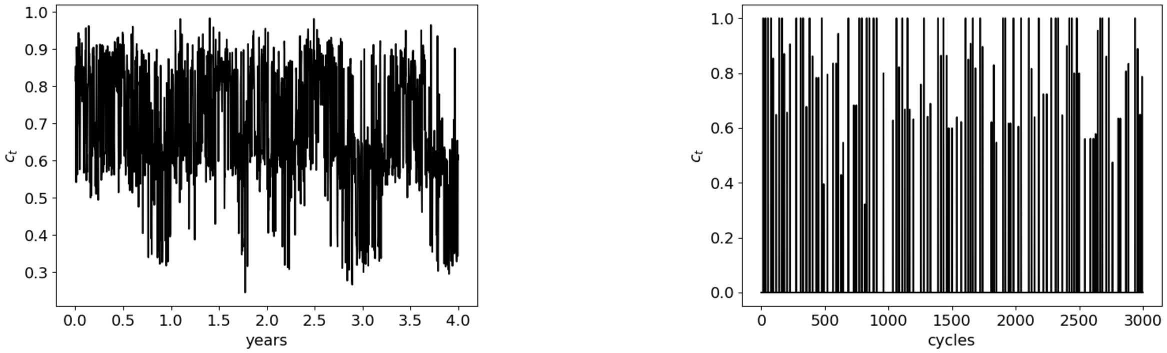
Figure 5.1: Modified PACER LIME model.

Airport data was collected and consolidated from the United States Environmental Protection Agency[1] and FrontierWeather[2]. Large and popular airports were chosen in the regions with matching data sets. Airports that lie within a 4.5 km range to the coast will always have a corrosivity index of 1 because of the ocean’s salinity. Otherwise, depending on their environmental factors, at any given time of the year airports will either have a ranking of A, B, or C (seasonality will play a role in that determination). To help illustrate the seasonality effect in the corrosivity index determination, Figure 5.2a presents the daily corrosivity index over four years of operation in a given airport.

On each flight mission two airports must be associated to a flight. To help structure this step in the synthetic simulation, we arbitrarily defined 10 routes. Hence, at each flight mission an aircraft is

[1] www.epa.gov

[2] www.frontierweather.com



(a) Daily C^{IDX} on a given airport.

(b) Aircraft C^{IDX} values per loading cycle.

Figure 5.2: Corrosivity index over time for: (a) - an airport (effect of seasonality), and (b) - an aircraft in the fleet (effect of seasonality and route structure).

randomly assigned to a given route, and therefore, to specific corrosivity indexes. Thus, the actual value of the corrosivity index for each aircraft on each flight is affected by the airports seasonality effect, and by the routes structure. Figure 5.2b exemplifies these effects on the aircraft corrosivity index by illustrating its evolution over the first 3,000 loading cycles of an airplane in the fleet.

Following all the previously discussed steps, the fleet simulation can be summarize as:

- an aircraft is randomly assigned to a route;
- at the beginning of each flight mission random draws are performed to define the corrosivity index (based on the route structure and the severity levels presented in Figure 5.1), and the mission mix configuration (see Table 5.2);
- the specific flight type for each mission is then randomly assigned following the mission mix probability distribution (see Table 5.2);
- based on the selected flight type, loads (i.e., ΔS and R values) are applied following the distributions detailed in Table 5.1.

Applying this steps we generate a novel data-set in which the crack propagation history of the synthetic fleet is as illustrated in Figure 5.3.

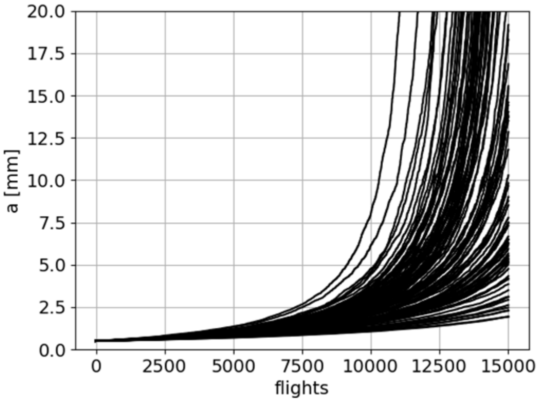


Figure 5.3: Crack propagation history in the synthetic fleet.

With this new data-set we retrained the recurrent neural network cell detailed in Section 4.2 considering the multi-layer perceptron configuration presented in Table 5.3, alongside the Mean Squared Error (MSE) loss function discussed in Eq. 5.1. It is worth mentioning that no attempts were made to further optimize the multi-layer perceptron architecture or its performance. The revised results for this bias estimation model will be discussed in the following sections in conjunction with obtained results for the three proposed cells discussed in this chapter.

$$Loss = MSE = \frac{1}{n} \times \sum_{i=1}^{n} |a_{PRED,i} - a_{OBS,i}|^2 \tag{5.1}$$

where as in Eq. 4.10, n is the number of observations (inspected aircraft), $a_{PRED,i}$ and $a_{OBS,i}$ are the predicted and observed crack lengths for the i^{th} inspection, respectively.

Table 5.3: Multi-layer perceptron (MLP) configurations used on each new recurrent neural network cell.

Layer #	**neurons/activation function**		
	Bias model MLP	$Bias_{\Delta K}$ model MLP	ΔK model MLP
0	10 / linear	5 / tanh	5 / tanh
1	5 / linear	1 / relu	1 / relu
2	1/ sigmoid		
Parameters[*]	131	38	51

Layer #	**neurons/activation function**	
	Log model C MLP	Log model m MLP
0	5 / tanh	10 / linear
1	1 / relu	5 / linear
2		1 / sigmoid
Parameters	27	97

[*] differences in the total number of parameters are due to differences in the number of inputs.

5.2 Reducing the Number of Inputs for the Data-Driven Layers: $Bias_{\Delta K}$ Modeling Approach

This functional form is a direct review of the bias estimator cell discussed throughout Chapter 4. The bias estimator present in Section 4.2 consisted of a general corrector term of functional form $MLP \rightarrow g(\Delta S_t, R_t, C_t^{IDX}, a_{t-1})$. Due to this functional form the correction terms proposed by the hybrid model to compensate for the missing physics of corrosion are directly impacted by the loading conditions (through ΔS_t and R_t values), current damage level (by a_{t-1}), and environmental conditions (through C_t^{IDX}). This implies that even though the functional form was conceived to compensate for environmental effects (i.e., the missing physics of corrosion), as formulated, it could potentially be influenced for instance, by unaccounted issues in the load estimation, and also attempt to compensate for its effects in crack propagation. In practice, as stated, the data-driven model detailed in Section 4.2 acts as a ‘‘black box’’, proposing correction terms without distinguishing the ‘‘root cause’’ for this correction.

In this section, the proposed functional form described by Eq. 5.2, aims to minimize the loading conditions impacts in the correction term. It is still derived as a bias term proposing bias values to compensate for the missing physics in the mechanical fatigue damage increment term ($\Delta a_{F,t}$).

$$\begin{aligned} a_t &= a_{t-1} + \Delta a_{F,t} + \Delta a_{CF,t} \\ \Delta a_{F,t} &= \frac{C_0}{(1-R)^{m(1-\gamma)}} (\Delta K_t)^m, \textit{ and} \\ \Delta a_{CF,t} &= MLP(\Delta K_t, R_t, C_t^{IDX}) \end{aligned} \tag{5.2}$$

where $\Delta a_{F,t}$ is damage increment due the mechanical fatigue, and $\Delta a_{CF,t}$ is the bias term related to the missing physics issue that the data-driven portion of the cell (i.e., the MLP) aims to learn during training.

The key modeling difference in this approach concerning the neural network cell presented in Section 4.2 is the inputs considered for the data-driven layers. In Section 4.2 the functional form proposed for the multi-layer perceptron considered previous estimates of crack length (a_{t-1}), far-field stresses (ΔS_t), stress ratio (R_t), and corrosion index (C_t^{IDX}) values as inputs, while here the data-driven layers only receive ΔK_t, R_t, and C_t^{IDX} as input information. Despite seemly looking like a minor modification to the cell, the physical interpretation, and numerical repercussion are very significant. While in the cell detailed in Section 4.2 the data-driven layer is acting as a generalized bias estimator, proposing general correction terms for the damage accumulation process, here the estimated bias values are more constrained in form. The revised functional form proposed here takes the stress intensity range ΔK_t as input, instead of observing directly the far-field stress ΔS_t values. This means that any issues concerning the estimation of these far-field stresses will not be directly observed by the bias estimator, but instead will indirectly manifest in the stress intensity range ΔK_t. Such features may hinder the ability of the proposed cell to identify any potential issues with the load estimation and thus inhibiting attempts for the data-driven model to correct these potential issues. The final result may end up being a bias estimator model more focused on the corrosion index effects in the crack propagation than with the loading effects in the damage accumulation. Hence, with this proposed functional form the hybrid model may indirectly be more sensitive to the corrosion index effects, and potentially be more effective in compensating for missing physics issues related to environmental effects (as being considered in this case study).

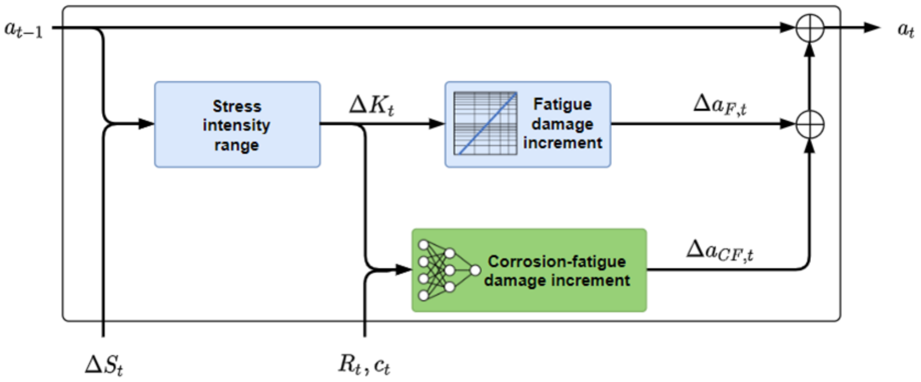


Figure 5.4: Recurrent neural network cell considering the $Bias_{\Delta K}$ model.

5.3 A More Physics-Constrained Cell: Log Modeling Approach

In this section, the functional form described by Eq. 5.3 is derived under significantly more constraining assumptions. As in the previously described hybrid cells, it still formulated as a bias estimator term for the mechanical fatigue damage increment ($\Delta a_{F,t}$), but contrary to the previous cells, the bias correction term ($\Delta a_{CF,t}$) is being directly calculated based on the curves presented in Figure 4.2b (where the physics-based layers estimate Δa_{MECH} considering material coefficients for pure air, while the data-driven layers estimate the ‘‘blue dashed lines’’ in the figure to infer Δa_{CORR}), with two multi-layer perceptrons being used to estimate the equivalent material coefficients C and m values.

$$\begin{aligned} a_t &= a_{t-1} + \Delta a_{F,t} + \Delta a_{CF,t} \\ \Delta a_{F,t} &= \frac{C_0}{(1-R)^{m(1-\gamma)}}(\Delta K_t)^m\textit{, and} \\ \Delta a_{CF,t} &= \Delta a_{F,t} \times (10^{\log MLP_C(R_t, C_t^{IDX}) + MLP_m(R_t, C_t^{IDX}) \times \log \Delta K_t}) - \Delta a_{F,t} \end{aligned} \tag{5.3}$$

where MLP_C and MLP_m are the multi-layer perceptrons tasked with estimating values for the material coefficients (i.e., Paris' law coefficients C and m, respectively).

This is a significant departure from the previously discussed cells from a modeling perspective. Here the proposed multi-layer perceptrons observe only the stress ratio (R_t) and the corrosivity index (C_t^{IDX}) at any given cycle in order to estimate values for the material coefficients. This implies that while this model will be heavily influenced by environmental effects and potentially capable of directly compensating for environmental effects and material properties variation in damage accumulation, it will most likely not be able to identify issues with load estimation and hence not being capable to directly compensate for its effects. It is important to highlight here that the keyword is direct compensation for the loading effects. Even though the loads are not directly observed by the data-driven model, its training is still guided by the crack lengths observed in the inspection. Any unexpected value for the crack lengths due to whatever unforeseen effects (e.g., loading issues, material capability variation, environmental effects, faulty sensors, etc.) would still be provided for the loss function during training and the hybrid model would still attempt to compensate for any unaccounted effect by proposing equivalent material coefficient values and correction terms. Nonetheless, the resulting hybrid model is potentially more sensitive to environmental effects (expressed through C_t^{IDX}) and material capability variation.

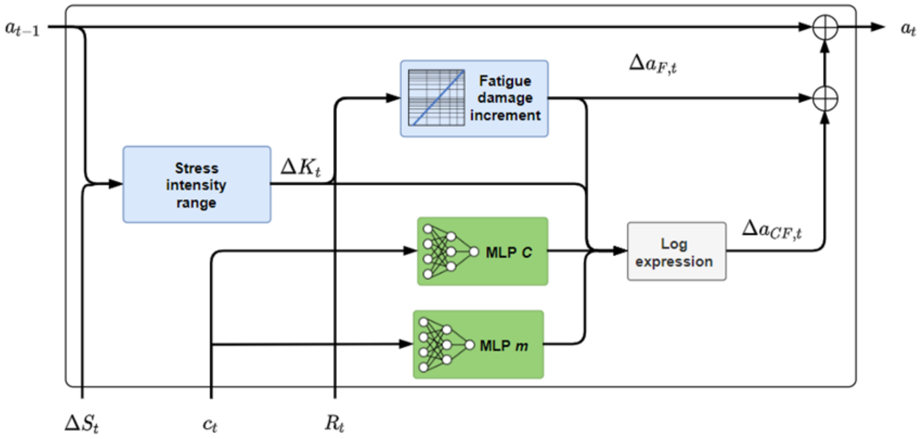


Figure 5.5: Recurrent neural network cell considering the *Log* model.

5.4 Focusing on the Loads Effects: ΔK Correction Modeling Approach

Lastly, in this section, we propose a functional form more focused on compensating for potential load estimation issues. As detailed in Eq. 5.4 in this hybrid model, the data-driven layers no longer act as bias estimators for the mechanical fatigue damage increment but rather aims to compensate for missing physics effects on the stress intensity range (ΔK_t) by proposing correction terms (bias values) for its value.

$$\begin{gathered}
a_t = a_{t-1} + \Delta a_t \\
\Delta a_t = \frac{C_0}{(1-R)^{m(1-\gamma)}}(\Delta K_t)^m \\
\Delta K_t = \Delta K_{F,t} + \Delta K_{CF,t} \\
\Delta K_{F,t} = F\Delta S_t\sqrt{\pi a_{t-1}}, \textit{ and} \\
\Delta K_{CF,t} = MLP(\Delta S_t, R_t, C_t^{IDX}, a_{t-1})
\end{gathered} \tag{5.4}$$

where $\Delta K_{F,t}$ is stress intensity range considering purely mechanical fatigue, and $\Delta K_{CF,t}$ is a bias term related to the potential missing physics that might be plaguing the loads estimation that the data-driven portion of the cell (i.e. the MLP) aims to learn how to compensate for during training.

This is a more extreme departure from the previously discussed cells from a modeling perspective. Here the focus is on the stress intensity range rather than on the damage increment itself. This is the least representative model for the considered case study of this chapter, given that in the proposed case study the source of the epistemic uncertainty is the missing physics of corrosion and not any issues with the loading estimation. However, as in the case of the *Log* modeling approach (see Section 5.3), the proposed cell is still capable of yielding accurate predictions by proposing equivalent stress intensity values to compensate for the missing physics of corrosion. Albeit this is not representative of the actual physics of degradation in the case study, the numerical results might be the same, as long as the hybrid model learns to stipulate a valid equivalency between the corrosion effects and the stress intensity range correction terms. As derived this might not be such a difficult task given that the multi-layer perceptron for the stress intensity range has as one of its inputs the corrosivity index C_t^{IDX}. Nevertheless, it is expected for the hybrid model resulting from this approach to be more sensitive to loading effects than to environmental or material capability effects.

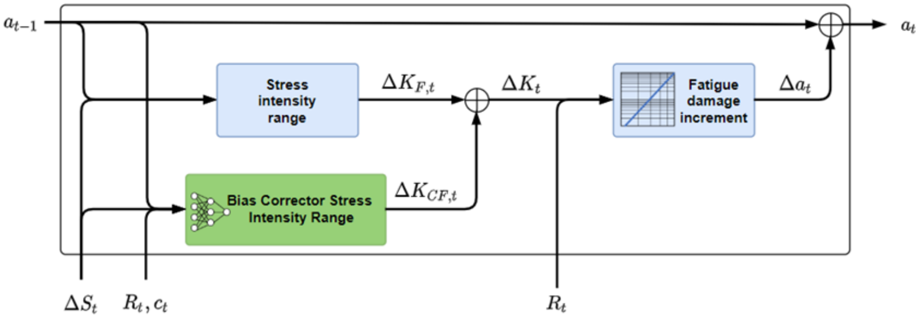


Figure 5.6: Recurrent neural network cell considering the ΔK model.

5.5 Diagnosis and Prognosis Analyses

Considering the revised data-set detailed on Section 5.1, all four recurrent neural network cells (the three cells presented in this chapter in addition to the bias cell discussed in Chapter 4) were trained considering crack length data for 10% of the fleet at 9,000 flight hours of operation (roughly around the second ‘‘C’’ check (see Table 2.3) in the fleet). Additionally, the auxiliary plane procedure described in section 4.2.3 for multi-layer perceptron hyperparameters initialization, were also adopted in the recurrent neural network cells training considering the multi-layer perceptron architectures detailed in Table 5.3, for all the considered formulations. All recurrent neural network cells discussed in this chapter were trained considering the Mean Squared Error (MSE) loss function illustrated in Eq. 5.1.

After all recurrent neural network cells were properly trained, the resulting hybrid models were used to perform crack length estimation after 9,000 flight hours of operation (diagnosis) and damage forecast after 12,500 flight hours ((around the third ‘‘C’’ check (see Table 2.3) and at half-point until the ‘‘D’’ check in the fleet)) of operation (prognosis) across the entire fleet (100 aircraft). Figure 5.7 presents the results of the crack length estimation at diagnosis for the entire fleet considering all proposed hybrid models. By comparing all four model predictions we can see that for relatively small crack lengths of 3 mm or lower, all four models seem to have a good agreement with the actual crack length values. This strongly indicates that, despite the different modeling approaches, all considered recurrent neural network cells are capable to capture the overall trend of the crack propagation in the fleet, and, in their own way, compensate for the missing physics of corrosion. Model behaviors start to diverge from one another when predicting larger crack lengths. For instance the ΔK *model* (see Section 5.4) is significantly overestimating crack lengths above 3 mm. This might be an indication that for larger crack lengths, the cumulative effects of compensating the missing physics of corrosion by adding a bias to the stress intensity range, start to be significant and starts to actively distort the neural network predictions. Nonetheless, even in this scenario, such a model would still be able to provide rough predictions, albeit extremely conservative, for the fleet crack lengths. All the remaining hybrid models that are based on providing a bias correction term to the mechanical fatigue damage increment, provide relatively good predictions regardless of the crack range.

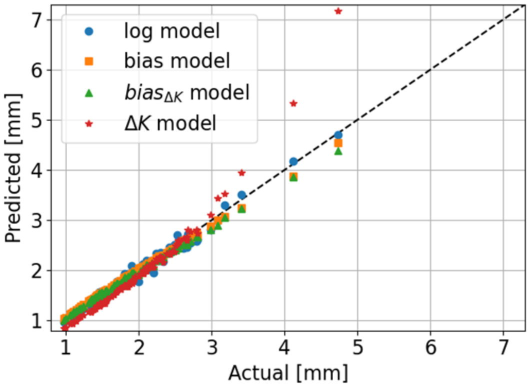
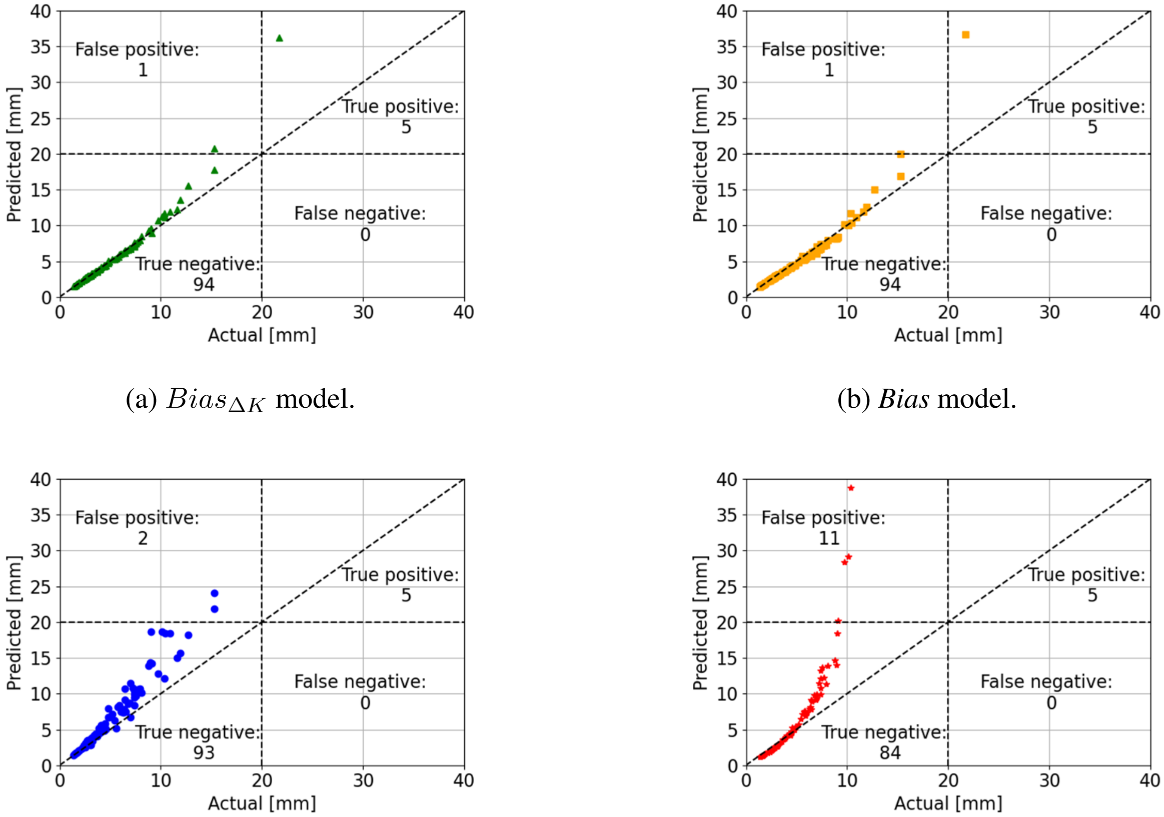


Figure 5.7: Recurrent neural network cells ($Bias_{\Delta K}$, Bias, Log, and ΔK models) prediction across the entire fleet after 9,000 flights of operation.

Figure 5.8 presents the results for the damage forecast analysis considering all of the proposed hybrid models. Using the same arbitrary threshold of 20 (mm) for repair/replacement flagging as shown in Chapter 4 discussion, at 12,500 flights, 5 aircraft in the fleet should be flagged. As seen in Figure 5.8 all proposed models are capable of accurately flagging the aircraft that requires action without presenting any false negatives (i.e., aircraft that should be flagged but were wrongly not). Their behavior differs in regards to wrongly flagging aircraft that do not require action (i.e. false positives). As seen in Figure 5.8a and 5.8b the so-called *Bias* and *Bias*$_{\Delta}K$ models are the more accurate models, presenting a minor cost of 1 false positive. The so-called *Log* model is marginally more conservative than the previous two models presenting 2 false-positive values as shown in Figure 5.8c As indicated by the overestimation results from the diagnosis analysis (see Figure

5.7), the so-called ΔK model end-up being significantly more conservative than the remaining models, presenting 11 false-positive flags as shown in Figure 5.8d. As previously discussed in safety-critical applications as in civil aviation, having no false negatives is extremely important (is highly undesirable to have an aircraft wrongly cleared to flight) for reliable and safe operation, while false positives will play a significant part in maintenance-related costs. Hence, is reasonable to conclude that despite the varying levels of conservativeness and the differences in formulation, all proposed models have a good performance in terms of fleet management.



(a) $Bias_{\Delta K}$ model.

(b) *Bias* model.

(c) *Log* model.

(d) ΔK model.

Figure 5.8: Recurrent neural network cells ($Bias_{\Delta K}$, *Bias*, *Log*, and ΔK models) prediction across the entire fleet after 12,500 flights of operation. Plot is clipped at 40 (mm) even though predictions can be higher than this value.

5.6 Summary

In this chapter, we revised the previously presented corrosion-fatigue case study to include seasonality effects in the corrosion index estimation through the consideration of a meteorological model and redefined the loading conditions to increase variability in the fleet. All these modifications were proposed aiming to hinder the task of distinguishing the true cause for the epistemic uncertainty in crack propagation.

The recurrent neural network cell presented in Chapter 4 were expanded into three novel hybrid cells, namely, *Bias*$_{\Delta K}$ *model*, *Log model*, and ΔK *model*, each derived based on distinct physics-based considerations and constraints. To evaluate the performance of the proposed cells, we presented a numerical case study focusing on diagnosis and prognosis analyses of the presented synthetic fleet. In the considered case, the physics-informed layers only account for purely mechanical fatigue. All proposed data-driven layers, need by their own means, to adjust the damage accumulation rate and compensate for the corrosion effects. Presented results indicate that despite the varying levels of the conservativeness of the hybrid models, all proposed hybrid cells are able to capture the overall trend of fleet crack propagation.

In the following chapter, we will focus our discussion on different forms of ranking and segregation of the proposed hybrid models, besides evaluating any possible gain in merging the hybrid models predictions.

CHAPTER 6: MODEL SEGREGATION AND FORECAST WINDOW - A CASE STUDY ON ENSEMBLES OF MODULAR HYBRID NEURAL NETWORKS

In this chapter, we will focus our attention on the decision-making effort related to the previously discussed hybrid models. Based on cross-validation errors, we will evaluate two possible routes for the previously derived model ‘‘pool,’’ either segregate the models aiming to select the model in the ‘‘pool’’ that better represents the observed data; or by employing a weighted ensemble derived from the base models in the ‘‘pool’’ improve the overall prediction accuracy.

Concerning the model selection task, the analysis is straightforward. We assume that cross-validation errors are directly related to the true error of the models, and provide a valid estimator of the prediction true error (which is never observable in real applications given that we would not have access to true crack length values) From sets of cross-validation errors, we evaluate the validity of this assumption and how well can the cross-validation errors be used to rank the models in the ‘‘pool.’’

Regarding the ensemble analysis, we adapted the weighted average surrogate procedure discussed in Viana et al. [195] to build weighted ensembles from the hybrid neural networks predictions and improve the overall accuracy of the damage forecast. Additionally, we briefly discuss how the uncertainty in the ensemble predictions can be used to evaluate the trust in the ensemble forecast. From this analysis, we aim to determine a time window in which we can safely forecast the crack propagation in the fleet.

Overall both analysis can be summarized by the following fundamental questions:

- Can we use the cross-validation errors to properly rank the base models?
- Based on cross-validation errors can we combine the model predictions?
- It is advantageous to combine base model predictions?
- Does the uncertainty in the ensemble relates to its prediction error?

The remainder of this chapter focuses on the discussions associated with these questions.

6.1 Model Ranking and Segregation

In this section, we evaluate the use of cross-validation errors to directly infer the ranking of the base models and guide the selection of the best base model, or at least to properly determine what is the ‘‘best’’ model in the pool. To carry such an analysis we first need to define a cross-validation procedure from which we will extract the cross-validation errors. A cross-validation error is an error at a data point when the model is fitted/trained to a subset of the data points not including such a point. As detailed in Section 4.2 we adopted cross-validation procedures as presented in Kohavi et al. [192]. To perform this analysis we adopted the following measures:

- We started by splitting the synthetic fleet into two portions, 60 aircraft to compose a test set, and the remaining 40 aircraft were used to derived training sets.
- Considering these 40 aircraft, 20 sample sets containing each 20 aircraft were defined.
- On each sample set, training of each of the four base models described in Chapter 5 is performed at 9,000 flights (i.e., each base model is retrained on each sample set composed of 20 aircraft each).

- After the training of the neural networks was completed, a cross-validation analysis, considering a ‘‘leave-one-out’’ strategy was carried, i.e., at each iteration of the cross-validation analysis 1 aircraft is removed from the training set, and all the base models are retrained considering the remaining 19 aircraft on the sample set. After the training considering the remaining 19 aircraft is finished, the predictions are checked against the aircraft left out. As previously detailed cross-validation errors are defined as $\mathbf{e}_{XV} = \mathbf{a}_{XV} - \mathbf{a}_{OBS}$, where $\mathbf{a}_{XV}$ are the cross-validation predicted crack lengths for the left out aircraft and $\mathbf{a}_{OBS}$ are the actual observed crack lengths of said aircraft. This procedure is repeated until all aircraft in the sample set as being removed once, i.e., at each sample set the cross-validation analysis consists of 20 iterations. Hence, at the end of the cross-validation analysis, 400 cross-validation error values will be generated for each base model.

As detailed the computational cost of such cross-validation analysis is relatively high but not prohibitive. Considering a laptop configured with an Intel Core i7-7820HQ CPU at 2.90GHz, 32GB of RAM, and NVIDIA Quadro M620 graphical processing unit running Windows 10, and 10 epochs on each cross-validation iteration, each iteration for each model takes on average 540 seconds. For all 1600 cross-validation iterations (20 iterations for each of the 20 sample sets and each of the 4 base models) considered in this chapter, the total run time was roughly 240 hours or 10 days. It is worth mentioning that with parallel computing this computational cost would be significantly reduced.

Performing the above procedure, three distinct sets of errors are generated:

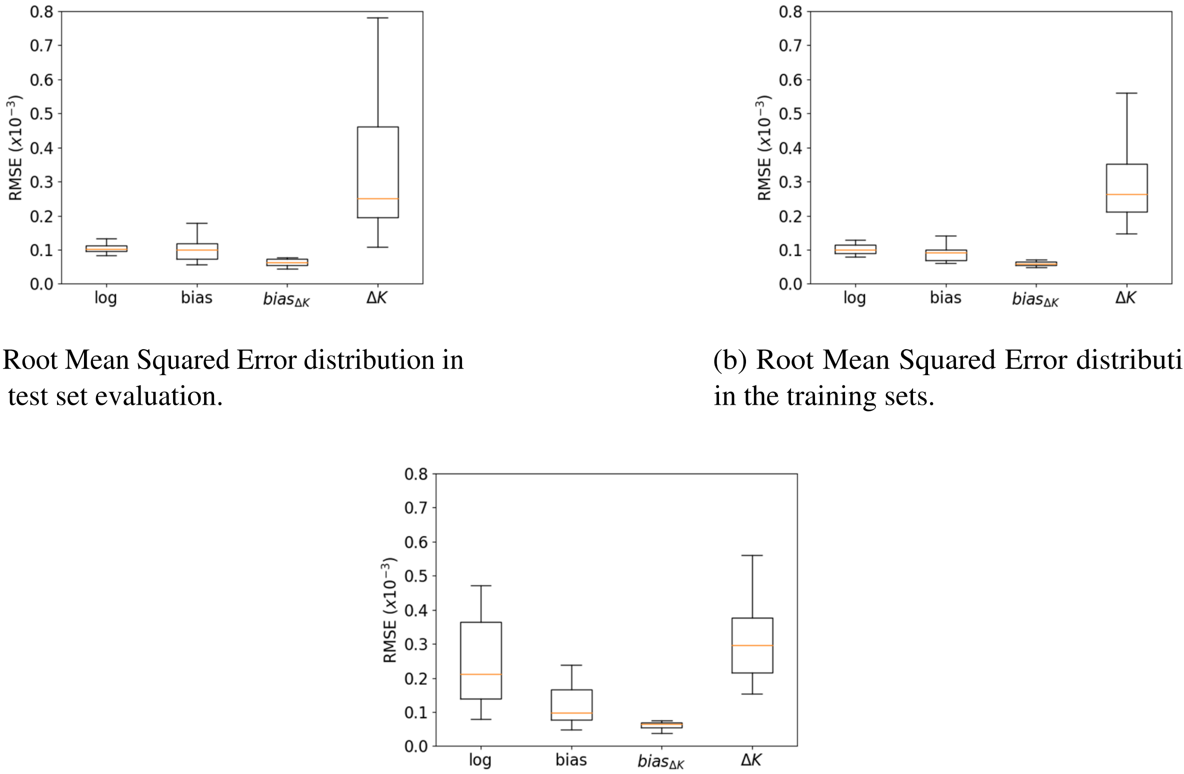
- a test error set, that in this analysis is the estimator of the actual prediction error for each base model. It is worth mentioning that such a set would not be viable in most real applications given that crack length information for the entire fleet would not be available.

- training error sets for each considered sample set. Most likely in a real application, only one or very few training error sets would be available given that this information is derived from crack length values obtained through inspection campaigns.

- cross-validation error sets, that in this contribution are being considered as valid estimators for the base models true prediction error. It is important to remind that in real application true errors would not be available and in this section, we evaluate the assumption that the cross-validation errors are good estimators of such information.

Figure 6.1 illustrates the error distribution for each base model on each of these sets. As shown by Figure 6.1b the training error sets distribution is very close to the error distribution of the test error set (see Figure 6.1a), with the more distinct difference concerning the spread of the ΔK model. Concerning the cross-validation errors distribution, we can see from Figure 6.1c that the spread of the *Log* and *Bias* models significantly increases when compared to the test set errors shown in Figure 6.1a. Considering the overall behavior of all error sets illustrated in Figure 6.1 it seems that the cross-validation errors can safely indicate the best model in the pool, but inferences of the remaining models ranking could be distorted.

To better evaluate the behavior indicated by Figure 6.1 we performed the confusion matrices analysis shown in Figure 6.2. Here we directly compared the ranking given by cross-validation errors against the ranking derived from the true prediction errors in the test set (in the figure 1^{st} means the best model, 2^{nd} indicates the second-best model, etc.). As it can be seen in Figure 6.2a using the cross-validation errors we can safely infer the best model in the pool (that for the case study considered in this chapter is the *Bias*$_{\Delta K}$ model) and for the vast majority of the cases also infer the correct ranking of the remaining models. Such results are strong indications that cross-validation errors can be used to safely segregate the models in the fleet and help to select the base model that better explains the observed crack lengths.



(a) Root Mean Squared Error distribution in the test set evaluation.

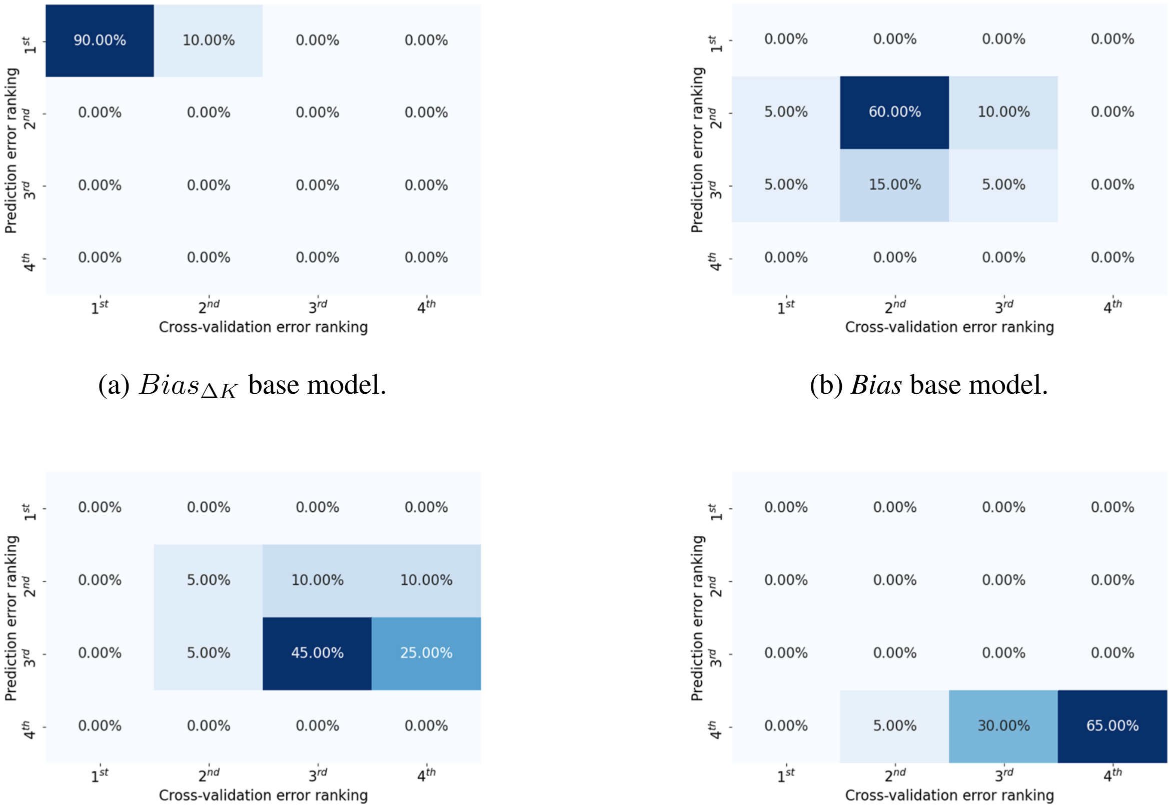
(b) Root Mean Squared Error distribution in the training sets.

(c) Root Mean Squared Error distribution in the cross-validation analysis.

Figure 6.1: Root Mean Squared Error (RMSE) behavior on the throughout the test set, training sets and cross-validation analyses.

6.2 Ensemble Analysis

In this section, we focus on the discussion of using cross-validation errors to combine base model predictions and increase the overall prediction accuracy. To do so we derived ensembles of the base neural network models by adapting the weighted average surrogate procedure detailed in Viana et al. [195]. A weighted average surrogate (WAS) intends to take advantage of n surrogates (that in this contribution are the previously derived base models, namely, *Bias*, $Bias_{\Delta K}$, *Log*, and ΔK models) in the hope of canceling errors in prediction through proper weighting selection in the



(a) $Bias_{\Delta K}$ base model.

(b) *Bias* base model.

(c) *Log* base model.

(d) ΔK base model.

Figure 6.2: Confusion matrices for the model ranking based on cross-validation errors.

linear combination of the models. Mathematically, this can be summarized as shown in Eq. 6.1.

$$\begin{gathered} a_{Ens.} = \sum_{i=1}^{n} \omega_i \times a_i = \boldsymbol{\omega}^T \boldsymbol{a}, \textit{ and} \\ \sum_{i=1}^{n} \omega_i = 1 \end{gathered} \tag{6.1}$$

where $a_{Ens.}$ is the predicted crack length by the resulting ensemble, ω_i is the weight associated with the ith base model, and a_i is the predicted response by the respective base model.

As detailed by Viana et al. [195], given a $\mathbf{C}$ matrix of base models cross-validation errors, the optimal weighted ensemble can be found through the minimization problem shown in Eq. 6.2.

$$\begin{aligned} &\min_{\boldsymbol{\omega}} \boldsymbol{\omega}^T \boldsymbol{C} \boldsymbol{\omega} \\ &s.t. : \mathbf{1}^T \boldsymbol{\omega} = 1 \end{aligned} \tag{6.2}$$

where $c_{ij} = \frac{1}{p} \boldsymbol{e}_i^T \boldsymbol{e}_j$, p is the number of data points in the cross-validation, and $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are the cross-validation errors of base models i and j, respectively.

The solution of the optimization detailed in Eq. 6.2 is obtained through Lagrange multipliers as presented by Eq. 6.3. To avoid issues with negative weights as well as weights larger than one, Eq. 6.3 is solved considering only the diagonal terms of matrix **C**.

$$\boldsymbol{\omega} = \frac{\boldsymbol{C}^{-1} \mathbf{1}}{\mathbf{1}^T \boldsymbol{C}^{-1} \mathbf{1}} \tag{6.3}$$

Figure 6.3 illustrates a possible set of weights for the neural network ensemble considering the cross-validation errors of a given sample set. For simplicity presented weights were rounded up. As we can see the derived weights highly favor the *Bias*ΔK model and hardly differentiate among the remaining base models. This a direct reflection of the behavior illustrated in the cross-validation error set considered in this example. A possible explanation for the ensemble weights to slightly favor the ΔK model over the *Log* model in the example shown in Figure 6.3, despite the ΔK model clearly overestimation of larger crack lengths, is most likely due to the fact of the best model in this case (the *Bias*ΔK model) slightly underestimation of such cracks.

In Figure 6.4 we are presented with a direct comparison between the weights distribution across the sample sets when test errors (Figure 6.4a) and cross-validation errors (Figure 6.4b) are used in the optimization problem shown in Eq. 6.2. As with the confusion matrices analysis of Figure 6.2 we

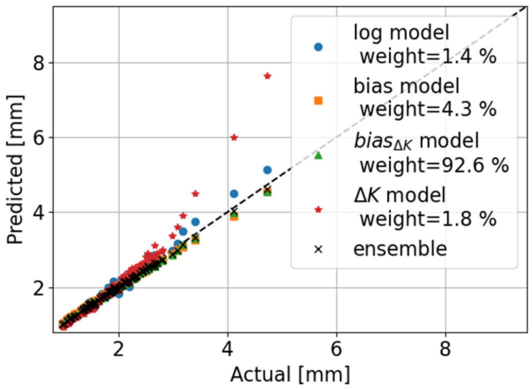


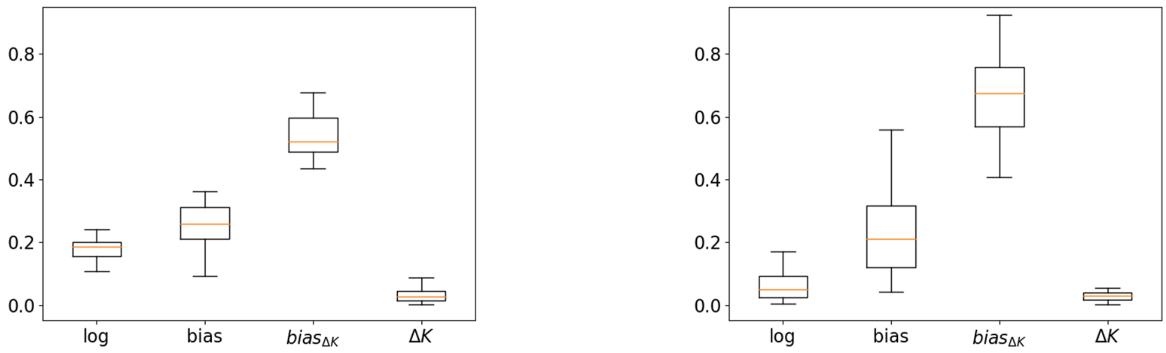
Figure 6.3: Example of an ensemble prediction. Weights derived considering the cross-validation errors in a given sample set.

can see that when using the cross-validation errors (see Figure 6.4b) the final ranking expressed by the model weights largely favor the Bias_{DeltaK} model and heavily minimize the $DeltaK$ model contributions. However, despite being minimal the contributions $DeltaK$ model is still relevant as illustrated in the discussion of previous results. By heavily overestimating larger crack lengths, $DeltaK$ model can help balance out crack lengths underestimation from better models in the ensemble.

The results illustrated by Figures 6.3 and 6.4 helps answer the questions related to using cross-validation errors to rank the base models and combine base model predictions. Now we focus our attention to address the advantages of using such ensembles in comparison to just using the best base model predictions. In order to answer this fundamental question we used the metric expressed in Eq. 6.4.

$$\%diff = 100 \times \left(\frac{RMSE_{Best} - RMSE_{Ens.}}{RMSE_{Best}} \right) \tag{6.4}$$

where RMSE refers to the root mean squared error, and $RMSE_{Best}$ is always the best base model based on test set errors on a sample set basis. Therefore it changes with the considered data set and



(a) Weights based on test set errors.

(b) Weights based on cross-validation errors.

Figure 6.4: Weights associated with which base model in the ensemble. Distributions derived by considering all 20 sample sets.

it is an idealization.

Using the metric defined in Eq. 6.4 we can evaluate the following scenarios:

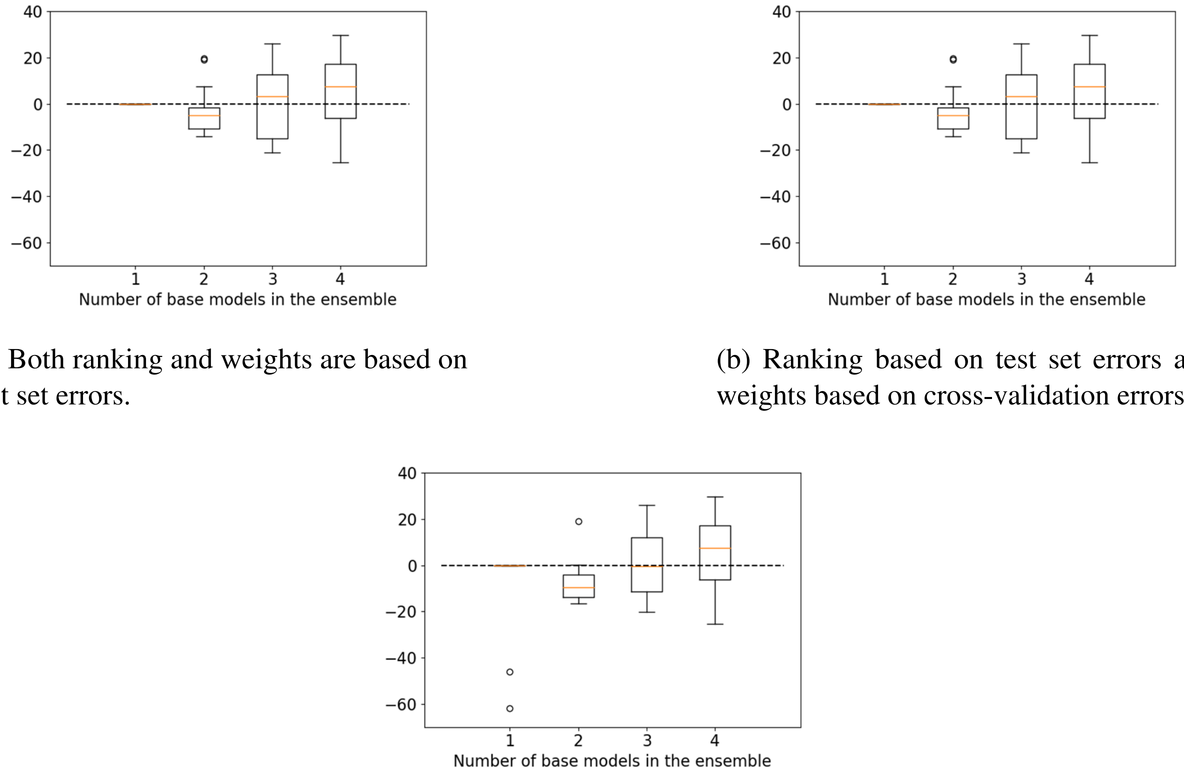
- *Ranking and ensemble weights based on test errors set:* this is an idealized scenario and illustrates the best case scenario possible, where true prediction errors are known and no uncertainty plagues the ranking task and the model weights. Such a scenario is not possible in real applications given that true prediction errors are unknown.

- *Ranking based on test errors and ensemble weights derived from cross-validation errors:* this scenario is also unrealistic given that the base model ranking is based on true prediction errors. However, this scenario helps us evaluate the impact of the cross-validation errors in the ensemble weights.

- *Ranking and ensemble weights derived from cross-validation errors:* this scenario is closest to real applications, with the exception that in real applications $RMSE_{Best}$ in Eq. 6.4 would not be known.

Figure 6.5 illustrates the results for the previously described scenarios concerning models ranking and ensemble prediction at 9,000 flights. Except for two outliers in Figure 6.5c we can see that when the number of base models is equal to 1 (i.e., the ensemble is composed only of the best base model) the metric described in Eq. 6.4 is zero. This illustrates once again how the ranking based on the cross-validation errors can safely and accurately identify the best base model in the pool ($RMSE_{Best}$ derived from the true prediction errors expressed here by the test set errors).

When the number of base models in the ensemble is larger than 1, the metric from Eq. 6.4 can either be positive or negative. Negative values indicate that for the considered weight combination in the ensemble, the predictions will be worse than the ones generated by the best base model, meaning thus that, at this condition, it does not pay off to combine the base models. Positive values indicate that the ensemble predictions are more accurate than the ones generated by the best base model and thus it is worth it to combine the model predictions. In the scenario where both model ranking and ensemble weights are derived from true prediction errors (see Figure 6.5a), negative values are indications that the generated ensembles are inheriting underlying biases from the base models (e.g., all base models have an overestimation trend), given that the ensemble weights are derived from an error minimization procedure (see Eq. 6.2), and in this scenario should present errors that are at least equal to the best base model in the ensemble. The overall behavior in Figures 6.5a and 6.5b are very similar showing once more that the cross-validation errors are good estimators for the true prediction error in the ensemble weights estimation procedure. Figure 6.5c show how the losses on accuracy when using the cross-validation errors for model ranking and ensemble weight estimation are minimal. Again, this is another clear indication that we can safely use the cross-validation errors for base model ranking and combining the base model predictions. Additionally, from Figure 6.5c we can safely conclude that on average it does pay off, in terms of accuracy, to combine the base model predictions and as the number of base models in the ensemble increases so does the gain associated with using the ensemble. Lastly, in Figure 6.5 we can clearly observe that the best overall results are obtained when considering all base models in the ensembles. This result further collaborates with the observation that even a relatively ‘‘poor’’ base model actively contributes to the ensemble accuracy, and suggests that there is no gain in removing the ‘‘worst” base model from the ensemble.



(a) Both ranking and weights are based on test set errors.

(b) Ranking based on test set errors and weights based on cross-validation errors.

(c) Both ranking and weights are based on cross-validation errors.

Figure 6.5: Evaluation on the gain of using an ensemble of the neural network predictions over using solely the predictions of the best base model.

6.3 A Heuristic for Ensemble Forecast

In this section, we focus our attention on addressing the last fundamental question proposed in this chapter: *does the uncertainty in the ensemble relates to its prediction error?* Since the ensemble is generated by combining the predictions of some base models it is reasonable to assume that the ensemble prediction starts to lose its validity when the actual value no longer lies within the prediction range covered by the ensemble base models. This does not mean that prior to this point in time the ensemble prediction error is not significantly large, it only means that at such condition is no longer possible for the ensemble to generate an accurate prediction. Figure 6.6 illustrate the key concepts behind this idea.

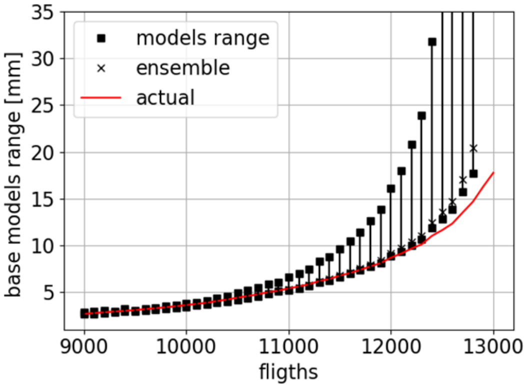


Figure 6.6: Example of base models range propagation in a given ensemble. At some point the base models range that generates the ensemble no longer contains the actual crack length value. This is the instant in which we should start to be concerned by the ‘‘validity’’ of the ensemble damage forecast, given that is no longer possible for the ensemble to yield an accurate prediction.

The issue however is that we only have access to actual crack length values in training (i.e., crack values coming from inspection campaigns), meaning that in the damage forecast we do not have any information concerning the actual crack length value or the ensemble true prediction error. However, in Figure 6.6 we can observe another important feature of the neural network ensemble for cumulative damage modeling. The base model prediction range grows over time, given that the cumulative error associated with each base model prediction grows with the continuous increment in the number of cycles. Hence, in this section, we aim to derive an uncertainty metric for the ensemble damage forecast based on this range growth. At any instant in the ensemble forecast, we know the ensemble range for each aircraft, and the ensemble range of the entire training set (that in this chapter consist of each sample set). Thus, a reasonable approach would be to combine such information in the desired uncertainty metric. Here, we propose the uncertainty metric given in Eq. 6.5, in which the growth rate of the ensemble range is approximated by a range ratio.

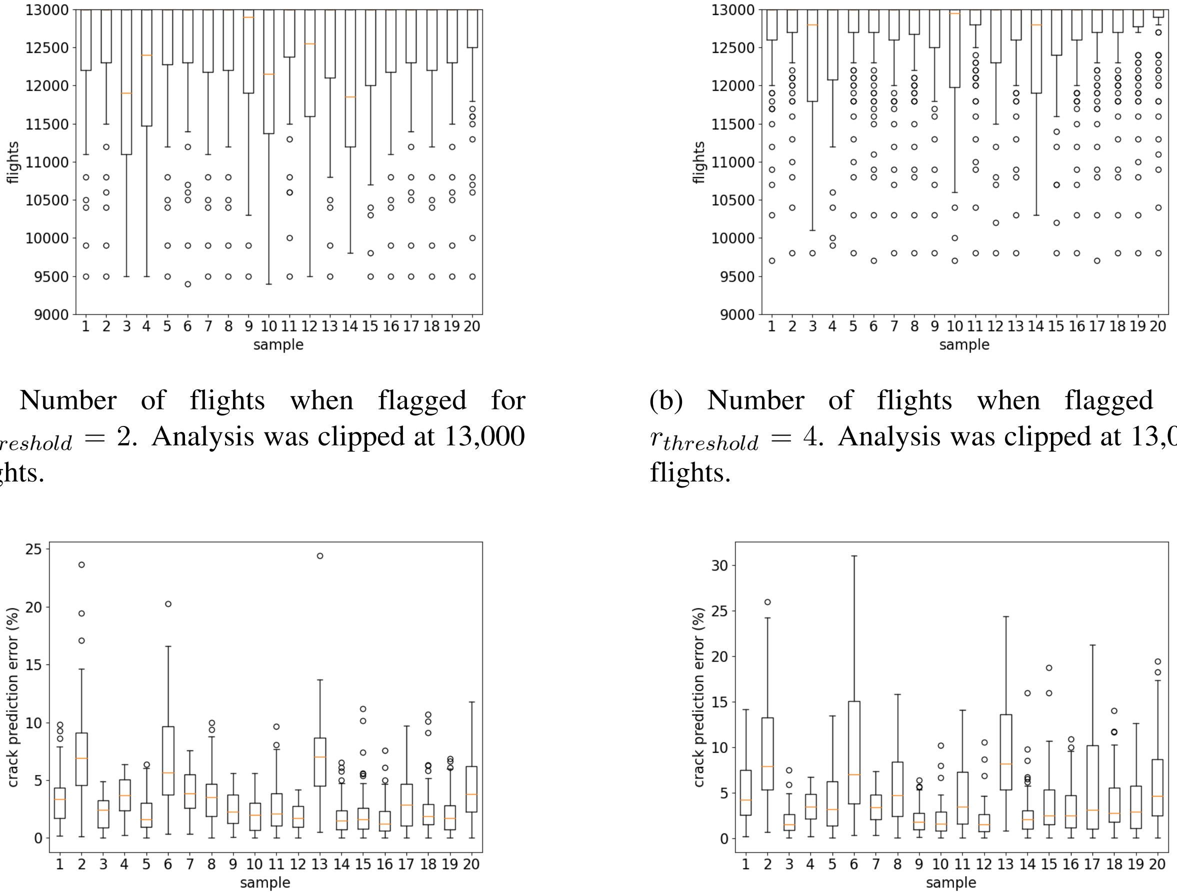
$$r = \frac{range_t}{range_{training}^{MAX}} \tag{6.5}$$

where $range_t$ is the current range of the ensemble base models predictions (see Figure 6.6) for a given aircraft, and $range_{training}^{MAX}$ is the maximum observed range during training in a sample set basis. Therefore, despite r being define on an aircraft basis and applied individually for each aircraft, $range_{training}^{MAX}$ is defined on a sample set basis. Additionally, r is indirectly evaluating the range growth rate on damage forecast for each aircraft in the fleet. Thus, by imposing an arbitrary threshold on r we define a window of damage forecast for each aircraft. On such a window we can not categorically affirm that the ensemble prediction is accurate or even valid, given that in real applications we never know the true crack length value in damage forecast, but we can express a level of confidence in the ensemble predictions.

Similar to the risk assessment analysis discussed in Chapter 3, the definition of the threshold value for r would depend of operators and practitioners attitudes towards risk. Since $range_{training}^{MAX}$ is a known value observed in the training set (e.g., 3 mm) by imposing a threshold for instance equals to 3, an aircraft would be flagged when its range is 3 times the value of the maximum training range (e.g., 9 mm), given that from Eq. 6.5 $range_t = r \times range_{training}^{MAX}$. Thus, operators and practitioners could also opt to impose such thresholds based on a maximum allowable range.

Figure 6.7 presents a comparison between two distinct values for r threshold ($r_{threshold} = 2$ and $r_{threshold} = 4$). Figures 6.7a and 6.7b illustrates the distribution for the number of flights in which the ensemble predictions for each aircraft was flagged on a sample set basis. It is clear that smaller thresholds impose more conservative flags, but on average, in both scenarios, this approach would allow for around 3,000 flights of damage forecast with worst-case scenarios revolving around 500 flights of extrapolation. It is worth mentioning that these aircraft with fewer flights of allowable damage forecast are most likely aircraft that deserve special attention on fleet management and should most likely be indeed flagged for mitigation actions (e.g., inspection, derating, repair/replacement, etc.). True prediction errors for each aircraft when flagged for considered sample sets in this case study are shown in Figures 6.7c and 6.7d. On both scenarios, average errors are small (roughly around 5 to 10%) with maximum errors around 25 to 30%. Once again these results indicate that the proposed metric allows for safe extrapolations and damage forecast, but by no means are direct measures of the ensemble true uncertainty. Reminding that on real applications the true prediction errors are unknown and such analysis as shown in Figures 6.7c and 6.7d would not be possible

Figure 6.8 illustrates a more detailed analysis of the ensemble accuracy when its crack length extrapolation is flagged. As seen in Figure 6.8a some aircraft are being very conservatively flagged with their predictions very close to the actual crack length value (the 45°dashed line). This suggests that for the scenario illustrated by the case study considered in this chapter such r threshold might be too conservative and that the actual crack value still lies within the ensemble range. Now on Figure 6.8b we start to observe a more clear departure from the ensemble predictions and the actual crack lengths. Such behavior seems to suggest that the ensemble is closer or above the true threshold in which its range no longer contains the actual crack length. But is important to highlight that even in this scenario prediction errors are not very large and we can be relatively confident in the ensemble



(a) Number of flights when flagged for $r_{threshold} = 2$. Analysis was clipped at 13,000 flights.

(b) Number of flights when flagged for $r_{threshold} = 4$. Analysis was clipped at 13,000 flights.

(c) Crack prediction error when flagged for $r_{threshold} = 2$.

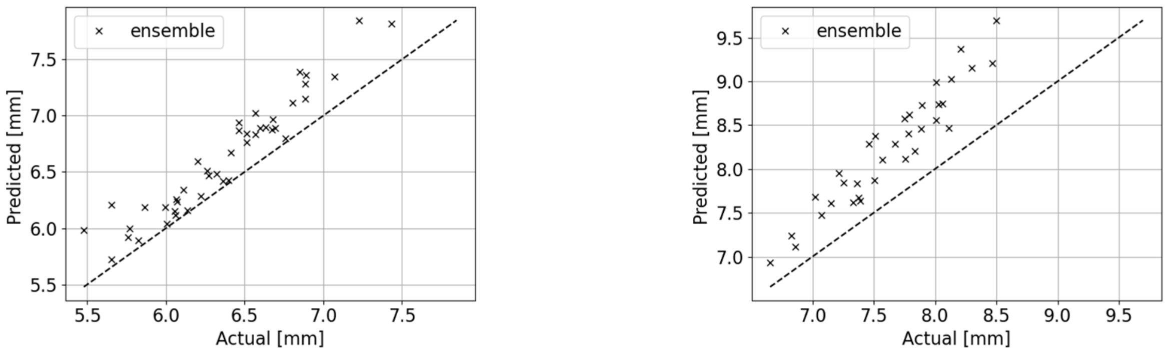
(d) Crack prediction error when flagged for $r_{threshold} = 4$.

Figure 6.7: Ensemble behavior throughout the samples in the instant that each aircraft crack extrapolation is flagged for different values of $r_{threshold}$.

prediction accuracy.

6.4 Summary

In this chapter, we evaluated the use of cross-validation errors to either properly rank the predictions of the previously defined hybrid models pool for the considered corrosion-fatigue case study, or



(a) $r_{threshold} = 2$.

(b) $r_{threshold} = 4$.

Figure 6.8: Ensemble prediction against actual crack length comparison in a given sample set at the instant that each aircraft crack extrapolation is flagged for different values of $r_{threshold}$.

through weighted ensembles combine the neural networks predictions and increase the overall prediction accuracy. Concerning the model ranking task, we evaluated the relation between the cross-validation errors and the unknown true prediction errors. Results showed that the cross-validation errors act as good estimators for the base models prediction error allowing for the proper ranking of the models.

Regarding the ensemble analysis, we adapted an existing weighted average procedure to build weighted ensembles from the hybrid neural networks predictions and improve the overall accuracy of the damage forecast. Additionally, we also evaluated the use of an indirect metric of the ensemble uncertainty to infer the confidence in the ensemble forecast and define a time window for crack length extrapolation in the fleet. Obtained results indicated that the use of cross-validation errors allows for proper ranking of the base models and to reliability derive model ensembles that on average increases the overall prediction accuracy. Additionally, depending on practitioners attitude towards risk the proposed uncertainty metric provides a good indicator for safely performing damage forecast in the fleet.

CHAPTER 7: SUMMARY AND FINAL REMARKS

Motivated by the growing availability of data and computational power as well as the advances in hybrid modeling frameworks, capable of merging elements of physics, machine learning, and statistical learning, in this dissertation, we focus on the development of novel approaches to minimize the impact of unforeseen factors in fleet management. To this end, proper comprehension of hardware degradation and accurate predictions of remaining useful life is crucial. Despite the advances in uncertainty propagation in prognosis and health management, uncertainty estimations on a fleet level are still a challenging task. Multiple sources of uncertainties can act asynchronously throughout the fleet usage history and must be properly handled by the fleet model.

In this dissertation, we focus on the challenges associated with accounting for the impacts of such unforeseen factors on two specific stages of a component service life; early-file and end-life. Two numerical case studies are derived to emulate two common issues in fleet management; manufacturing issues leading to an infant mortality problem, and unexpected exposure to harsher environments by operators, accelerating wear-out and significantly reducing component's useful life.

In Chapter 3, we studied early life failures as applied to fleet management. Depending on the scale of the problem, early failures can have a significant impact on the safety, availability, and operational profit of industrial equipment. We used Bayesian networks to model fleet reliability as well as to characterize and quantify a manufacturing-related problem (material capability degradation). We designed a simple numerical experiment where:

- degradation in material capability is used to characterize infant mortality, and
- fleet commissioning is a function of time.

We have studied:

- The effect of degradation in material capability: results confirmed that it dramatically reduces fleet reliability.
- The effect of fleet commissioning over time: results highlighted how it masks the increase in fleet unreliability.
- Fleet size and number of failures interaction and its effects on the infant mortality characterization: in the considered numerical experiment, the proposed framework was able to characterize the percent of the fleet plagued by poor material capability. Nevertheless, results illustrated how the fleet size can hinder the characterization of the problem, especially for small fleet operators.
- Bayesian models for assessing possible mitigation approaches: the proposed framework was able to evaluate fleet recommissioning as a risk mitigation procedure (i.e. derating assets to reduced load levels). Presented results indicate that the proposed model could be used as a tool for fleet management and aid operators to decide on a proper action course. For example, recommissioning mitigates the loads and therefore slows the failure rate down (at the cost of performance loss).

The results motivate us to extend the study in several aspects. For example, to have a better understanding of the impact that early failure in fleets of assets, we suggest extending the study and include, among other factors:

- Improved physics of failure models: not only by separating the cycles spent in initiation and propagation, but also improving the stress models to account for geometry and boundary conditions.

- Other sources of uncertainty, such as:
  - field inspection: qualitative inspection (distress ranking) instead of damage measurements
  - service level (repair versus replace failed units).

In Chapter 4 we proposed modeling damage accumulation by simultaneously using physics-informed and data-driven layers within a recurrent neural network cell. The major achievement of such a hybrid model is the ability to use the data-driven layers to compensate for the missing physics in reduced-order models and accurately estimate damage accumulation. In our numerical experiments, this reduced the need for observing the output of interest at all times. In other words, the physics-informed layers reduce the need for large datasets found in purely data-driven approaches.

In order to evaluate the performance of the proposed framework, we presented a numerical case study focusing on corrosion-fatigue crack propagation. In the considered case, the physics-informed layer only accounts for purely mechanical fatigue. The data-driven layer, on the other hand, adjusts the damage accumulation rate and compensates for the corrosion effects (i.e., working as a bias corrector).

We designed a numerical experiment where we simulated *(a)* a fleet of 150 aircraft flying different mission mixes (which implies in the variation of both mechanical loads and exposure to corrosive environments), and *(b)* inspection of 10% of the fleet at the end of the 5^{th} year of operation. With the help of this numerical experiment, we have studied:

1. *Initialization of the neural network parameters:* we proposed a simple strategy for initializing the data-driven portion of our hybrid model (a multi-layer perceptron). We argue that engineers and scientists would be able to prescribe the first-order effects of input variables in

the damage accumulation rate. We observed that when neural network parameters are poorly initialized, their optimization is not guaranteed.

2. *Complexity of neural network architecture and the number of available observations during training:* we performed an empirical study varying the depth and activation functions used in the multi-layer perceptron of our hybrid model. For this particular problem and data set, we found a marginal influence of the chosen architecture in the final prediction capability of the recurrent neural network. We believe this is because the ratio between available data and the number of trainable parameters remained high (i.e., a large number of data points compared to hyperparameters) across the multi-layer perceptrons studied here. In training, we observed data for 15 aircraft flying on average 174,000 cycles (around 2,610,000 observed input conditions) while the number of trainable parameters in the considered multi-layer perceptrons ranged from 31 to 1,241.

3. *Use of the hybrid models for diagnosis and prognosis:* we used the obtained model to predict damage across the fleet close (short-term analysis) and further (long-term analysis) of the inspection point. We learned that the model can successfully forecast damage both in the short-term (one year after partial fleet inspection) and in the long-term (up to five years after partial fleet inspection). The model was slightly conservative, which resulted in a manageable number of false positives (aircraft with damage wrongly tagged above an arbitrary threshold) and no false negatives (aircraft mistakenly tagged with damage below the arbitrary threshold).

Despite the many desirable features, the results indicate that the proposed methodology has the following deficiencies (potential topics for future research):

1. *High sensitivity of training results concerning the initial neural network hyperparameters:* even using the proposed auxiliary planes for the neural network initialization, a poorly

initialization of such parameters can potentially derail the training. Therefore, optimization with multiple initial hyperparameter values is still needed.

2. *Model predictions do not interpolate crack length at the training (or test/validation) points:* depending on the application, this can lead to either undesired conservatism or underprediction.

The obtained results motivate us to extend the study in several aspects. For example, we suggest studying the effect of improved physics of failure models (e.g., by including pit growth in the cumulative damage). We also suggest addressing multiple sources of uncertainty within the model and proposing ways to handle them using deep neural networks. For example, one can study the uncertainty in the loads model (from cycle counting to estimation of far-field stresses through finite element) and/or the scatter in material properties (including uncertainty and limitations in coupon data). Finally, one can also study how these physics-informed neural networks could be used to help decision-making in fleet management of industrial assets (inspection optimization, fleet recommissioning, and repair/replacement).

Chapter 5 extended the discussions presented in Chapter 4 and defined novel hybrid models to address the epistemic uncertainty in the corrosion-fatigue case study. Varying physics-based considerations and constraints were applied to derive recurrent neural network cells more sensitive to different aspects of the crack propagation problem. To evaluate the performance of the proposed cells, we revised the previously presented corrosion-fatigue case study to include seasonality effects in the corrosion index estimation through the consideration of a meteorological model and redefined the loading conditions to increase variability in the fleet. All these modifications were proposed aiming to hinder the task of distinguishing the true cause for the epistemic uncertainty in crack propagation.

A numerical experiment focusing on diagnosis and prognosis analyses of the derived synthetic fleet was presented. In the considered numerical analysis, the physics-informed layers only accounted for purely mechanical fatigue damage. All proposed data-driven layers, had by their own means, to adjust the damage accumulation rate and compensate for the corrosion effects. Presented results indicate that despite the varying levels of the conservativeness of the hybrid models, all proposed hybrid cells are able to capture the overall trend of fleet crack propagation.

In Chapter 6, we evaluated the use of cross-validation errors to either properly rank the predictions of the previously defined hybrid models pool for the considered corrosion-fatigue case study, or by employing weighted ensembles to combine the neural networks predictions and increase the overall prediction accuracy. Concerning the model ranking task, we evaluated the assumption that cross-validation errors can properly relate to the unknown true prediction errors and behave as valid estimators of such errors. Results showed that the cross-validation errors act as good estimators for the models prediction error allowing for the proper ranking of the models and selection of the base model that better explains the observed crack lengths.

Regarding the ensemble analysis, we evaluated the use of an existing weighted average procedure to build weighted ensembles from the hybrid neural networks predictions and improve the overall accuracy of the damage prediction. Additionally, we also evaluated the use of an indirect metric of the ensemble uncertainty to infer the confidence in the ensemble forecast and define a time window for crack length extrapolation in the fleet. Obtained results indicated that the use of cross-validation errors allows for proper ranking of the base models and to reliability derive model ensembles that in the majority of the cases increased the overall prediction accuracy. Additionally, depending on the selected threshold for the range growth metric the proposed uncertainty metric safely guides the stipulation of a time window for damage forecast in the fleet.

To improve the performance of the proposed ensemble procedure we suggest the following studies:

- Increase in the overall number of base models for the ensemble, and evaluated possible effects of the number of base models in the ensemble accuracy convergence, i.e., there is a number of base models from which the ensemble accuracy can no longer increase?

- Evaluate the use of actual statistical metrics or metrics derived from statistical process analysis the properly define the ensemble uncertainty and the confidence in its predictions.

- Investigate the ensemble performance when multiple sources of uncertainty (either epistemic or random in nature) plague the damage accumulation process.

- Extend the proposed framework to consider complex problems with combined failure modes simultaneously acting on the fleet.

- Evaluate if the proposed neural network ensemble is capable of automatically learn to identify the root cause of the unexpected observations and properly detect such unforeseen factors.

LIST OF REFERENCES

[1] MMPDS collaborators. MMPDS - 12: Metallic Materials Properties Development and Stadardization, 2017.

[2] IATA contributors. Airline Cost Management Group (ACMG). Online (retrieved 13 Oct 2020), 2015.

[3] Darli Rodrigues Vieira and Paula Lavorato Loures. Maintenance, repair and overhaul (mro) fundamentals and strategies: An aeronautical industry overview. *International Journal of Computer Applications*, 135(12), feb 2016.

[4] J. B. De Jonge, D. Schutz, H. Lowak, and J. Schijve. A standardized load sequence for flight simulation tests on transport aircraft wing structures. NLR TR 73029U, Amsterdam, 1973. Report from National Aerospace Laboratory.

[5] Marco Fioriti, Valeria Vercella, and Nicole Viola. Cost-estimating model for aircraft maintenance. *Journal of Aircraft*, 55(4):1564--1575, 2018.

[6] GE contributors. Truechoice commercial services. Online (retrieved 12 Oct 2020), 2020.

[7] Siemens contributors. Energy services. Online (retrieved 12 Oct 2020), 2020.

[8] Lufthansa Technik contributors. Aviatar: How it works. Online (retrieved 08 July 2020), 2020.

[9] Gastops contributors. Remote monitoring. Online (retrieved 08 July 2020), 2020.

[10] Uptake contributors. Services - Accelerate your time to value. Online (retrieved 12 Oct 2020), 2020.

[11] Dimitrios Serpanos and Marilyn Wolf. *Industrial Internet of Things*, pages 37--54. Springer International Publishing, 2018.

[12] Alberto Diez-Olivan, Javier Del Ser, Diego Galar, and Basilio Sierra. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. *Information Fusion*, 50:92 -- 111, 2019.

[13] Zhiheng Zhao, Peng Lin, Leidi Shen, Mengdi Zhang, and George Q. Huang. Iot edge computing-enabled collaborative tracking system for manufacturing resources in industrial park. *Advanced Engineering Informatics*, 43:101044, 2020.

[14] Laura Swanson. Linking maintenance strategies to performance. *International Journal of Production Economics*, 70(3):237 -- 244, 2001.

[15] Bernard Schmidt and Lihui Wang. Cloud-enhanced predictive maintenance. *The International Journal of Advanced Manufacturing Technology*, 99(1-4):5--13, jun 2016.

[16] Peng Wang, Robert X. Gao, Dazhong Wu, and Janis Terpenny. A computational framework for cloud-based machine prognosis. *Procedia CIRP*, 57:309 -- 314, 2016.

[17] D. Roach. Real time crack detection using mountable comparative vacuum monitoring sensors. *Smart Structures and Systems*, 5(4):317--328, 2009.

[18] Kenneth O. Hill and Gerald Meltz. Fiber bragg grating technology fundamentals and overview. *Journal of Lightwave Technology*, 15(8):1263--1276, 1997.

[19] John C. Aldrin, Jeremy S. Knopp, Eric A. Lindgren, and Kumar V. Jata. Model-assisted probability of detection evaluation for eddy current inspection of fastener sites. In *AIP Conference Proceedings*, volume 1096, pages 1784--1791. AIP, 2009.

[20] B. W. Drinkwater and P. D. Wilcox. Ultrasonic arrays for non-destructive evaluation: A review. *NDT & E International*, 39(7):525 -- 541, 2006.

[21] Michael Van Hoye. Fluorescent penetrant crack detection. Patent: US4621193A, Nov 1986.

[22] Jian Gao, Xin Chen, Oguzhan Yilmaz, and Nabil Gindy. An integrated adaptive repair solution for complex aerospace components through geometry reconstruction. *The International Journal of Advanced Manufacturing Technology*, 36(11-12):1170--1179, feb 2007.

[23] M. Brandt, S. Sun, N. Alam, P. Bendeich, and A. Bishop. Laser cladding repair of turbine blades in power plants: from research to commercialisation. *International Heat Treatment and Surface Engineering*, 3(3):105--114, sep 2009.

[24] Zhu Zheng-Qing, Zhang Yun, and Chen Zhi-Tong. A repair strategy based on tool path modification for damaged turbine blade. *The International Journal of Advanced Manufacturing Technology*, 106(7-8):2995--3006, jan 2020.

[25] Robert S. Fredell. Damage tolerant repair techniques for pressurized aircraft fuselages. Technical Report ADA286298, Air Force Wright Aeronautical Laboratories, 1994.

[26] Cong N. Duong and Chun H. Wang. *Composite Repair: Theory and Design*. Elsevier, Amsterdam, The Netherlands, 2007.

[27] D. Kwon, M. R. Hodkiewicz, J. Fan, T. Shibutani, and M. G. Pecht. IoT-based prognostics and systems health management for industrial applications. *IEEE Access*, 4:3659--3670, 2016.

[28] Kamran Javed, Rafael Gouriveau, and Noureddine Zerhouni. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mechanical Systems and Signal Processing*, 94:214 -- 236, 2017.

[29] Michael P. Brundage, Thurston Sexton, Melinda Hodkiewicz, KC Morris, Jorge Arinez, Farhad Ameri, Jun Ni, and Guoxian Xiao. Where do we start? guidance for technology

implementation in maintenance management for manufacturing. *Journal of Manufacturing Science and Engineering*, 141(9), jul 2019.

[30] T. P. Raptis, A. Passarella, and M. Conti. Data management in industry 4.0: State of the art and open challenges. *IEEE Access*, 7:97052--97093, 2019.

[31] Bjørn Grung and Rolf Manne. Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 42(1):125 -- 139, 1998.

[32] Julie Josse, Jérôme Pagès, and François Husson. Multiple imputation in principal component analysis. *Advances in Data Analysis and Classification*, 5(3):231--246, mar 2011.

[33] Stéphane Dray and Julie. Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657--667, 2015.

[34] Maria Lucia Parrella, Giuseppina Albano, Michele La Rocca, and Cira Perna. Reconstructing missing data sequences in multivariate time series: an application to environmental data. *Statistical Methods & Applications*, 28(2):359--383, 2019.

[35] Mohamed El Esawey. Using spatio-temporal data for estimating missing cycling counts: a multiple imputation approach. *Transportmetrica A: Transport Science*, 16(1):5--22, 2020.

[36] Lijun Sun and Xinyu Chen. Bayesian temporal factorization for multidimensional time series prediction. *arXiv preprint arXiv:1910.06366*, 2019.

[37] B. K. Beaulieu-Jones and J. H. Moore. Missing data imputation in the electronic health record using deeply learned autoencoders. In *Biocomputing 2017*, Kohala Coast, USA, Jan 2017. WORLD SCIENTIFIC.

[38] Christopher K. I. Williams, Charlie Nash, and Alfredo Nazábal. Autoencoders and probabilistic inference with missing data: An exact solution for the factor analysis case. *arXiv preprint arXiv:1801.03851*, 2018.

[39] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018.

[40] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, and Yuan Xiaojie. Multivariate time series imputation with generative adversarial networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 1596--1607. Curran Associates, Inc., 2018.

[41] Matthew J. Daigle and Kai Goebel. Model-based prognostics with concurrent damage progression processes. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(3):535--546, 2013.

[42] Kamal Medjaher, Diego Alejandro Tobon-Mejia, and Noureddine Zerhouni. Remaining useful life estimation of critical components with application to bearings. *IEEE Transactions on Reliability*, 61(2):292--302, 2012.

[43] Matthew J. Daigle and Chetan Shrikant Kulkarni. Electrochemistry-based battery modeling for prognostics. In *2013 Annual Conference of the PHM Society*, page 13 pages, New Orleans, USA, Sep 2013. PHM Society.

[44] Brian Bole, Chetan S. Kulkarni, and Matthew J. Daigle. Adaptation of an electrochemistry-based Li-ion battery model to account for deterioration observed under randomized use. In *2014 Annual Conference of the PHM Society*, page 9 pages, Fort Worth, USA, September 2014. PHM Society.

[45] M. Chookah, M. Nuhi, and M. Modarres. A probabilistic physics-of-failure model for prognostic health management of structures subject to pitting and corrosion-fatigue. *Reliability Engineering & System Safety*, 96(12):1601 -- 1610, 2011.

[46] Shun-Peng Zhu, Hong-Zhong Huang, Weiwen Peng, Hai-Kun Wang, and Sankaran Mahadevan. Probabilistic physics of failure-based framework for fatigue life prediction of aircraft gas turbine discs under uncertainty. *Reliability Engineering & System Safety*, 146:1 -- 12, 2016.

[47] Chetan S. Kulkarni and Matteo Corbetta. Health management and prognostics for electric aircraft powertrain. In *AIAA Propulsion and Energy 2019 Forum*, pages AIAA--2019--4474, Indianapolis, USA, Aug 2019. AIAA.

[48] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341--1390, oct 1996.

[49] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67--82, 1997.

[50] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation -- a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1--14, 2011.

[51] Samir Khan and Takehisa Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241--265, 2018.

[52] Rodney K. Singleton, Elias G. Strangas, and Selin Aviyente. Extended Kalman filtering for remaining-useful-life estimation of bearings. *IEEE Transactions on Industrial Electronics*, 62(3):1781--1790, 2014.

[53] Lingli Cui, Xin Wang, Yonggang Xu, Hong Jiang, and Jianping Zhou. A novel switching unscented Kalman filter method for remaining useful life prediction of rolling bearing. *Measurement*, 135:678--684, 2019.

[54] A. Soualhi, G. Clerc, and H. Razik. Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique. *IEEE Transactions on Industrial Electronics*, 60(9):4053--4062, Sep. 2013.

[55] A. J. Torabi, M. J. Er, X. Li, B. S. Lim, and G. O. Peen. Application of clustering methods for online tool condition monitoring and fault diagnosis in high-speed milling processes. *IEEE Systems Journal*, 10(2):721--732, 2016.

[56] D. A. Tibaduiza, L. E. Mujica, and J. Rodellar. Damage classification in structural health monitoring using principal component analysis and self-organizing maps. *Structural Control and Health Monitoring*, 20(10):1303--1316, 2013.

[57] K. Prabakaran, S. Kaushik, R. Mouleeshuwarapprabu, and A. B. Singh. Self-organizing map based fault detection and isolation scheme for pneumatic actuator. *International Journal of Innovation and Applied Studies*, 8(3):1361--1369, 09 2014.

[58] Andrea Giantomassi, Francesco Ferracuti, Alessandro Benini, Gianluca Ippoliti, Sauro Longhi, and Antonio Petrucci. Hidden markov model for health estimation and prognosis of turbofan engines. In *Volume 3: 2011 ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, Parts A and B*, Washington, DC, USA, Aug 2011. ASME.

[59] Haitao Zhou, Jin Chen, Guangming Dong, Hongchao Wang, and Haodong Yuan. Bearing fault recognition method based on neighbourhood component analysis and coupled hidden markov model. *Mechanical Systems and Signal Processing*, 66-67:568 -- 581, 2016.

[60] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812--820, 2015.

[61] Z. Tian and M. J. Zuo. Health condition prediction of gears using a recurrent neural network approach. *IEEE Transactions on Reliability*, 59(4):700--705, 2010.

[62] F. Wang, J. Du, Y. Zhao, T. Tang, and J. Shi. A deep learning based data fusion method for degradation modeling and prognostics. *IEEE Transactions on Reliability*, pages 1--15, 2020.

[63] J. Liu, V. Vitelli, E. Zio, and R. Seraoui. A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data. *IEEE Transactions on Reliability*, 64(4):1203--1213, 2015.

[64] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237--285, may 1996.

[65] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26--38, 2017.

[66] Ibrahim Ahmed, Marcos Quinones-Grueiro, and Gautam Biswas. Complementary meta-reinforcement learning for fault-adaptive control. In *2020 Annual Conference of the PHM Society*, Virtual Event, Nov 2020. PHM Society.

[67] L. Yi, X. Deng, L. T. Yang, H. Wu, M. Wang, and Y. Situ. Reinforcement learning-enabled partial confident information coverage for iot-based bridge structural health monitoring. *IEEE Internet of Things Journal*, pages 1--1, 2020.

[68] Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318--2331, 2017.

[69] Tian Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-

Bianchi, and R. Garnett, editors, *31st Advances in Neural Information Processing Systems*, pages 6572--6583. Curran Associates, Inc., 2018.

[70] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686 -- 707, 2019.

[71] Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Fusing physics-based and deep learning models for prognostics. *arXiv preprint arXiv:2003.00732*, 2020.

[72] Felipe A. C. Viana, Renato G. Nascimento, Arinan Dourado, and Yigit A. Yucesan. Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. *Computers and Structures*, 245:106458, 2021.

[73] Wayne Nelson. Accelerated life testing - step-stress models and data analyses. *IEEE Transactions on Reliability*, R-29(2):103--108, June 1980.

[74] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29(7):603--614, July 1984.

[75] John T. Renwick and Paul E. Babson. Vibration analysis---a proven technique as a predictive maintenance tool. *IEEE Transactions on Industry Applications*, IA-21(2):324--332, March 1985.

[76] J.J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, 8(6):3--11, December 1988.

[77] C. Joseph Lu and William O. Meeker. Using degradation measures to estimate a time-to-failure distribution. *Technometrics*, 35(2):161--174, May 1993.

[78] Jay Lee. Measurement of machine performance degradation using a neural network model. *International Journal of Modelling and Simulation*, 16(4):192--199, January 1996.

[79] A. Ray and S. Tangirala. Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, 4(4):443--451, July 1996.

[80] H. R. DePold and F. D. Gass. The application of expert systems and neural networks to gas turbine prognostics and diagnostics. *Journal of Engineering for Gas Turbines and Power*, 121(4):607--612, October 1999.

[81] Y. Li, S. Billington, C. Zhang, T. Kurfess, S. Danyluk, and S. Liang. Adaptive prognostics for rolling element bearing condition. *Mechanical Systems and Signal Processing*, 13(1):103--113, January 1999.

[82] G. Vachtsevanos and P. Wang. Fault prognosis using dynamic wavelet neural networks. In *2001 IEEE Autotestcon Proceedings. IEEE Systems Readiness Technology Conference. (Cat. No.01CH37237)*. IEEE, 2001.

[83] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Robust performance degradation assessment methods for enhanced rolling element bearing prognostics. *Advanced Engineering Informatics*, 17(3-4):127--140, July 2003.

[84] Andrew K. S. Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483--1510, 2006.

[85] Jian-Da Wu and Chiu-Hong Liu. Investigation of engine fault diagnosis using discrete wavelet transform and neural network. *Expert Systems with Applications*, 35(3):1200--1213, October 2008.

[86] Huiguo Zhang, Rui Kang, and Michael Pecht. A hybrid prognostics and health management approach for condition-based maintenance. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, December 2009.

[87] J.Z.Sikorska, M.Hodkiewicz, and L.Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803--1836, 2011.

[88] A. Mosallam, K. Medjaher, and N. Zerhouni. Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5):1037--1048, June 2016.

[89] Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1--11, April 2018.

[90] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani. Failure prognosis and applications -- a survey of recent literature. *IEEE Transactions on Reliability*, pages 1--21, 2019.

[91] Yigit A. Yucesan and Felipe A. C. Viana. A physics-informed neural network for wind turbine main bearing fatigue. *International Journal of Prognostics and Health Management*, 11(1):17 pages, 2020.

[92] Arinan Dourado and Felipe A. C. Viana. Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue. *ASME Journal of Computing and Information Science in Engineering*, 20(6):061007 (10 pages), 2020.

[93] Renato G. Nascimento and Felipe A. C. Viana. Cumulative damage modeling with recurrent neural networks. *AIAA Journal*, 58(12):5459--5471, 2020.

[94] Felipe A. C. Viana, Renato G. Nascimento, Yigit Yucesan, and Arinan Dourado. Physics-informed neural networks package. https://github.com/PML-UCF/pinn, Aug 2019.

[95] Arinan Dourado and Felipe A. C. Viana. Python scripts for physics-informed neural networks for corrosion-fatigue prognosis, v0.0.1. https://github.com/PML-UCF/pinn_corrosion_fatigue, Aug 2019.

[96] Yigit A. Yucesan and Felipe A. C. Viana. Python scripts for wind turbine main bearing fatigue life estimation with physics-informed neural networks, Aug 2019.

[97] H. E. G. Powrie and C. E. Fisher. Engine health monitoring: Towards total prognostics. In *IEEE Aerospace Conference*, volume 3, pages 11--20 vol.3, Aspen, USA, 1999.

[98] M.J. Roemer, E.O. Nwadiogbu, and G. Bloor. Development of diagnostic and prognostic technologies for aerospace health management applications. In *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*. IEEE, 2001.

[99] Abdo Abou Jaoude. The paradigm of complex probability and analytic linear prognostic for unburied petrochemical pipelines. *Systems Science & Control Engineering*, 5(1):178--214, 2017.

[100] Mayank Shekhar Jha, Mathieu Bressel, Belkacem Ould-Bouamama, and Genevieve Dauphin-Tanguy. Particle filter based hybrid prognostics of proton exchange membrane fuel cell in bond graph framework. *Computers & Chemical Engineering*, 95:216--230, December 2016.

[101] J.M. Ko and Y.Q. Ni. Technology developments in structural health monitoring of large-scale bridges. *Engineering Structures*, 27(12):1715--1725, October 2005.

[102] M. Mehrjoo, N. Khaji, H. Moharrami, and A. Bahreininejad. Damage detection of truss bridge joints using artificial neural networks. *Expert Systems with Applications*, 35(3):1122--1131, October 2008.

[103] Romain Pasquier, James-A. Goulet, and Ian F.C. Smith. Measurement system design for civil infrastructure using expected utility. *Advanced Engineering Informatics*, 32:40--51, 2017.

[104] Michael P. Enright, Stephen J. Hudak, R. Craig McClung, and Harry R. Millwater. Application of probabilistic fracture mechanics to prognosis of aircraft engine components. *AIAA Journal*, 44(2):311--316, feb 2006.

[105] Enrico Zio and Francesco Di Maio. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, 95(1):49--57, January 2010.

[106] David Siegel, Wenyu Zhao, Edzel Lapira, Mohamed AbuAli, and Jay Lee. A comparative study on vibration-based condition monitoring algorithms for wind turbine drive trains. *Wind Energy*, 17(5):695--714, January 2013.

[107] Amir Rasekhi Nejad, Peter Fogh dgaard, Zhen Gao, and Torgeir Moan. A prognostic method for fault detection in wind turbine drivetrains. *Engineering Failure Analysis*, 42:324--336, July 2014.

[108] Jay Lee, Shanhu Yang, Edzel Lapira, Hung An Kao, and Nelson Yen. Methodology and framework of a cloud-based prognostics and health management system for manufacturing industry. *Chemical Engineering Transactions*, 33:205–210, July 2013.

[109] Tangbin Xia, Yifan Dong, Lei Xiao, Shichang Du, Ershun Pan, and Lifeng Xi. Recent advances in prognostics and health management for advanced manufacturing paradigms. *Reliability Engineering & System Safety*, 178:255--268, October 2018.

[110] Geert Craessaerts, Josse De Baerdemaeker, and Wouter Saeys. Fault diagnostic systems for agricultural machinery. *Biosystems Engineering*, 106(1):26--36, May 2010.

[111] Rasool Khodabakhshian. Maintenance management of tractors and agricultural machinery: Preventive maintenance systems. *Agricultural Engineering International: CIGR Journal*, 15(4):147--159, 2013.

[112] Kristy F. Tiampo and Robert Shcherbakov. Seismicity-based earthquake forecasting techniques: Ten years of progress. *Tectonophysics*, 522-523:89--121, February 2012.

[113] Sudarshan Ganesh, Qinglin Su, Le Bao Dan Vo, Nolan Pepka, Benjamin Rentz, Lucas Vann, Nima Yazdanpanah, Thomas O'Connor, Zoltan K. Nagy, and Gintaras V. Reklaitis. Design of condition-based maintenance framework for process operations management in pharmaceutical continuous manufacturing. *International Journal of Pharmaceutics*, 587:119621, September 2020.

[114] B. S. Dhillon. Medical equipment reliability: a review, analysis methods and improvement strategies. *International Journal of Reliability, Quality and Safety Engineering*, 18(04):391--403, 2011.

[115] Hassana Mahfoud, Abdellah El Barkany, and Ahmed El Biyaali. Medical maintenance performance monitoring: a roadmap to efficient improvement. *International Journal of Productivity and Quality Management*, 22(1):117, 2017.

[116] K. Feldman, T. Jazouli, and P.A. Sandborn. A methodology for determining the return on investment associated with prognostics and health management. *IEEE Transactions on Reliability*, 58(2):305--316, June 2009.

[117] S. Wu and D. Clements-Croome. Optimal maintenance policies under different operational schedules. *IEEE Transactions on Reliability*, 54(2):338--346, June 2005.

[118] N. Iyer, K. Goebel, and P. Bonissone. Framework for post-prognostic decision support. In *2006 IEEE Aerospace Conference*. IEEE, 2006.

[119] K. R. McNaught and A. Zagorecki. Using dynamic bayesian networks for prognostic modelling to inform maintenance decision making. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, Dec 2009.

[120] Karen B. Marais and Matthew R. Robichaud. Analysis of trends in aviation maintenance risk: An empirical approach. *Reliability Engineering and System Safety*, 106:104--118, 2012.

[121] Maria Grazia DeGiorgi, Stefano Campilongo, and Antonio Ficarella. A diagnostics tool for aero-engines health monitoring using machine learning technique. *Energy Procedia*, 148:860--867, 2018.

[122] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1--9, 2008.

[123] J. Xu, Y. Wang, and L. Xu. Phm-oriented integrated fusion prognostics for aircraft engines based on sensor data. *IEEE Sensors Journal*, 14(4):1124--1132, 2014.

[124] Z. Wang, J. Zarader, and S. Argentieri. A novel aircraft engine fault diagnostic and prognostic system based on svm. In *2012 IEEE International Conference on Condition Monitoring and Diagnosis*, pages 723--728, 2012.

[125] Zeqi Zhao, Bin Liang, Xueqian Wang, and Weining Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety*, 164:74--83, 2017.

[126] Celestino Ordóñez, Fernando Sánchez Lasheras, Javier Roca-Pardiñas, and Francisco Javier de CosJuez. A hybrid arima--svm model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 346:184--191, 2019.

[127] Dong Dong, Xiao-Yang Li, and Fu-Qiang Sun. Life prediction of jet engines based on lstm-recurrent neural networks. In *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, pages 1--6. PHM Society, 2017.

[128] Feng Lu, Jindong Wu, Jinquan Huang, and Xiaojie Qiu. Aircraft engine degradation prognostics based on logistic regression and novel os-elm algorithm. *Aerospace Science and Technology*, 84:661--671, 2019.

[129] S.J Findlay and N.D Harrison. Why aircraft fail. *Materials Today*, 5(11):18 -- 25, 2002.

[130] S.K. Bhaumik, M. Sujata, and M.A. Venkataswamy. Fatigue failure of aircraft components. *Engineering Failure Analysis*, 15(6):675 -- 694, 2008.

[131] T. K. Goswami and D. W. Hoeppner. Pitting corrosion fatigue of structural materials. In *Structural Integrity in Aging Aircraft*, page 47. ASME, New York, 1995.

[132] G. S. Chen, K. C. Wan, M. Gao, R. P. Wei, and T. H. Flournoy. Transition from pitting to fatigue crack growth -- modeling of corrosion fatigue crack nucleation in a 2024-T3 aluminum alloy. *Material Science Engineering*, pages 126--132, 1996.

[133] R. M. V. Pidaparti and M. J. Palakal. Neural network approach to fatigue-crack-growth predictions under aircraft spectrum loadings. *Journal of Aircraft*, 32(4), 1995.

[134] D. L. DuQuesnay, P. R. Underhill, and H. J. Britt. Fatigue crack growth from corrosion damage in 7075-T6511 aluminium alloy under aircraft loading. *International Journal of Fatigue*, 25(5):371--377, 2003.

[135] Iyyer Nagaraja, Sarkar Subhasis, Merrill Robert, and Phan Nam. Aircraft life management using crack initiation and crack growth models--p-3c aircraft experience. *International Journal of Fatigue*, 29(9-11):1584--1607, 2007.

[136] Chenzhao Li, Sankaran Mahadevan, You Ling, Sergio Choze, and Liping Wang. Dynamic Bayesian network for aircraft wing health monitoring digital twin. *AIAA Journal*, 55(3):930--941, 2017.

[137] Aleksandar Grbovic and Bosko Rasuo. Fem based fatigue crack growth predictions for spar of light aircraft under variable amplitude loading. *Engineering Failure Analysis*, 26:50--64, 2012.

[138] Isaac Asher, Liping Wang, Genghis Khan, You Ling, and Felipe A. C. Viana. Developing a probabilistic load spectrum for fatigue modeling. In *AIAA Scitech 2017 Forum*, pages AIAA--2017--1562, Grapevine, USA, Jan 2017. AIAA.

[139] Q.Y. Wang, R. M. Pidaparti, and M. J. Palakal. Comparative study of corrosion-fatigue in aircraft materials. *AIAA Journal*, 39(2):325--330, 2001.

[140] C Boller. Ways and options for aircraft structural health management. *Smart Materials and Structures*, 10(3):432--440, jun 2001.

[141] Sung W.Choi, Eun-Jung Song, and H.Thomas Hahn. Prediction of fatigue damage growth in notched composite laminates using an artificial neural network. *Composite Science and Technology*, 63(5):661--675, 2003.

[142] R.Jones, A.J.Kinloch, and W.Hua. Cyclic-fatigue crack growth in composite and adhesively-bonded structures: The faa slow crack growth approach to certification and the problem of similitude. *International Journal of Fatigue*, 88:10--18, 2016.

[143] Theodoros Loutas, Nick Eleftheroglou, and Dimitrios Zarouchas. A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data. *Composite Structures*, 161(1):522--529, 2017.

[144] Pier Carlo C. Berri, Matteo Davide Lorenzo Dalla Vedova, and Laura Mainini. Real-time fault detection and prognostics for aircraft actuation systems. In *AIAA Scitech 2019 Forum*, pages AIAA--2019--2210, San Diego, USA, Jan 2019. AIAA.

[145] Carl S Byington, Matthew Watson, and Doug Edwards. Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720)*, volume 6, pages 3581--3589. IEEE, 2004.

[146] Andrea De Martin Giovanni Jacazio and Massimo Sorli. Enhanced particle filter framework for improved prognosis of electro-mechanical flight controls actuators. In *Fourth European Conference of the Prognosis and Health Management Society*, page 10 pages, Utrecht, The Netherlands, 2018. PHM Society.

[147] EJ Cross, P Sartor, K Worden, and P Southern. Prediction of landing gear loads using machine learning techniques. *Structural Health Monitoring*, pages 1056--1063, 2012.

[148] Geoffrey Holmes, Pia Sartor, Stephen Reed, Paul Southern, Keith Worden, and Elizabeth Cross. Prediction of landing gear loads using machine learning techniques. *Structural Health Monitoring*, 15(5):568--582, july 2016.

[149] Joshua Hoole, Pia Sartor, Julian D Booker, Jonathan E Cooper, Xenofon Gogouvitis, and R Kyle Schmidt. Comparison of surrogate modeling methods for finite element analysis of landing gear loads. In *AIAA Scitech 2020 Forum*, page 0681, Orlando, USA, 2020.

[150] Chao Tong, Xiang Yin, Jun Li, Tongyu Zhu, Renli Lv, Liang Sun, and Joel JPC Rodrigues. An innovative deep architecture for aircraft hard landing prediction based on time-series sensor data. *Applied Soft Computing*, 73:344--349, 2018.

[151] P. Phillips, D. Diston, A. Starr, J. Payne, and S. Pandya. A review on the optimisation of aircraft maintenance with application to landing gears. In *4th World Congress on Engineering Asset Management*, pages 68--76, Athens, Greece, September 2009. Springer London.

[152] Paul Phillips, Dominic Diston, and Andrew Starr. Perspectives on the commercial development of landing gear health monitoring systems. *Transportation Research Part C: Emerging Technologies*, 19(6):1339 -- 1352, 2011.

[153] Chad Forrest, Clint Forrest, and Doug Wiser. Landing gear structural health monitoring (shm). *Procedia Structural Integrity*, 5:1153 -- 1159, 2017. 2nd International Conference on Structural Integrity, ICSI 2017, 4-7 September 2017, Funchal, Madeira, Portugal.

[154] S. Haider. Overview of prognostics and health management for landing gear maintenance. In *2019 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1--7, Orlando, USA, Jan 2019.

[155] C. Wilkinson, D. Humphrey, B. Vermeire, and J. Houston. Prognostic and health management for avionics. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, volume 5, pages 3435--3447, 2004.

[156] L. V. Kirkland, T. Pombo, K. Nelson, and F. Berghout. Avionics health management: searching for the prognostics grail. In *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, volume 5, pages 3448--3454, 2004.

[157] T. D. Batzel and D. C. Swanson. Prognostic health management of aircraft power generators. *IEEE Transactions on Aerospace and Electronic Systems*, 45(2):473--482, 2009.

[158] J. A. Rosero, J. A. Ortega, E. Aldabas, and L. Romeral. Moving towards a more electric aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 22(3):3--9, 2007.

[159] A. Boglietti, A. Cavagnino, A. Tenconi, S. Vaschetto, and P. di Torino. The safety critical electric machines and drives in the more electric aircraft: A survey. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 2587--2594, 2009.

[160] X. Liu, L. Liu, D. Liu, L. Wang, Q. Guo, and X. Peng. A hybrid method of remaining useful life prediction for aircraft auxiliary power unit. *IEEE Sensors Journal*, march 2020.

[161] National Transportation Safety Board. Ntsb report aab-13-02. Online (retrieved 4 March 2021), 2013.

[162] Victor Chan and William Meeker. A failure-time model for infant-mortality and wearout failure modes. *IEEE Transactions on Reliability*, 48(4):377--387, 1999.

[163] J Schijve. Statistical distribution functions and fatigue of structures. *International Journal of Fatigue*, 27(9):1031--1039, 2005.

[164] Vijay Rathod, Prakash Yadav, Ajay Rathore, and Rakesh Jain. Probabilistic modeling of fatigue damage accumulation for reliability prediction. *Journal of Quality and Reliability Engineering*, 2011, 2011.

[165] Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101--117, 2001.

[166] Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91(1):110--131, 1990.

[167] Jun Takeuchi and Yukio Kosugi. Neural network representation of finite element method. *Neural Networks*, 7(2):389--395, 1994.

[168] Yi-Jen Wang and Chin-Teng Lin. Runge-kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions on Neural Networks*, 9(2):294--307, 1998.

[169] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, 55(7):2215--2227, 2017.

[170] Maziar Raissi and George Em Karniadakis. Hidden physics models: machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125 -- 141, 2018.

[171] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172--A198, 2018.

[172] J. Nagoor Kani and Ahmed H. Elsheikh. Reduced-order modeling of subsurface multi-phase flow models using deep residual recurrent neural networks. *Transport in Porous Media*, 126(3):713--741, 2018.

[173] Maziar Raissi. Deep hidden physics models: deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1--24, 2018.

[174] Yang Yu, Houpu Yao, and Yongming Liu. Physics-based learning for aircraft dynamics simulation. In *2018 Annual Conference of the PHM Society*, Philadelphia, USA, Nov 2018. PHM Society.

[175] Ali Fatemi and Linsheng Yang. Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials. *International Journal of Fatigue*, 20(1):9 -- 34, 1998.

[176] Dan M. Frangopol, Maarten-Jan Kallen, and Jan M. van Noortwijk. Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials*, 6(4):197--212, 2004.

[177] J. L. Bogdanoff and F. Kozin. *Probabilistic models of cumulative damage*. John Wiley & Sons, 1985.

[178] Toshio Nakagawa and Masaaki Kijima. Replacement policies for a cumulative damage model with minimal repair at failure. *IEEE Transactions on Reliability*, 38(5):581--584, 1989.

[179] Pe Paris and Fazil Erdogan. A critical analysis of crack propagation laws. *Journal of Basic Engineering*, 85(4):528--533, 1963.

[180] Norman E. Dowling. *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue*. Pearson, 2012.

[181] P. Shi and S. Mahadevan. Damage tolerance approach for probabilistic pitting corrosion fatigue life prediction. *Engineering Fracture Mechanics*, 68:1493--1507, 2001.

[182] F. Menan and G. Henaft. Synergistic action of fatigue and corrosion during crack growth in the 2024 aluminum alloy. *Procedia Engineering*, 2:1441--1450, 2010.

[183] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[184] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735--1780, 1997.

[185] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2067--2075, Lille, France, 07--09 Jul 2015. PMLR.

[186] Jerome T. Connor, R. D. Martin, and L. E. Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2):240--254, 1994.

[187] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer, editors, *28th International Conference on Machine Learning*, pages 1017--1024, Bellevue, USA, June 2011. ACM.

[188] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645--6649, 2013.

[189] Hasim Sak, Andrew Senior, and Francoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, pages 338--342, Singapore, September 2014.

[190] Sucheta Chauhan and Lovekesh Vig. Anomaly detection in ECG time signals via deep long short-term memory networks. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1--7, Oct 2015.

[191] Paul Goodwin and Richard Lawton. On the asymmetry of the symmetric MAPE. *International Journal of Forecasting*, 15(4):405 -- 408, 1999.

[192] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI'95, pages 1137--1143. Morgan Kaufmann Publishers Inc., 1995.

[193] Robert Summitt and Frederick C. Fink. PACER LIME: An environmental corrosion severity classification system. Technical Report AFWAL-TR-80-4102, Air Force Wright Aeronautical Laboratories, 1980.

[194] P.R. Roberge, R.D. Klassen, and P.W. Haberecht. Atmospheric corrosivity modeling -- a review. *Materials & Design*, 23(3):321 -- 330, 2002.

[195] Felipe A. C. Viana, Raphael T. Haftka, and Valder Steffen. Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization*, 39(4):439--457, Jan 2009.