Generic approach towards a diagnostic and prognostic data suitability assessment

Bieber, M.T.

DOI
10.4233/uuid:8adff5c2-e817-459d-b895-ade85c802536

Publication date
2023

Document Version
Final published version

Citation (APA)
Bieber, M. T. (2023). *Generic approach towards a diagnostic and prognostic data suitability assessment*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:8adff5c2-e817-459d-b895-ade85c802536

GENERIC APPROACH TOWARDS A DIAGNOSTIC AND PROGNOSTIC DATA SUITABILITY ASSESSMENT

GENERIC APPROACH TOWARDS A DIAGNOSTIC AND PROGNOSTIC DATA SUITABILITY ASSESSMENT

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 21 december 2023 om 10:00 uur

door

Marie BIEBER

Master of Science in Mathematik,
Technische Universität Wien, Wenen, Oostenrijk,
geboren te Wenen, Oostenrijk.

Dit proefschrift is goedgekeurd door de

promotor: Dr. B.F. Santos
copromotor: Dr. W.J.C. Verhagen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Dr. B.F. Santos,	Technische Universiteit Delft
Dr. W.J.C. Verhagen,	Technische Universiteit Delft

Onafhankelijke leden:

Prof. dr. ir. B. De Schutter	Technische Universiteit Delft
Dr. O. Fink	EPFL, Switzerland
Prof. dr. ir. J. Hoekstra	Technische Universiteit Delft
Prof. dr. ir. T. Tinga	University of Twente
Prof. dr. ir. R. Vingerhoeds	ISAE - SUPAERO, France

Keywords: condition-based maintenance, diagnostics, prognostics, metrics, data suitability

Printed by: todo

Front & Back: todo Beautiful cover art that captures the entire content of this thesis in a single illustration.

todo ISBN 000-00-0000-000-0

todo An electronic version of this dissertation is available at http://repository.tudelft.nl/.

Nothing in life is to be feared. It is only to be understood.
Now is the time to understand more, so that we may fear less.

Marie Curie

CONTENTS

ACKNOWLEDGEMENTS

Even though my name is on the cover of the book you hold in your hands, it certainly is not a work of just my own. In fact, credit has to be given to many - supporting me on this journey of doing a PhD, laughing with me, giving me honest feedback (not just regarding my work) and helping me navigate the world of academia.

Thank you Wim, for knowing which are the right words in every situation, for in a week which made me feel frustrated and depressed with just two sentences bringing me back to feel alive and completely ready for the challenges ahead - elevated and ready to run with it. Thank you Bruno, for bringing me back on track every now and then, for coming up with creative and interesting suggestions that always lead to interesting discussions, for challenging me in the right way helping me to grow.

Thank you, to the entire ATO Team and all its varying members in the course of the four years I spent in Delft, for being an always welcoming distraction from sitting alone in front of the computer, especially after doing so for quite a long time at home with no distraction at all, for being open for discussions even if, or especially if, they went totally off topic into crazy directions, for making sure I ate cake in regular time intervals, for not being super mad when I crashed into the office for the fifth time in search for good coffee and last but not least for spending hours or even days with me in one particularly good Greek restaurant.

Thank you Vivian, for making sure that I had friends even though it was a difficult time to find friends outside of the research group, for dragging me out of the house often enough to not become a lonesome rider and for being open for spontaneous and crazy ideas whenever they came along. Thank you Farzam, for introducing me to Dough - a very tasty if less helpful in the purchase of a PhD- drink, for telling me numerous times I am crazy to do a PhD and for helping me to put things into perspective at times.

A big thank you to my family, whom I feel grateful for every day of my life and who possesses the incredible skill of making me feel like my research is the most interesting and relevant thing in the world, even though I bore them to death with it. To my parents, who always have my back and tell me how great I am doing even if no one else seems to agree with them and without whom I can safely say I would not be where I am today (and this is meant in more than one way). To my brothers, Sebastian and Konstantin, who light up my day, joke with me, make me work out harder, make me party harder, hype me up and most importantly make me laugh - mainly about myself. To my grandparents, who are all four the best people to look up to, learn from and who are my heroes in many ways. To Hanna, who is simply there for me when I need her, always up for philosophical (and less philosophical) discussions in Delirium, hard climbing sessions, crazy hikes or Harry

Potter comfort nights even if it costs her her last nerves and a good nights sleep. To Rosi, who always was and is a joy to talk to and spend time with and who is not only shy to say what she thinks about me - great life lessons indeed - but also brings me down to earth more often than I would wish for.

And last but not least, thank you Matthäus, one of the only ones and perhaps the only one who experiences my downs, at whom I can cry my heart out, who really has given too much of himself in his never ending attempt to sooth me, comfort me, listen to me and hold me when I need it and whom I love with all my heart.

SUMMARY

Machine learning can arguably solve many problems we face today and make many aspects of our lives easier. Take, for example, the use of system data in a machine learning model that is able to detect and predict system faults and failures. Such early warnings and failure detection can be helpful in two ways: First, to understand systems failures better and thereby improve systems accordingly, and second, to plan related maintenance and repair actions in advance. Especially for safety-critical systems, such as aircraft and satellite systems, such approaches are important. Failures of such systems can lead to not only operational interruptions and, thereby, major costs and delays, but also compromise the safety of operations. Of course, airlines and satellite operators take steps towards ensuring smooth and safe operations of their systems. Almost every system is monitored continuously. Operational and system health-related data is collected in regular time intervals. And still, for both satellites and aircraft most of the maintenance and system life-time calculation is done using simple statistical models not exploring the full range of available data. The monitored data is mostly used to track causes of failures or faults that already occurred. But how can we move away from this practice and put the vast amount of data to a better use? This is where machine learning approaches come into play. When used to build the above-mentioned machine learning models that are able to detect and predict system failures, such models can help to calculate system reliability dynamically over a system's life and plan maintenance accordingly. This practice is called condition-based maintenance. The major question now is: Why are both airlines and satellite operators so reluctant to use such a practise if it is so promising?

There are many reasons that complicate the application of machine learning approaches for failure detection and prediction - one of the main reasons being linked to underlying system data. Even though most systems are monitored continuously, the available data is not guaranteed to capture failures or even degradation. This can be due to the data itself (e.g. its quality, missing data or missing sensors), but it can also be due to the nature of system degradation and failures. Some failures might, for example, happen spontaneously or without measurable degradation symptoms, which makes it impossible to predict them. Furthermore, many existing machine learning techniques are implemented, validated, tested and sometimes even tailored to existing publicly available datasets in literature, mostly simulated ones. Therefore, a way to identify if system data is suitable for failure or anomaly detection or prediction in the first place would be needed. This is what we provide in this thesis in the form of a generic diagnostic (failure detection) and prognostic (failure prediction) framework. There are two main aspects of the framework: First, it provides a guideline for the development of detection or prediction models for systems. Second, it gives an indication whether system data is suitable for data-driven diagnostic and prognostic approaches.

This dissertation starts by systematically defining the requirements for a generic diagnostic and prognostic framework. Defining such requirements helps to approach the development and implementation of the framework in a structured way. Therefore, based on those requirements in Chapter 3 a Generic Diagnostics Framework is presented. It is applied in two case studies: First, the performance of the framework is benchmarked against existing approaches by applying it to an open source dataset. Second, it is used to detect satellite system anomalies. The framework not only proves to be helpful in the choice of diagnostic methodologies, but also shows the challenges with using real world data for data-driven diagnostics. In Chapter 4, the Generic Prognostic Framework is introduced. This results in a Generic Diagnostic and Prognostic Framework, which can be used for a diagnostic or prognostic data suitability assessment, depending on the use case. Again, it is applied to two systems: a simulated turbofan engine dataset and an aircraft cooling unit dataset, both collected from a commercial aircraft. The results show that the obtained accuracy is comparable to what has been achieved in literature and provide insights into the adaptivity and generalizability of the framework, especially with respect to real aircraft data. Finally, we aim to understand how the Generic Prognostic Framework can be translated towards assessing the suitability of data for prognostics. The focus lies on prognostics for this purpose, but the shown methodologies can be adapted towards diagnostics as well. Several representative metrics are used within the framework to guide the decision of whether system data is suitable for prognostics or not. The thereby adapted framework is applied in three case studies to complex systems with different underlying data quality, i.e. while some of the systems have data of high quality that can be used to build prognostics models, others do not. The results show two interesting findings: First, the choice of optimization metric has an impact on the output of the generic prognostic framework and on the overall prognostic performance. Second, such a first prognostic assessment can give a rough indication of whether or not it makes sense to use system data to train prognostic models.

Now one might wonder: How is the conducted research overall useful? A generic framework as presented in this dissertation can not only provide guidelines for further development, but also give an indication whether system data are suitable for diagnostics or prognostics. Of course, one has to keep in mind that such a framework can never be truly generic - it is impossible to include all possible steps for diagnostics or prognostics, let alone all available machine learning methodologies. The scale of the problem is simply huge and still widely researched and therefore growing. In addition, the data suitability assessment is based on metrics measuring the model quality. When used for a specific application, such as aircraft maintenance, it might be more suiting to use metrics measuring other aspects, like the cost of not detected failures, as well. All in all, the framework presented in this dissertation is still a big step towards the application of diagnostics and prognostics in the aerospace domain due to two major advantages: First, it can provide guidelines where further development should go and second, it can indicate which systems to include in a PHM/ CBM solution and which system data are suitable to train machine learning based models.

SAMENVATTING

Machine learning kan aantoonbaar veel problemen oplossen waar we mee te maken hebben en veel aspecten van ons leven eenvoudiger maken. Neem bijvoorbeeld het gebruik van systeemgegevens in een machine learning model dat systeemfouten en -storingen kan detecteren en voorspellen. Dergelijke vroegtijdige waarschuwingen en foutdetectie kunnen op twee manieren nuttig zijn: Ten eerste om systeemfouten beter te begrijpen en daardoor systemen dienovereenkomstig te verbeteren en ten tweede om gerelateerde onderhouds- en reparatieacties van tevoren te plannen. Vooral voor veiligheidskritische systemen, zoals vliegtuig- en satellietsystemen, zijn dergelijke benaderingen belangrijk. Storingen in deze systemen kunnen niet alleen leiden tot operationele onderbrekingen en daarmee tot grote kosten en vertragingen, maar ook tot gevaarlijke situaties. Natuurlijk nemen luchtvaartmaatschappijen en satellietoperatoren maatregelen om een probleemloze werking van hun systemen te garanderen. Bijna elk systeem wordt continu gemonitord en operationele data en gegevens over de gezondheid van het systeem worden regelmatig verzameld. Toch wordt voor zowel satellieten als vliegtuigen het grootste deel van het onderhoud en de berekening van de levensduur van het systeem gedaan op basis van eenvoudige statistische modellen die niet het volledige scala van beschikbare gegevens onderzoeken. De gemonitorde data worden meestal gebruikt om inzicht te krijgen in oorzaken van storingen of fouten die al zijn opgetreden. Maar hoe kunnen we van deze praktijk afstappen en de enorme hoeveelheid van data beter gebruiken? Dit is waar machine learning methoden om de hoek komen kijken. Wanneer deze technieken worden gebruikt om modellen te bouwen die systeemstoringen kunnen detecteren en voorspellen, kunnen ze helpen bij het plannen van onderhoud en het dynamisch berekenen van de betrouwbaarheid van het systeem. Deze praktijk wordt "condition-based maintenance"genoemd. De grote vraag is nu: Waarom zijn zowel luchtvaartmaatschappijen als satellietexploitanten zo terughoudend in het gebruik van een dergelijke praktijk als deze zo veelbelovend is?

Er zijn veel redenen die de toepassing van machine learning benaderingen voor foutdetectie en -voorspelling bemoeilijken - een van de belangrijkste redenen heeft te maken met de onderliggende systeem data. Hoewel de meeste systemen continu worden gemonitord is er geen garantie dat de beschikbare data storingen of zelfs degradatie registreren. Dit kan te maken hebben met de data zelf (de kwaliteit, ontbrekende data, ontbrekende sensoren, etc.). Sommige storingen kunnen bijvoorbeeld spontaan of zonder meetbare degradatie symptomen optreden, waardoor het onmogelijk is om ze te voorspellen. Bovendien zijn veel bestaande technieken voor machine learning geïmplementeerd, gevalideerd, getest en soms zelfs toegesneden op bestaande, publiekelijk beschikbare datasets in de literatuur, meestal gesimuleerde datasets. Daarom is er behoefte aan een manier om te bepalen of systeem data überhaupt geschikt zijn voor het detecteren of voorspellen van storingen of anomalieën. Dit is wat we in dit proefschrift

presenteren in de vorm van een generiek diagnostic (foutdetectie) en prognostic (foutvoorspelling) framework. Er zijn twee belangrijke aspecten van het framework: Ten eerste biedt het een richtlijn voor de ontwikkeling van detectie- of voorspellingsmodellen voor systemen. Ten tweede geeft het een indicatie of systeem data geschikt zijn voor datagestuurde diagnostic en prognostic benaderingen.

We beginnen met het systematisch definiëren van vereisten voor een Generic Diagnostic and Prognostic Framework. Het definiëren van requirements helpt om de ontwikkeling en implementatie van het framework op een gestructureerde manier aan te pakken. Daarom wordt op basis van de requirements in hoofdstuk 3 een generiek diagnostisch raamwerk gepresenteerd. Het wordt toegepast in twee case studies: Ten eerste wordt de prestatie van het raamwerk vergeleken met bestaande benaderingen door het toe te passen op een open source dataset. Ten tweede wordt het gebruikt om afwijkingen in satellietsystemen te detecteren. Het raamwerk blijkt niet alleen nuttig te zijn bij de keuze van diagnostische methoden, maar laat ook de uitdagingen zien bij het gebruik van industriele data voor datagestuurde diagnostiek. In hoofdstuk 4 wordt het Generic Prognostic Framework geïntroduceerd. Dit resulteert in een Generic Diagnostic and Prognostic Framework, dat kan worden gebruikt voor een diagnostische of prognostische beoordeling van de geschiktheid van data, afhankelijk van de use case. Het wordt opnieuw toegepast op twee systemen: een gesimuleerde dataset van een turbofanmotor en een dataset van een koeleenheid van een vliegtuig. De resultaten laten zien dat de verkregen nauwkeurigheid vergelijkbaar is met wat in de literatuur is bereikt en geven inzicht in de aanpasbaarheid en generaliseerbaarheid van het raamwerk, vooral met betrekking tot echte vliegtuig data. Tot slot willen we begrijpen hoe het Generic Prognostic Framework kan worden gebruikt om in te schatten of en in welke mate data geschikt is voor prognostische doeleinden. De focus ligt hierbij op prognose, maar de getoonde methodologieën kunnen ook worden aangepast voor diagnostiek. Binnen het framework worden verschillende representatieve meetindicatoren gebruikt om te bepalen of systeem data geschikt zijn voor prognostics of niet. Het framework wordt in drie case studies toegepast op complexe systemen met verschillende onderliggende data kwaliteit, d.w.z. terwijl sommige systemen data van hoge kwaliteit hebben die gebruikt kunnen worden om prognostische modellen te bouwen, hebben andere systemen dat niet. De resultaten laten twee interessante bevindingen zien: Ten eerste heeft de keuze van de meetindicator voor optimalisatie invloed op de output van het Generic Prognostic Framework en op de algehele prognostic prestaties. Ten tweede kan zo'n eerste prognostische beoordeling een ruwe indicatie geven of het al dan niet zinvol is om systeem data te gebruiken om prognostic modellen te trainen.

Nu kun je de vraag stellen: Hoe is het uitgevoerde onderzoek in het algemeen relevant? Een Generic Framework als gepresenteerd in dit proefschrift kan niet alleen richtlijnen geven over waar verdere ontwikkeling naartoe zou moeten gaan, maar het kan ook helpen om te begrijpen of we überhaupt in staat zijn om diagnostische of prognostische modellen te leveren. Natuurlijk moet men in gedachten houden dat een dergelijk framework nooit echt generiek kan zijn - het is onmogelijk om alle stappen voor diagnostiek of prognostiek op te nemen, laat staan alle beschikbare methoden voor machinaal leren.

De omvang van het probleem is enorm, wordt nog steeds op grote schaal onderzocht en groeit daarom nog steeds. Bovendien is de beoordeling van de geschiktheid van data gebaseerd op metrieken die de kwaliteit van het model meten. Bij gebruik voor een specifieke toepassing, zoals vliegtuigonderhoud, is het misschien beter om ook metrieken te gebruiken die andere aspecten meten, zoals de kosten van niet gedetecteerde fouten. Al met al is het raamwerk dat in dit proefschrift is gepresenteerd nog steeds een grote stap in de richting van de toepassing van diagnostiek en prognostiek in de lucht- en ruimtevaart vanwege twee grote voordelen: Ten eerste kan het richtlijnen geven voor verdere ontwikkeling en ten tweede kan het aangeven welke systemen opgenomen moeten worden in een PHM/ CBM-oplossing en welke systeemgegevens geschikt zijn om op machine learning gebaseerde modellen te trainen.

1

INTRODUCTION

1.1. RESEARCH BACKGROUND

An increasing amount of systems installed on aerospace vehicles like aircraft and satellites are monitored continuously: Health related data is collected through sensors and can be used to detect system malfunctions, anomalies or even failures. In this way, unexpected failures can be avoided, anticipated on or, at minimum, be reacted to proactively at occurrence. In case of aircraft systems such information can be used to schedule maintenance and prevent the failure from happening. For satellite systems the benefit of system health monitoring lies in a better understanding of failure rates, a more accurate reliability and availability assessment and, in some cases, even a maintenance action conducted from the operational center on ground.

So, using system monitoring data for maintenance or a system health assessment has substantial benefits. The question now is: How can system data be translated into a useful health assessment or even failure prediction? There certainly is not merely one correct answer to this question. It also entails many more questions, such as the question of what "useful" means in this context. In the past years, a vast amount of literature has been published on data-driven solutions for system health assessment and remaining useful life predictions (Zio, 2022). Data-driven approaches make use of such condition monitoring data to assess system health or estimate remaining useful life (Kan et al., 2015) either by statistical methods or by artificial intelligence methodologies (An et al., 2015; Peng, Dong, et al., 2010). However, as we later on establish in more detail, most of the published studies are conducted using simulated data sets. There is not much literature available dealing with and digging into the challenges of using real-life aerospace system-related data sets. A shortcoming, which can partly be attributed to the lack of publicly available real-life data sets. Applying existing data-driven models or frameworks, which have been tested on simulated data sets, to real-life aerospace systems comes with challenges. The resulting models are, in many cases, worse performing due to a loss of accuracy. In some cases, this issue can be solved by applying more advanced model tuning techniques. However, more often it has nothing to do with the

models themselves, but instead can be linked back to underlying system data, to data quality or to system failure behaviour. Therefore, we argue that the better question to ask is: How can we tell whether system data is suitable for a useful health assessment or failure predictions? Before we look in further detail into how we will address this, we give an overview over existing data-driven solutions used for real-life aerospace systems and how they are translated to further actions, such as maintenance tasks.

Today, most airlines and maintenance providers, such as Maintenance, repair and overhaul providers (MROs), still apply traditional maintenance approaches and maintain their systems in a preventive or corrective way (Gerdes et al., 2016). In preventive maintenance, systems are checked at regular time intervals. Those intervals are pre-determined with simple statistical models developed using historical data. Preventive maintenance has two disadvantages, however: First, it can affect aircraft safety when underlying models fall short of detecting degradation or failures due to the fact that they do not receive information about the actual system health. Second, it can lead to still healthy or only slightly degraded systems being replaced (M. Scott et al., 2022). Corrective maintenance refers to systems operated in a run-to-failure way. A redundancy is created, and systems are only repaired after they fail. However, corrective maintenance can lead to unexpected aircraft-on-ground events and thereby cause delays.

In order to provide a solution to those downsides of classical maintenance, the concept of Condition-Based Maintenance (CBM) was introduced (Broer et al., 2022; Montero Jimenez et al., 2020; Peng, Liu, et al., 2010). The main principle of CBM is to collect system-related health data through condition monitoring and recommend according maintenance actions (Jardine et al., 2006). It has the capability to improve reliability, safety, and availability while reducing the life-cycle operational costs of components (J. Zhang et al., 2018). To standardize the implementation of CBM systems, the Open System Architecture (OSA)-CBM standard framework was developed in 2001 by an industry-led team consisting of members such as Boeing, Rockwell Automation, or Oceana Sensor Technologies (Swearingen et al., 2007). According to the framework, a CBM strategy consists of the following steps:

1. Data Acquisition
2. Data Manipulation
3. State Detection
4. Health Assessment
5. Prognostics
6. Advisory Generation

In the first three steps, condition monitoring data is collected, processed, and used to build models that identify abnormal behavior. Steps 4 to 6 combine the monitored data with methodologies to assess components' health, predict the future health state and make maintenance decisions based on the results. Both diagnostics and prognostics play an important part in CBM. Diagnostics detect, isolate, and identify faults, while prognostics attempt to predict failures before they occur (Jardine et al., 2006).

As can be seen in the OSA-CBM framework, CBM starts with the step of data acquisition. This step is crucial as the key requirement for data-driven algorithm development is the availability of data of sufficient quality characterizing system behavior in all phases of normal and faulty operation (Elattar et al., 2016). Data availability is a requirement that, for many applications, cannot or is not sufficiently fulfilled. Frequently it is discovered that the collected data is inaccurate, incomplete, or redundant (Y. Chen et al., 2013). While much literature exists on developing diagnostic and prognostic solutions, a much smaller number of authors focuses on the data suitability aspect (Coble, 2010). The literature that exists on data suitability aspects in a CBM framework typically focuses on analyzing the data and trends within the data themselves. The problem with such an approach is that most of the existing data-driven diagnostic and prognostic models are based on machine learning techniques, and it is usually not clear beforehand (and for some even not after applying the algorithm) which features should be used to train the models (Y. Liu & Goebel, 2018). In addition, time is needed to set up data pipelines, enable continuous data collection and monitoring, develop and test data preprocessing tools, define straightforward or combined features and develop and adapt machine learning models. All of this, most of the times, happens without knowing if it is even worth the effort.

The second development in the field of diagnostics and prognostics is as follows: The main focus in the past years has been on developing more advanced and more accurate models and algorithms. Standard data sets are often used as these enable comparative evaluation of multiple models, which make the approaches application- and system-specific (Lewis & Groth, 2022a). Therefore, most existing diagnostic or prognostic models are system specific and their translation towards application in aerospace systems is lacking. As a consequence, especially for aerospace systems, the system-related data is still often not explored to its full extent and, in many cases, only after a fault, anomaly or failure occurred (Yang et al., 2021). If data is used during operations, then often simple methodologies are applied, such as the out-of-limits (OOL) method, which compares monitored satellite telemetry data with a manually pre-defined threshold (Xu et al., 2023).

1.2. RESEARCH MOTIVATION

Consider, for example, an airline operating different types of aircraft and aiming to introduce diagnostics and prognostics on a broad basis. Now, in recent years, the following developments could be seen in diagnostics and prognostics: First, typically, data-driven models are developed on and tailored to specific data sets- often simulated open source data sets, which are publicly available (Coble, 2010). To return back to the example: If the airline CEO has heard of those developments and now has an interest to make use of diagnostic and prognostic solutions for airline maintenance, she or he might approach the data scientists with this suggestion. The data scientists of the airline, however, are aware of the following: Each aircraft can be considered a complex system with multiple subsystems and components. A dedicated diagnostic and prognostic model is needed for each of these subsystems or components. Each of those models needs to be devel-

oped, tested, and validated. The costs and time required for such an undertaking are high, especially when considering the scale of the problem. Multiple models are needed for complex systems with a multi-level hierarchy and system dependencies. What would be more desirable is a generic prognostic framework that chooses the most accurate diagnostic or prognostic approach from a set of algorithms given component data.

For this purpose, data suitability studies and methodologies have been developed. As highlighted in Section 1.1, data-driven diagnostic and prognostic approaches rely on data covering all phases of normal and faulty operations as well as degradation scenarios under certain operating conditions (Elattar et al., 2016). Not only is such data necessary for model development, but also, as (Jia et al., 2022) points out: Underlying system data and its quality have a major impact on the performance of data-driven diagnostic and prognostic solutions. (Coble & Wesley Hines, 2009) state that defining a suitable, accurate health index, which can be a measure of system degradation, is key to successfully applying prognostics within CBM. An optimization approach is presented to develop a prognostic model based on prognostic parameter suitability metrics. (Y. Chen et al., 2013) suggest a method to evaluate data quality before the modeling by clustering the data into different system conditions. (Omri et al., 2021) propose a set of data quality requirements, especially for step 4 in a CBM framework, the health assessment and fault detection. They propose a 'detectability' metric to assess the suitability of data for fault detection.

A limitation of existing data suitability assessments is that they are often done apart from the actual implementation of data-driven approaches for diagnostics or prognostics. In previous papers, statistical methodologies and pre-defined metrics, depending on the underlying trends in data, are used to assess the data quality. However, Artificial intelligence (AI) based methodologies are in some cases able to detect failures even though the underlying data degradation is not visible or statistically traceable (Braglia et al., 2012). In addition, the quality of the AI models is often influenced by steps taken before the implementation of the actual technique, such as data pre-processing or feature engineering (Lecun et al., 2015). Therefore, diagnostic and prognostic techniques should be integrated into a proper assessment of the suitability of system data in a CBM framework.

When observing the existing literature, we identify two gaps:

1. Data suitability approaches presented in the literature are often focused on the data rather than on the data within a CBM framework.

2. Existing approaches are often tailored to specific systems.

To fill those gaps, what we suggest in this thesis is an integrated framework to assess the data suitability and aspects such as the ability to apply diagnostic and prognostic approaches to systems based on underlying data. Note that we refer to it as "diagnostic" and prognostic framework, while in fact regarding diagnostics it only covers the aspect of anomaly detection. Diagnostics, as (Jardine et al., 2006) point out, covers a number of steps, fault detection as it is addressed through anomaly detection being one of the

most important ones. With this perspective in mind, in the following we keep referring to it as "diagnostics" framework. In addition to providing a data suitability assessment, the framework is generic, can be adapted to various systems and provides guidelines for choosing diagnostic or prognostic methodologies. It outputs good-performing anomaly detection models, respectively remaining useful life estimating models and gives an indication of which techniques to use given a specific dataset. Multiple metrics for the performance assessment of the models are implemented to make the framework more robust. With this one of the challenges of applying anomaly detection and prognostic methodologies in real practice of complex systems is addressed.

1.3. RESEARCH AIM

1.3.1. RESEARCH QUESTION

This thesis fills the in Section 1.2 presented gaps: It presents a way to assess the quality of system data for the development of diagnostic or prognostic models and a way to provide diagnostic or prognostic methodology choices based on the system data. The main research question asked in the thesis is as follows:

How can system data be used to assess the application of diagnostic and prognostic methodologies for failure detection and prediction?

1.3.2. RESEARCH SCOPE

To answer the above research question is a challenging undertaking, especially when considering the vast amount of machine learning and deep learning algorithms available, as well as the fact that systems are operated differently in various conditions and in varying environments. All those factors influence how to approach the task of CBM, of detecting/ predicting system failures. No two systems are exactly the same and so far, we did not even mention all the challenges connected to data quality, data collection, sensors and data storage. In other words- we cannot provide a one size fits all model for diagnostics or prognostics; there is no free lunch. However, it might be worth to at least have cheaper supper (Calikus et al., 2020) - especially when faced with the task of doing a first diagnostic or prognostic assessment for a range of systems and make decisions for further model development based on such an assessment. This could help with deciding where to put further resources in the investigation of diagnostic or prognostic solutions.

What could such a "cheaper supper" look like? In order to provide a diagnostic and prognostic assessment based on system data, such a solution would need to entail both a range of representative diagnostic as well as prognostic data-driven techniques. What is more, diagnostic and prognostic approaches consist of several steps and for each of the steps a range of methods are available. An example for such a step is data manipulation (see Section 1.1) - as well as other steps included in the previously presented OSA-CBM framework. When developing a model, depending on the underlying algorithm, it might be beneficial to use several of those methods, while omitting others. Therefore, in addition to including a range of diagnostic and prognostic methods, a set of techniques for

each of the steps included in the design of a diagnostic or prognostic model needs to be contained in such a solution.

When considering the above indicated complexity of designing a generic diagnostic or prognostic solution, it becomes clear that we cannot design a truly "generic" framework. To make this clearer, consider the following: (Nassif et al., 2021) who summarized what they found in reviewing 290 research articles on machine learning for anomaly detection (which is only a part of diagnostics) from the years 2000 to 2020, found 28 different machine learning methods and 21 different methodologies for feature selection/ extraction. Similarly, (Zio, 2022) in a recent literature review on prognostics and health management identified 16 different types of machine learning techniques for fault detection and 15 different diagnostics and prognostics algorithms. Considering those numbers and considering the possibility to combine algorithms, which multiplies the numbers, shows the scale of the problem. The number of methods that could be included in a generic solution is considerable and when taking into account the additional steps that can be taken to arrive at diagnostics or prognostics models, it even increases. It seems like a big undertaking to provide a "generic" assessment for diagnostics or prognostics. However, it also can be considered a valuable undertaking, especially when the goal is to arrive at a first assessment rather than a fully developed accurate model.

In order to provide such a solution, a generic diagnostic and prognostic framework with the capability to select optimal diagnostic and prognostic settings is developed in this thesis. We start by setting a scope, defining according requirements for such a framework and develop it over the next chapters. A framework with the capability to diagnose or prognose systems based on system data is presented. It provides a data suitability evaluation for developing diagnostic and prognostic models within a CBM strategy. In other words, the framework is used to link diagnostic and prognostic approaches back to the question of data suitability. It is tested on both aircraft and satellite system data and validated on a simulated data set.

1.3.3. CONTRIBUTION

The main contributions of the thesis are as follows:

- A set of requirements for a generic diagnostic and prognostic framework is systematically derived and presented.
- A generic diagnostic and prognostic framework is developed within the set scope and based on the defined requirements.
- The framework is adapted and used on various system data. This includes simulated and real data, fault-related and component run-to-failure data, and satellite and aircraft systems case studies.
- The resulting outcomes are compared, and the value and limitations of the generalizability of the framework are highlighted. This gives further insight into the challenges of diagnostic and prognostic methodologies, especially when applying them to real system data.

- A system data suitability assessment is presented based on the generic framework and tested on real-life aircraft data sets.

1.4. THESIS OUTLINE

The aim of the thesis is to provide a way to use system data to assess the applicability of diagnostic and prognostic methodologies for failure detection and prediction. From the main research question (RQ), the following sub-questions are derived: **Main RQ: How can system data be used to assess the application of diagnostic and prognostic methodologies for failure detection and prediction?**

- **RQ 2: What are the requirements for a generic diagnostic and prognostic framework that is applicable to different components in various applications?**
- **RQ 3: In what way can diagnostic methodologies be integrated into such an adaptive framework that, when applied to given system data, provides a diagnostic assessment?**
- **RQ 4: How can prognostic methodologies estimating a system's remaining useful life be integrated into such an adaptive framework that, when applied to given system data, it provides an assessment of the prognosability?**
- **RQ 5: How can the quality of prognostics on a system be evaluated within a CBM strategy?**

This is addressed in two main parts as visualized in Figure 1.1: In Part 1, a generic diagnostic and prognostic framework is developed and implemented in case studies on aircraft and satellite systems. In Part 2, the framework is used to assess the ability to diagnose or prognose systems based on the underlying system data. The implementation of such a generic diagnostic and prognostic framework is approached from a systems engineering perspective based on ideas presented by (Li, Verhagen, et al., 2020) to address RQ 2. In Chapter 3, a generic diagnostic framework is developed and tested in a satellite system case study to answer RQ 3. Based on RQ 4, the framework is then extended in Chapter 4 to assess the ability to prognose systems and applied to an aircraft case study.

Using diagnostic and prognostic approaches in a CBM context requires a proper assessment of the quality of predictions (Zio, 2022). This thesis also uses this quality assessment to quantify the suitability of system data for diagnostics and prognostics. For this purpose, diagnostic and prognostic metrics are used. A particular focus is put on the assessment of prognostics. An effort to standardize prognostic metrics has been made by (Saxena, Goebel, et al., 2008b), (Saxena et al., 2010). An overview of existing metrics to evaluate prognostic performance is presented in (Ochella & Shafiee, 2021). The authors point out that prognostic metrics should capture three aspects of predictions: accuracy, precision, and timeliness. In this thesis, together with the generic diagnostic and prognostic framework, we attempt to link identified metrics capturing all three aspects, accuracy, precision, and timeliness, back to the data quality, which is addressed in Chapter 5 through RQ 5. Finally, the conclusion and a summary of the findings are presented in Chapter 6.

1

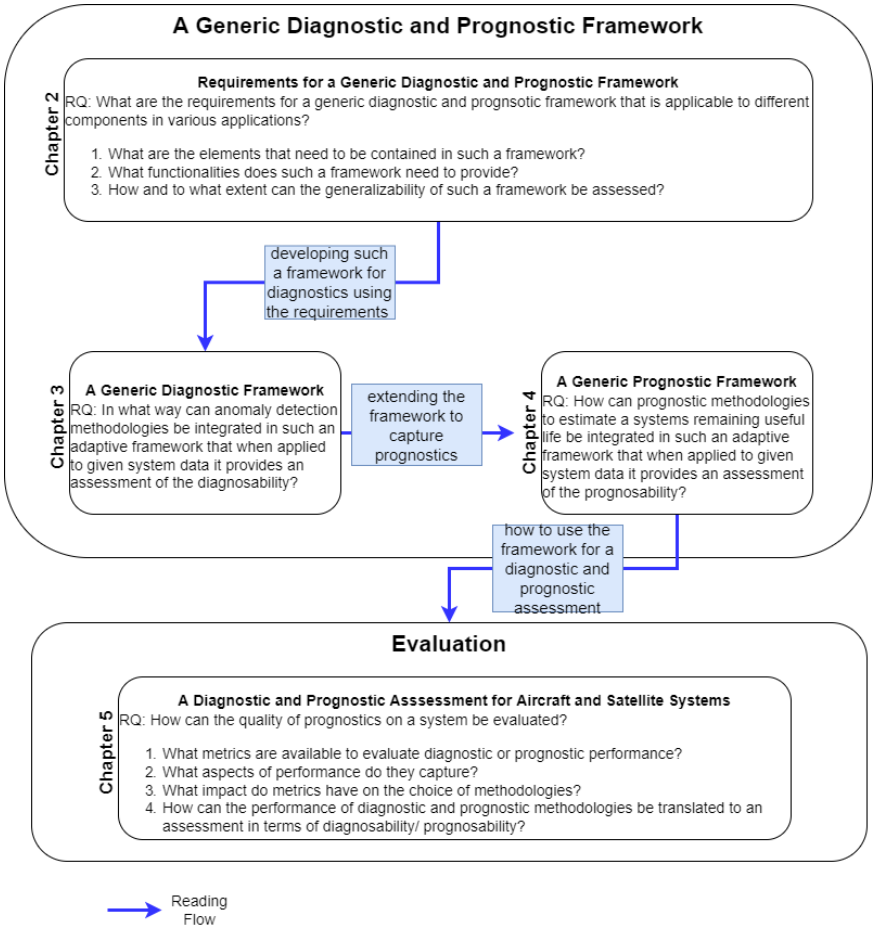


Figure 1.1: Thesis outline and reading flow

2

Requirements for a Generic Framework for Diagnostics and Prognostics

There are two main aspects of the diagnostic and prognostic framework presented in this thesis: First, it provides a guideline towards the choice of diagnostic and prognostic algorithms for complex systems. Second, it assesses whether system data is suitable for data-driven diagnostic and prognostic approaches. In this chapter the requirements for such a framework are defined in a systematic way using system engineering methodologies.

2.1. INTRODUCTION

A variety of mechanical and electrical systems in aerospace are constantly monitored by numerous sensors and system telemetry data is collected continuously. Within a Prognostics and Health Management (PHM) or CBM framework such data can be used for diagnostics or prognostics (M. J. Scott et al., 2022). Diagnostics aim at using data-driven methods to detect failures or anomalies in systems and prognostics aim at predicting such instances before they occur (Swearingen et al., 2007).

When designing and developing data-driven diagnostic or prognostic models, choices have to be made, e.g. regarding suiting algorithms or data pre-processing methods. Those decisions, however, require expertise and time. Since the quality of resulting models depends on those choices, this is a crucial step in the development of diagnostics or prognostics solutions. Consider the example of a satellite, which consists of many subsystems. As highlighted in Chapter **??**, developing diagnostic or prognostic models for each of those subsystems in this way would therefore not only be costly but also time intensive. Furthermore, in some cases it results in models not able to predict or even detect failures or anomalies due to a lack of training data or low data quality.

Therefore, what would help is a framework that automatically creates diagnostic and prognostic models, given system data, and thereby provides both- a data suitability assessment and guidance in further development. However, such a framework is difficult to design, due to several reasons, such as systems exhibiting different failure behaviour, having multiple failure modes, acting differently in various operating conditions and failure data capturing the entire range of failure behaviour being scarce or not available at all. All in all, the development of data-driven diagnostic and prognostic systems is a difficult undertaking.

So, before we dive into the challenging task of designing such a diagnostic and prognostic framework, we approach this from a systems engineering perspective in a systematic way and come up with a set of underlying functional requirements for such a framework. We therefore, starting from our identified need, provide a systems engineering based definition of requirements for a generic diagnostic and prognostic framework for specific stakeholders. The question we ask ourselves is: How can the development of a generic diagnostic and prognostic framework that can be used for a first data suitability assessment be approached in a systematic way? In order to understand this, we first identify a list of stakeholders and define in what environment and by which stakeholders such a framework would be used, before we move to the definition of functional requirements.

The remainder of this chapter is structured as follows: Section 2.2 presents the current state of the art, in Section 2.3 we present the methodology and in Section 2.4 the stakeholders and the functional requirements are defined. The resulting requirements are discussed and further directions for research are indicated in Section 2.5. Finally, we conclude our findings in Section 2.6.

2.2. LITERATURE REVIEW AND BACKGROUND

In the following we give an overview over existing literature on diagnostic and prognostics within the PHM context in Section 2.2.1 and introduce publications in this field providing a systems engineering approach towards the design of diagnostic and prognostic frameworks in Section 2.2.2.

2.2.1. DIAGNOSTICS AND PROGNOSTICS IN THE PHM CONTEXT

PHM and CBM aims at utilizing system data to automatically detect and predict system failures and plan according maintenance actions (Swearingen et al., 2007). There are multiple steps in a typical PHM approach, such as data acquisition or data manipulation, but the two steps we focus on in this work are diagnostics and prognostics. In a PHM approach, system's health is monitored through sensors continuously and the collected information are processed by algorithms to perform diagnostics and prognostics. While diagnostics aim at detecting failures, prognostics deal with predicting failures (Peng, Dong, et al., 2010; J. Zhang et al., 2018). Data-driven diagnostic and prognostic methods can further be split into statistical and AI-based methods. Mainly due to major advances that were made in the past year in the field of AI in general, especially AI-based methodologies have grown more popular recently and many studies have been conducted regarding those. Several literature reviews on such methodologies applied in PHM exist (Diez-Olivan et al., 2019; Elattar et al., 2016; M. J. Scott et al., 2022; J. Zhang et al., 2018).

2.2.2. A SYSTEMS ENGINEERING PERSPECTIVE ON THE DESIGN OF A GENERIC DIAGNOSTIC AND PROGNOSTIC FRAMEWORK

A coherent and thorough overview over the basic principles of Systems engineering (SE) can be found in (Blanchard et al., 1990). In the following, we give an overview over existing literature on applying those principles to determine requirements or some form of standardization for PHM approaches.

An early effort of coming up with requirements for a PHM system has been made by (Brown et al., 2007). The authors highlight and point out the challenges and efforts of applying PHM to the Joint Strike Fighter (JSF) and mention how they translate underlying requirements into the development of PHM solutions. (Saxena et al., 2012) provide requirements for prognostic algorithmic performance of PHM systems. They argue that requirements towards prognostic performance can be retrieved from high level functional requirements limitations and demonstrate their findings in an example. Their focus lies on prognostic algorithms and specifically the evaluation of prognostic models within a PHM system. A set of data quality requirements for PHM applications, especially for fault detection is provided by (Omri et al., 2021). (Li, Verhagen, et al., 2020a, 2020b) provide a systematic PHM architecture design methodology. A thorough definition of requirements for a PHM system is given and the principles of systems engineering are used to provide guidelines for how to systematically design a PHM system.

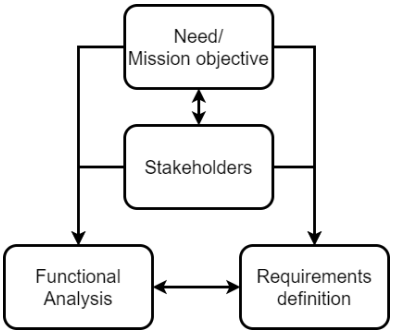


Figure 2.1: Systems engineering: From a need towards requirement definition

2.3. METHOD

We make use of basic principles of SE as explained in more detail in the following sections and shown in Figure 2.1.

2.3.1. NEED/ MISSION OBJECTIVE

In systems engineering the so-called 'Conceptual Design' is the first phase of the system design and development process (Blanchard et al., 1990). The idea is to determine the function, form, cost and development schedule of a system. The starting point for this phase is the identification of a need or mission objective.

2.3.2. STAKEHOLDERS

A stakeholder "is a group or individuals that is affected by or has a stake in the product or project" ("NASA Systems Engineering Handbook Rev. 2.", 2018). It is important to consider stakeholders in the concept design phase of systems engineering as their expectations and needs are influencing and should be reflected in the system requirements.

According to (Viscio et al., 2015) stakeholders can be categorized in the following five categories:

- Sponsors, who define the mission statement, fix bounds on schedule and fees,
- operators, who are in charge of controlling space and ground assets,
- end-users, i.e. people/ entities that receive mission products and capabilities,
- customers, who pay fees to utilize a specific mission's product or service and
- developers, who are those who develop the product based on the mission statement.

2.3.3. FUNCTIONAL ANALYSIS

In a SE approach, one major way to elicit and define requirements is through the application of functional analysis. A 'function' refers to a specific or discrete action (or series of actions) that is necessary to achieve a given objective" (Blanchard et al., 1990). In the functional analysis in an iterative way, functions are defined and translated into top-level system requirements (see Section 2.3.4) and vice versa. The system in this step is defined in 'functional' terms, i.e. with respect to what functionalities it should fulfill in order to meet the top-level requirements and especially the identified need. The question to be asked in this step is therefore more the 'what?' (What should the system be able to do?) rather than the 'how?' (How does the system achieve the required functionality?).

Usually this is done by the means of Functional Flow Block Diagram (FFBD)s. In such a diagram all activities throughout the system life cycle are covered and the relationship or sequences between those is reflected.

2.3.4. REQUIREMENTS DEFINITION

We approach the definition of the requirements as mentioned above through an iterative process using Functional Analysis. Starting from the identified need, we therefore derive a set of top-level requirements and iteratively refine those using the Functional Analysis performed. (Viscio et al., 2015) differ several categories of requirements: mission, functional, configuration, interface, environment, operational, logistic support, performance, design, physical and product assurance and safety related requirements. We aim at building a tool which can be used to assess data suitability. We focus on the development and implementation of this tool in further consequence and less the application to specific user applications. The for this purpose relevant requirements are functional and performance related requirements. Therefore, in this work, we will focus on those two types of requirements.

2.4. RESULTS

2.4.1. MISSION OBJECTIVE/ NEED

As highlighted in Section 2.1, our aim is to provide a systematic (SE-based) way to design and development a Generic diagnostic and prognostic framework (GDPF). Therefore, the need can be formulated as follows:

Provide a framework with the capability to perform diagnostic and prognostic assessment of different aerospace system data for subsequent PHM development.

2.4.2. STAKEHOLDERS

We differ between five types of stakeholders (Section 2.3.2). In our case, there are the following stakeholders:

- Sponsors: airlines, aircraft manufacturers, satellite system designers, space agencies, such as European Space Agency (ESA), private companies launching space missions

- Operators: maintenance engineers, who perform diagnostics and prognostics on an ongoing basis and maintenance planners, who use the framework to plan and schedule maintenance tasks

- End-users/ customers: airlines, Maintenance Return and Overhaul providers, satellite operators

- Developers: data scientists, data analysts developing diagnostic or prognostic models, future missions departments making decisions about in which future research to put time and money

For this purpose, the framework is developed from a developers perspective, i.e. data scientist, data analysts or future missions and projects teams at airlines or space agencies - in other words, players who make decisions about how to develop models further, for which systems to use diagnostic or prognostic approaches and how to assess those for further use in a PHM context. There are several reasons for this choice. First, the developers perspective represents the first step in the development of diagnostic and prognostic methods, before considering deployment of such or actual application for maintenance or other activities within a PHM framework. Second, as already mentioned above, a list of requirements or constraints is and has to be application and sometimes even system specific and therefore would narrow down the analysis to specific systems and applications. However, we want to provide with our research a first assessment, which can adaptively be applied to different systems in aerospace applications.

Next, we might ask what the need of the developer, as identified stakeholder is or what questions such a stakeholder might ask. We identified the following set of questions:

- How do we identify systems with available data suitable for diagnostics/ prognostics?

- How can we get a first diagnostic/ prognostic assessment of such systems?

- Is it possible to build simple (i.e. not too complex) yet robust (i.e. reaching a minimum required performance for system data) models?

- How can such models be assessed, especially towards further applications within a PHM framework?

2.4.3. REQUIREMENTS DEFINITION AND FUNCTIONAL ANALYSIS

As mentioned in Section 2.3.3, the definition of requirements and performing the functional analysis is an iterative process. Since we handle it as such, we present the results for both in a single section. First, starting from the need/ mission objective defined in Section 2.4.1 and keeping the stakeholders, identified in Section 2.4.2, in mind, we define the top-level requirements.

There are four top level requirements as shown in Figure 2.2:

- **R1.0 Utilize aerospace time-series and telemetry data of systems**: The framework should be able to process time-series telemetry and sensor data related to aerospace systems health.
- **R2.0 Capability to perform diagnostics**: The framework should provide the capability of outputting diagnostic models trained using the system input data.
- **R3.0 Capability to perform prognostics**: The framework should provide the capability of outputting prognostic models trained using the system input data.
- **R4.0: Assessment for subsequent PHM use**: The framework should provide as an output metrics/ evaluation criteria that can be further used by developers to make design decisions and understand if the data is suitable for diagnostics or prognostics.

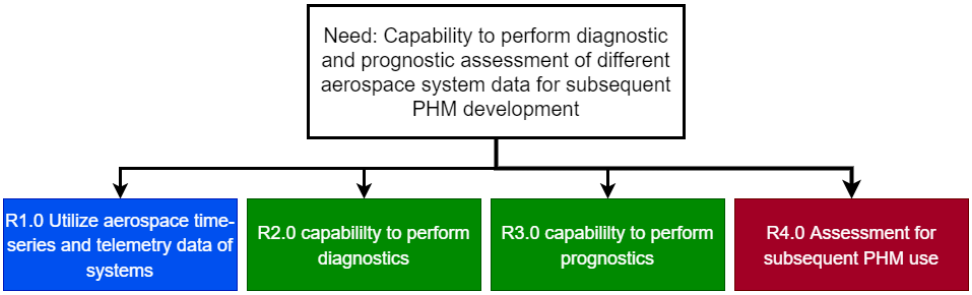


Figure 2.2: Top level requirements for the Generic Diagnostic and Prognostic Framework.

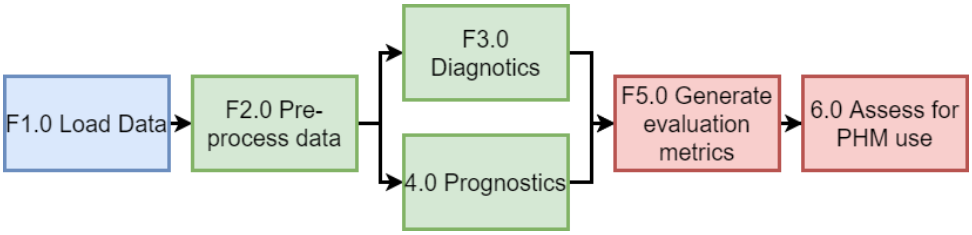


Figure 2.3: High-level Functional Analysis for the Generic Diagnostic and Prognostic Framework.

In Figure 2.3 the according Functional flow block diagram is shown including the high-level functionalities the framework needs to be able to provide and perform. The colours in the figure indicate which function relates to which top-level requirement. According to the OSA-CBM standard framework (Swearingen et al., 2007), the main functionalities for such a generic diagnostic and prognostic framework are as follows:

- F1.0 Load system data,
- F2.0 pre-process input data
- F3.0 build diagnostics models,

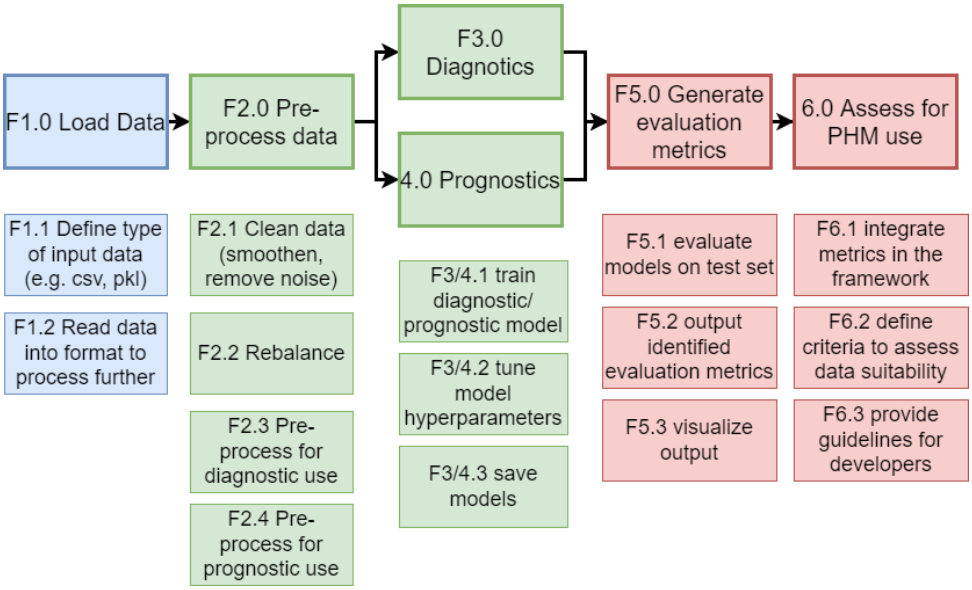


Figure 2.4: Detailed Functional Analysis for the Generic Diagnostic and Prognostic Framework.

- F4.0 build prognostics models,
- F5.0 generate evaluation metrics and
- F6.0 assess the models for PHM use.

Figure 2.4 provides a more detailed overview over the functionalities.

Based on the top-level requirements and the identified functionalities, further requirements are defined. Figures 2.5, 2.6 and 2.7 show the according detailed set of requirements derived from top level requirements 1.0, 2.0 and 3.0 and 4.0 respectively. The requirements 2.0 and 3.0 regarding the diagnostic and prognostic models are handled simultaneously because they are in essence the same.

The requirements derived from top level requirement R1.0, which states the framework should be capable of utilizing time-series and telemetry data of systems, as presented in Figure 2.5, are as follows:

- **R1.1 Capability to handle missing/incomplete or incorrect data:** The framework should include solutions to identify and pre-process missing, incomplete or incorrect data.
- **R1.2 Capability to handle imbalanced data:** The framework should be able to handle imbalanced data, e.g. when the number of failures or anomalies in the provided system dataset is low.
- **R1.3 Capability to handle multi-variate time series data:** The framework needs

to be able to handle and pre-process time-series data containing multiple features and variables that are continuously measured over time.

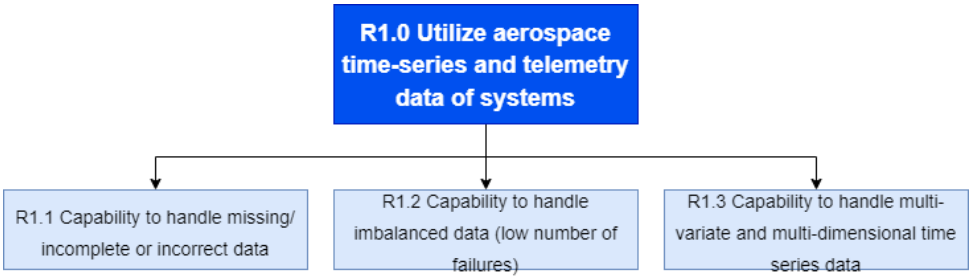


Figure 2.5: Detailed list of requirements derived from top level requirement 1.0 for the Generic Diagnostic and Prognostic Framework.

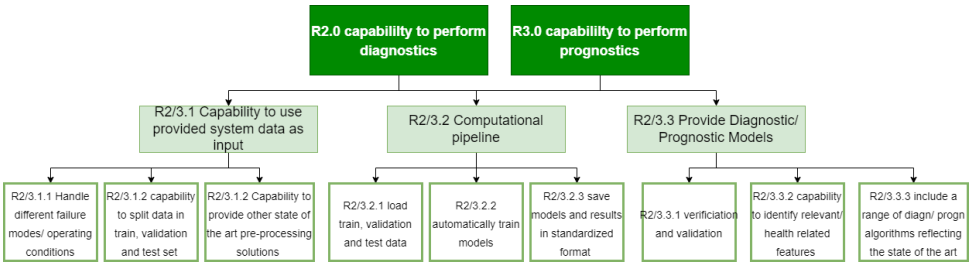


Figure 2.6: Detailed list of requirements derived from top level requirement 2/3.0 for the Generic Diagnostic and Prognostic Framework.

The requirements derived from top level requirement R2.0 and R3.0, related to the diagnostic and prognostic models are shown in Figure 2.6 and are as follows:

- **R2/3.1 Capability to use provided system data as input**: The framework needs to be capable to pre-process the provided system data in a way that it can further be used by the diagnostic and prognostic models contained in the framework. In more detail, this includes the following sub-requirements:
  - R2/3.1.1 The framework is able to handle different failure modes and operating conditions.
  - R2/3.1.2 It needs to be capable to split the data in train, validation and test set.
  - R2/3.1.3 It needs to provide and include other state of the art pre-processing solutions.
- **R2/3.2 Computational pipeline**: The framework needs to perform the required functionalities in an automated way, i.e.

  - R2/3.2.1 it needs to be able to automatically load train, test and validation data,
  - R2/3.2.2 automatically train the diagnostic and prognostic models and
  - R2/3.2.3 save the models and results in a standardized format, so that in a further step the evaluation metrics can be automatically calculated and extracted.

- **R2/3.3 Diagnostic/ prognostic models**: The framework needs to be able to output diagnostic/ prognostic models. This includes the following:
  - R2/3.3.1 It should provide a verification and validation solution for the diagnostic and prognostic models.
  - R2/3.3.2 It needs to be capable to identify relevant/ health related features.
  - R2/3.3.3 The framework should include a range of diagnostic/ prognostic algorithms reflecting the state of the art and different model types (such as machine learning based methods, or statistical algorithms).

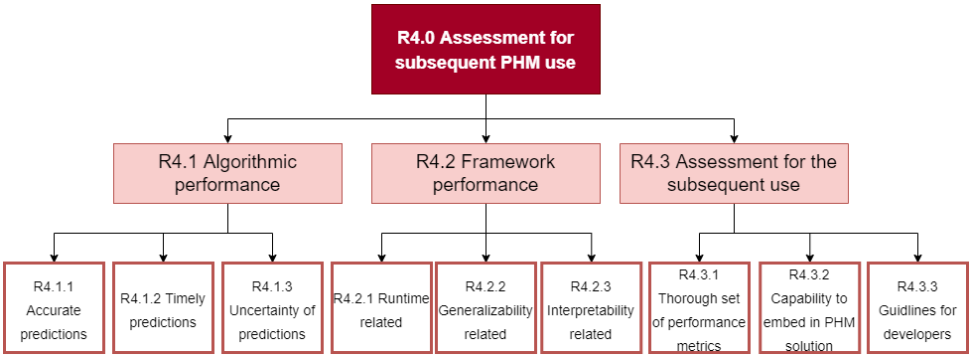


Figure 2.7: Detailed list of requirements derived from top level requirement 4.0 for the Generic Diagnostic and Prognostic Framework.

Finally, in Figure 2.7, the requirements derived from top level requirement R4.0, related to the assessment within a PHM framework, are shown. They include the following:

- **R4.1 Algorithmic performance**: The framework needs to be able to assess the diagnostic and prognostic model performance in a comprehensive and suiting manner.
  - R4.1.1 It needs to be able to measure and output model accuracy.
  - R4.1.2 It needs to be able to assess whether the predictions are produced far enough in advance for subsequent use in PHM applications (e.g. for scheduling maintenance).
  - R4.1.3 The framework should measure and return the prediction uncertainty.

- **R4.2 Framework performance**: The framework should meet performance requirements, which are mostly linked to constraints set by the stakeholder. Below we provide an example of three such requirements.
  - R4.2.1 The framework needs to run within a specified time limit.
  - R4.2.2 The framework needs to be generalizable, i.e. it needs to be adaptive (applicable to various aerospace systems), generic (include a thorough set of methodologies) and robust (produce reliable and consistent results).
  - R4.2.3 The outputs of the framework need to be interpretable to be used within a PHM setting.
- **R4.3 Assessment for the subsequent use**: The framework should provide an assessment of the system data with regards to whether or not they are suitable for diagnostics or prognostics in PHM.
  - R4.3.1 It therefore needs to include a thorough set of performance metrics.
  - R4.3.2 It should be capable to be embedded in a PHM solution.
  - R4.3.3 It should provide guidelines for further diagnostic/ prognostic model development and deployment.

2.5. DISCUSSION AND LIMITATIONS

A set of requirements has been systematically derived and presented for a generic diagnostic and prognostic framework. This was done on basis of the identified need to provide a framework to perform a diagnostic and prognostic assessment of different aerospace system data. However, there are two main limitations we want to point out. The first concerns the completeness of the presented requirements. As already mentioned in Chapter 1, the question of what a generic framework needs to include is manifold and complex and often depends on factors, such as the underlying system and its failure behaviour. It is in part due to this that we cannot guarantee a completeness of such a framework, which translates back to the requirements: How can we, for example, be sure that the requirements we impose upon data pre-processing methods cover all aspects that need to be covered? This is especially true when considering that new methods, techniques and ways to perform tasks (such as data pre-processing) are introduced and further developed over time. The second concerns the validation of the set of requirements. In order to validate requirements in a systems engineering setting, one needs to ensure three points: First that the set of requirements is consistent. Second that a practical system can be built that satisfies the requirements and third that it is possible to prove that the system satisfies the requirements. As to the second and third point, we will come back to those in Chapter 6 and validate the requirements based on the developed framework. However, we did not perform a consistency check of the requirements, which should be the next step in its validation. Despite those two limitations, the list of requirements is as complete as possible and can provide guidance in the development of a generic framework for diagnostics and prognostics.

2.6. CONCLUSION

While existing data-driven prognostic and diagnostic solutions yield promising results, it takes time and expertise to tune them to a specific system dataset. A generic framework to perform this task automatically is needed to provide a quick diagnostic and prognostic assessment. In this chapter, we approach the development of such a generic diagnostic and prognostic framework in a systematic way and use system engineering methodologies for this purpose. We define a need statement, identify possible stakeholders and based on those perform a functional analysis to define requirements. This work can be used as a formal baseline for the further development and implementation of a framework that takes as an input system data and outputs a diagnostic and prognostic assessment for the provided input data.

3

A Generic Framework for Diagnostics of Complex Systems

Based on the requirements defined in the previous chapter, in this chapter we introduce a generic framework for diagnostics. In addition to diagnostic techniques, it also includes methodologies for data pre-processing and feature engineering. This framework is applied in two case studies. First, the performance of the framework is benchmarked against existing approaches by applying it to an open source dataset. Second, it is used to detect satellite system anomalies. Thereby an initial verification and validation of the framework is performed, with particular attention towards its generalizability. Furthermore, the framework not only proves to be helpful in the choice of diagnostic methodologies, but also shows the challenges with using real world data for data-driven diagnostics.

This chapter is based on on the publication: Bieber, Marie, Verhagen, Wim JC, Cosson, Fabrice, and Santos, Bruno F . "Generic Diagnostic Framework for Anomaly Detection—Application in Satellite and Spacecraft Systems." Aerospace 10(8), (2023): 673. (Bieber et al., 2023).

3.1. INTRODUCTION

Spacecraft consist of many complex systems, with each system's functional and operational availability contributing to the overall spacecraft availability. Failures and faults of a single system can lead to major operational interruptions and substantial costs. Therefore, spacecraft operators go to great lengths to ensure the high reliability of all systems and subsystems (J. Chen et al., 2021). Currently, reliability and availability calculations of most systems are based on historical data and statistical analysis (Fuertes et al., 2016). While spacecraft systems are equipped with sensors recording telemetry and system behaviour in regular time intervals, the vast amount of available data is still not fully explored (Hundman et al., 2018). However, together with operational and technical system data, such sensor data can be used to detect, diagnose and predict faults and failures and plan according to actions.

Fault or anomaly detection is typically seen as the first major step in prognostics and health management (PHM). It aims at identifying data deviating from what is considered normal, expected or likely behaviour (Zio, 2022). Several anomaly detection approaches exist, ranging from statistics or signal processing techniques to machine learning (Basora et al., 2019). As mentioned above, most existing approaches for spacecraft rely on statistical models. However, as Zeng et al. (Zeng et al., 2022) point out, statistical models for anomaly detection rely on historical data, which makes them inflexible towards new failure modes or change(s) in operating conditions, leading to thresholds often not being exceeded and is associated with time-consuming development. Furthermore, faults occur randomly for some systems, and failure modes are diverse. Therefore it can be challenging to collect sufficient historical data representing all types of faults (Z. Chen et al., 2023). With this in mind, machine learning models have gained popularity over the past few years and have been widely developed for anomaly detection in other engineering applications. For example, Shao et al. (Shao et al., 2023) developed an unsupervised machine learning-based anomaly detection approach for application to wind turbines. An online adaptive transfer learning model for unsupervised anomaly detection for steam turbines is presented by Chen et al. (Z. Chen et al., 2023).

Over the past years, especially fuelled by the increased number of small satellites (cubesat) launches, there has been an increase in published research on telemetry data and its usage for anomaly detection for satellite systems. Chen et al. (J. Chen et al., 2021) present a real-time onboard satellite anomaly detection system based on Bayesian neural networks, which characterise uncertainty and re-evaluate samples with high uncertainty. Hundman et al. (Hundman et al., 2018) achieve high performance for spacecraft anomaly detection with an LSTM network mainly due to their non-parametric, dynamic and unsupervised technique to set the threshold. An anomaly detection approach considering parameter interactions is suggested in (Zeng et al., 2022). The drawback of those anomaly detection approaches as well as the ones presented in the previous paragraph for other applications, is that they aim for more complexity in algorithms instead of trying to find out which methods work best for the underlying data or simply understanding if the data is suitable for anomaly detection at all. In other words, a fundamental underlying assumption is present regarding anomalies and the associated data's suitability for

anomaly detection approaches. This assumption is not necessarily true: it can, for example, be the case that failures occur suddenly or there are so many failure modes and operational conditions to consider that much more data would be required to train machine learning models. In addition, it could also be the case that available data does not capture degradation, for instance, because the sensor properties do not represent the underlying physical degradation process.

Therefore, as Fink et al. (Fink et al., 2020) point out in their article addressing challenges and future directions for deep learning in PHM applications, what is needed are anomaly detection approaches which are both applicable and adaptable to different systems and failures. Such a framework is presented in this paper: The Generic Diagnostic Framework (GDF) takes as input system data and outputs the optimal combination of data preprocessing and anomaly detection methods as expressed in terms of predefined metrics. It thereby provides a quick diagnostic assessment for the underlying system and, at the same time, gives an indication of which AI-based methods are worth pursuing further (if applicable).

There are two things worth noting regarding the in this paper presented framework: First, it is referred to as "diagnostic" framework, while in fact it is a "Generic Anomaly Detection Framework". Diagnostics, as Jardin et al. (Jardine et al., 2006) point out, incorporates the steps of fault detection, isolation and identification. Anomaly detection only deals with a part of it, namely fault detection. The purpose of the framework, however, is to be adaptive and it can easily be extended incorporating multiple more methods also for fault isolation and identification. Therefore, we will continue to refer to it as "Generic Diagnostic Framework" in the remainder of the paper. Second, we claim it to be "generic". When considering the scale of the problem, the amount of machine learning methods available and challenges, such as those related to using real-life data, it becomes clear that such a framework can never truly be "generic". In recent reviews on machine learning methods for anomaly detection, the scale of the problem becomes clear: Choi et al. (Choi et al., 2021) who focus on deep learning methods only, for example, list 27 methods in total. Nassif et al. (Nassif et al., 2021) who summarized what they found by looking at 290 research articles on machine learning from the years 2000 to 2020, found 28 different machine learning methods and 21 different methodologies for feature selection/ extraction. And Zio (Zio, 2022) lists 16 methods only for the step of fault detection, just to give a few examples. However, the purpose of the framework is to provide a quick assessment and further guidance for the development and employment of diagnostic methods based on system data. Furthermore, as demonstrated in three case studies, it is generic in the sense that it is capeable of taking into account different systems and can be adapted quickly.

We pursue the following three objectives: First, to provide an adaptive framework which outputs good-performing anomaly detection models and gives an indication of which techniques to use given a specific dataset. Second, to make the framework robust by including multiple metrics for the performance assessment of the anomaly detection models. Third, to improve the anomaly detection models further by including thresh-

olding methodologies. Our contributions can be summarised as follows:

- A robust and adaptive framework for automatically creating anomaly detection models is presented.
- The framework is applied in three case studies, including benchmark datasets for satellite and spacecraft systems and a real-life satellite dataset provided by the European Space Agency (ESA).

The remainder of the paper is structured as follows: Section 3.2 gives an overview of existing literature on anomaly detection with a special focus on space applications and generic methods. In Section 3.3, the generic diagnostic framework is introduced. Section 3.4 presents the conducted case studies and the discussion, and Section 3.5 summarises the main findings and indicates directions for further research.

3.2. Literature Review and Background

In the following, we present a general overview over literature on anomaly detection in Section 3.2.1, provide an introduction into adaptive anomaly detection methods in Section 3.2.2 and finally focus on advances for adaptive anomaly detection methods applied to space applications in Section 3.2.3.

3.2.1. Anomaly Detection

Anomaly detection has been studied widely and finds application areas in many domains. The term 'anomaly detection' or 'outlier detection' refers to finding data patterns that are not aligned or do not conform to expected behaviour (Nassif et al., 2021). (Chandola et al., 2007) differ three types of anomalies:

- point anomalies, which are punctual occurrences of anomalous data with respect to the remaining data,
- contextual anomalies, which are instances that show anomalous behaviour in a specific context, e.g. instances with relatively larger/smaller values in their context but not globally and
- collective anomalies are anomalies consisting of a set of related data instances (e.g., occurring at a specific time range) that are anomalous with respect to the entire data set.

Taxonomy of Anomaly Detection Methods

Data-driven anomaly detection techniques can be classified into statistical and AI-based methods. As pointed out in Section 3.1, in this study, we focus on AI-based methods, in particular machine-learning (ML) methods. Recent reviews, such as (Choi et al., 2021; Khan et al., 2021; Nassif et al., 2021), provide an overview of such techniques. (Basora et al., 2019) provide a comprehensive summary of advances in anomaly detection applied to aviation. Based on (Basora et al., 2019), we classify AI-based anomaly detection techniques in four categories, as shown in Figure 3.1:

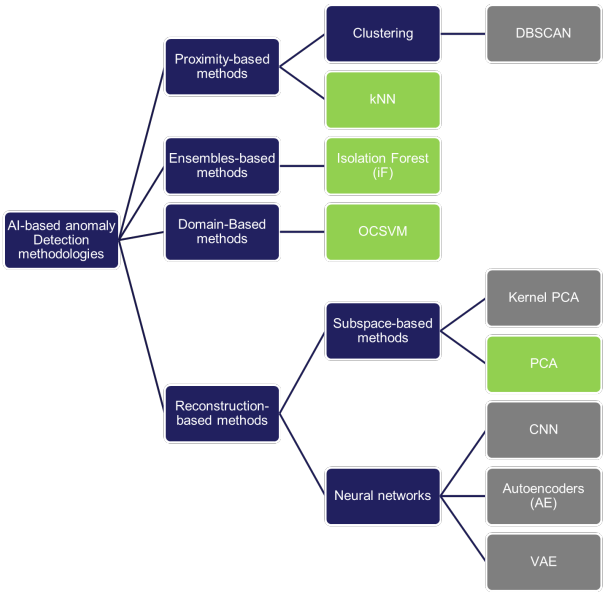


Figure 3.1: Taxonomy of AI-based anomaly detection methodologies. The methods marked in green are the ones included in the Generic Diagnostic Framework.

- proximity-based methods, which rely on the definition of a distance/ similarity function between two data instances,
- ensemble-based methods, which use ensembles of AI algorithms for anomaly detection,
- domain-based methods, which define boundaries or domains to separate normal data from anomalies and
- reconstruction-based methods, which embed data in a lower dimension to separate normal instances from anomalous ones.

THRESHOLDING

The outputs of anomaly detection techniques are scores and labels as defined in (Chandola et al., 2007). Scores are assigned to each instance, depending on whether it is an anomaly. Thus, scores can be viewed as a ranked list of anomalies. Those scores are, in further instances, used to assign labels to each data instance. Labels are binary values and simply classify a data instance as normal or anomalous. In order to calculate labels using the scores, thresholding techniques are used. Setting an appropriate threshold influences the quality of an anomaly detection model and is always a trade-off (Basora et al., 2021a; Choi et al., 2021). If it is set too high, anomalies will be missed, and if it is set too low, the rate of false positives will become high. Typically used methodologies

for thresholding are Area Under Curve Percentage (AUCP) (Ren et al., 2019), Median Absolute Deviation (MAD) (N. & Pawar, 2015), Modified Thompson Tau Test (MTT) (Rengasamy et al., 2021), Variational Autoencoders (VAE) (Xiao et al., 2020), Z-Score (Bagdonavičius & Petkevičius, 2020) or Clustering based techniques (Klawonn & Rehm, 2011).

3.2.2. ADAPTIVE ANOMALY DETECTION METHODS

We claimed in Section 3.1 that in many cases, the techniques presented in the literature are tuned to specific applications or even datasets. Still, there have been some efforts in the past to create more generic methods. (C. Zhao & Shen, 2022) present an adaptive open set domain generalisation network using local class cluster-based representation learning and class-wide decision boundary-based outlier detection. In (Alam et al., 2019), a simple yet robust way to detect anomalies in arbitrary time series by detecting seasonal patterns and identifying critical anomaly thresholds is presented. A meta-framework to create unsupervised anomaly detectors is introduced by (Calikus et al., 2020). The output is a suitable anomaly detection model of temporal streaming data. Several methods for anomaly detection are included, however, not all proved to be resilient against noise and different anomaly types in the data. In addition, several papers have been published guiding or even enabling automatic machine learning model development. (Akiba et al., 2019), for example, present an open-source solution for automatic hyperparameter selection. Such tools are powerful and provide easily adaptive solutions for machine learning model development. However, they are very generic and in order to adapt them to specific applications choices have to be made with regards to machine learning or feature engineering methods.

3.2.3. ADAPTIVE ANOMALY DETECTION METHODS FOR SPACE APPLICATIONS

Efforts to develop more adaptive anomaly detection models for spacecraft systems using telemetry data have been made, for example, at the German Space Operation Center (GSOC). A statistical-based anomaly detection approach, called "automated telemetry health monitoring system" (ATHMoS), is presented in (O'meara et al., 2016). The authors explored the application of deep neural networks within ATHMoS in (O'meara et al., 2018). An autoencoder was applied for automatic feature extraction, and a Long short-term memory (LSTM)-Recurrent Neural Network (RNN) structure was used for anomaly detection. The authors found, however, that due to the complexity of the methods and the black-box nature of the outputs, such approaches are challenging to apply to satellite telemetry data, especially when trying to link the output to the raw sensor signal. For this purpose one could make use of existing techniques in other domains. For example, a visual representation technique linking the output of Bayesian Recurrent Neural Networks back to input signals to identify faults is presented in (Sun et al., 2019). (Freeman et al., 2022) provide guidelines on choosing anomaly detection methods based on characteristics in time series (such as seasonality, trend or missing time steps). Several anomaly detection methods are compared, and current challenges of anomaly detection methods for time series are provided. What the above-presented methods have in common, though, is that they tend to focus on the data rather than on the more complex dynamics of using the data within a PHM framework.

		Prediction outcome		
		p	**n**	**total**
Actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Table 3.1: Confusion Matrix

3.3. METHODOLOGY

Using machine learning methods for anomaly detection, we aim to understand if system data is suitable for anomaly detection in the first place. For this purpose, we make use of a Generic diagnostic framework (GDF), which is an extension of the Generic Prognostic Framework presented in (Bieber & Verhagen, 2022). While the underlying idea and concept remain the same, we extend the framework to include anomaly detection methods. The basic idea is that taking system data as an input, the framework optimises the choice of data pre-processing techniques in combination with anomaly detection and thresholding methods simultaneously. The details of this process are explained in Section 3.3.2. Such an optimisation relies heavily on the choice of suitable metrics. We argue that using a single metric for our purpose is insufficient since a single metric cannot capture the quality of a resulting machine learning model to a full extent. This is explained in more detail in Section 3.3.1.

3.3.1. METRICS FOR ANOMALY DETECTION

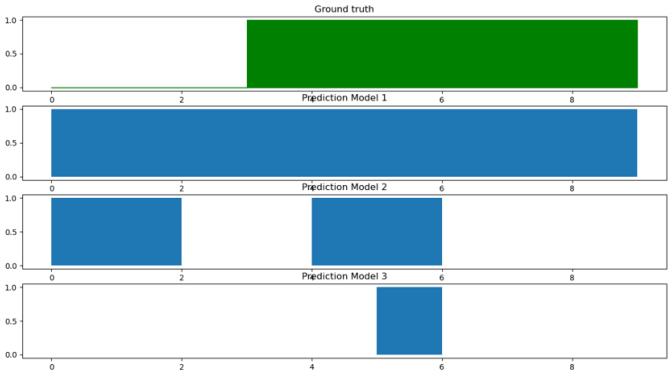
The anomaly detection problem is a classification problem in ML. Classification problems output binary values, and therefore, each resulting prediction can be one of the four: A true positive, if the true value was predicted correctly, a false positive, if an anomaly was predicted but none occurred, a false negative, if an anomaly occurred but was not predicted, or a true negative, in case no anomaly occurred and none was predicted. This can be visualised in the form of a confusion matrix as in Figure 3.1.

The typically used metrics for classification problems are precision (P) and recall (R), computed as follows:

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN}, \tag{3.1}$$

with TP denoting the number of true positives, FN the number of false positives and FN the number of false negatives. The precision is the fraction of relevant anomalies among retrieved ones, while the recall is the fraction of retrieved relevant anomalies. Using precision and recall, the F1 score can be calculated as their harmonic mean, i.e.

$$F1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{3.2}$$



	F1	F1pa	FC
M1	0.75	0.75	0.75
M2	0.36	0.80	0.57
M3	0.29	1.00	1.00

Figure 3.2: Examples of anomaly detection model outputs and their resulting F1, F1pa and FC scores.

One can argue that the F1 score is not an optimal metric for anomaly detection, as it tends to produce low scores, even though the anomaly was detected (G. Y. Kim et al., 2022). This can be seen in Figure 3.2, where the F1 score for anomaly detection model 3 is only 0.29, although the anomaly was detected. For this reason, a new metric has been introduced by (Hundman et al., 2018): the F1 point adjust (F1pa). An in-depth definition and description can be found in (S. Kim et al., 2022). The basic idea behind it is that if at least one moment in a contiguous anomaly segment is detected as an anomaly, the entire segment is then considered to be correctly predicted as an anomaly. This is referred to as event-based scoring. The F1 point adjust score is then calculated with the adjusted predictions.

However, also the F1pa does not come without criticism. (S. Kim et al., 2022) point out that it overestimates the quality of anomaly detection models. Anomaly detection model 2 in Figure 3.2, for example, receives an F1pa score of 0.8 while predicting an anomaly where none occurred. In order to compensate for this behaviour, the composite F1 score (FC) has been introduced by (Garg et al., 2021). The FC score is calculated similarly to the F1 score by taking the harmonic mean of precision and recall. The recall is event-based calculated instead of instance-based, whereas the precision is calculated

instance-based.

As it becomes clear from our line of argumentation, no single metric is able to capture the quality of diagnostic models to a full extent. No metric is flawless; suitable metrics should be chosen carefully. Of course, such a choice should be made application specific and with the purpose of the anomaly detection model output in mind. Because we aim to provide an adaptive framework, which is not application specific, we do not pick a single metric but instead optimise towards all three presented metrics, F1 score, F1pa score and FC score. This is explained in more detail in the next Section 3.3.2.

3.3.2. THE GENERIC DIAGNOSTIC FRAMEWORK

The GDF, visually represented in Figure 3.3, outputs for given system data and, in terms of pre-defined metrics, an 'optimal' anomaly detection model for the system. We assume that the underlying system data is time series data and comes in the form of sensor readings/ telemetry values, which are continuously recorded over a certain period of time. An example of what such data could look like can be found in Sections 3.4.2 and 3.4.4. The GDF includes a range of data pre-processing techniques, anomaly detection, and thresholding techniques. The choice of the respective techniques is approached as a multi-objective optimisation problem, simultaneously allowing to optimise towards all three selected metrics, F1 score, F1pa score and FC score. To be more precise, the problem of finding the respective combination of techniques can be formulated as the following optimisation problem: The objective function is to maximize the F1, F1pa and FC scores of the anomaly detection algorithm together with data pre-processing and thresholding techniques on the system data set. The output of such an optimisation is a Pareto front, which consists of multiple individuals outperforming the remaining individuals in terms of the chosen metrics. A detailed explanation of the workings and dynamics of the framework and the multi-objective optimisation problem can be found in (Bieber & Verhagen, 2022), in which the Generic Prognostic Framework is presented, which is the basis for the GDF presented here. In the following, we go into more detail concerning the genetic algorithm which is used to solve the optimisation problem in Section 3.3.2, the data pre-processing in Section 3.3.2, the anomaly detection methods in Section 3.3.2 and the thresholding techniques in Section 3.3.2 included in the framework.

MULTI-OBJECTIVE GENETIC ALGORITHM

Genetic Algorithms are based on the concepts of natural selection and genetics (Holland, 1992). Due to their flexibility, Genetic algorithm (GA)s are able to solve large optimisation problems. In addition, since GAs are a population-based approach, they are well-suited for multi-objective optimisation problems, like in our case, simultaneously optimizing towards three different metrics (F1 score, F1pa and FC score) (Stanovov et al., 2017). This is what makes them good candidates for our optimisation problem. A population of solutions are created, and their respective fitness values are computed in every generation (Konak et al., 2006). We make use of the Non-dominated Sorting Genetic Algorithm II (NSGA-II, introduced in (Deb et al., 2002)). It ranks candidate solutions with the fast non-dominated sorting method and uses a crowding distance as a diversity mechanism. The algorithm is well-tested, has been used in many applications and is

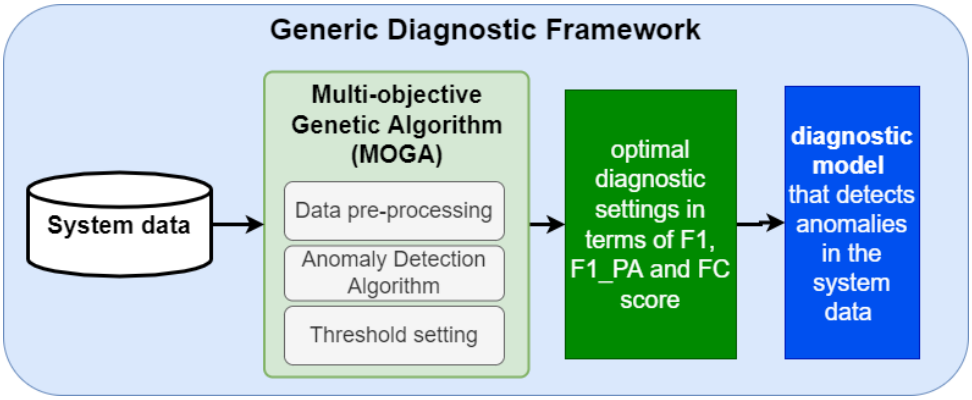


Figure 3.3: Elements of the Generic Diagnostic Framework.

efficient.

Algorithm 1: Genetic Algorithm

```
start;
t ← 0;
initialize population P(t);
evaluate fitness of each individual in P(t);
while termination condition not fulfilled do
    t ← t + 1;
    s1, s2 ← select individuals from P(t);
    x1, x2 ← create offspring by crossover operation on s1, s2;
    x̂1, x̂2 ← mutate x1, x2;
    evaluate fitness of x̂1, x̂2 if fitness of x̂1, x̂2 higher than least fittest individuals
     in P(t) then
        replace least fittest individuals with x̂1, x̂2;
    else
        pass;
    end
end
```

A GA consists of several steps as presented in Algorithm 2. The process is as follows:

- A population is initialised, composed of a set of individuals (i.e., solutions to the optimization problem).
- The best-fitted individuals are selected based on a fitness metric which represents the objective.
- In the following step, the selected individuals undergo a cross-over and mutation process to produce new children for a new generation of individuals.

- This process is repeated over a number of generations until the algorithm converges or a stopping criterion is achieved.

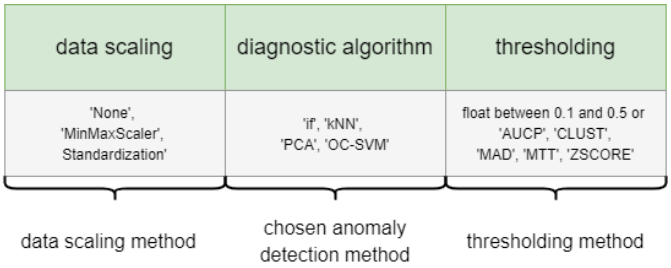


Figure 3.4: GDF individual.

The Multi-objective Genetic Algorithm (MOGA) takes as an input the system data and outputs the set of Pareto optimal solutions. A solution combines a data re-balancing technique, an anomaly detection method and a thresholding technique. Therefore, an individual of the MOGA takes the form as shown in Figure 3.4.

DATA PRE-PROCESSING

Data pre-processing is an essential step in the application of data-driven diagnostic methodologies. Commonly used data pre-processing methods for time series data are data standardization or normalization and signal-processing methods, such as time-domain analysis, frequency-domain analysis, time–frequency analysis and sliding windows to de-noise data (R. Liu et al., 2018). Furthermore, machine learning algorithms are often combined with a feature extraction or feature selection algorithms. Since the framework is supposed to be adaptive to different systems and data pre-processing heavily depends on the nature of the data and the underlying system. In addition, failure behaviour dynamics and the way system degradation is represented in the underlying data influence the selection of those methods. In order to make the framework as adaptive as possible, we only include the minimum amount of required data pre-processing techniques. However, data normalisation and standardisation are necessary steps when applying ML algorithms, especially when the input data is multi-dimensional like in our case. Therefore, the two included methods for the data scaling are 'Standardization' and 'MinMaxScaler', or normalisation. Standardization, or also Z-Score normalisation, results in variables with the properties of a standard normal distribution. Normalisation, or the MinMax scaler, scales the input data to a pre-defined range, in this case, $[0,1]$. Note that the cost of having this bounded range - in contrast to standardization - is that we can end up with smaller standard deviations, which can suppress the effect of anomalies. We also include the option 'None', in which no scaling method is chosen.

ANOMALY DETECTION

The anomaly detection methodologies represented in the framework should capture as many different techniques with different underlying dynamics as possible. For this rea-

son, we based the selection of the methods on the taxonomy of AI-based anomaly detection methods in Section 3.2. In Figure 3.1 we differed four categories of anomaly detection methods, namely proximity-based, ensemble-based, domain-based and reconstruction-based methodologies. In the framework one representative method of each of the four categories is included. Those are

- k-Nearest Neighbors (KNN) as presented in (Angiulli & Pizzuti, 2002), which measures the distance between data points and classifies the points with the highest distance from the other instances as anomalous,
- Isolation Forests (iF) as introduced by (F. T. Liu et al., 2008), which build tree structures to isolate data points (which are considered as anomalies),
- Principal component analysis (PCA), which performs a linear dimensionality reduction into a lower dimensional space to compute outlier scores and
- One Class-Support Vector Machines (OC-SVM), which estimate the support of a high-dimensional distribution and thereby define non-linear boundaries around the region of the normal data (separating the remaining points as anomalies).

In order to define initial settings for each of the four techniques, as a first step the hyperparameters are tuned for each. Table 3.2 contains the respective parameters and tested values.

Note that all our experiments are conducted in Python and for the anomaly detection methods, the PyOD toolbox is used (Y. Zhao et al., 2019).

THRESHOLDING

As highlighted in Section 3.2, thresholding methods can help improve the quality of anomaly detection methods. In the PyOD toolbox, every anomaly detection method returns outlier scores but also has an integrated thresholding method calculating the labels. We include both the default threshold setting provided by the PyOD algorithms and additional thresholding techniques in the framework. In a MOGA individual, see Figure 3.4, the default threshold methods are represented by the float options for the threshold settings (0.1 to 0.5). This is because PyOD calculates the thresholds based on the contamination rate, which is the rate of expected anomalies in a dataset. In the optimisation process of the MOGA, this can be regarded as an additional hyperparameter for the anomaly detection methods used being tuned. In order to provide a truly unsupervised and adaptive framework, several thresholding methods apart from the pre-implemented ones are included in the framework. Those are

- the AUCP,
- a clustering-based method (CLUST),
- the MAD,
- the MTT and
- the Z-Score (Z-Score).

Table 3.2: The hyperparameters and tested values for the four anomaly detection methods.

Method	Hyper parameter	Description	Tested Values
Isolation Forest	`max_samples`	Size of the tree, number of samples to draw from X to train each base estimator	100, 300, 500, 700
	`n_estimators`	number of trees in the ensemble (default is 100 trees)	100, 200, 300, 400, 500
	`max_features`	number of features to draw from X to train each base estimator (default value is 1.0)	5, 10, 15
KNN	`n_neighbors`	Number of neighbors to use for k neighbors queries	1,4,8,12,16
	`p`	Parameter for Minkowski metric	1,2,3
	`method`	• 'largest': use the distance to the kth neighbor as the outlier score • 'mean': use the average of all k neighbors as the outlier score • 'median': use the median of the distance to k neighbors as the outlier score	'largest', 'mean', 'median'
	`algorithm`	Algorithm used to compute the nearest neighbors: • 'ball_tree' will use BallTree • 'kd_tree' will use KDTree • 'auto' will attempt to decide the most appropriate algorithm based on the values passed to fit method	'auto', 'ball_tree', 'kd_tree'
PCA	n_components	number of components to keep	Np.arrange(1,20,2)
OC-SVM	`kernel`	Specifies the kernel type to be used in the algorithm used to pre-compute the kernel matrix.	'rbf', 'poly', 'sigmoid', 'linear'
	`nu`	upper bound on the fraction of training errors and a lower bound of the fraction of support vectors	0.1, 1, 10, 100, 1000
	`gamma`	Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.	np.arange(0,1,0.2)

The AUCP makes use of the area under the curve (AUC) to calculate the outlier labels using the outlier scores (Ren et al., 2018). The AUC is defined as

$$AUC = lim_{x \to inf} \sum_{i=1}^{n} f(x)\delta x, \tag{3.3}$$

with $f(x)$ the curve, δx the incremental step size of rectangles whose areas are summed up and n the number of points in the outlier scores. The curve is obtained by calculating the probability density function of the outlier scores (values between 0 and 1) is calculated using a kernel density estimation. The incremental step size δx is set to $\frac{1}{2n}$. Then the AUC is continuously calculated in steps from left to right of the data range starting from 0 and arriving at a number of AUCs, namely $AUC_0, \cdots, AUC_k$. To obtain the threshold, another variable, lim is introduced as follows:

$$lim = \bar{x} + |\bar{x} - \tilde{x}|, \tag{3.4}$$

where $\bar{x}$ is the mean outlier score and $\tilde{x}$ the median outlier score. The threshold is defined as:

$$thres = AUC_j, \text{ with } j = min\{k \in \{1, \cdots, n\} | AUC_k > lim \cdot AUC\}, \tag{3.5}$$

with lim as defined in Equation 3.4 and AUC as defined in Equation 3.3. In other words, the threshold is set to the first AUC that is greater than the total AUC of the pdf multiplied by the lim.

The clustering-based method used in this study creates clusters of the outlier scores using hierarchical clustering and classifies objects within clusters as "normal" and objects outside as "outliers" (Lara et al., 2020).

The MAD introduced in (Archana et al., 2015) is motivated by the fact that the median is more robust against outliers than the mean. The threshold in this case is calculated as follows:

$$Tmin = median(X) - a * MAD \tag{3.6}$$

$$Tmax = median(X) + a * MAD, \tag{3.7}$$

with $MAD = 1.4826 * median(|X - median(X|)$, a a user variable, set to 3 in our case and X the outlier scores.

The Modified Thompson Tau test is a modified univariate t-test that eliminates outliers that are more than a number of standard deviations away from the mean (Sonneveld, 1997). The Tau critical value is defined as

$$\tau = \frac{t \cdot (n-1)}{\sqrt{n}\sqrt{n-2+t^2}}, \tag{3.8}$$

with n the number of outlier scores and t the student t-value. The method works iteratively and recalculates the Tau critical value after each outlier removal until the dataset

no longer has data points that fall outside the criterion, which is set to 3 standard deviations in this case.

Finally using the Z-Score as thresholding technique (see (Bagdonavicius & Petkevicius, 2019) for further details) is based on the assumption that the outlier scores, x, are normally distributed with a mean μ and variance σ^2, i.e. $x \sim \mathcal{N}(\mu, \sigma^2)$. In this case the underlying Z-Score can be calculated as

$$Z = \frac{x - \mu}{\sigma}. \tag{3.9}$$

Data is then labelled as "normal" if the following criterion holds:

$$|ZScore| \leq a, \tag{3.10}$$

with a an input variable, set to $a = 3$ in our case.

The above mentioned methods are implemented using the PyThres library, a toolkit for thresholding outlier detection.

3.4. CASE STUDIES AND RESULTS

The GDF presented in Section 3.3 is applied to three satellite and spacecraft system datasets: The first two, presented in Section 3.4.2 and Section 3.4.3 are publicly available and commonly used datasets in literature and the third, presented in Section 3.4.4 is a real-life satellite system dataset provided by ESA. We try to understand whether the GDF provides a robust diagnostic assessment for all the datasets by comparing the results to baseline machine learning algorithms. A thorough assessment of the dynamics of the framework and the way the metrics influence choices is given by comparing the multi-objective optimisation framework to a single-objective approach. The single-objective optimization problem can be formulated as follows: The objective function is to maximize the F1 score (respectively the F1pa score) of the anomaly detection algorithm together with data pre-processing and thresholding techniques on the system data set. We argue (see Section 3.3) that including thresholding methodologies makes the GDF more adaptive and provides significantly better results, which is shown by comparing two versions of the GDF: One including thresholding methods and one without them. First, in Section 3.4.1, we give an overview of the settings used within the GDF and how it was applied to the three datasets.

3.4.1. APPLICATION OF THE GDF TO THE DATASETS

Several hyperparameters need to be set for the MOGA (see (Bieber & Verhagen, 2022) for more details). In Algorithm 2, it can be seen that cross-overs from two other individuals create new individuals. The cross-over rate is the probability with which two individuals are crossed and is set to 0.5. Furthermore, individuals can be mutated to evolve over time. The mutation rate is the probability of mutating an individual and is set to 0.1. The algorithm is run either until it converges to an optimal solution or a stopping criterion

is achieved, and we set the maximum number of generations to 20. The number of individuals in the population is set to 50.

Each of the below-presented datasets consists of multiple subsets corresponding to components. The subsets are split into train and test data, respectively. An anomaly detection model is trained on each of the train datasets and tested on each of the test datasets, and the final score is computed using the mean of all the scores on each sub-dataset. The results are compared to baseline models. The four baseline models are PCA, iF, KNN and OC-SVM trained on the dataset without applying any prior hyperparameter tuning. In other words, they are obtained using the four anomaly detection algorithms with the default settings as implemented in the Python PyOD package.

3.4.2. SMAP DATASET

The data from the NASA Soil Moisture Active Passive (SMAP) satellite is a publicly available expert-labelled telemetry anomaly data set (Hundman et al., 2018). It contains 54 multidimensional time-series sub-datasets. Each sub-dataset is split into a train and test set. An example of telemetry values can be seen in Figure 3.5.

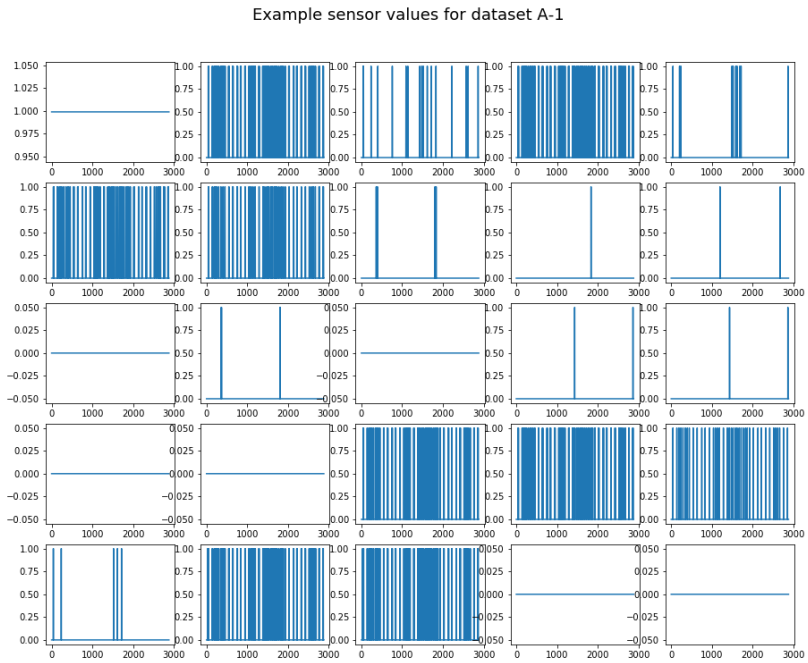


Figure 3.5: Example telemetry values of sub-dataset A-1 in the SMAP dataset.

As a first step, the initial diagnostic algorithms are determined by performing hyperparameter tuning as presented in Section 3.3.2. This results in initial anomaly detection models with settings presented in Table 3.3.

Table 3.3: Hyper parameter settings of initial anomaly detection methods for SMAP dataset.

Algo	Hyperparam	Chosen value
PCA	n_components	5
iF	n_estimators	100
	max_samples	100
	max_features	10
KNN	n_neighbors	13
	p	1
	method	'median'
	algorithm	'auto'
OC-SVM	nu	0.1
	Gamma	0.6
	kernel	'sigmoid'

RESULTING PARETO FRONT COMPARED AGAINST BASELINE

The output of the GDF is a Pareto front consisting of multiple individuals with different settings for the data pre-processing, anomaly detection and thresholding techniques. Table 3.4 contains the Pareto front for the SMAP dataset.

Table 3.4: Pareto front individuals and scores for SMAP dataset.

settings	F1	F1pa	FC
normalization KNN MAD	0.213	0.588	0.319
normalization KNN 0.04	0.249	0.582	0.34
normalization KNN ZSCORE	0.19	0.676	0.364
standardization KNN MAD	0.21	0.598	0.317
standardization KNN 0.04	0.249	0.582	0.34

Figure 3.6 shows the range of the three different scores (F1, F1pa and FC) for all individuals and the individuals in the pareto front. It can be seen in Table 3.4 that for this dataset,

the choice of the anomaly detection method is KNN as it beats the other anomaly detection methods in all cases. Furthermore, the individuals in the Pareto front are in terms of all metrics very close to each other. For example, the F1 scores range from 0.19 to 0.249 and the FC scores from 0.317 to 0.364. This can also be seen in Figure 3.6. One more notable thing is that it seems as if the threshold setting has the biggest influence on the scores. E.g. the F1pa score for using normalization together with KNN and MAD is 0.588 while the F1pa score for the same settings but using the ZSCORE is 0.676. We will go into more detail on this in Section 3.4.2.

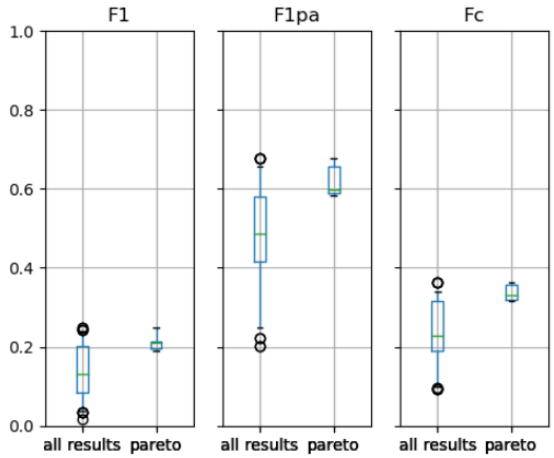


Figure 3.6: Scores of all individuals and Pareto front individuals for SMAP dataset.

Table 3.5: Baseline models and scores for the SMAP dataset.

Algorithm	**F1**	**F1_pa**	FC
ocsvm	0.183	0.565	0.276
KNN	0.239	0.427	0.301
if	0.095	0.457	0.175
PCA	0.0	0.0	0.0

Table 3.5 shows the results of the baseline models. For a better assessment, a comparison of the baseline models to the best-performing individuals of the Pareto front in terms of the respective scores can be found in Table 3.6. When looking closer at Table 3.5 and the results in terms of F1 score, it becomes clear why the KNN was chosen. The performance of other algorithms is much worse. While the OC-SVM outperforms the other algorithms in terms of F1pa score, Table 3.6 reveals that apparently, the thresholding improves the results in terms of F1pa score, resulting in the performance of all individuals of the Pareto

front outperforming all baseline models in terms of F1pa score. For the FC score, the results are similar to those of the F1pa score, but here the KNN baseline model already outperforms the OC-SVM model.

Table 3.6: Comparison of baseline models to respective best performing Pareto front individuals for SMAP dataset.

	Baseline	GDF
Settings	KNN	KNN 0.04
F1	0.239	0.249
Settings	OC-SVM	Normalization KNN ZSCORE
F1pa	0.565	0.676
Settings	OC-SVM	Normalization KNN ZSCORE
FC	0.276	0.364

COMPARING MULTI-OBJECTIVE OPTIMISATION WITH SINGLE-OBJECTIVE OPTIMIZATION

Performing single-objective optimisation and setting the metrics to both F1 score and F1pa, results in the following individuals chosen by the GDF:

- When optimising towards an F1 score, the best individual has the following settings: normalisation, KNN, 0.04 with an F1 score of 0.249.
- When optimising towards the F1pa score, the best individual has the following settings: normalisation, KNN, and ZSCORE with an F1pa score of 0.676.

In this case, Figure 3.6 already shows that the resulting scores within the Pareto front do not cover a wide range (e.g. the lowest FC score is 0.317, which is quite close to 0.364, the top score). Following this observation, we expect the results of single-objective optimisation to be very close to those of the MOGA, which they are. In most cases, increasing F1pa causes the F1 score to decrease. So, all in all, while in this case, single objective optimisation would form a formidable alternative to using the MOGA, optimising towards a single metric always means a compromise in terms of another metric. Therefore, the metric should be chosen with care.

THE EFFECT OF INCLUDING THRESHOLDING METHODS

Table 3.7 shows the results of using the GDF just using the default settings of the PyOD algorithms (which set the contamination rate to 0.1) for the label computation.
To make the effect of this clearer, Table 3.8 shows the best individual output by the GDF with default thresholding and when including the selected thresholding techniques.
While in terms of F1 score, the thresholding techniques have little effect on the quality of the results (see Table 3.8), including more elaborate thresholding methods improves the scores quite a bit in terms of F1pa and FC score.

Table 3.7: Pareto front when default thresholding techniques are included for SMAP dataset.

settings	**F1**	F1pa	FC
normalization PCA	0.136	0.49	0.221
normalization KNN	0.242	0.427	0.302
standardization KNN	0.242	0.427	0.302
standardization ocsvm	0.103	0.522	0.206

Table 3.8: Comparison of best individuals when using default thresholding vs using selected thresholding for SMAP dataset.

	no thresholding	GDF incl thresholding
Settings	standardization/ normalization KNN	KNN 0.04
F1	0.242	0.249
Settings	Standardization OC-SVM	Normalization KNN ZSCORE
F1pa	0.522	0.676
Settings	standardization/ normalization KNN	Normalization KNN ZSCORE
FC	0.302	0.364

3.4.3. MSL DATASET

Another publicly available spacecraft telemetry dataset that contains expert-labelled anomalous data is data from the Mars Science Laboratory (MSL) rover, Curiosity. Similarly, the SMAP dataset consists of 27 sub-datasets, each containing telemetry values of 25 sensors (Challu et al., 2022). The hyperparameter tuning to arrive at the initial diagnostic algorithms results in the settings listed in Table 3.9.

RESULTING PARETO FRONT COMPARED AGAINST BASELINE

The Pareto front for the MSL dataset is presented in Table 3.10.
Figure 3.7 shows the range of the three different scores (F1, F1pa and FC) for all individuals and the individuals in the Pareto front.
The performance of individuals in the Pareto front for the MSL data, as can be seen in Figure 3.7, covers a wider range than for those for the SMAP dataset. For example, the F1 ranges from 0.107 to 0.259 and the F1pa from 0.524 to 0.734. In addition, in this case, it is less clear which anomaly detection method is the best since three of the four anomaly detection techniques, iF, PCA and KNN are represented in the Pareto front. Using KNN

Table 3.9: Hyper parameter settings of initial anomaly detection methods for MSL dataset.

Algo	Hyperparam	Chosen value
PCA	n_components	1
iF	n_estimators	500
	max_samples	100
	max_features	5
KNN	n_neighbors	13
	p	1
	method	'largest'
	algorithm	'auto'
OC-SVM	nu	0.1
	Gamma	0
	kernel	'linear'

Table 3.10: Pareto front individuals and scores for MSL dataset.

settings	F1	F1pa	FC
normalization PCA AUCP	0,107	0,734	0,313
normalization if AUCP	0,184	0,626	0,306
normalization if MAD	0,184	0,626	0,306
normalization if 0,08	0,184	0,626	0,306
normalization KNN CLUST	0,233	0,620	0,36
normalization KNN ZSCORE	0,249	0,553	0,346
normalization KNN MAD	0,259	0,524	0,338

results in the highest scores in terms of F1 but the lowest in terms of F1pa. The IF models receive medium scores in terms of both F1 and F1pa but score lowest in terms of FC, and the PCA models score highest in terms of F1pa but lowest in F1.

Table 3.11 shows the results of the baseline models. A comparison of the baseline models to the best-performing individuals of the Pareto front in terms of the respective scores can be found in Table 3.12.

Again, we see that in terms of F1 score, there is no significant improvement, but the individuals in the Pareto front score much higher in terms of F1pa score and FC score. The F1pa score, as can be seen in Table 3.12, is improved from 0.559 (for the baseline model IF) to 0.734 (when using normalization, PCA and AUCP).

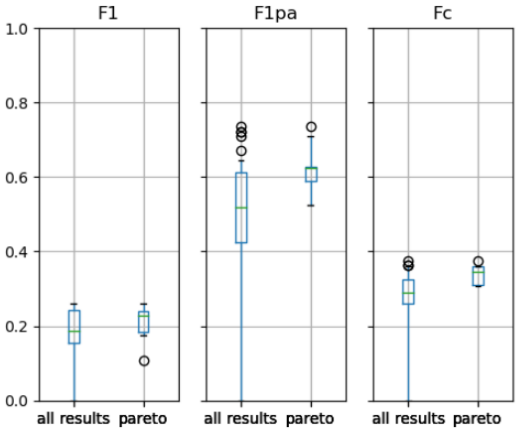


Figure 3.7: Scores of all individuals and pareto front individuals for MSL dataset.

Table 3.11: Baseline models and scores for the MSL dataset.

Algorithm	**F1**	**F1_pa**	FC
ocsvm	0.208	0.53	0.324
KNN	0.251	0.488	0.324
if	0.144	0.559	0.238
PCA	0.166	0.554	0.261

Table 3.12: Comparison of baseline models to respective best performing Pareto front individuals for MSL dataset.

	Baseline	GDF
Settings	KNN	Normalization KNN MAD
F1	0.251	0.259
Settings	IF	Normalization PCA AUCP
F1pa	0.559	0.734
Settings	OCSVM and KNN	Normalization KNN CLUST/MAD/ZSCORE
FC	0.324	0.36

COMPARING MULTI-OBJECTIVE OPTIMISATION WITH SINGLE-OBJECTIVE OPTIMISATION

Performing single-objective optimisation and setting the metrics to both F1 score and F1pa results in the following individuals chosen by the GDF:

- When optimising towards an F1 score, the best individual has the following settings: normalisation, KNN, and MAD with an F1 score of 0.259.
- When optimizing towards an F1pa score, the best individual has the following settings: normalisation, PCA, and AUCP with an F1pa score of 0.734.

In the case of the single objective optimisation for the MSL dataset, it can be observed that the GA outputs normalisation, KNN and MAD when optimising towards the F1 score, which results in the lowest scoring individual contained in the Pareto front (see Table 3.10) in terms of F1pa score. The same is true and vice versa: The best-performing individual in terms of F1pa score is the lowest-scoring individual in terms of F1 score. Therefore, it becomes visible here that optimising towards a single metric comes at the cost of a lowered score in terms of another metric.

THE EFFECT OF INCLUDING THRESHOLDING METHODS

Table 3.13 shows the results of using the GDF with the default settings of PyOD for the label computation.

Table 3.13: Pareto front when default thresholding techniques are included for MSL dataset.

settings	**F1**	F1pa	FC
normalization if	0.184074	0.596667	0.282963
normalization KNN	0.255185	0.503333	0.325185
standardization if	0.181481	0.587407	0.291852

Table 3.14 shows the best individuals output by the GDF with default thresholding and when including the selected thresholding techniques.
Similarly, as for the SMAP dataset, in Table 3.14 it can be seen that the biggest difference by including elaborate thresholding methods is achieved in terms of F1pa and FC score. Compared to the results of the baseline models (see Table 3.12), the scores improve slightly when including data pre-processing techniques.

3.4.4. SATELLITE REACTION WHEEL DATASET

The third dataset used in this study contains telemetry data from reaction wheels (RWL) operated on ESA Earth Observation satellites in a two-satellite constellation. Each of the two satellites carries four reaction wheels. A substantial amount of health related RWL data has so far been collected during this mission, which can be utilised for anomaly detection. In the operation time, however, only six anomalies occurred, which, together with anomaly reports, were used to create the test dataset for this study. Each RWL is equipped with 10 sensors recording health-related telemetry values. An example of such telemetry sensor readings can be seen in Figure 3.8.

Table 3.14: Comparison of best individuals when using default thresholding vs using selected thresholding for MSL dataset.

	GDF no thresholding	GDF incl thresholding
Settings	Normalization KNN	Normalization KNN MAD
F1	0.255	0.259
Settings	Normalization IF	Normalization PCA AUCP
F1pa	0.597	0.734
Settings	Normalization KNN	Normalization KNN CLUST/MAD/ZSCORE
FC	0.325	0.36

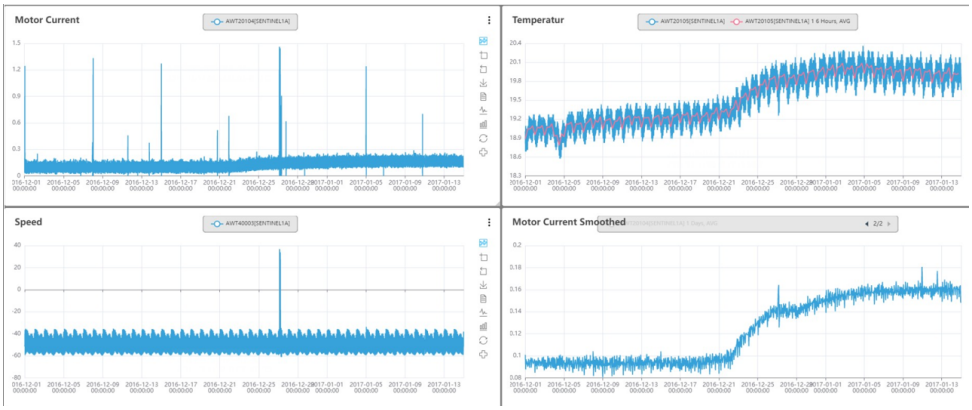


Figure 3.8: Example telemetry values for the ESA dataset.

The hyperparameter tuning to arrive at the initial diagnostic algorithms results in the settings listed in Table 3.15.

RESULTING PARETO FRONT COMPARED AGAINST BASELINE

Table 3.16 contains the output of the GDF applied to the ESA dataset, i.e. the individuals in the Pareto front. Figure 3.9 shows the range of the three different scores (F1, F1pa and FC) for all individuals and the individuals in the Pareto front.
Applying the GDF to the ESA dataset results in the largest Pareto front of the three datasets. It is not so surprising, therefore, that the range of performance of individuals in the Pareto front is quite high (see Figure 3.9), e.g. the F1 score ranges from very close to 0 to 0.623. It can also be seen that the highest performance in terms of F1pa score results in a very poor F1 score: For example, the individual KNN MAD has an F1 score of 0.0314

Table 3.15: Hyper parameter settings of initial anomaly detection methods for ESA dataset.

Algo	Hyperparam	Chosen value
PCA	n_components	1
iF	n_estimators	100
	max_samples	400
	max_features	10
KNN	n_neighbors	5
	p	1
	method	'mean'
	algorithm	'auto'
OC-SVM	nu	0.1
	Gamma	0.8
	kernel	'rbf'

Table 3.16: Pareto front individuals and scores for ESA dataset.

settings	F1	F1pa	FC
normalization PCA 0,02	0.459	0.971	0.841
normalization PCA 0,04	0.489	0.949	0.8
normalization PCA ZSCORE	0.113	0.983	0.903
normalization if 0,02	0.476	0.939	0.817
normalization KNN MAD	0.031	1.0	1.0
normalization KNN ZSCORE	0.079	0.983	0.921
normalization KNN 0,06	0.607	0.839	0.794
normalization KNN 0,08	0.616	0.827	0.78
normalization KNN 0,14	0.621	0.791	0.741
standardization PCA 0,04	0.489	0.949	0.8
standardization PCA ZSCORE	0.113	0.983	0.903
standardization KNN 0,06	0.607	0.837	0.79
standardization KNN 0,08	0.617	0.826	0.776
standardization KNN 0,12	0.619	0.804	0.754
standardization KNN 0,18	0.623	0.77	0.724
standardization KNN ZSCORE	0.059	1.0	0.994
standardization ocsvm MAD	0.531	0.933	0.897
standardization ocsvm CLUST	0.601	0.907	0.841

and the individual using KNN with the ZSCORE an F1 score of 0.059, while both of these

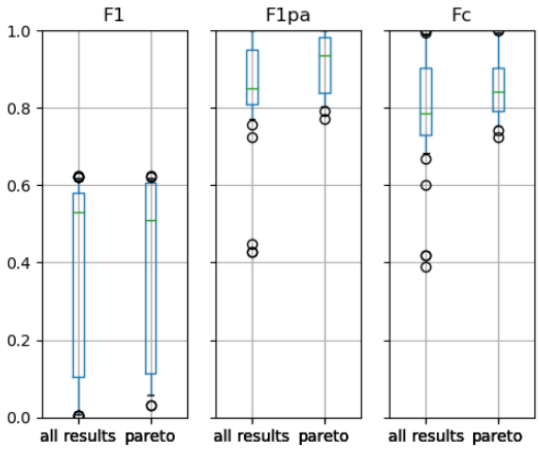


Figure 3.9: Scores of all individuals and Pareto front individuals for ESA dataset.

individuals have an F1pa score of 1.0. It can be said that, in general, increasing the F1pa score comes at the cost of lowering the F1 score (see Table 3.16). Similarly, increasing the FC score results in lower F1 scores. Furthermore, the thresholding techniques do not seem to have a particularly strong effect on the scores when using KNN for anomaly detection (see Table 3.16). Table 3.17 shows the results of the baseline models. The comparison of the baseline models to the best-performing individuals of the Pareto front in terms of the respective scores can be found in Table 3.18. When comparing the Pareto front individuals to the baseline models, we again see that in terms of F1, there is not much improvement in the results. Still, significant improvement is visible in terms of F1pa and FC scores.

Table 3.17: Baseline models and scores for the ESA dataset.

Algorithm	**F1**	**F1_pa**	FC
ocsvm	0.54	0.849	0.736
KNN	0.613	0.814	0.766
if	0.539	0.841	0.737
pca	0.336	0.597	0.499

COMPARING MULTI-OBJECTIVE OPTIMISATION WITH SINGLE-OBJECTIVE OPTIMISATION

Performing single-objective optimisation and setting the metrics to both F1 score and F1pa results in the following individuals chosen by the GDF:

- When optimising towards an F1 score, the best individual has the following settings: normalisation, KNN, 0.14 with an F1 score of 0.621.

Table 3.18: Comparison of baseline models to respective best performing Pareto front individuals for ESA dataset.

	Baseline	GDF
Settings	KNN	Standardization KNN 0.18
F1	0.613	0.623
Settings	OC-SVM	KNN MAD/ZSCORE
F1pa	0.849	1.0
Settings	KNN	KNN MAD
FC	0.766	1.0

- When optimising towards the F1pa score, the best individual has the following settings: normalisation, KNN MAD with an F1pa score of 1.0.

Here, the effect of including multiple metrics in the optimisation is visible because many individuals score high F1pa scores in the Pareto front. Therefore, considering the F1 score in addition to the F1pa gives a good insight into performance (see the previously pointed out very poor performing individuals in terms of F1 score). Again, the FC score is mostly in alignment with the F1pa score, i.e. increasing the F1pa score usually simultaneously increases the FC score.

THE EFFECT OF INCLUDING THRESHOLDING METHODS

Table 3.19 presents the results of using the GDF including default thresholding techniques. Again, to give a clearer insight into the results, Table 3.20 shows the best individuals returned by the GDF with default thresholding and with the additional thresholding techniques.

Table 3.19: Pareto front when default thresholding techniques are included for ESA dataset.

settings	**F1**	F1pa	FC
normalization PCA	0.536	0.879	0.744
normalization if	0.547	0.839	0.747
normalization KNN	0.617	0.816	0.769
normalization ocsvm	0.54	0.841	0.74
standardization PCA	0.536	0.879	0.744
standardization if	0.547	0.839	0.747

Table 3.20: Comparison of best individuals when using default thresholding vs using selected thresholding for ESA dataset.

	GDF no thresholding	GDF incl thresholding
Settings	Normalization KNN	Standardization KNN 0.18
F1	0.617	0.623
Settings	PCA	KNN MAD/ZSCORE
F1pa	0.879	1.0
Settings	Normalization KNN	KNN MAD
FC	0.768	1.0

Compared to Table 3.17, we see that using pre-processing data methods on the ESA dataset does not improve the results as much as for the MSL and SMAP datasets. Furthermore, we see that again, in terms of F1pa and FC scores, including thresholding techniques result in much better anomaly detection models. In contrast, in terms of F1 score, the effect is less significant.

3.4.5. DISCUSSION

In this section, we present the findings of the results regarding the three main objectives as highlighted in Section 3.1 and at the beginning of Section 3.4 based on the results. The results show that the framework is adaptive to different datasets and outperforms the baseline algorithms in all three case studies (see Tables 3.6, 3.12 and 3.18). Furthermore, the framework indicates as to which methods to focus further on and which methods perform well for a given dataset. For the SMAP dataset, the results presented in Section 3.4.2, a single anomaly detection method (KNN) can be singled out from the four input techniques. For the MSL dataset, presented in Section 3.4.3, this is not so clear, both iF and KNN could be considered and similarly for the ESA dataset (see Section 3.4.4), the Pareto front is much bigger, which makes it harder to choose the 'best' set of methods. This points out the importance of choosing suitable metrics for evaluating the models.

Including three different metrics in the framework makes it more robust, which is especially visible in the results on the ESA dataset (see Table 3.16). In this case the best results in terms of F1pa score receive the lowest score in terms of F1 score. In general, higher F1pa and FC scores result in lower F1 scores. Mostly the FC score is aligned with the F1pa score, but that is not always true. For example, for the MSL dataset in Table 3.10, we see that the highest scoring individual in terms of F1pa score (normalisation PCA and AUCP) reaches an F1pa score of 0.734 and an FC score of 0.313. In contrast, the highest-scoring individual in terms of FC score (KNN and ZSCORE) with an FC score of 0.346 has an F1pa score of only 0.553.

Finally, including thresholding techniques improves the results significantly. Throughout all three datasets, which becomes visible in Tables 3.8, 3.14 and 3.20, both the F1pa and FC score can be improved by a margin when using thresholding techniques. For example, in the ESA dataset (Table 3.20), the FC score is improved from 0.768 to 1.0 and the F1pa score from 0.876 to 1.0 by including thresholding techniques in the framework.

3.5. CONCLUSION

A Generic Diagnostic Framework has been presented with the capability to automatically chose optimal data pre-processing, anomaly detection and thresholding techniques simultaneously given system data. Overall, thresholding methods play an important role in anomaly detection and can significantly influence the quality of resulting models. In addition, the optimisation metrics affect the choice of methods, and the optimisation towards a single metric is always a trade-off. Therefore particular care should be taken when choosing suitable metrics to evaluate the anomaly detection models.

A next step in the development of the GDF could be to include more metrics in the model assessment or even perform a more thorough assessment towards applications. What could also be an interesting direction for further research is to look into systems operated in different operating conditions. Especially for satellite systems, for which failures or even anomalies are scarce, it would be an asset to be able to train models on systems in different satellite constellations, operated in similar conditions. Furthermore, the framework could be extended to include a wider range of techniques, e.g. by including more elaborate data pre-processing methods, deep learning anomaly detection methods or statistical algorithms.

All in all, the framework offers a quick way to assess system data of complex systems towards their suitability for anomaly detection approaches. Based on the outputs, further decisions can be taken, and development and expertise can be streamlined in fruitful directions.

4

A Generic Framework for Prognostics of Complex Systems

While in Chapter 3, a generic framework is developed for diagnostics, in this chapter the focus lies on prognostics. Similarly as in the previous chapter, the underlying requirements introduced in Chapter 2 are used to develop such a generic prognostic framework. The framework incorporates steps necessary for prognostics, including data pre-processing, feature extraction and machine learning algorithms for remaining useful life estimation. It is applied to two systems; a simulated turbofan engine dataset and an aircraft cooling unit dataset. The results show that the obtained accuracy of the remaining useful life estimates are comparable to what has been achieved in literature and provide insights into the adaptivity and generalizability of the framework, especially with respect to real aircraft data.

This chapter is based on on the publication: Bieber, Marie, and Wim JC Verhagen. "A Generic Framework for Prognostics of Complex Systems." Aerospace 9.12 (2022): 839. (Bieber & Verhagen, 2022).

4.1. INTRODUCTION

Over the last few years the field of prognostics has undergone substantial growth as evidenced by advances in algorithms, models and their applications (M. Scott et al., 2022). Prognostics is the process of estimating a system's Remaining useful life (RUL) (Elattar et al., 2016), usually following fault detection and/or diagnosis, and is usually considered part of a condition-based maintenance strategy. Prognostics enables operators to react to faults before failures occur, leading to a minimization of systems downtime, lowered operational cost and increased reliability (Lei et al., 2018; Zio, 2022).

Prognostic approaches can be classified into three types: Physics-based, data-driven and hybrid approaches (Peng, Dong, et al., 2010). Physics-based approaches can be applied in cases in which the underlying degradation phenomenon can be mathematically modelled. There are quite some examples of physics-based models that were successfully applied in practical cases, such as Li-Ion batteries (J. Zhang & Lee, 2011) and other structures subject to fatigue degradation (Brownjohn et al., 2011). Data-driven approaches are used when it is difficult to obtain a degradation model or when there is no knowledge about the system physics. And finally hybrid approaches combine available information about underlying physical knowledge and data. Examples for such approaches are (M. Baptista et al., 2019) combining Kalman Filtering with data-driven methods, (Downey et al., 2019) integrating a physical model and the least square method to estimate RUL of industrial equipment or (Lyathakula et al., 2022) using a physics-based fatigue damage degradation model and combining it with an neural network-based to model the damage progression in bonded joints. When considering complex systems which are subject to multiple degradation mechanisms, fault modes and operating conditions, accurate physics-based models are often not available (Zio, 2022). Therefore, data-driven prognostic techniques making use of monitored system condition data and failure data can be applied in such a case. They are mostly based on statistical or artificial intelligence (AI) methods. The requirement for such algorithms is the availability of data characterizing system behaviour that covers all phases of normal and faulty operation and all degradation scenarios under different operating conditions. Recent developments in sensing technologies, data storage, data processing, IT systems and computational power have been major drivers of data-driven prognostic approaches, leading to an increase in available methods and algorithms in the state of the art.

Most of the existing literature on data-driven prognostics focuses on the development of more advanced and more accurate models and algorithms. For this purpose, standard data sets are often used as these enable comparative evaluation of multiple models. This is a valid approach when the aim is the development of better-performing methods for those specific data sets. However, it also makes the approaches application- and system-specific. When applying those methodologies on 'real' systems, it can be the case that simple algorithms outperform very complex ones. Furthermore, tuning a complex algorithm to reach a better performance generally takes a lot of time and skill, which is often not available. Consider, for example, an airline operating different types of aircraft and aiming to introduce prognostics on a broad basis. Each aircraft can be considered as a complex system with multiple subsystems and components. For each of these sub-

systems or components a dedicated prognostic model is needed and the costs for the airline to hire data scientists that develop, test and validate a single model for each of the components would be immense. Therefore, what would be more desirable is a generic prognostic framework that chooses the most accurate prognostic approach from a set of algorithms given component data.

Prior studies proposing such frameworks have yielded promising results. An autonomous diagnostics and prognostics framework (DPF) is suggested by (Baruah et al., 2006). It consists of several steps, including data pre-processing, clustering to distinguish operating conditions and finally diagnostics and prognostics steps. A limitation of the approach is the fact that some parameters, including the number of observations for initialisation and optimization of cluster adaption rates have to be set manually and it can be tricky to tune the algorithm in an optimal way. Another limitation is the fact that a classification is performed (i.e. at any time it is determined if the component is faulty or not), rather then a remaining useful life estimation. To account for this, (Voisin et al., 2010) provide a generic prognostic framework that can be instantiated to various applications. However, their approach is very formal and no specific machine learning algorithms are used in this framework. Again, this is a limitation, as it is up to the user to define proper techniques. To overcome this problem, (An et al., 2015) provide guidelines to help with the selection of appropriate prognostic algorithms depending on the application. Another way to address this is by using ensembles of machine learning approaches that combine multiple prognostic algorithms with an accuracy-based weighted-sum formulation (Hu et al., 2010). Still, a problem remains: this addresses only prognostics but not the steps needed before, namely the data pre-processing and diagnostics. This is overcome by (Trinh & Kwon, 2020), who suggest a prognostics method based on an ensemble of genetic algorithms that includes all the steps, from the data pre-processing until the RUL estimation. With this it provides a truly generic framework for prognostics. The authors of the paper validated their framework by applying it to three commonly used and available data sets and comparing its performance to other existing approaches. However, their findings are limited to simulated data sets.

This development makes sense, especially when one considers the problems and challenges arising with using real-life data: As (Zio, 2022) points out, often collected sensor signals are collected under changing operational and environmental conditions. On top of that they are often incomplete, unlabeled, data are missing or scarce. Therefore, extracting informative content for the diagnostics and prognostics can be a challenging task. Still, this points towards a problematic trend: Many prognostic method developments in recent literature are not tested on real-life industrial cases. While many methods show highly promising results (M. Scott et al., 2022), they may face significant limitations when applied towards real-life cases. However, it is not often that these limitations are identified and addressed in literature. Nevertheless, several studies using real aircraft data have been published. Fault messages of an aircraft system have been used in (M. Baptista et al., 2017) to compare data-driven approaches for aircraft maintenance to the more classically used experience-based maintenance. An anomaly detection method for condition monitoring for an aircraft cooling system unit is presented in (Basora et

al., 2021b). On the same dataset, two more studies have been conducted on remaining useful life estimation: First, a clustering approach was used to determine degradation models and failure thresholds and together with a particle filter algorithm this results in RUL estimates (Mitici & De Pater, 2021). Second, a HI construction approach integrating physics based and data-driven methods was applied to the same data set to estimate the systems RUL (Rosero et al., 2022).

Still, applications for generic prognostic frameworks are limited to simulated data sets. We therefore present a generic framework and apply it to both a simulated dataset as well as a 'real' data set of operating aircraft within an airline. The aim is to provide for both a guidance in the choice of prognostic methodologies for a given dataset and a systems data suitability analysis from a prognostics perspective. We thereby also address the challenge of applying prognostic methodologies in real practice of complex systems and provide an assessment of whether or not a system is prognosable given the system data. A genetic algorithm is used to find the optimal combination of methodologies and associated hyperparameter settings for each step in the process of generating prognostics. With respect to the current academic state of the art, our novel contributions include:

- The presentation of a generic prognostic framework with the capability to not only estimate a system's RUL, but also give an assessment towards the ability to perform prognostics on such a system. A system is defined to be 'prognosable' if meaningful and accurate data-driven prognostic models can be developed based on available operational, contextual and failure data. Meaningful refers to the fact that the models are able to capture degradation trends and learn failure behaviour, while the term accurate pertains to the prediction quality in terms of one or multiple defined prognostic metrics.
- The implementation of the framework on both real aircraft data as well as a simulated data set.
- An identification of the challenges faced with using prognostic approaches on a real aircraft data set opposed to using simulated data.

The remainder of this paper is organized as follows. Section 4.2 introduces the generic prognostic framework. In Section 4.3, the aircraft systems, underlying data and failure modes are described and the results of the case study are presented. Subsequently, the adaptivity of the framework, the difficulties with applying it to a real dataset and the question of how to determine the ability to perform prognostics on a system are discussed. Finally, in Section 4.4 we conclude by highlighting the most important findings and limitations and providing directions for further research.

4.2. THE GENERIC PROGNOSTIC FRAMEWORK

In essence, the Generic prognostic framework (GPF) as shown in Figure 4.1 - originally introduced by (Trinh & Kwon, 2020) and extended here - takes as an input system data and outputs a trained prognostic model with the capability of predicting system remaining useful life at any time of operation. To be more precise, we define the GPF to be a tool

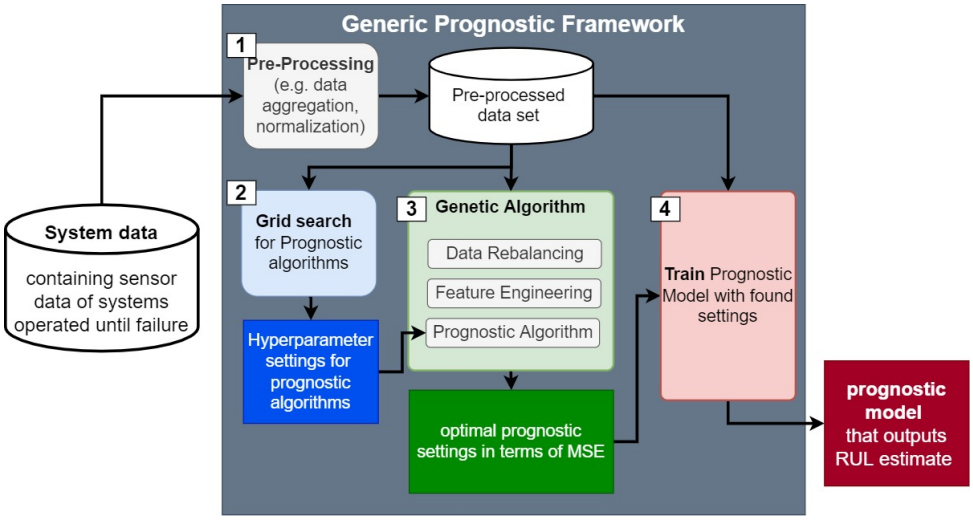


Figure 4.1: The Elements of the Generic Prognostic Framework.

that contains modelling techniques covering multiple aspects of a data-driven prognostics approach and, given a system data set, selects the best techniques for each case. This means that in addition to incorporating different methodologies, the framework includes a selection step in which the best set of techniques is chosen relative to prognostic performance.

There are multiple steps that have to be implemented in a prognostics framework (such as data pre-processing methods or feature engineering techniques) before the actual prognostics algorithm that performs the remaining useful life prediction on the data set is executed. Therefore, a generic prognostic framework does not only need to provide the flexibility of choosing the 'best' prognostic algorithm, but also it has to incorporate the previous steps. Note that we distinguish prognostic algorithms from prognostic models: when using the term 'prognostic algorithm' we refer to a certain selected technique used to perform prognostics, e.g., Random forest (RF) or neural networks, and by 'prognostic model' we indicate the derived predictor (as output of the prognostic algorithm and feature engineering methodologies) that takes system data as an input and outputs the RUL estimate.

The GPF treats the selection of the according techniques as an optimization problem: The objective is to select the optimal methodology (in terms of Mean squared error (MSE), defined in Equation 5.1) with the optimal hyper parameter settings for each element of prognostics included in the framework (such as data rebalancing). We implement this in four steps shown in Figure 4.1. In step 1 the selected system data is pre-processed. As the GPF is a generic framework that is adaptive by nature to different data sets, the data pre-processing techniques applied are kept to a minimum. Further details about the pre-processing applied are given in Section 4.2.1. In step 2 the hyper param-

eters for prognostic algorithms are tuned by grid search as further explained in Section 4.2.2. Step 3 aims to solve the optimization problem that can be formulated as follows: find the optimal combination to generate predictions for a given dataset, where optimality is evaluated through minimisation of the MSE, given a set of re-balancing, feature engineering techniques and prognostic algorithms. A detailed explanation of this process and the according techniques is given in Section 4.2.3. Finally, in step 4, the settings are used to build the prognostic model to output the RUL estimate. The framework as suggested in this paper can be used in multiple ways, two of which are of primary importance in the context of our research: Either it can provide a quick assessment of the ability to perform prognostics based on the input data or it can be used to perform an automatic selection of feature engineering settings. This is further explained in Section 4.2.4. To guide through the following sections and make the dynamics of the GPF clearer, we make use of a small example dataset. It is split in a training and test set as it would for a machine learning application as considered in this paper. The example training set is presented in Table 4.1 and the respective test set can be found in Table 4.2.

Table 4.1: Sample train data set.

Current mean	current min	current max	speed mean	speed min	speed max	high current count	RUL	id
0.00	0.0	0.0	0.00	0	0	751	0	11
1.19	0.0	2.1	4035	0	5024	967	1	11
2.15	2.1	2.2	4998	4976	5024	42	2	11
2.11	2.1	2.2	4997	4976	5016	83	3	11
2.18	1.8	2.4	4822	4472	5024	2223	4	11
2.15	1.8	2.4	4516	4448	5024	39267	5	11
1.84	1.6	2.2	4547	4456	4840	1693	6	11
2.13	2.1	2.2	4996	4976	5008	12	7	11
1.49	0.0	2.4	4564	0	5032	1910	0	3
2.43	2.4	2.5	4639	4576	4720	39	1	3
2.43	2.4	2.5	4557	4536	4584	9	2	3
2.40	2.4	2.5	4497	4472	4552	104	3	3
2.24	2.1	2.4	4493	4464	4528	846	4	3
2.13	1.9	2.2	4493	4456	4528	1017	5	3

4.2.1. STEP 1: DATA PRE-PROCESSING

We make the following assumptions for the system data:

- The system is operated until failure.
- System data is related to operational properties of the system, captured e.g. through sensors and is available from the begin of operations until failure.
- The remaining useful life (RUL) of the system is known at any time of operations, i.e. in machine learning terms, a labelled data set is available.

Table 4.2: Sample test data set.

Current mean	current min	current max	speed mean	speed min	speed max	high current count	RUL	id
1.08	0.0	2.2	3225	0	5024	567	0	25
2.11	2.1	2.2	4996	4968	5032	41	1	25
2.12	2.1	2.2	4998	4984	5008	10	2	25

- In addition, the data must represent all phases of operation, i.e. normal as well as faulty behaviour and degradation under different operating conditions.

This results in data sets similar to those presented in Table 4.1 and 4.2 consisting of several trajectories, identified by ids (in the example, ids 11, 3 and 25) each representing a single system. The systems are operated until failure, i.e., until their RUL has reached 0. In each time step, several operational conditions are given, such as current and speed in the example data set. To evaluate and validate the prognostic models, the data is split into training and test data. The splits are such that trajectories are kept in the same data sets and 10% of the trajectories (ids) are used for testing. This is demonstrated in the example data sets, in which ids 11 and 3 are used for training and id 25 is used for testing. Further data pre-processing steps depend on the underlying data sets. For those used in our case studies, we explain the steps in Section 4.3.

4.2.2. STEP 2: GRID SEARCH TO TUNE PROGNOSTIC ALGORITHMS

Once the system data has been selected for the prognostic framework, the first step in the proposed GPF is to select the prognostic algorithms. Note, that the strength and the focus of the framework lies in providing a quick prognostic assessment rather than providing the 'best' possible prognostic assessment. Therefore, it suffices to use simple and easily implementable machine learning techniques, acknowledging that such algorithms may often provide first insights in the nature of the predictions.

In this paper, for this purpose we choose two different machine learning methodologies, a RF regression and a Support vector machine (SVM). Random Forests were introduced by (Breiman, 1996), (Breiman, 2001) and are based on the concept of bagging, where ensemble trees are grown by a random selection (without replacement) from the examples in the training set. Support vector machines, introduced by (Vapnik, 1995), make use of basis functions that are centred on the training data points and then selecting a subset of these during training. The two selected algorithms are well-established and offer potential advantages in terms of interpretability and explainability, which is necessary to understand systems retrospectively and prospectively (Ward & Habli, 2020). This may assist in the adoption of these algorithms for a variety of applications, potentially even covering safety-critical components. They thereby also provide the possibility to establish first baseline models for a quick prognostic assessment. Those two methodologies are chosen as representative machine learning algorithms. Both RF and SVMs

have shown to be adaptive to different datasets even without applying a thorough hyper parameter selection and are therefore good candidates to establish a first baseline. However, the framework can easily be extended to include further methodologies or algorithms.

For the chosen algorithms on a validation set, a grid search is performed to find the optimal hyper parameter settings. Since the aim of the grid search in this case is to establish quick baseline models that can consequently be used as in input in the following step of the framework, we only search a limited set of parameters. The according hyper parameters and their possible settings explored during the grid search are given in Table 4.3. The found settings are the ones then used as initial settings for the prognostic algorithms in the genetic algorithm that is presented in the next section.

Table 4.3: The hyper parameters and combination of settings explored during the grid search for each of the prognostic algorithms.

Prognostic algorithm	**Hyper parameter**	**Description**	**Possible settings**
rf	n estimators	number of trees	{200, 800, 1400}
	max features	maximum number of features to consider when looking for the best split	{'auto', 'sqrt', 'log2'}
	min samples leaf	minimum number of samples required to be at a leaf node	{1, 2, 4}
SVM	C	learning rate	{0.001, 0.01, 0.1, 10}
	gamma	kernel coefficient	{0.001, 0.01, 0.1, 1}

4.2.3. STEP 3: GENETIC ALGORITHM

As highlighted before, we treat the problem of finding the prognostic settings as an optimization problem: The objective function is to minimize the MSE (Equation 5.1) of the prognostic algorithm together with data re-balancing and feature engineering techniques on the pre-processed data set. The MSE at time t is defined as

$$MSE(t) = \frac{1}{t}\sum_{i=1}^{t}(RUL_i - \hat{RUL}_i)^2, \tag{4.1}$$

with RUL_i the true RUL value and $\hat{RUL}_i$ the predicted RUL value at timestep i.

The reason we chose the MSE for the evaluation of the prognostics is twofold: First, as a score which captures accuracy, the MSE gives a good indication over how well the al-

gorithms perform with respect to predicting the RUL. Second, despite the fact that it is important to not rely on one metric to evaluate predictions (Lewis & Groth, 2022a), we found that the majority of the literature considering the simulated turbofan engine dataset uses the MSE or Root mean squared error (RMSE) to evaluate RUL predictions. For this reason, it makes sense for us to apply it in this case study as well to have results that are comparable with the state of the art and thereby can be validated against existing approaches.

The concepts of natural selection and genetics inspired the field of evolutionary strategies and genetic algorithms. Genetic algorithms are based on the concepts of natural selection and genetics (Holland, 1992). Due to their flexibility, GAs are able to solve global optimization problems and optimize several criteria at the same time, like in our case the simultaneous selection of data re-balancing, feature engineering and prognostic algorithm techniques (Stanovov et al., 2017). This is what makes them good candidates for our optimization problem.

A GA consists of several steps as presented in Algorithm 2 and Figure 4.2. The process is as follows:

- A population is initialized, composed by a set of individuals (i.e., solutions to the optimization problem).
- The best fitted individuals are selected based on a fitness metric which represents the objective.
- In a following step, the selected individuals undergo a cross-over and mutation process to produce new children for a new generation of individuals.
- This process is repeated over a number of generations until the algorithm converges or a stopping criterion is achieved.

A population consists of individuals, which in turn consists of a set of chromosomes. Each individual represents a solution to the optimization problem and is associated with a fitness. In our case, an individual consist of three chromosomes corresponding to choices of methodologies for data re-balancing, feature engineering and prognostic algorithms as it is shown in Figure 4.3. The details of the setup for each of the respective steps are given in the following subsections. For the example included in this section, the solution space of the optimization problem corresponds to 32 possible solutions. The fitness of each individual is given by the MSE at time t (Equation 5.1) resulting from the prognostics performed with the individual settings on the underlying data set.

In the following subsections we give an overview of the multiple techniques considered by the GA for the data re-balancing, feature engineering, and prognostic algorithm. To guide through the process, we make use of the example introduced in Section 4.2.1.

DATA RE-BALANCING

Data pre-processing or data manipulation is usually done as a step previous to applying data-driven approaches for two reasons: first, to reduce the number of features in order

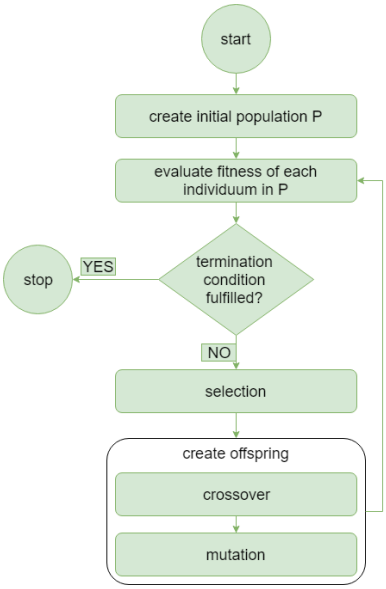


Figure 4.2: Genetic Algorithm process.

Algorithm 2: Genetic Algorithm

start;
$t \leftarrow 0$;
initialize population $P(t)$;
evaluate fitness of each individual in $P(t)$;
while *termination condition not fulfilled* **do**
  $t \leftarrow t+1$;
  $s_1, s_2 \leftarrow$ select individuals from $P(t)$;
  $x_1, x_2 \leftarrow$ create offspring by crossover operation on s_1, s_2;
  $\hat{x_1}, \hat{x_2} \leftarrow$ mutate x_1, x_2;
  evaluate fitness of $\hat{x_1}, \hat{x_2}$ **if** *fitness of* $\hat{x_1}, \hat{x_2}$ *higher than least fittest individuals in* $P(t)$ **then**
    replace least fittest individuals with $\hat{x_1}, \hat{x_2}$;
  else
    pass;
  end
end

to achieve a more efficient analysis and second, to adapt the dataset to suit the selected method (Jović et al., 2015). Steps typically involved are data cleaning, normalization and feature engineering (Elattar et al., 2016). In cases of imbalanced datasets, oversampling can be introduced in addition (Branco et al., 2019). A comprehensive overview of feature

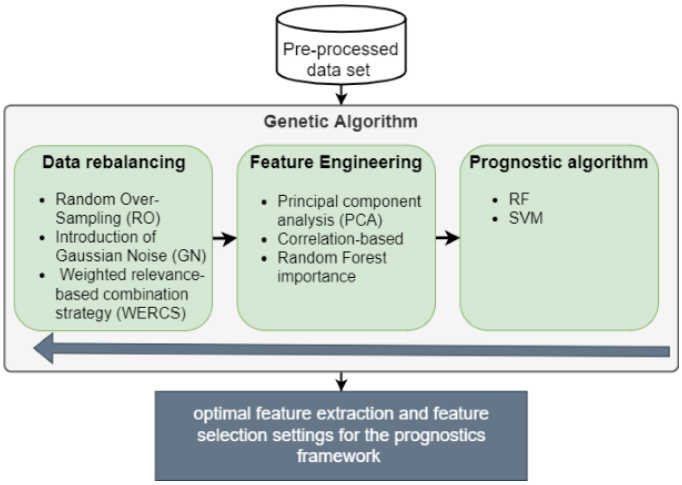


Figure 4.3: The prognostic steps and methodologies included in the Genetic Algorithm.

engineering steps and according methodologies is given by (Jović et al., 2015). In the generic prognostic framework two steps of data pre-processing are addressed, namely data re-balancing and feature engineering, including feature extraction and selection methods.

The data re-balancing step is done first, to address the problem of imbalanced distributions in prognostic datasets. In this framework, three methodologies to address this issue, introduced by (Branco et al., 2019) are included, namely

- Random Over-Sampling (RO),
- Introduction of Gaussian Noise (GN),
- Weighted relevance-based combination strategy (WERCS).

The presented methodologies are suiting for regression problems, such as RUL estimation. While we do not go into details about them and refer interested readers to (Branco et al., 2019), we introduce the underlying basic concepts in the following paragraph. The main idea behind re-balancing methods for continuous target variables is the construction of bins based on a relevance function. The relevance function maps the values of the target variable into a range of importance, where 1 corresponds to maximal importance and 0 to minimum relevance. With this, the bins classify the data in normal (BIN_N) and relevant samples (BIN_R). In our setup, we use a sigmoid relevance function as defined in (Gado et al., 2020) and shown in Figure 4.4 with a relevance threshold, t_r of 0.5. Furthermore, we set all values with a RUL of less then the threshold $cl = 10$ to be of importance, set the oversampling rate to 0.9 and the undersampling rate to 0.1.

- **Random oversampling:** Random oversampling is often used to deal with imbalanced classification tasks. Samples from the rare class are randomly selected and

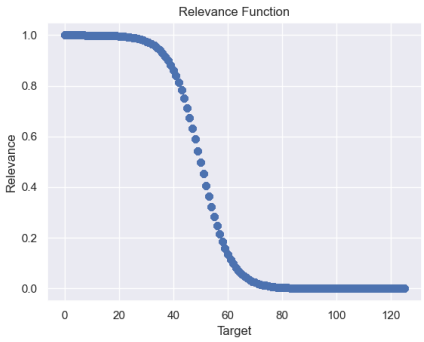


Figure 4.4: Example of a sigmoid relevance function similar to the one used for the rebalancing task.

replicated in a new updated data set. In (Branco et al., 2019) this strategy is adapted to regression tasks in the following way: The bins are constructed as above and while the samples in BIN_N remain unchanged a number of replicas of samples is added in BIN_R. The number of replicas is determined by the variable $over$, specifying the added percentage. While no information is discarded this way, the likelihood of overfitting increases.

- **Gaussian Noise:** Here, the re-balancing is done in two ways, under-sampling the normal cases and generating new cases based on the relevant target variable.
- **WEighted Relevance-based Combination Strategy (WERCS):** The idea behind this method is to combine over- and under-sampling strategies dependent only on the relevance function to avoid the definition of bins of relevance or the need of setting a relevance threshold, but it only uses the information of the relevance function.

Figure 4.5 shows the resulting dataset sizes on the demonstration dataset presented in Table 4.1 with the relevance threshold $cl = 1$.

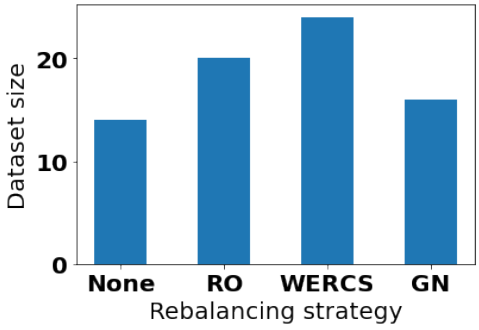


Figure 4.5: The dataset sizes for the different rebalancing strategies when applied to the demonstration example.

FEATURE ENGINEERING

In case of feature engineering the field of available literature and proposed methodologies is much wider and more diverse. Often, the terms feature selection and feature extraction are used in this context meaning various things. To be clear on this, we use the definitions used by (Jović et al., 2015). In feature extraction, either the entire set of features or a subset of features are transformed by mapping the original feature space to a new feature space with lower dimensions. Examples of feature extraction methodologies are Principle component analysis (PCA), kernel PCA, or techniques based on hierarchical clusterings, such as feature agglomeration (FAG). The scope of this analysis are RUL estimation models for mechanical or electrical systems with run-to-failure data, it is assumed that underlying signals come in the form of time-series data. A widely used feature extraction technique for time-series data is PCA which projects the data into a lower dimensional space through its singular value decomposition. Due to the fact that it has been so widely and successfully applied to prognostic approaches for time-series data, PCA is included in the GPF.

On the other hand, in feature selection a subset of features is chosen from the original feature set without transformation. Feature selection methods can be classified into four types (Hoque et al., 2014),

- filter-based approaches, selecting a subset of features without using a learning algorithm,
- wrapper approaches, evaluating the accuracy produced by use of the selected features in regression or classification,
- embedded approaches, performing feature selection during the process of training and sepcific to applied learning algorithms, and
- hybrid approaches, combining filter and wrapper methods.

In the GPF, we include a filter and an embedded approach. The filter approach is a correlation based approach, which chooses the best features based on univariate statistical tests. The embedded approach is based on the random forest importance, i.e., it chooses the features identified as most important by a random forest estimator.

PROGNOSTIC ALGORITHMS

Finally, the according prognostic algorithm needs to be chosen and applied to the data transformed by the previous steps. The underlying set of algorithms with according hyper parameters consists of a RF regression and a SVM for which the hyper parameters were found during the grid search step as presented in Section 4.2.2.

GENETIC ALGORITHM PARAMETERS

The previous paragraphs gave an overview over the form of an individual of the GA. Of course, also for the GA hyper parameters need to be set. The termination condition is chosen as the maximal number of generations. The probability with which an individual is mutated is set to 0.1, the probability for cross-over to 0.5 and the population size

to 20 as presented in (Trinh & Kwon, 2020). With this, we are ready to run the GA and apply it to system data to find the 'optimal' settings of feature engineering methodologies and according hyper parameters. Now the next step is to use those settings to build the prognostic model.

4.2.4. STEP 4: TRAINING THE PROGNOSTIC MODEL

The output of the GA is the 'best individual', i.e. the set of methodologies and hyper parameter settings that lead to the best performance on the data set in terms of MSE. This individual is now used to build a prognostic model. As an input this model takes a new data set of according system data and it outputs the RUL estimation.

All the models are implemented in Python. For the implementation we use the skikit-learn package in Python (Pedregosa et al., 2011). For the re-balancing techniques, the resreg python package is used (Gado et al., 2020).

4.3. CASE STUDY AND RESULTS

In Section 4.1, we pointed out that our aim is to provide a generic prognostic framework with the capability of providing RUL estimation models and determine the ability to perform prognostics on a system based on given operational and failure data. To understand if the framework is adaptive to different systems and to get insights into how the results can be used towards determining if a system is prognosable, the following steps are taken:

- The GPF is implemented in two different case studies involving a simulated and a real aircraft system, respectively.
- The results of the GPF are compared to two baseline machine learning algorithms, RF and SVM.
- The observed values are used in a comparative evaluation of the GPF and its capability to assess if a system is prognosable is analyzed.

In Section 4.3.1, the framework is implemented and validated on a simulated turbofan engine dataset. Section 4.3.2 presents the results of applying the framework to an aircraft cooling unit. Finally, in Section 4.3.3 the results of the case studies are discussed and the generalizability and adaptivity of the GPF are assessed.

4.3.1. SIMULATED TURBOFAN CASE STUDY

The first case study is conducted on a simulated turbofan engine dataset widely used for prognostic approaches in literature. We introduce the dataset in more detail in Section 4.3.1. Subsequently, we explain how we applied the GPF on the dataset in Section 4.3.1, after which we go into details of how the verification and validation was conducted using this dataset in Section 4.3.1 and finally we present the results in Section 4.3.1.

SIMULATED TURBOFAN ENGINE DATASET

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data consists of four data sets, each containing simulated run-to-failure data for turbofan engines

(Frederick et al., 2007) (Saxena, Goebel, et al., 2008a). The data sets differ mainly in the number of fault modes ('modes') and operating conditions ('conditions') as listed in Table 5.1. Each engine is considered to be from a fleet of engines of the same type and each time series, also often referred to as trajectory, is from a single unit. The engines are operated until failure, i.e. the time series capture the operations of each unit until it fails. In the test set, the time series ends at some point before the failure and the objective is to estimate the RUL, or in other words the number of remaining operational cycles before failure. There are 21 sensor measurements and each row in the data set contains the measurements corresponding to operations during one time cycle for a certain unit.

Table 4.4: Characteristics of the four turbofan engine data sets, note that the difference between the four data sets lies within the number of fault modes ('modes') and operating conditions ('conditions')

Data set	#modes	#conditions	#Train units	#Test units
#1	1	1	100	100
#2	1	6	260	259
#3	2	1	100	100
#4	2	6	249	248

APPLICATION OF THE GPF TO THE DATASET

In order to train the prognostic models we require a labelled data set, i.e., we assume that the RUL is known at any time. In the C-MAPSS data set the units are operated until failure, which means that the RUL can simply be calculated as the time to failure. In this case study, we set the maximum number of generations of the GA to 10 and vary the number of individuals in a population between 20, 30 and 50.

VERIFICATION AND VALIDATION OF THE GPF

Due to fact that it has been so extensively studied and there is a lot of material, especially on the C-MAPSS dataset FD001 in literature, we use it to conduct a validation of the GPF. Furthermore, we take this opportunity to mention that every element and step of the GPF was verified using unit tests and testing of the entire blocks of the GPF. The validation is done for each of the methodologies included in the GPF, to be more precise data rebalancing methods, feature engineering techniques and prognostic algorithms. What we present in the following is an extract of the validation of the feature engineering and prognostic algorithms.

In Section 4.3.1 we already mentioned that there are 21 features, corresponding to sensor readings. Seven of those are constant all the component life, leaving us with 14 features of interest, namely sensors $2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, 21$. It has been found that of those 14 the sensors $7, 8, 9, 12, 16, 17$ and 20 are the most valuable ones for RUL estimations (Wang et al., 2008), which is mostly in alignment with what (Jia et al., 2019)

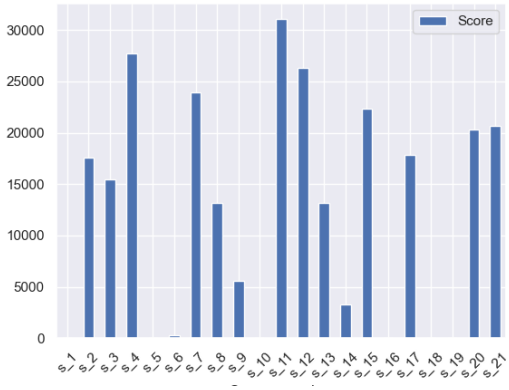


Figure 4.6: The features selected by the PCA and their relevance scores (the higher the more relevant).

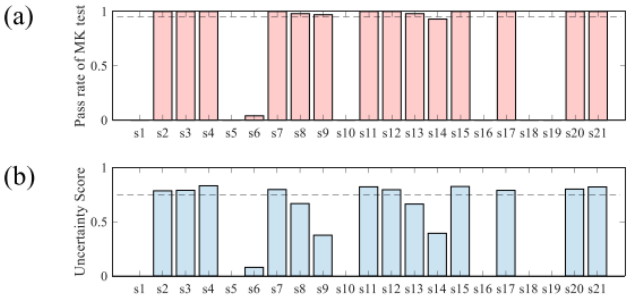


Figure 4.7: The most relevant features selected based on two different relevance scores by (Jia et al., 2019).

Table 4.5: The selected most relevant features of the C-MAPSS FD001 dataset by the methodologies included in the GPF and in existing literature.

Generic prognostic framework			**Literature**	
PCA	**Correlation-based**	**Importance-based**	**Paper #1 (Wang et al., 2008)**	**Paper #2 (Jia et al., 2019)**
s2, s3, s4, s7, s11, s12, s15, s17, s20, s21	s4, s7, s11, s12, s15, s21	s4, s9, s11, s12	s7, s8, s9, s12, s16, s17, s20	s2, s3, s4, s7, s11, s12, s15, s17, s20, s21

found. They pointed out that sensors 2, 3, 4, 7, 11, 12, 15, 17, 20 and 21 are the most relevant for RUL predictions as shown in Figure 4.7. In the GPF we include three basic feature engineering methodologies as explained in Section 4.2.3: PCA, correlation-based and importance-based feature engineering. Table 4.5 gives an overview over the result-

ing selected features and shows that all the three methodologies included in the GPF are aligned and also select the same features as in the two selected papers.

Table 4.6: Reference papers to validate the output of the prognostic algorithms in the GPF.

Paper ID	Reference
1	(C. Zhang et al., 2017)
2	(Babu et al., 2016)
3	(Jia et al., 2019)

Table 4.7: Comparision of the GPF performance to three selected papers in literature.

Dataset	Metric	Paper #1	Paper #3	RF in the GPF
FD001	RMSE	20.23	17.91	18.16
	Score	802.23	479	578.20
FD002	RMSE	30.01	29.59	29.15
	Score	84068	70465	65114
FD003	RMSE	22.34	20.27	20.76
	Score	1000.51	711.13	743.03
FD004	RMSE	29.62	31.12	30.00
	Score	22250	46567	26247.53

In order to validate the outputs of the prognostic algorithms and the GPF itself, we select three papers from literature presented in Table 4.6 to compare the metrics reached when using the SVM and RF of the GPF to the results reached in the respective papers on all four C-MAPSS datasets. Note that all of those papers use a piecewise linear RUL function (well explained in (Heimes, 2008)), which has been shown to result in much better predictions and in order to make the results comparable we do so too. Therefore the results presented in the following are not comparable with the results reached using the linear RUL function as presented in Section 4.3.1. Furthermore, two metrics are used to compare the results, the root mean-squared error (RMSE) which is simply the square root of the MSE and the score function as defined in (Saxena, Goebel, et al., 2008c). The resulting metrics of the three selected papers (in case of using the RF only two selected papers) and of the GPF are summarized in Tables 4.7 and 4.8. On all four datasets the results reached by the RF and SVM of the GPF in terms of RMSE and the score function are in the same range as the algorithms presented in the three papers in literature.

Table 4.8: Support vector machine RMSE and score on the three papers of literature and using the GPF.

Dataset	Metric	Paper #1	Paper #2	Paper #3	SVM in the GPF
FD001	RMSE	20.58	20.96	40.72	24.25
	Score	852.07	1381.5	7703	2312.64
FD002	RMSE	36.27	42	52.99	30.15
	Score	521461	589900	316483	19827.94
FD003	RMSE	23.3	21.05	46.32	23.69
	Score	1108.68	1598.3	22541	2472.71
FD004	RMSE	40.77	45.35	59.96	32.24
	Score	46611	371140	141122	10248.59

Table 4.9: A comparison of applying different rebalancing methodologies and the resulting MSEs on dataset FD001.

Settings			MSE
rebalancing	**feature engineering**	**prognostic algorithm**	
RO	None	rf	1657,90
None	None	rf	1650,41
GN	None	rf	1656,45
WERCS	None	rf	1658,01

COMPARATIVE STUDY ON DATASET FD001

In this section we present a short comparative study to show the effect of the different rebalancing, feature engineering and prognostic algorithm settings on dataset FD001. The aim is to understand what impact the different settings have on the resulting prognostic model in terms of MSE. In Tables 4.9, 4.11 and 4.13 the resulting MSEs for the different rebalancing, feature engineering and prognostics algorithm settings are presented. A visual representation of the scores is given in Figure 4.8 a) to c).

It can be seen that the rebalancing methodologies do not really affect the prognostic models in terms of MSE. As Table 4.9 and Figure 4.8 a) show the MSE varies only between 1650,41 when using no rebalancing and 1658,01 when using WERCS as rebalancing methodology. Different feature engineering settings together with no rebalancing have a higher impact on the MSE as Table 4.11 and Figure 4.8 b) show. The worst performing method is using PCA together with a RF, while correlation and importance based methods perform similarly with an MSE of 1769,25 and 1775,82 respectively. The prognostic algorithms presented in Table 4.13 and Figure 4.8 c) impact the resulting scores as well: While the RF based model achieves an MSE of 1650,99, the SVM based model only reaches an MSE of 1775,05. All in all, the results are surprising on first sight, because one would expect applying rebalancing or feature engineering techniques to

Table 4.10: A comparison of applying different feature engineering methodologies and the resulting MSEs on dataset FD001.

Settings			MSE
rebalancing	**feature engineering**	**prognostic algorithm**	
None	correlation	rf	1769,25
None	importance	rf	1775,82
None	None	rf	1650,88
None	PCA	rf	2105,58

Table 4.11: A comparison of applying the different prognostic algorithms and the resulting MSEs on dataset FD001.

Settings			MSE
rebalancing	**feature engineering**	**prognostic algorithm**	
None	None	rf	1650,88
None	None	SVM	1775,05

improve the prognostic models. However, it seems as if increasing the complexity of prognostic models does not necessarily lead to an improvement in terms of MSE. Random forests are known to be adaptive themselves. Therefore it is not too astonishing that simply using a RF on the unaltered dataset outperforms the methods in which we make changes to the dataset in terms of size or dimensionality. Especially, since the underlying models are trained and tested on dataset FD001, which is considered the simplest of the C-MAPSS datasets since it only contains one failure mode and operating conditions (see Table 5.1).

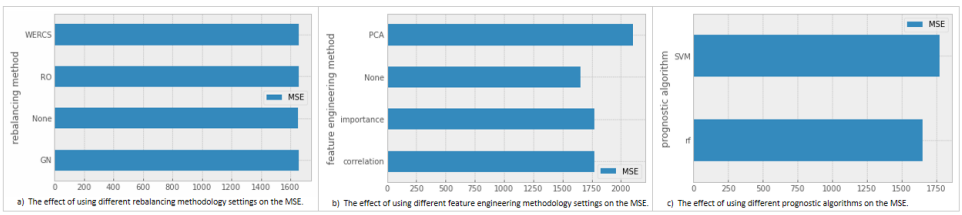


Figure 4.8: A comparison of applying the different prognostic settings on dataset FD001.

RESULTS SIMULATED TURBOFAN DATA

The GPF is applied to all four datasets and to evaluate how the performance, it is compared to pure RF and SVM models. "Pure" here refers to the models obtained by training the RF and SVM algorithms with the settings found in the grid search (see step 2 in Section 4.2.2) directly, i.e. skipping step 3, applying the GPF. The resulting metrics are summarized in Table 4.13 and Figure 4.9. Unsurprisingly the GPF outperforms methods in almost every case. Only for dataset FD004 the GPF makes the choice to use RF directly without including a data rebalancing or a feature engineering method and therefore reaches the same MSE as simply using RF. In general, choices in rebalancing/ fea-

ture engineering do not seem to have a big impact on the quality of resulting predictions (in terms of MSE), as can be seen from Table 4.13. The MSEs are all very close to each other.

Table 4.12: The resulting MSEs of using the GPF versus purely using RF or SVM.

Dataset	Algorithm		
	GPF (50 individuals)	**RF**	**SVM**
FD001	1649.92	1650.41	1775.05
FD002	1877.882809	1974.466387	2152.961399
FD003	4170.12	4239.47	4650.67
FD004	4559.05	4559.05	5238.34

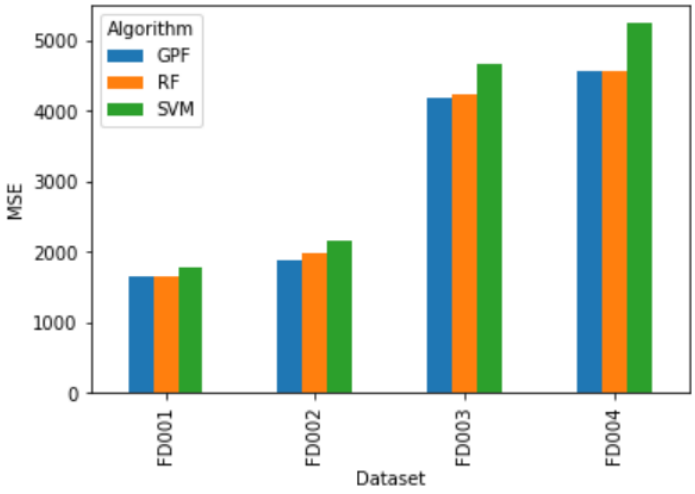


Figure 4.9: The MSE of GPF versus purely using RF or SVM for the four CMAPSS datasets.

More insight in the quality of predictions can be gained by observing Figures 4.10, 4.11, 4.12 and 4.13 showing the resulting predictions and the ground truth on six randomly selected trajectories of the test set for the GPF, the RF and the SVM models. Note that the Figures show six randomly selected trajectories of the test set and they might not be a representative choice as the performance varies between different trajectories. Still, the Figures give some insight into how well the models are able to capture degradation. By and large, the trends are captured quite well. Throughout all four datasets FD001 - FD004 it can be observed that the true RUL is better approximated by predictions for longer trajectories, i.e. trajectories operating for longer than 100 time cycles. For shorter trajectories throughout all datasets the algorithm is not able to predict RUL accurately

or capture degradation trends. For dataset FD001, the least complex dataset, the trends are in most cases very close to ground truth. For most trajectories the RF outperforms SVM (see Figure 4.10 a), d) and e)). Although for trajectories 15 and 87, shown in Figure 4.10 b) and f), this is not the case, those are also the cases with the trajectories only running for a bit more than 70, respective 30 time cycles. In dataset FD002 for trajectories 7, 15 and 46 represented in Figure 4.11 a), b) and e) the RUL prediction is very close to the ground truth and for most of the other trajectories the RUL towards the end of the component life is predicted quite accurately. In dataset FD003 the predictions seem more unstable. Still the degradation is captured quite well, especially towards the end of life. As mentioned before for dataset FD004, the GPF chose as the optimal prognostic settings RF without any feature engineering or rebalancing method, therefore only two lines visible in the plots. On the chosen trajectories it seems as if the SVM outperforms RF quite often, although most of the trajectories are quite short (none is longer than 175 time cycles), so the set of trajectories might not be a good representation of the overall performance.

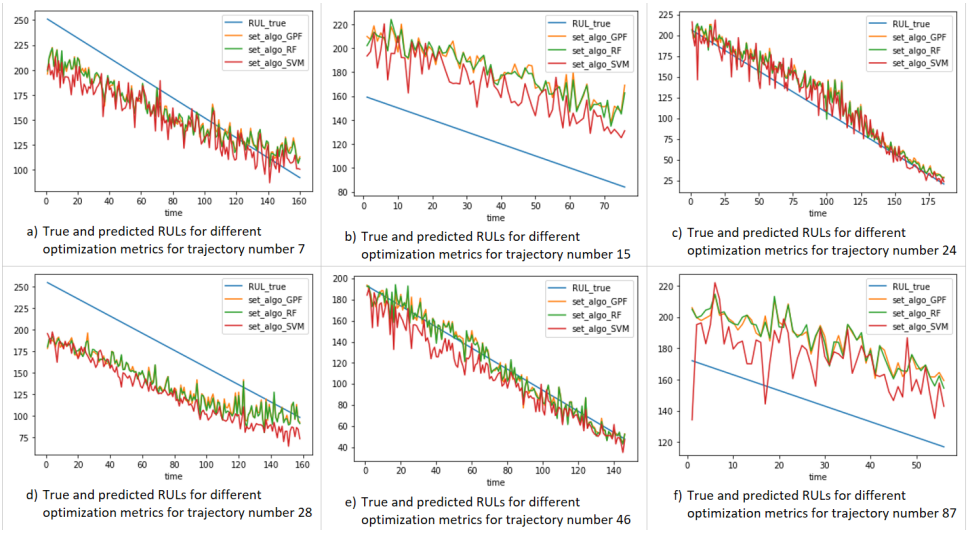


Figure 4.10: True and predicted value on dataset FD001 for six different trajectories when using the GPF, RF and SVM.

Table 4.13 shows the chosen prognostic settings when running the GPF on the four C-MAPSS data sets with 20, 30 and 50 individuals. We see a consistency of the choices of the GPF over the population size. Furthermore in those cases where the choices of methodologies differ, than it is only minor changes in the settings, e.g. a different selection of rebalancing method for datasets FD001 and FD003. Furthermore, we note that the GPF consistently chooses the RF over the SVM, only for dataset FD003 it selects the SVM, which together with the suiting feature engineering and data rebalancing methods even outperforms the RF. This shows the importance of including such steps when developing prognostic models. While the differences in terms of MSE in this case are minor, it can be the case that they are bigger for a different dataset.

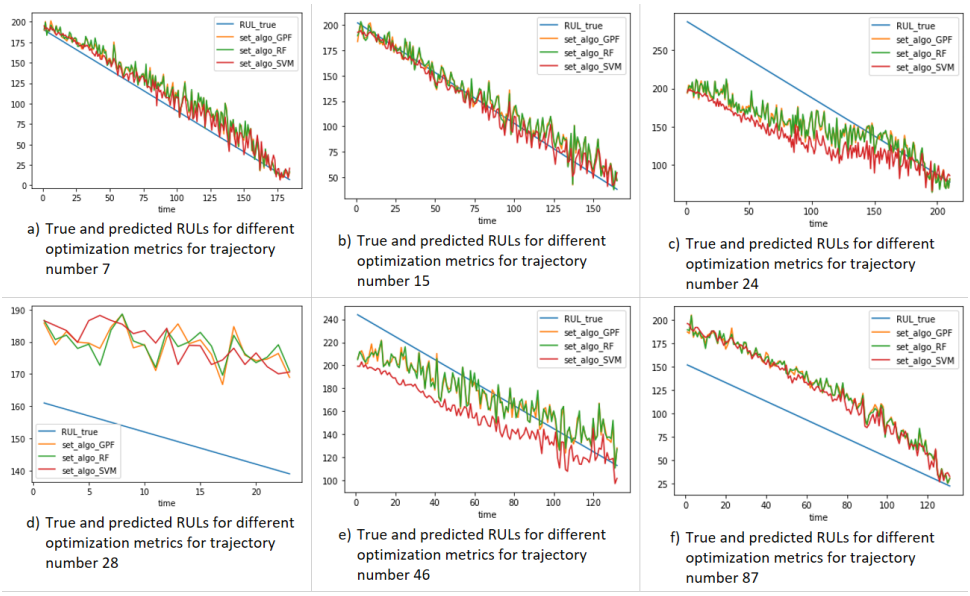


Figure 4.11: True and predicted value on dataset FD002 for six different trajectories when using the GPF, RF and SVM.

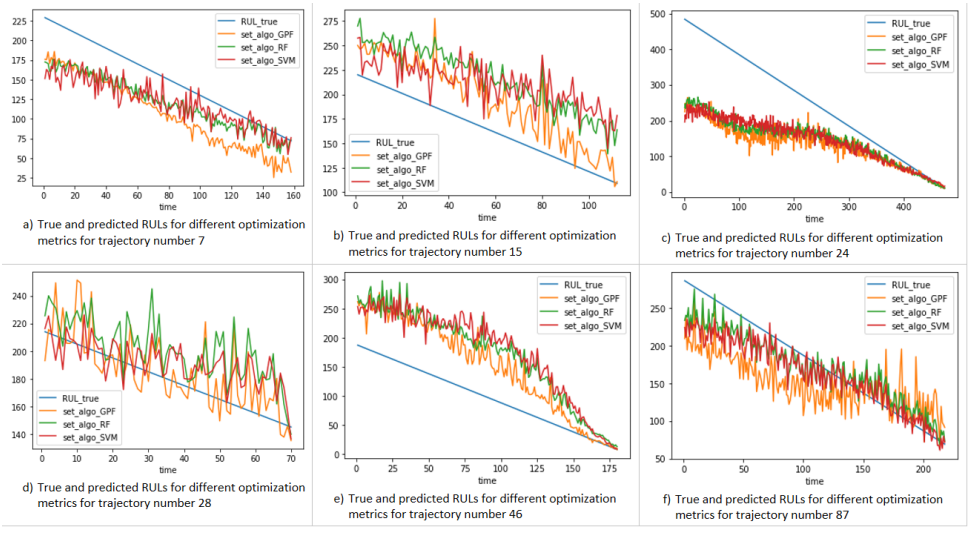
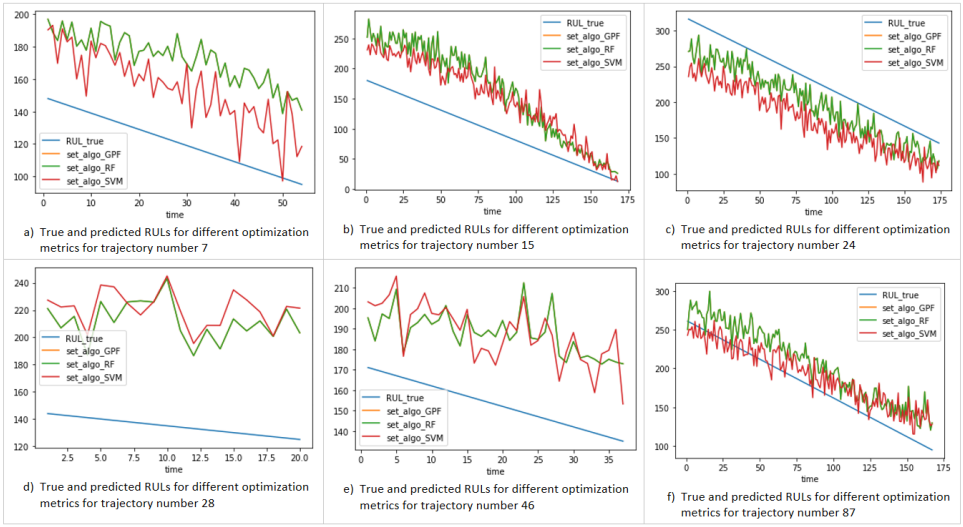


Figure 4.12: True and predicted value on dataset FD003 for six different trajectories when using the GPF, RF and SVM.

All in all, on the C-MAPSS dataset even simple methodologies, such as applying RF and SVM without any feature engineering or data rebalancing yield quite promising results and although the GPF improves performance, it does not significantly add to the predic-



a) True and predicted RULs for different optimization metrics for trajectory number 7

b) True and predicted RULs for different optimization metrics for trajectory number 15

c) True and predicted RULs for different optimization metrics for trajectory number 24

d) True and predicted RULs for different optimization metrics for trajectory number 28

e) True and predicted RULs for different optimization metrics for trajectory number 46

f) True and predicted RULs for different optimization metrics for trajectory number 87

Figure 4.13: True and predicted value on dataset FD004 for six different trajectories when using the GPF, RF and SVM.

Table 4.13: The resulting prognostic settings when running the GPF with populations of 20, 30 and 50 individuals on the four C-MAPSS datasets.

Dataset	Population size	Rebalancing	Feature engineering	Prognostic Algorithm
FD001	**20**	WERCS	None	RF
	30	RO	None	RF
	50	RO	None	RF
FD002	**20**	GN	None	RF
	30	GN	None	RF
	50	GN	None	RF
FD003	**20**	None	importance	SVM
	30	None	importance	SVM
	50	GN	importance	SVM
FD004	**20**	None	None	RF
	30	None	None	RF
	50	None	None	RF

tion quality.

4.3.2. AIRCRAFT SUPPLEMENTAL COOLING UNITS

As a second case study, we consider cooling units (CUs) installed on aircraft operated in a modern and widely-used airline. They are part of the cooling system, which cools the aircraft galleys. On each considered aircraft, four CUs are installed in the cooling system and each consists of a condenser, a flash tank, an evaporator and a compressor.

During flights, one or more CUs can be in operation at the same time, but in general the aircraft tries to spread loads equally over the four CUs. The system is maintained in a run-to-failure way, in the sense that if one of the CUs fails, the entire system is repaired and replaced.

COOLING UNITS DATASET

The dataset provided by the airline contains both, sensor data and contextual data. On each cooling unit nine sensors are installed (i.e. 36 sensors in total for the four CUs) measuring different system properties continuously during flights at a rate of 1Hz, resulting in 26.4 GB of sensor data for two and a half years of operation corresponding to 18295 flights. In addition to the sensor measurements, the data contains information such as a flight ID, the plane tail (a unique identifier for each aircraft), the departure date and time, the flight phase and the row number specifying the exact time of the measurement. Note that every flight cycle consists of 14 flight phases from departure to landing. The contextual data contains information about failures and replacements, documenting when failures, each identified by a failure ID, on which cooling unit happened. More information about how maintenance is performed on the CUs and how the dataset was constructed can be found in (Basora et al., 2021b).

APPLICATION OF THE GPF TO THE DATASET

In order to apply the GPF to the cooling unit dataset provided by the airline, several basic data pre-processing steps were conducted. First of all, the sensor measurements together with the information about failures have to be translated into run-to-failure trajectories, similarly as those contained in the C-MAPSS dataset. For each aircraft (identified by plane tail), based on the contextual datasets containing replacement time and date for each failure ID, the trajectories can be constructed using the flight IDs, departure date and time, flight phase and row number. As a next step, the nine sensor measurements for each CU are aggregated per flight phase by their mean, minimum and maximum value. This is done on the one hand for smoothing the dataset and reduce the noise and on the other hand to reduce the size of the dataset to make it more applicable for the GPF. The aim is, after all, to provide a quick prognostic assessment rather than a perfect prognostic model. In Table 4.14 the resulting 24 trajectories are listed including the number of data points (after the aggregation) and the number of flight cycles of operation until failure.
Using the resulting trajectories, for each the RUL is calculated in the same way as on the C-MAPSS dataset in Section 4.3.1 as a linear function of time, in this case measured in flight cycles until failure. Figure 4.14 shows the mean RUL for the 24 trajectories of the CU dataset. For some sensors there are `nans` or missing values in the dataset. Since they account for only 2.12% of data, we simply remove them from the dataset. The last step which has to be performed to apply the GPF to the cooling unit dataset is to split the data into a train and test set. Now, with only 24 trajectories it seems a natural choice to apply cross validation. What we use in this work is a leave-one-out cross validation approach and with a number of test set size of roughly 10% of the train set size, this corresponds to selecting 2-3 random trajectories for the test set and keeping the others in the train set. To be more precise, for the selection of the prognostic methodologies, i.e. step 3 of the GPF as described in Section 4.2.3, we use the following approach: For each generation

Table 4.14: The 24 trajectories of the CUs, the number of flight cycles in operation and the number of data points after aggregation.

Failure ID	Plane Tail	# data points	Flight Cycles
111	dlkzncgy	24593	2236
18	wnjxbqsk	16623	1511
114	enwslczm	12877	1170
116	iefywfmy	11845	1077
115	iefywfmy	11746	1068
118	dlkzncgy	10519	957
112	dlkzncgy	10244	932
108	trmblwny	8998	818
109	tjyjdtaf	8921	811
113	lbhkyjhi	8836	803
105	dlkzncgy	7119	648
31	iefywfmy	6770	616
22	iilvtkok	13440	611
110	iilvtkok	6255	569
107	ibauqnxj	5403	491
117	cntxlxyh	5391	490
23	iilvtkok	4966	452
25	lbhkyjhi	3358	305
26	tjyjdtaf	2751	250
28	tjyjdtaf	2192	199
24	lbhkyjhi	1763	160
2	ibauqnxj	1661	151
11	rgwwyqtt	517	47
17	wnjxbqsk	88	8

of the genetic algorithm, when the next population of individuals is selected, the genetic algorithm also creates a new train and test set - based on the above described leave-one-out cross validation approach. In addition to that, during training the prognostic models in step 3 of the GPF, the trajectories are cut n flight cycles before failure, where n is set to $50, 100, 200$ or 500. This means that only the last n flight cycles before failure are used for the training, which is useful for two reasons: First, this reduces the dataset size and therefore also the computational time needed. Second and more importantly though, this reduces noise introduced by long running trajectories that do not contain much information about degradation behaviour and condenses the information on the failure dynamics. Note that in step 4 of the GPF (Section 4.2.4), when training the prognostic model using the by the GPF chosen settings, as opposed to using cross-validation the train and test sets are fixed and the test set consists of the three trajectories with failure IDs 108, 113 and 116.

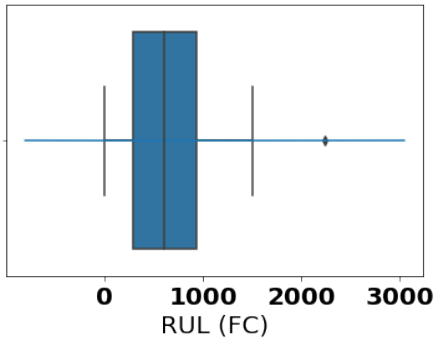


Figure 4.14: The mean RUL for all trajectories of the cooling unit dataset.

RESULTS COOLING UNIT DATASET

Table 4.15 and Figure 4.15 show the resulting MSE of using the GPF, compared to using purely RF and purely SVM on the cooling unit dataset for different cut settings. We can see that the GPF always outperforms the RF and SVM by margins, i.e. including feature engineering and/ or rebalancing methods seems to have a significant impact on the prediction quality. Table 4.16 showing the resulting prognostic settings found by the GPF and the results below introduce further details and make this even clearer. The best results in terms of MSE are achieved when cutting 500 FC before failure. This is not surprising, since the dataset behind it contains the most information. The cutting can be seen as some kind of classification of the data in healthy and faulty behaviour. Therefore, they do have quite some influence on the quality of predictions as can be clearly seen in Figure 4.15. However, while the MSE is lower when cutting 50 Flight cycles (FC) before failure (see Table 4.15), this is not really comparable to the slightly higher MSEs when cutting 100 or 200 FC before failure, since the MSE punishes false predictions closer to the end of life of a component less than false predictions at the beginning.

Table 4.15: MSE of using GPF, only RF or SVM for different cut settings (cut 50, 100, 200 or 500 FC before failure)

Settings		**MSE**		
Population size	**cut**	**GPF**	**SVM**	**RF**
20	50	121133	252559	256327
20	100	160608	228725	235180
20	200	176610	191486	186678
20	500	12818	43626	75002

Table 4.16 contains the by the GPF chosen prognostic settings for different cut settings and the corresponding MSEs. In all cases rebalancing methods are chosen and in most cases, feature engineering methods are also included by the GPF to arrive at the optimal prognostic output. Still, the MSE is remarkably high in all cases even when it is low compared to using only RF or SVM.

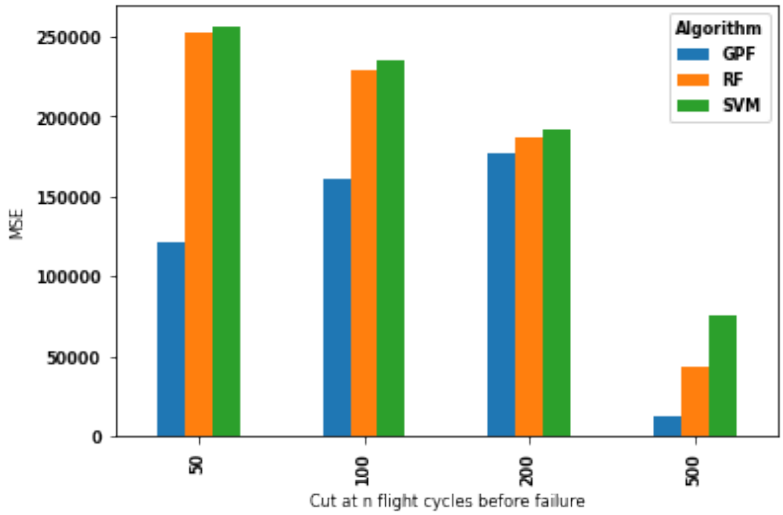


Figure 4.15: MSE of using GPF, only RF or SVM for different cut settings (cut 50, 100, 200 or 500 FC before failure)

Table 4.16: Chosen prognostic settings and MSE for different cut settings (cut 50, 100, 200 or 500 FC before failure)

Population size	cut	Rebalancing	Feature engineering	Prognostic Algorithm	MSE	percentage of data
20	50	RO	importance	SVM	169933	7,15
20	100	GN	importance	SVM	160608	12,74
20	200	GN	PCA	SVM	119626	27,40
20	500	WERCS	None	rf	12818	55,98

The resulting predictions and the ground truth on three trajectories of the test set for the GPF, using only the RF and only the SVM for predictions when using different cut settings (cutting 50, 100, 200 and 500 FC before failure) are displayed in Figures 4.16, 4.17,4.18 and 4.19. Cutting 50 FC before failure results in quite unstable predictions, which do not depict any degradation trends at all. When including a bit more points and cutting 100 FC before failure this changes. In fact, using Gaussian Noise to do rebalancing and applying the random forest importance feature selection methodology, improves the prediction quality in such a way that now a trend is captured compared to the RF and SVM models predictions (see 4.17). Table 4.15 reflects this behaviour in the lowered MSE of using the GPF as compared to using only RF or SVM. For cutting 200 FC before failure, the predictions seem to be less stable, perhaps due to the additional noise introduced through the data. This changes again when cutting 500 FC before failure as displayed in Figure 4.19. In this case the predictions become more stable again and especially the GPF captures the degradation trend quite well.
All in all, the two main points we find when applying the GPF to the cooling unit dataset

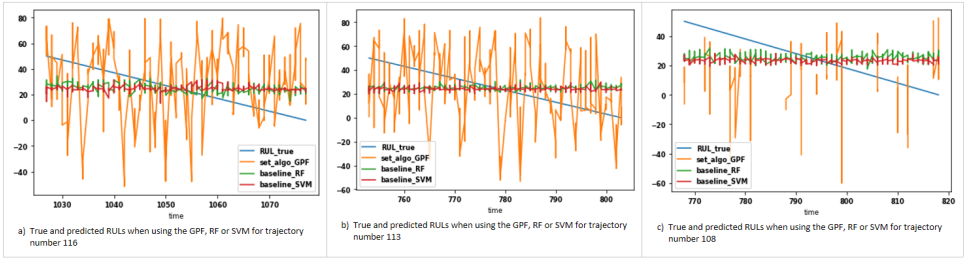


Figure 4.16: True and predicted values for three different trajectories of the SCU test set when using the GPF, RF and SVM (cut 50 FC before failure).

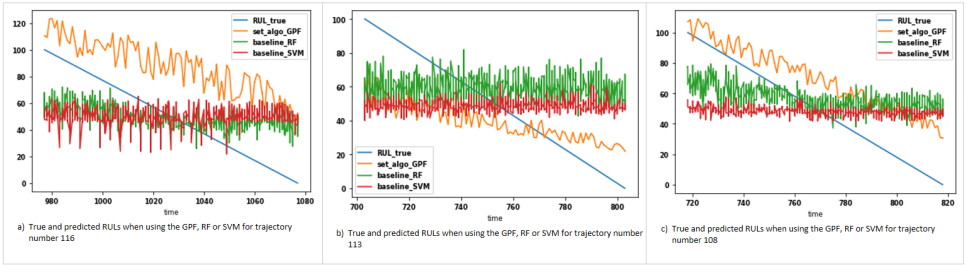


Figure 4.17: True and predicted values for three different trajectories of the SCU test set when using the GPF, RF and SVM (cut 100 FC before failure).

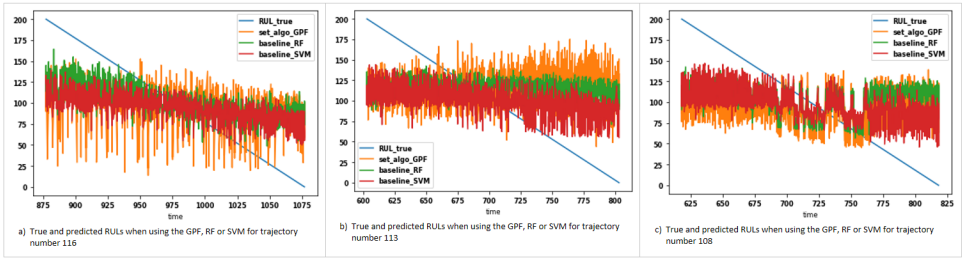


Figure 4.18: True and predicted values for three different trajectories of the SCU test set when using the GPF, RF and SVM (cut 200 FC before failure).

can be summarized as follows: First, the GPF outperforms using simple machine learning methods by margins (in terms of MSE). Second, the impact of when to 'cut' the data before failure in the train set is high, which can be seen as the impact of labelling data as 'healthy'/ 'faulty'. A next step can be to use a piecewise linear function similarly to existing approaches on the C-MAPSS dataset, like the one presented in (Jayasinghe et al., 2018), or to use a health indicator flagging data as 'healthy' or 'faulty'.

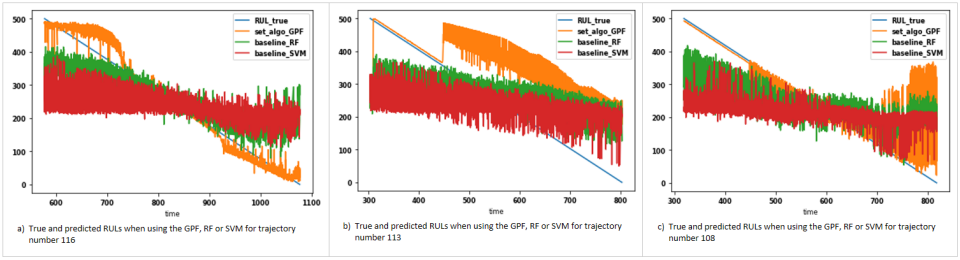


Figure 4.19: True and predicted values for three different trajectories of the SCU test set when using the GPF, RF and SVM (cut 50 FC before failure).

4.3.3. COMPARATIVE EVALUATION AND DISCUSSION OF THE RESULTS

In Section 4.1 we put forward the idea of applying the GPF to assess the ability to perform prognostics on a system, i.e., to find out whether it makes sense to put more time into training prognostic models and applying more advanced prognostic methodologies on the system data. Now that we have the results of applying the framework to both a simulated prognostic dataset, which is known to be suitable for RUL estimation models, and a real aircraft cooling unit dataset, we can compare and draw some conclusions. First, in section 4.3.3 we highlight the similarities and differences between prognostics on simulated and real data sets. Second, we go into details on using the GPF to determine the ability to perform prognostics on a system based on the underlying data in Section 4.3.3 and thirdly, we discuss limitations and directions for further research in Section 4.3.3.

SIMILARITIES AND DIFFERENCES BETWEEN SIMULATED AND REAL DATA

The GPF was applied to both a simulated aircraft turbofan dataset and a cooling unit dataset provided by an airline. The results presented for the C-MAPSS dataset in Section 4.3.1 and for the cooling unit dataset in Section 4.3.2 highlighted some of the challenges that arise when using prognostics on a real aircraft data set opposed to using prognostic approaches on simulated data. Three main points can be discerned. First, the much smaller number of failures leads to a smaller dataset. When using data-driven prognostic methodologies this can lead to less stable predictions and in some cases to models that are not able to predict RUL reliably at all. This can be seen when comparing e.g. Figure 4.11 to Figure 4.18 showing the true and predicted values for trajectories of the test set of the C-MAPSS dataset FD002 and the cooling unit dataset when cut 200 FC before failure respectively. Not only this, but throughout Figures 4.10 to 4.13 the degradation trend is much better captured than for the cooling unit dataset (Figures 4.16- 4.19). Second - and this point is closely linked to the previous one - including additional steps such as data pre-processing, data rebalancing or feature engineering to predict RUL can improve the quality of the predictions. This is true even when the methodologies are not tailored towards the dataset, but only applied in a basic way as it is done through the GPF. The impact of including such methodologies is much higher for the cooling unit dataset compared to the turbofan dataset. This becomes clear from the optimal choice of methodologies presented in Table 4.16 for the cooling unit and in Table 4.13 for the C-MAPSS dataset. And this leads to the third point we noticed when applying the GPF to

both datasets: While the GPF outperforms the basic machine learning models in every case for both simulated and real data (see Table 4.13 and Figure 4.9, respectively Table 4.15 and Figure 4.15), it still has much higher potential for improvement for the cooling unit dataset.

USING THE GPF TO DETERMINE THE ABILITY TO PERFORM PROGNOSTICS ON A SYSTEM

As noted in the previous section, especially applying the GPF to real data seems to result in predictions of much better quality for the cooling unit dataset. This indicates the GPF provides a more thorough prognostic assessment as simply applying a RF or SVM would do. Since it is straightforward to apply the framework, it can not only give an indication over which prognostic methodologies might be the most effective on a given dataset, but also it can give an indication of the ability to perform prognostics on a system. In Section 4.1 we defined the ability to perform prognostics on a system to mean that meaningful and accurate data-driven prognostic models can be developed based on given underlying operational and failure data for a system. To be more precise, the assessment of whether or not a system is prognosable is approached from a data suitability point of view. The aim is to understand if, based on the system data, we are able to retrieve first simple prognostic models. If this is not the case, the system data might not be of sufficient quality and size to train a prognostic model. It is not very surprising that the simple prognostic methodologies included in the GPF result in quite accurate predictions in terms of MSE and visually compared against the true RUL on all four simulated data sets. The C-MAPSS datasets are after all created for prognostics and it has been shown over the past decade multiple times in literature that even with simple methodologies the RUL of the underlying turbofan engine can be accurately estimated. For the cooling unit dataset this is a bit more complicated. There are several additional challenges when working with a real dataset, as covered in the previous Section 4.3.3. Other authors who have worked on the cooling unit dataset noticed the same: In their paper in which they present an anomaly detection method and apply it to the dataset, (Basora et al., 2021b) point out that the prediction of fault occurrences proved a challenge, especially due to the fact that fault dynamics are different from one case to another. This situation is not improved by the small number of faults and the lack of knowledge of failure modes. Still, other authors found that applying prognostic methodologies to the same dataset results in quite accurate RUL predictions (see (de Pater & Mitici, 2021) and (Rosero et al., 2022)).

All in all, we would therefore from previous works in literature conclude that the system based on the collected data is prognosable. The only remaining challenge is to extend the dataset and especially collect more data concerning faults. This is in alignment with the results presented in Section 4.3.2 and becomes especially visible in Figures 4.17 and 4.19. Based on this and our findings in the previous Sections 4.3.1 and 4.3.2, the output of the GPF can be used to tell if a system is prognosable when keeping the following in mind: Even if the MSE does contain some information on the ability to perform prognostics on a system, this information does not suffice to make real implications. Depending on the dataset the resulting MSE can differ significantly and it does not give a real indication if the degradation trend is captured or not. E.g. Table 4.15 shows that the GPF results in a lower MSE than the classic machine learning algorithms, but Figure 4.16 shows us that

exactly like the RF and SVM models, it doesn't seem to capture any trends at all on the test dataset. For this purpose, visual representations of the predictions can be helpful.

LIMITATIONS AND FURTHER RESEARCH

The findings of this research are subject to several limitations, which point out directions for further research.

- As mentioned in the previous section, the use of the MSE in isolation does not give a sufficient insight into the performance of prognostics. In addition to visualisation of trajectories, several other metrics can be used to given additional insight into prognostic performance, such as the prediction horizon.
- Only a limited amount of methods are included in this application of the GPF, which limits the assessment of the ability to perform prognostics on a system. Nevertheless, the GPF can easily be extended to include alternative methods such as neural networks and their myriad variations. This will however have implications on the computational runtime of the GPF; careful balancing might be required between prognostic performance and computational performance in view of organisational objectives (i.e., obtaining a first assessment of a dataset or obtaining the best performing model).
- For now we apply hyperparameter tuning only to determine initial settings for the prognostic algorithms. However, further research could go into the investigation of using different hyperparamter selection methodologies or pre-select them according to the selected prognostic algorithm.
- The amount of available data is an important consideration when considering the applicability of data-driven methods and by extension the GPF. While this framework has the capability to work with large amounts of input data, a lack of (labelled) failure data may lead to difficulties in accurately predicting future failure events.

While the data availability and quality, especially of failure related data, is one of the biggest challenges when applying prognostics to real system data (Zio, 2022), this is also one of the main points we aim to address with the presented framework. Applying the GPF to system data results in an assessment of the suitability of the underlying data for prognostics. While such an assessment can help in the decision of further prognostic development effort, it also can provide insights into possible arising requirements for further or more dense failure related or sensor data.

4.4. CONCLUSION

We have presented a generic prognostic framework with two major aims: 1) provide an approach capable of identifying a suiting prognostic model given related system data; 2) provide a way to assess system data in terms of the ability to perform prognostics on a system. To substantiate both points, we have applied the framework towards two datasets: the synthetic C-MAPPS dataset and a real aircraft system dataset, enabling a

comparative evaluation. This is in contrast to existing literature, which focuses exclusively on either synthetic or real data, with the latter being much less prevalent. Additionally, as pointed out in the introduction, recent advances in prognostic method developments lack convincing proof regarding generalizibility, i.e., suitability for application beyond synthetic datasets such as C-MAPSS towards real-life industrial cases.

The results of our study suggest that the generic prognostic framework can be adapted to various systems and provides potential towards valid remaining useful life estimates for aircraft systems. Furthermore, the framework provides a means to quickly assess the ability to perform prognostics based onsystem data. In addition to that, we highlight the limitations and challenges with applying prognostics to real-life datasets.

Future research will focus on expanding and testing the methods included in the framework. Furthermore, the influence of a variety of metrics on prognostic performance will be assessed more thoroughly.

5

THE IMPACT OF METRICS ON THE CHOICE OF PROGNOSTIC METHODOLOGIES

In Chapter 1, the introduction of the thesis, the main research objective is formulated as to "provide a framework to assess the application of diagnostic and prognostic methodologies for failure detection and prediction based on system data". This is reflected in the in Chapter 2 defined requirements for such a framework. Chapter 3 and Chapter 4 introduce frameworks for diagnostics and prognostics. Through applying the respective frameworks in case studies to both, simulated and real world data, the frameworks are proven to be generalizable, adaptive and applicable. What remains now is to address the word 'assess' in the main objective. Or put in other words: In this chapter we aim to understand how the in Chapter 4 presented generic prognostic framework can be translated towards a prognostic assessment. Note, that we focus on prognostics here, but the shown methodologies can be adapted towards diagnostics as well. First, the impact of metrics on the choice of prognostic methodologies is characterized. Then, several representative metrics are used within the prognostic framework to guide the decision whether system data is suitable for prognostics or not. The thereby adapted framework is applied in three case studies to complex systems with different underlying data quality, i.e. while some of the systems have data of high quality that can be used to build prognostics models, others do not. The results show two interesting findings: First, the choice of optimization metric has an impact on the output of the generic prognostic framework and on the overall prognostic performance. Second, such a first prognostic assessment can give a rough indication of whether or not it makes sense to use system data to train prognostic models.

5.1. INTRODUCTION

Within the framework of Condition-Based Maintenance, prognostics enable assessment of equipment health and prediction of the RUL (Elattar et al., 2016). Using prognostics in such a context requires properly assessing the quality of predictions (Brunton et al., 2021; Zio, 2022). An effort to standardize prognostic metrics has been made by (Saxena, Celaya, et al., 2008), (Saxena et al., 2010). The metrics commonly used in prognostics are highlighted, and several ways to classify them are presented as ways to interpret and use the metrics. A comprehensive overview of existing metrics to evaluate prognostic performance is given by (Ochella & Shafiee, 2021). A single metric, such as the MSE, can arguably not characterize the quality of RUL predictions sufficiently for a thorough assessment within a CBM framework (Saxena et al., 2014). Instead, the design of prognostic metrics has to be linked to the application and decision-making process (Bi et al., 2017; Sankararaman et al., 2014). In addition, as highlighted in Figure 5.1, metrics are needed to define requirements and thoroughly evaluate prognostic performance (Saxena, Celaya, Saha, Saha, & Goebel, 2009).

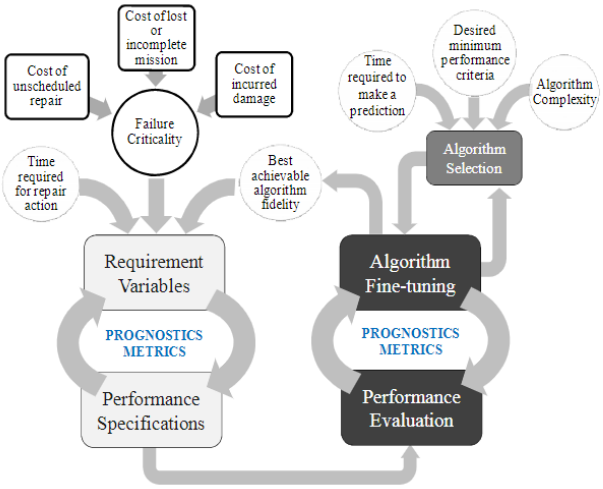


Figure 5.1: Prognostic metrics are needed to define requirements and evaluate performance (Saxena, Celaya, Saha, Saha, & Goebel, 2009)

(Goebel et al., 2017) state that a meaningful prediction has three attributes, namely correctness, timeliness, and confidence (see Section 5.2.1). Performance evaluation of prognostic methodologies should enhance all three of those aspects. However, the vast majority of literature published in the field of prognostics uses only a single metric, which is often one linked to the correctness of the method (Saxena, Celaya, Saha, Saha, & Goebel, 2009). Still, previous works on including more advanced metrics or defining more advanced metrics have been done in the literature. For example, (Amigó et al., 2011) introduce a measurement to combine several metrics and indicate how robust the measured differences are to changes in the relative weights of the individual metrics. (M. L. Baptista

et al., 2022) show that prognostic metrics correlate with a SHapley Additive exPlanations (SHAP) model's explanation. A performance metric to assess performance, effectiveness and efficiency of health monitoring models of complex engineering systems is suggested by (Lewis & Groth, 2022b).

In addition to a suitable prognostic assessment technique, the question remains of how to translate this towards a prognostic assessment. Such an assessment is and must be application dependent. This study focuses on applying prognostics within a CBM framework for aircraft maintenance. A number of publications have been made on the topic of integrating prognostic models in aircraft maintenance planning (de Pater & Mitici, 2021; Pater et al., 2022). A framework for aircraft maintenance design with reliability and cost-efficiency objectives has been provided in (Lee & Mitici, 2022). To use prognostic models as input for maintenance planning, those models need to be developed, which is time-consuming and requires expertise. However, what would be desirable was if, instead of spending months on developing prognostic models, there was a way to assess system data towards their suitability for prognostics relatively quickly. One of the main guiding works in the literature on this topic is perhaps the work by (Coble & Wesley Hines, 2009), in which prognostic parameters are retrieved from the system data to do a prognostic assessment before applying actual prognostic methodologies. A method to evaluate data quality before the modelling by clustering the data into different system conditions is suggested in (Y. Chen et al., 2013). (Omri et al., 2021) propose a set of data quality requirements, especially for health assessment and fault detection. They propose a 'detectability' metric to assess the suitability of data for fault detection. (Atamuradov et al., 2020) present a hybrid feature evaluation with a combined metric. A framework for RUL prediction, including a physics-informed failure mode recognition model that can be applied to different systems with different failure modes, is presented first in (Jiao et al., 2020) and extended by (Xiong et al., 2023).

Two challenges arise from the above-presented literature: One, tuning prognostic algorithms without understanding which metrics are needed to assess the algorithm is difficult. Similarly, it is tricky to understand the full impact of choosing prognostic metrics without considering the prognostic algorithm. Two, while the presented data suitability methodologies are demonstrated in several case studies, they are lacking the link toward prognostic algorithms. Furthermore, often statistical methodologies and pre-defined metrics are used to assess the data quality. This is problematic for several reasons: First, data suitability for prognostics can only truly be assessed when attempting to train a model capable of predicting the system's RUL. Second, AI-based methodologies are in some cases able to detect failures even though the underlying data degradation is not visible or statistically traceable, i.e., statistical methods might not really give us insight into the data suitability for prognostics (Braglia et al., 2012). Third, in order to go beyond a statistic-based data assessment, prognostic performance metrics should be translated toward a data suitability assessment.

To address the challenges listed above, we, therefore, in this paper, investigate the impact of metrics on the choice of prognostic methodologies. On top of that, we explore how

the performance of prognostic methodologies can be translated to an assessment of the suitability of the system for prognostics. Therefore, the following novel contributions to state of the art are made:

1. An integrated framework is presented that selects the optimal prognostic settings, where optimality is assessed in terms of three selected prognostic metrics, representing: correctness, confidence, and timeliness of predictions.
2. A study on the impact of different metrics on the choice of prognostic methodologies is conducted.
3. The resulting outcome is used to define the term 'system data suitability' for prognostics such that it not only includes the data characteristics but also takes into account the data suitability in a CBM framework.
4. An example of the data suitability assessment for aircraft data sets of different quality is given.

The remainder of the paper is structured as follows: Section 5.2 explains the generic prognostic framework used in this study and how it can be used to assess the prognostic suitability of system data. To investigate the impact of prognostic metrics and validate the presented data suitability assessment, in Section 5.3 two case studies are conducted: One on a simulated turbofan dataset and one on a real aircraft system. In Section 5.4 the two research questions are addressed based on the results obtained in the two case studies, and the limitations and directions for further research are highlighted. Section 5.5 concludes the paper and highlights the main findings.

5.2. METHODOLOGY

To select the optimal set of prognostic methodologies, a Generic Prognostic Framework (GPF) as presented in (Bieber & Verhagen, 2022) is used, which contains three steps of prognostics and according to representative techniques. This means that in addition to incorporating different methodologies, the framework includes a selection step in which the best set of techniques is chosen. Note that the essence of the work presented in this paper lies in assessing and optimizing the set of prognostic techniques. The way we measure and evaluate the chosen techniques defines the prognostic settings and, further consequences, the quality of the predictions. In order to evaluate the prognostic performances, we, therefore, use different prognostic metrics to account for different aspects of prediction evaluation. Those metrics integrated into the GPF give us insight into the quality of predictions and thereby help to choose appropriate prognostic methods.

The generic prognostic framework consists of three phases (coloured blocks in Figure 5.2). In phase one, which is highlighted in green, a Genetic algorithm is applied to find the optimal prognostic settings. This is done using multi-objective optimization based on three different metrics, which are explained in more detail in Section 5.2.1. In phase two, highlighted in red and further explained in Section 5.2.2, a prognostic model is trained, which then has the capability to output RUL estimates. In the final step of the

framework, phase three, highlighted in blue, a data suitability assessment is performed. Based on the resulting accuracies in terms of the selected prognostic metrics, thresholds are defined to determine if the system data is suitable for prognostics. A detailed explanation is given in Section 5.2.3.

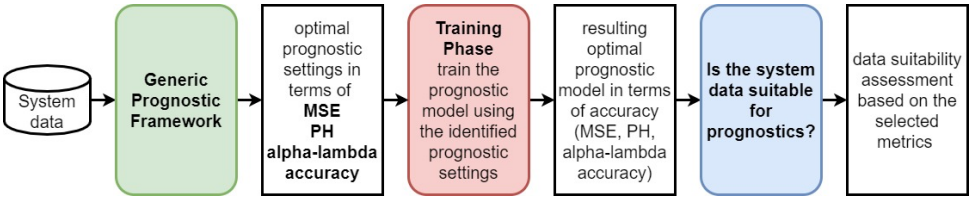


Figure 5.2: The generic prognostic framework flow.

5.2.1. GENERIC PROGNOSTIC FRAMEWORK

The framework used in this work is a modified version of the Generic Prognostic Framework (GPF) presented in (Bieber et al., 2021; Bieber & Verhagen, 2022). It differs from the original framework mainly in optimizing three different prognostic metrics simultaneously. Therefore, we only give a short overview of the elements and functionalities of the generic prognostic framework and refer the reader to the previous work for more details about the GPF. The GPF consists of three blocks corresponding to three selected steps in prognostics: data rebalancing, feature engineering, and the prognostic algorithm itself, as displayed in Figure 5.3. The three blocks each contain several representative methodologies for each of the selected steps in prognostics. Imbalanced data occurs when one class of data (e.g., faulty behaviour) is under-represented when compared to the other class(es) (e.g., healthy behaviour). Data rebalancing methods make use of the concepts of undersampling and oversampling: the former consists in removing majority examples while the latter replicates the minority examples (Santos et al., 2018). Three data rebalancing methodologies as introduced in (Branco et al., 2019) are included:

- Random Over-Sampling (RO),
- Introduction of Gaussian Noise (GN) and
- Weighted relevance-based combination strategy (WERCS).

The feature engineering methodologies in the framework are PCA, correlation-based feature, and importance-based feature selection representing, respectively, feature extraction, filter-based feature selection, and embedded feature selection techniques. In order to get a first prognostic assessment through the framework, the prognostic algorithms included are a Random Forest Regression (RF) and a Support Vector Regression (SVM). The two selected algorithms are well-established and offer potential advantages in terms of interpretability and explainability (Ward & Habli, 2020). However, they will generally not offer performance on the level of bespoke, advanced models developed for specific applications.

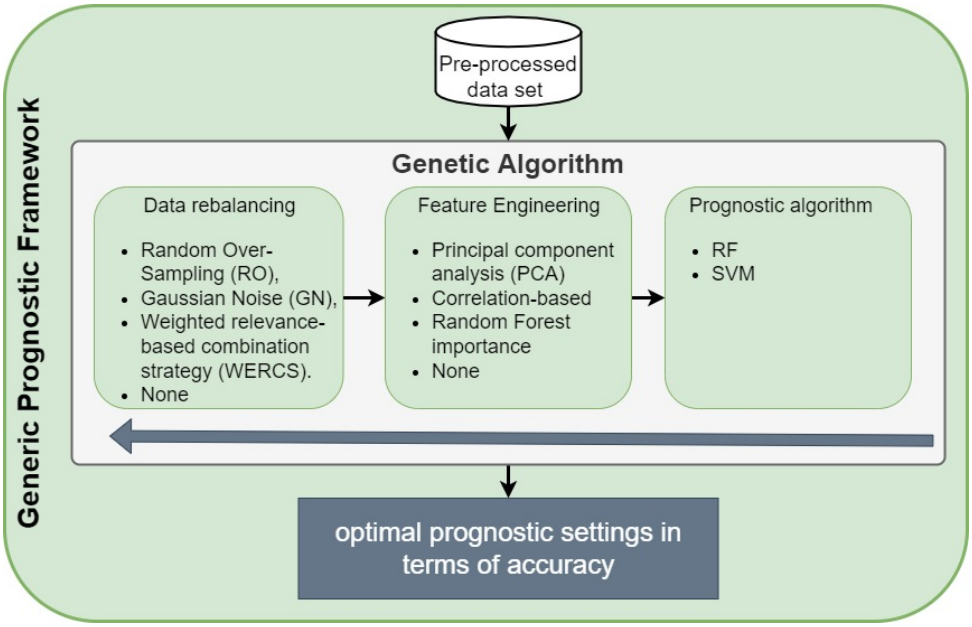


Figure 5.3: The elements of the generic prognostic framework

The GPF selects optimal sets of methodologies for each of the three steps in the prognostic framework. Here, 'optimal' refers to the best in terms of the mean-squared error (MSE), Prognostic Horizon (PH), and $\alpha-\lambda$ score. In other words, we treat the problem of finding the prognostic settings as a multi-objective optimization problem: The objective function is to simultaneously minimize the MSE, maximize the PH, and maximize the $\alpha-\lambda$ score of the prognostic algorithm together with data re-balancing and feature engineering techniques on the pre-processed data set. To solve the optimization problem, we use a genetic algorithm (GA). These algorithms are based on the concepts of natural selection and genetics (Holland, 1992). Due to their flexibility, GAs can solve global optimization problems and optimize several criteria at the same time, like in our case, the simultaneous selection of data re-balancing, feature engineering, and prognostic algorithm techniques (Stanovov et al., 2017).

Basically, there are two approaches to multi-objective optimization: The first is to create a single optimization objective by combining the individual objective functions. The second is to move all but one objective to the constraint set (Konak et al., 2006). This approach results in a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by another solution. Due to the fact that GAs are a population-based approach, they are well-suited for multi-objective optimization problems. Sets of solutions are returned in every generation; therefore, multiple solutions can easily be returned (Konak et al., 2006). In fact, a majority of the multi-objective optimization problems in current literature are solved using evolutionary approaches (Jones et al., 2002). Several multi-objective approaches for GAs have been suggested in the lit-

erature, and a comprehensive overview can be found in (Konak et al., 2006). We use the Non-dominated Sorting Genetic Algorithm II (NSGA-II, introduced in (Deb et al., 2002)). It ranks candidate solutions with the fast non-dominated sorting method and uses a crowding distance as a diversity mechanism. The algorithm is well-tested, has been used in many applications and is efficient, which makes it a good candidate for this study.

The NSGA-II, in our case, takes the system data as input and outputs the set of Pareto dominant solutions. A solution is Pareto dominant if there does not exist any other feasible solution that dominates it (Hua et al., 2021). In this case, a solution is a combination of a data re-balancing technique, a feature engineering methodology, and a prognostics algorithm. If the algorithm identifies that applying no re-balancing or feature engineering technique results in better prognostic outputs, the GPF returns 'None' for the according to the block. The three different metrics integrated into the framework are the mean squared error (MSE), prognostic horizon (PH), and the $\alpha - \lambda$ metric. The metrics account for the three attributes of meaningful predictions, i.e., correctness (MSE), timeliness (PH), and confidence ($\alpha - \lambda$ metric) (Saxena, Celaya, et al., 2008; Saxena, Celaya, Saha, Saha, & Goebe, 2009).

The MSE at time t it is given as

$$MSE(t) = \frac{1}{t}\sum_{i=1}^{t}(RUL_i - \hat{RUL}_i)^2, \tag{5.1}$$

where RUL_i is the true RUL value and $\hat{RUL}_i$ the predicted RUL value at timestep i.

The prognostic horizon (PH) is defined as

$$PH(t,\alpha) = RUL(t_{i_\alpha}), \tag{5.2}$$

with $RUL(t_{i_\alpha})$ the true RUL at time t_{i_α} and $i_\alpha := min\{k \in p | \forall\, j \geq k : \alpha_j^- \leq \hat{RUL}(t_j) \leq \alpha_j^+\}$, where

- p is the set of all time indices where predictions are made,
- $\hat{RUL}(t_j)$ is the prediction at time index $j \in p$
- and the a bounds are defined as $\alpha_j^- := RUL(t_j) - \alpha$ and $\alpha_j^+ := RUL(t_j) + \alpha$.

, The prognostic horizon is the smallest RUL in which the predicted RUL is still within the specified α bounds. The best score for the PH is obtained when the predicted RUL always falls within the specified accuracy zone, while the worst score is obtained when the predicted RUL is never within the accuracy zone. The PH indicates whether the predicted estimates are within the specified limits, especially towards the end of life (EoL), so that predictions can be considered trustworthy during a specified time span before the system's EoL is reached. It becomes clear that the longer the PH is, the more time becomes available to act based on a prediction. It, therefore, gives an indication of the

timeliness of an algorithm, in the sense that during a time span before the system's EoL, the predictions can be used to plan according to actions. In the case studies presented in Section 5.3, we set $\alpha = 40$ flight cycles, which is the time needed to schedule maintenance for an aircraft in case it is needed.

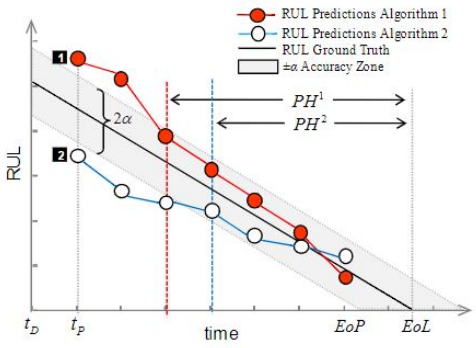


Figure 5.4: Example of calculating the Prognostic Horizon for two prognostic algorithms ((Saxena, Celaya, Saha, Saha, & Goebe, 2009)).

Figure 5.4 gives an example of two prognostic algorithms, Algorithm 1 (represented in red) and Algorithm 2 (represented in blue). The shaded area is the desired accuracy zone, i.e., the condition checked through Equation 5.2 (and more precisely when finding i_a). It can be seen that Algorithm 2 exhibits a poorer performance in terms of the prognostic horizon than Algorithm 1, which has a longer prognostic horizon, i.e., $PH^1 > PH^2$.

And finally, the $\alpha - \lambda$ metric is as in (Biggio et al., 2021) defined as

$$\alpha - \lambda := \begin{cases} 1, & \text{if } (1-\alpha)\lambda^* \leq \lambda_p \leq (1+\alpha)\lambda^* \\ 0, & otherwise, \end{cases} \tag{5.3}$$

with $\lambda^* = RUL(t_\lambda)$ the ground truth, $\lambda_p = RUL(t_{\lambda_p})$ the prediction and α an arbitrary chosen accuracy. The two input parameters for the metrics are α, which determines the required level of confidence for the predictions, and λ, which represents a fraction of time between the point when the algorithm starts predicting (t_P) and the actual failure or End of Life (EoL).

The $\alpha - \lambda$ metric, therefore, measures the prediction quality by determining whether the prediction falls within specified limits at particular times, which -as mentioned above- are presented as a percentage of the total ailing life of the system. To be more precise, the question it seeks to answer is whether the prediction accuracy of the RUL model is within $\alpha \cdot 100\%$ of the actual RUL at a specified time instance t_λ (depending on λ). The output is binary (true or false), stating if the desired condition (Equation 5.3) is met at the specific time instance. It is more stringent than the PH because it requires the predictions to stay

within a cone of accuracy, i.e., bounds that shrink as time passes. This is also visible in the presented examples when comparing Figure 5.5 to Figure 5.4).

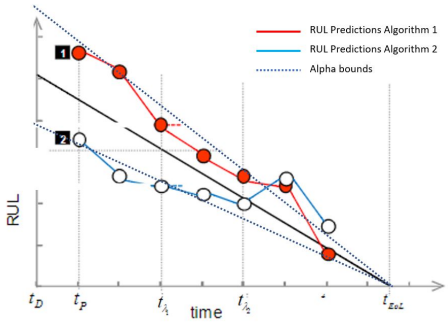


Figure 5.5: Example of calculating the $\alpha - \lambda$ metric for two prognostic algorithms (modified from (Saxena, Celaya, Saha, Saha, & Goebe, 2009)).

To demonstrate how the $\alpha - \lambda$ metric is calculated, Figure 5.5 shows an example of calculating it for two different prognostic algorithms, Algorithms 1 (represented in red) and Algorithm 2 (represented in blue). The black line shows the true RUL and the dotted lines show the specified α bounds, i.e., $(1-\alpha)\lambda^*$ and $(1+\alpha)\lambda^*$. When now calculating the $\alpha - \lambda$ metric for λ_1, as becomes visible in Figure 5.5, for Algorithm 1, it would be 1, while for Algorithm 2, which predicts an RUL outside the α bounds, it would be 0. For λ_2, i.e., at t_{λ_2}, both Algorithms yield predictions within the bounds, and therefore the $\alpha - \lambda$ metric is 1 for both.

The $\alpha - \lambda$ metric can be evaluated and averaged over the whole trajectory with N time steps (i.e., for the entire interval $[t_P, EoL]$), arriving at $\overline{\alpha - \lambda}$, which lies between 0 and 1. It, therefore, returns the confidence that the predictions fall into the α bounds over the entire period of time. This is why it is a good candidate to represent prediction confidence. In the example in Figure 5.5, for Algorithm 1, which performs visibly better in terms of $\alpha - \lambda$ metric than Algorithm 2, $\overline{\alpha - \lambda}(Algo1) = (6*1+1*0)/7 = 0.857$, whereas for Algorithm 2 the metric over the entire time interval is $\overline{\alpha - \lambda}(Algo2) = (3*1+4*0)/7 = 0.429$.

5.2.2. TRAINING PHASE

The output of the GA is the 'best individual', i.e., the set of methodologies and hyperparameter settings that lead to the best performance on the dataset. Note that in order to save computational power and arrive at a solution more quickly, the GPF only takes a reduced dataset as an input for the optimization (Bieber & Verhagen, 2022). In this step, the prognostic model is trained on the full dataset using the optimal settings returned by the GPF. Therefore, the output of this step is a trained prognostic model, which takes as input system data and outputs the remaining useful life (RUL).

5.2.3. DETERMINING IF A SYSTEM IS SUITABLE FOR PROGNOSTICS

Once the GPF has identified a set of optimal prognostic models in terms of MSE, PH and $\alpha-\lambda$ score and outputs the according scores, phase three (indicated in blue in Figure 5.2) starts. Based on the output models, this phase aims to identify whether the system data are suitable for prognostics. Of course, the question of whether it makes sense to apply prognostic approaches for given data highly depends on the user, the application, and the underlying requirements. As highlighted in Section 5.1 we aim to assess data suitability in a prognostic context. This means we go beyond a simple statistical assessment and instead translate prognostic metrics of basic prognostic machine learning models trained on the underlying system data into a data suitability assessment. The definition of 'system data suitability for prognostics' depends on user inputs, which can be adapted accordingly. The user needs to set bounds for each of the criteria measured:

- In terms of correctness, MSE_{max}, the upper MSE limit,
- in terms of timeliness, $PH_{min}(a)$, the minimum number of time steps before failure at which the failure needs to be known to take according to actions, which is based on a, the maximum value (measured in time steps) that the prediction is allowed to deviate from the true value,
- and in terms of confidence, $(\alpha-\lambda)_{min}$, where $0 < (\alpha-\lambda)_{min} < 1$, the minimum ratio of predictions within the α bounds.

System data is defined to be suitable for prognostics if

$$MSE(t = \text{end of life}) \leq MSE_{max} \tag{5.4}$$

$$\wedge\ PH(t_j) \geq PH_{min}(a)\ \forall j \in p \text{ and specified } a \tag{5.5}$$

$$\wedge\ \overline{\alpha-\lambda} \geq (\alpha-\lambda)_{min}. \tag{5.6}$$

Only when all three of the conditions are met for a given prognostic model does that model satisfies the data suitability criteria. This can be applied to each model in the set of optimal models returned by the GPF. If a single prognostic model is found that fulfils the above requirements (Equations (5.4)-(5.6)), then the system data is assumed to be suitable for prognostics.

5.3. RESULTS

There are two main aims of the conducted study: First, we want to understand the impact of prognostic metrics on the methodology selection in the different steps of the prognostic framework. Second, an example evaluation is performed for different input system data to understand if the systems are suitable for prognostics. For this purpose, two case studies were conducted: The first case study in Section 5.3.1 is conducted on a simulated turbofan dataset commonly used in literature and known to be suitable for prognostics. The second case study in Section 5.3.2 uses a real-world aircraft dataset.

5.3.1. CASE STUDY: SIMULATED TURBOFAN DATASET

The C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) data set contains simulated run-to-failure data for turbofan engines (Frederick et al., 2007). Using

this tool, 4 data sets were created (Saxena, Goebel, et al., 2008a). The data sets differ mainly in the number of fault modes and operational conditions simulated in the experiments. An overview is given in Table 5.1. For our purpose, we use two of the four datasets: First, dataset FD001, is considered the simplest one as it only contains one fault mode and operating condition. And second dataset FD002, is considered to be more complex due to the different operating conditions. Each engine is considered to be from a fleet of engines of the same type, and each time series, also often referred to as trajectory, is from a single unit. The engines are operated until failure, i.e., the time series captures the operations of each unit until it fails. In the test set, the time series ends at some point before the failure, and the objective is to estimate the RUL, or, in other words, the number of remaining operational cycles before failure. There are 21 sensor measurements, and each row in the data set contains the measurements corresponding to operations during a one-time cycle for a certain unit.

The framework is applied to both datasets and in the following, the according results are presented. We compare the resulting prognostic models to baseline models, namely using only RF and SVM, respectively, without any data rebalancing or feature engineering. In all cases, we run the genetic algorithm for 20 generations with a population of 30 individuals.

Table 5.1: Characteristics of the four turbofan engine data sets (Ramasso & Saxena, 2014)

Data set	#Fault modes	#Conditions	#Train units	#Test units	relative #Train units	relative #Test units
#1	1	1	100	100	0.485%	0.485%
#2	1	6	260	259	0.484%	0.762%
#3	2	1	100	100	0.405%	0.603%
#4	2	6	249	248	0.407%	0.602%

RESULTS ON DATASET FD001

First, we present the output of the genetic algorithm, i.e. the Pareto front for dataset FD001. Table 5.2 contains the set of individuals in the Pareto front, with their respective choices of methodologies for the data rebalancing, feature engineering and prognostic algorithm. In addition, the according metrics (MSE, PH and alpha-lambda score) for the trained prognostic models are given.

The results in Table 5.2 show that most of the Pareto optimal solutions use SVM as a prognostic algorithm. Note that the SVM-based solutions outperform RF-based solutions when using feature engineering or rebalancing techniques together with SVM. At the same time, the RF performs well without using any data rebalancing or feature engineering methodologies. The term 'outperforms' here refers to in the sense of a lower

Table 5.2: The resulting best prognostic settings and metrics when running the MOGA GPF with 30 individuals for data set FD001.

rebalancing	**feature engineering**	**prognostic algorithm**	**MSE**	**PH**	$\overline{\alpha-\lambda}$
None	None	rf	1647,10	144.34	0.524206
GN	PCA	SVM	1774,21	129.47	0.536729
None	PCA	SVM	1759,08	132.25	0.536652
RO	None	SVM	1757,86	132.68	0.529704
WERCS	None	SVM	1755,32	130.92	0.531689

$\overline{\alpha-\lambda}$ score in terms of best MSE and PH, using only RF proves to be the optimal technique for FD001. Furthermore, the $\overline{\alpha-\lambda}$ scores are all very close to each other. Finally, it can be seen that increasing the performance in terms of MSE also, in most cases, increases the performance in terms of PH, while it usually results in lower $\overline{\alpha-\lambda}$ scores.

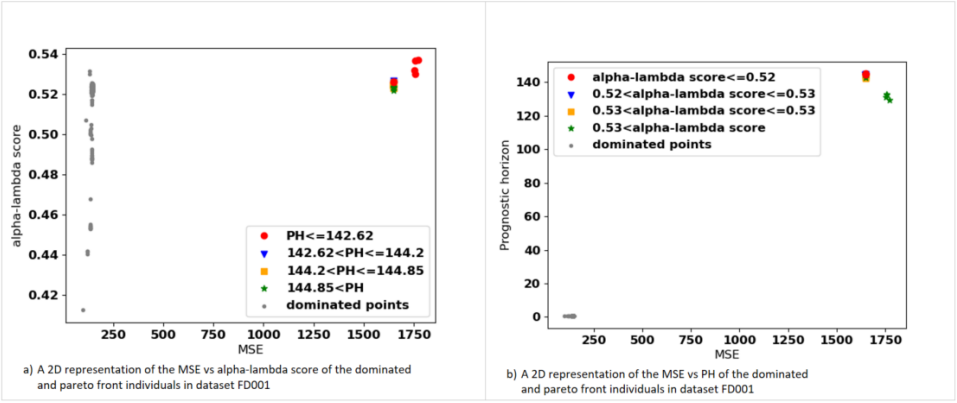


Figure 5.6: A 2D representation of the resulting scores for individuals in the Pareto front and dominated individuals found when running the MOGA GPF with 30 individuals for data set FD001.

Figure 5.6 shows a two-dimensional representation of both individuals in the Pareto front and dominated individuals. The following is observed:

- In Figure 5.6 a) it can be seen that a good score in terms of MSE can be reached without decreasing the performance in terms of alpha-lambda score too much (only from around 0.54 to 0.42)
- Figure 5.6 b) shows that this comes at the cost of reducing the PH to almost 0.
- Therefore, in Table 5.2, only individuals with an MSE of around 1750 are in the Pareto front.

- In this case, when using only the MSE as an optimization metric it would result in models with a poor score in terms of timeliness.

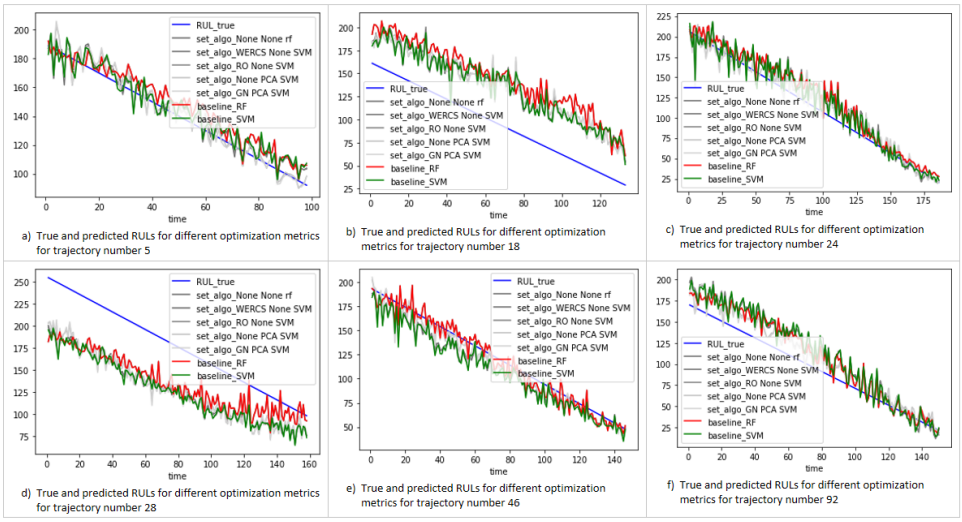


Figure 5.7: Predictions of best-found settings vs. the two baseline scenarios (only RF and only SVM) on example trajectories for a population size of 30 on data set FD001.

Figure 5.7 shows for six randomly selected trajectories in the test set, the true RUL and the predicted values for the individuals in the Pareto front and the baseline models (using purely RF and SVM). For the selected trajectories of dataset FD001 the resulting prognostic models seemingly all perform very well, as do the baseline models. This is especially true for Trajectories 5 (Figure 5.7 a)), 24 (Figure 5.7 c)), 46 (Figure 5.7 e)) and 92 (Figure 5.7 f)). Only in Figure 5.7 a) for Trajectory 5, it can clearly be seen that the GPF-based models outperform the baseline algorithms, especially towards the end-of-live.

RESULTS ON DATASET FD002

Similarly, the results for the runs on data set FD002 are shown in Table 5.3, which contains both the choice of methodologies and the scores in terms of the selected metrics. What can clearly be seen is that the Pareto front contains more individuals than the one for FD001 (10 respectively, 5 individuals). Again, similarly to dataset FD001, in most cases, SVM results in better solutions than RF. While adding a resampling or feature engineering step can improve the predictions in single metrics, the Pareto front contains both the baseline scenarios, using purely RF and SVM.

Figure 5.8 shows a two-dimensional representation of both individuals in the Pareto front and dominated individuals. In the figure, it can clearly be seen that increasing the performance in terms of MSE (decreasing the MSE) results in a better lambda-alpha score (Figure 5.8 a)), but a lower PH (Figure 5.8 b)), which can also be observed in Table

Table 5.3: The best prognostic settings and metrics when running the MOGA GPF with 30 individuals for data set FD002.

rebalancing	feature engineering	prognostic algorithm	MSE	PH	$\overline{\alpha-\lambda}$
None	None	rf	1873,40	117,11	0.463387
RO	None	rf	1865,68	116,50	0.460887
WERCS	None	rf	1872,08	118,64	0.462034
GN	correlation	SVM	2241,46	122,10	0.439204
None	importance	SVM	2262,72	124,53	0.430084
None	None	SVM	2152,96	120,97	0.452355
RO	importance	SVM	2557,72	134,95	0.400665
RO	None	SVM	2188,97	122,18	0.438469
WERCS	importance	SVM	2510,06	132,32	0.404166
WERCS	None	SVM	2189,37	122,00	0.442205

5

5.3.

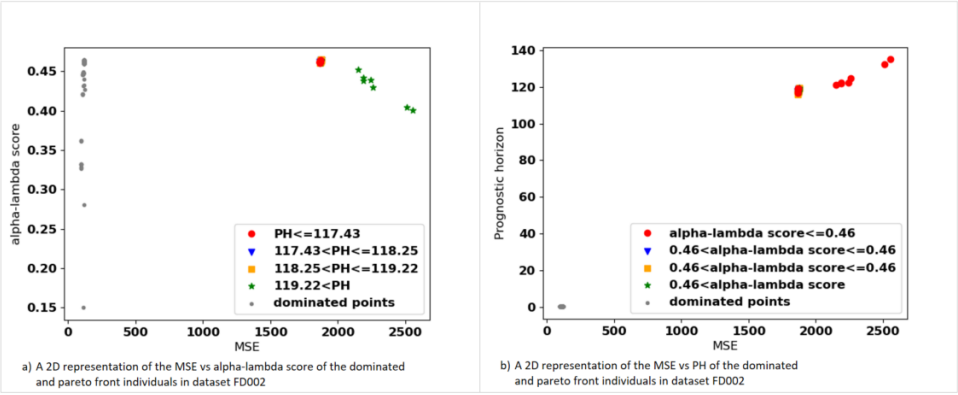


Figure 5.8: A 2D representation of the resulting scores for individuals in the Pareto front and dominated individuals found when running the MOGA GPF with 30 individuals for data set FD002.

Figure 5.9 shows six randomly selected trajectories in the test set, the true RUL, and the predicted values for the individuals in the Pareto front and the baseline models. Here, as opposed to in FD001, the quality of results varies much more between the different selected trajectories. For Trajectories 5 (Figure 5.9 a)), 46 (Figure 5.9 e)) and 92 (Figure 5.9 f)), the models predict the RUL quite accurately, especially towards the end of life. This is not true for Trajectories 18 (Figure 5.9 b)) and 28 (Figure 5.9 d)), for which the prognostic models are not able to accurately predict RUL. Note that the baseline algorithms (only RF and only SVM) are contained in the Pareto front. Therefore, it is no surprise

that their predictions' quality is relatively high compared to the other chosen settings of Pareto front individuals.

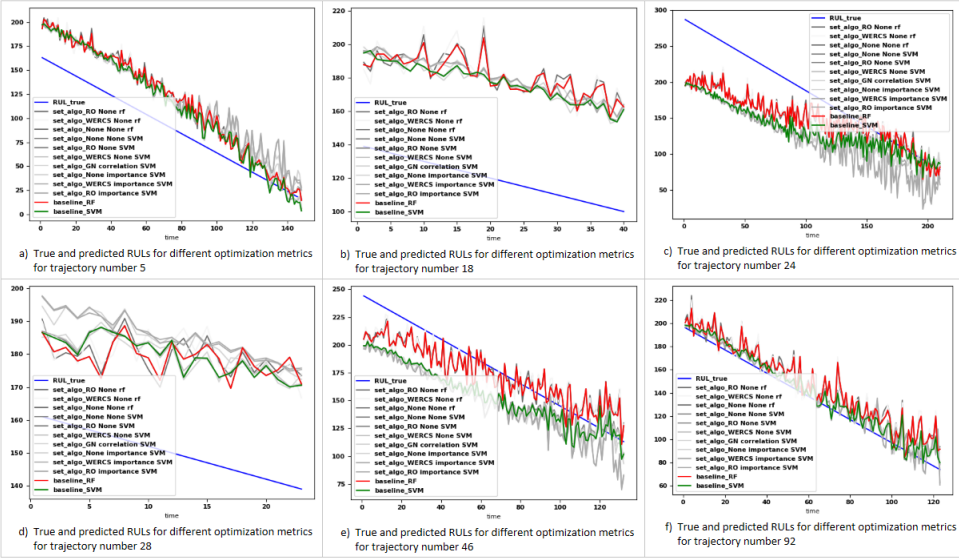


Figure 5.9: Predictions of best-found settings vs the two baseline scenarios (only RF and only SVM) on example trajectories for a population size of 30 on data set FD002.

5.3.2. CASE STUDY: AIRCRAFT SYSTEM DATA

In the second case study, the GPF is applied to an aircraft pump package installed close to the landing gear. The pump package consists of two redundant pumps: pump 1 and pump 2. The assumption is made that pump 1 and pump 2 failures are independent. Failures happen on the two power boards, presumably due to short circuits.

On each of the pumps, sensors have been installed that measure the following properties:

- The motor current,
- the motor speed,
- the motor temperature,
- the reservoir fluid level and
- the junction temperature (of the liquid).

Next to those sensors, the static air temperature (sat) and the calibrated airspeed (cas) reported on the aircraft level are used as input. The sensor measurements are made every second. The per-second data is aggregated per flight phase by mean, maximum, and minimum to remove noise from the raw sensor data. A flight consists of twelve flight

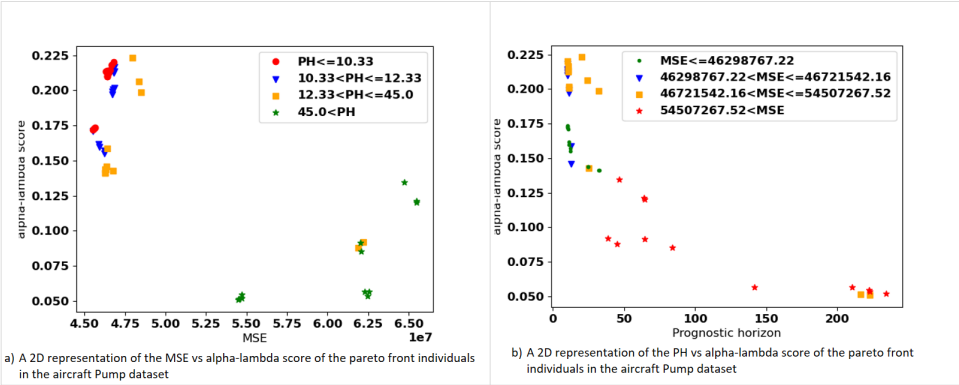
Table 5.4: The resulting best prognostic settings and metrics when running the MOGA GPF with 30 individuals for the aircraft Pump dataset.

rebalancing	**feature engineering**	**prognostic algorithm**	**MSE**	**PH**	$\overline{\alpha-\lambda}$
GN	None	rf	4,64E+07	29,94	0,1417
GN	correlation	rf	4,59E+07	11,25	0,1606
None	None	rf	4,63E+07	16,64	0,1507
None	PCA	rf	6,24E+07	191,61	0,0556
None	correlation	rf	4,56E+07	10,23	0,1725
None	importance	rf	5,46E+07	223,75	0,0523
RO	None	rf	4,85E+07	32,00	0,1985
WERCS	None	rf	4,82E+07	22,17	0,2147
WERCS	PCA	rf	6,52E+07	58,33	0,1253
WERCS	correlation	rf	4,66E+07	10,48	0,2104
WERCS	importance	rf	6,20E+07	57,88	0,0893

phases, from taxi-out until taxi-in. The aggregated data set contains around 35000 flight phases in total. Of those data, 10% are maintained in the test set, and the rest forms the train set.

The results for the runs on the Pump data set are presented in Table 5.4. It contains both the choices of methodologies for the three selected steps in prognostics and the according scores. Again, in this case, the Pareto front contains more individuals than the one of FD001. With its 11 individuals, the size is comparable to that of dataset FD002. As opposed to the simulated aircraft turbofan dataset, in this case, using RF results in better solutions than SVMs. In fact, no SVM solution is contained in the Pareto front. For the RF, almost every combination of rebalancing, feature engineering, and the prognostic algorithm is contained, resulting in very similar scores in terms of MSE. However, differences can be seen in terms of the other metrics. The PH ranges from 10.48 when using WERCS, correlation-based feature selection and RF to 223,75 when using no rebalancing, importance-based feature selection, and RF. The $\overline{\alpha-\lambda}$ ranges from 0.0523 when using the previous settings to 0.2104 when using WERCS, correlation-based feature selection, and RF.

Figure 5.10 shows a two-dimensional representation of individuals in the Pareto front and their according scores in relation to each other. In Figure 5.10 a), it can be seen that optimizing towards a low MSE simultaneously results in a lower PH but increases the $\overline{\alpha-\lambda}$ score. Figure 5.10 b) shows the link between the $\overline{\alpha-\lambda}$ score and PH in a clearer way: Increasing the PH at the same time decreases the $\overline{\alpha-\lambda}$ score. This can also be observed in Table 5.4: The highest scoring solution in terms of PH is also the lowest scoring in terms of $\overline{\alpha-\lambda}$ score.



a) A 2D representation of the MSE vs alpha-lambda score of the pareto front individuals in the aircraft Pump dataset

b) A 2D representation of the PH vs alpha-lambda score of the pareto front individuals in the aircraft Pump dataset

Figure 5.10: A 2D representation of the resulting scores for individuals in the Pareto front found when running the MOGA GPF with 30 individuals for the Pump data set.

5

5.4. DISCUSSION

The aim of the conducted case studies was to explore the two main research questions introduced in Section 5.1:

- What impact do metrics have on the choice of methodologies?
- How can the performance of prognostic methodologies be translated to an assessment of the system's suitability for prognostics?

Section 5.4.1 analyses the impact of metrics on the choice of prognostic methodologies. In Section 5.4.2 the system data of the two conducted case studies is analysed using the definition of data suitability given in Section 5.2.3. Finally, in Section 5.4.3 addresses the limitations of the presented study, and directions for further research are given.

5.4.1. THE IMPACT OF METRICS ON THE CHOICE OF PROGNOSTIC METHODOLOGIES

The results of applying the GPF to the simulated turbofan datasets FD001 and FD002 are presented in Section 5.3.1. When trying to understand the impact of the metrics on the choice of prognostic methodologies, it is of interest to take a closer look at both Tables 5.2 and 5.3 listing the chosen methodologies in the Pareto front and Figures 5.6 and 5.8 showing the links between the different metrics. It can be seen that using a different optimization metric can have an impact as big as a different choice of the prognostic algorithm used. For example, for FD001 in Table 5.2 we see that optimizing towards confidence results in using SVM for the prognostic model while optimizing towards correctness results in using RF for this purpose. In FD002 the dynamics are a bit different. Still, the underlying outcome is the same: When optimizing towards confidence, the GPF

chooses RF as the optimal prognostic algorithm while optimizing towards timeliness results in the GPF choosing SVM. Those dynamics are visualized in Figures 5.11 and 5.12. This can also be observed in the aircraft Pump data case study: In Table 5.4, we see that instead of in the choice of prognostic algorithm, the impact metrics have on the selection of techniques is reflected in the rebalancing and feature engineering settings. An example of this effect is given in the example in Section 5.3.2 for the selection the GPF makes to reach the highest PH or highest $\overline{\alpha - \lambda}$ score.

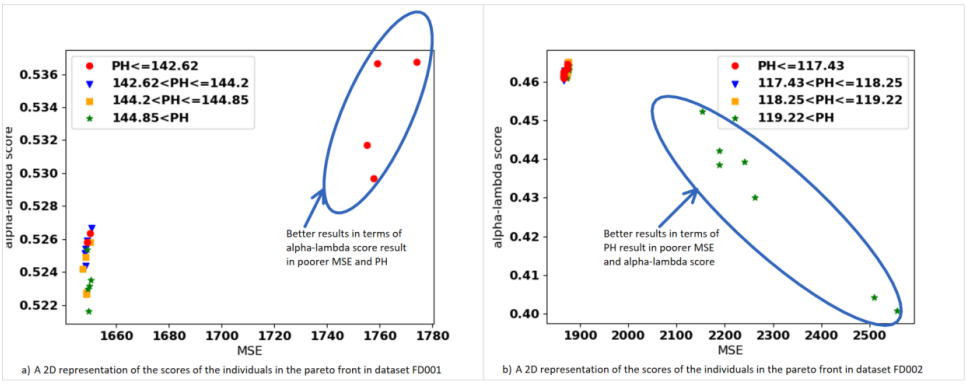


Figure 5.11: Comparison of the alpha-lambda score vs MSE in the 2D representation of the Pareto points for datasets FD001 and FD002.

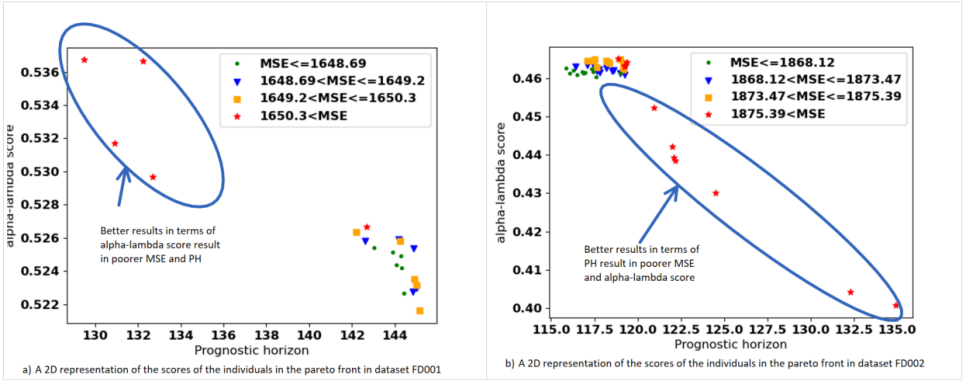


Figure 5.12: Comparison of the alpha-lambda score vs PH in the 2D representation of the Pareto points for datasets FD001 and FD002.

Therefore, increasing the performance in terms of a single metric comes at the cost of decreasing the performance in terms of another metric, i.e. the metrics do have an influence on the chosen prognostic settings. With the term 'prognostic settings' we refer to the combination of data rebalancing, feature engineering, and prognostic algorithm that is used to arrive at a prognostic model. This means that, when making choices for

prognostic methodologies, it is important to consider which metric to use for evaluation. To summarize, the following main points are raised:

- The choice of the optimal metric depends on the underlying data set and objective of prognostics, for example in what context they are used.
- A single metric is often not enough to make fully informed choices regarding which prognostic methodology to use.
- Optimizing towards different prediction attributes, i.e. correctness, timeliness, or confidence, results in different prognostic models and is often a trade-off.

5.4.2. EVALUATION OF THE SYSTEMS SUITABILITY FOR PROGNOSTICS

In this section, we answer the question of how to assess data suitability for prognostics using the GPF. This is achieved by applying the in Section 5.2.3 introduced definition and methodology in both the case studies to assess the according systems data suitability. Be aware that in the following, we provide a suggestion of how to set the boundaries, which is tailored to this case study. We put the focus on aircraft and look at it from the perspective of an MRO/ airline/ aircraft maintenance provider. Such a stakeholder uses the output of the prognostic models to plan and schedule maintenance tasks. Furthermore, we assume that the airline operates short-haul flights mainly with an average aircraft usage of 4 FC per day. As mentioned in Section 5.2.1, the assumption is made that a failure needs to be known at least 40 Flight cycles (FC) in advance (for the case studies on simulated turbofan engine data, we assume that FCs correspond to time cycles) in order to schedule maintenance. Based on this assumption, we set the following bounds for the criteria specified in Section 5.2.3:

- The upper MSE limit, $MSE_{max} = 2000$ FC.
- The minimum number of time steps before failure at which the failure needs to be known to take according to actions, $PH_{min}(a) = 40$ FC, which corresponds to 10 days of operation, with $a = 40$ FC.
- It is assumed that for this case it is sufficient that 45% of the predictions lie within the α bounds. Therefore, the minimum ratio of predictions within the α bounds, $(\alpha - \lambda)_{min} = 0.45$.

Based on the definition introduced in Section 5.2.3, we observe that for all solutions contained in the Pareto front for dataset FD001, the three conditions (Equations 5.8- 5.9) hold, i.e.:

$$MSE(t = \text{end of life}) \leq 2000 \tag{5.7}$$

$$\wedge\ PH(t_j) \geq 40\ \forall j \in p \text{ and } a = 40 \tag{5.8}$$

$$\wedge\ \overline{\alpha - \lambda} \geq 0.45. \tag{5.9}$$

And since all the conditions are fulfilled, according to the definition given in Section 5.2.3, dataset FD001 proves to be suitable for prognostics. Figure 5.7 underlines this

visually as it can be seen that the predicted value of almost all the models is close to the true RUL. For dataset FD002, Table 5.3 shows that there are three individuals in the Pareto front, satisfying all three above criteria (Equations 5.8-5.9). Those three individuals are the ones using RF as a prognostic algorithm, together with no rebalancing and feature engineering, together with random oversampling as a rebalancing methodology, and together with WERCS as a rebalancing technique respectively. The definition of data suitability in Section 5.2.3 states that only a single solution in the Pareto front is required to fulfill the conditions in order for the system data to be suitable for prognostics. As a result, also dataset FD002 turns out to be suitable for prognostics. Visually an indication of this can be seen in Figure 5.9.
For the aircraft pump dataset, however, both the MSE and the $\overline{\alpha - \lambda}$ score are too high, respectively too low for all solutions in the Pareto front. Therefore, the pump dataset is not suitable for prognostics according to the here presented definition.

5.4.3. LIMITATIONS AND FURTHER RESEARCH

The presented definition of data suitability is only dependent on prognostic metrics, meaning that the assessment of suitability is based merely on the quality of the prognostic model. Of course, as a stakeholder, one could be interested in metrics not just linked to the prognostic model itself. Therefore, a possible direction for further research would be to extend the data suitability assessment towards a more thorough assessment based on stakeholder needs. This could, for example, be to include a calculation of costs associated with wrong predictions. Depending on how 'wrong' the predictions are (in terms of selected prognostic metrics) this can then further be reflected in setting the thresholds for the data suitability assessment presented in Section 5.2.3.

In addition, the GPF only includes a limited set of methodologies and steps integrated into prognostics. Of course, those were carefully chosen to represent the most important groups of methodologies and be relatively simple, while still powerful. The framework could be extended to include more advanced methodologies, such as deep learning techniques or even diagnostic approaches.

The prognostic horizon (PH) used in the data suitability assessment depends on the parameter a, which we treat as a user constraint in this study and set to 40FC, representing the time needed to schedule and plan maintenance. In a further study, a range of values for a could be tested to see the effect on the prognostic model assessment. Such a sensitivity analysis could be conducted taking scheduling approaches into account, i.e., assessing a range of parameters and their effect on prognostic performance not only in terms of prognostic algorithms but also in terms of, e.g., costs for re-scheduling maintenance. Such an analysis would produce a more thorough assessment of the according values, model qualities, and implications for subsequent CBM use.
Finally, the user is required to specify boundaries for each metric. This can be a challenging task. A way to overcome this could be to implement, as mentioned above, a more thorough assessment, e.g., in terms of costs. Having said that, the approach presented here still goes beyond what has been done in literature so far, adding a novelty here. So far, as highlighted in Section 5.1, most studies regarding data suitability focused

merely on the system data and their structure and statistical properties. However, when using machine learning approaches, it can be the case that even without trends being visible in the system data, the models can detect or even predict anomalies (R. Liu et al., 2018). The approach presented here does not only provide an integrated way of assessing data suitability by taking into account prognostic machine learning algorithms. It also integrates metrics to capture the three aspects of prognostics namely correctness, timeliness, and confidence and thereby enables a more thorough assessment of the model quality.

5.5. CONCLUSION

The objective of the presented study is twofold: The first aim is to investigate the impact metrics have on prognostics. The second aim is to provide the means for a data suitability assessment for prognostics. To account not only for different prognostic algorithms but also for other steps involved in prognostics, such as data rebalancing and feature engineering, we use a generic prognostic framework that chooses the optimal settings for the three steps of data rebalancing, feature engineering, and prognostic algorithm. A multi-objective optimization is conducted to reflect a selection of metrics, which account for all the aspects of prediction evaluation, including correctness (MSE), timeliness (PH), and confidence ($\overline{\alpha-\lambda}$ score). The results show the following: First, the choice of optimization metric has a big impact on the output of the generic prognostic framework. This means that depending on the objective and motivation of using prognostics, a suitable metric should be carefully chosen. It can also make sense to use a combination of metrics to reflect multiple prediction evaluation aspects. Especially the Prognostic horizon can play an important role for airlines that want to schedule maintenance time and are dependent on predictions arriving early enough to schedule a corrective action. Therefore this should be taken into consideration when developing and evaluating prognostic methodologies. Second, the framework presented can be used together with a definition we provided to assess a system's suitability for prognostic based on the system data. All in all, this study both highlights the importance of choosing proper prognostic metrics and their impact on the prognostic outputs and gives directions for practitioners as to whether or not it makes sense to invest time and money in the development of prognostic systems based on the available system data.

6

CONCLUSION

The previous chapters guided through the development, use and deployment of a generic diagnostic and prognostic framework, which takes as an input aerospace system data and outputs diagnostic or prognostic models for the underlying system.

In this Chapter we draw an overall conclusion based on the findings in the previous chapters, provide answers to the research questions in Section 6.1, perform a compliance check to see if the presented GDPF is in alignment with the requirements (presented in Chapter 2) in Section 6.2, point out the limitations and indicate directions for further research in Section 6.3 and provide an overview over our contributions in Section 6.4.

6.1. RESEARCH QUESTIONS AND ANSWERS

In Chapter 1 we identified two gaps in existing literature with regards to diagnostic and prognostic approaches in aerospace applications. First, data suitability approaches are often focused on the data rather than on the data within a CBM framework. Second, many of the existing approaches are tailored to specific systems. Based on those gaps, the main research question is:

How can system data be used to assess the application of diagnostic and prognostic methodologies for failure detection and prediction?

From the main research question (RQ) the following sub questions are derived:

- **RQ 2: What are the requirements for a generic diagnostic and prognostic framework that is applicable to different components in various applications?**
- **RQ 3: In what way can anomaly detection methodologies be integrated into such an adaptive framework that, when applied to given system data, provides a diagnostic assessment?**

	RQ2	RQ3	RQ4	RQ5
Primary research activity	Define requirements for a generic diagnostic and prognostic framework using SE methods	Case study on satellite system data	Case study on AC system data	Sensitivity analysis with respect to different prognostic metrics
Contribution	The formal definition of a GDPF and its limitations and scope	A generic diagnostic framework	A generic prognostic framework	A data suitability assessment for prognostic solutions in a PHM context
Primarily addressed in	Chapter 2	Chapter 3	Chapter 4 and 5	Chapter 5

Table 6.1: Summary of research activities and contributions with regards to each research question in this dissertation

- **RQ 4: How can prognostic methodologies estimate a systems remaining useful life be integrated into such an adaptive framework that, when applied to given system data, it provides an assessment of the prognosability?**
- **RQ 5: How can the quality of prognostics on a system be evaluated within a CBM strategy?**

Table 6.1 summarizes how each of the research questions was addressed in this thesis, what underlying research has been conducted and where to find the according results.

Before we started developing a generic framework, in Chapter 2, we set according scopes and limitations for such a framework and defined underlying requirements in a systematic way. Note, that the identified stakeholder for the framework as presented in this thesis, is a developer, such as a data scientist, who can further use it as a data suitability assessment or as guidance for decisions on choices of diagnostic and prognostic techniques. The identified requirements can be found in Figure 6.1. In Section 6.2 we provide a summary of the requirements and go into detail on if and how the presented generic framework is compliant with the requirements.

In Chapter 3, the generic prognostic framework presented in Chapter 4 was extended to a generic diagnostic and prognostic framework (GDPF) and applied to real-life satellite data. Given the nature of the input data and the fact that most satellite systems have a very high reliability and very rarely fail or anomalies or failures occur rarely, we decided to focus specifically on anomaly detection. The framework contains different anomaly

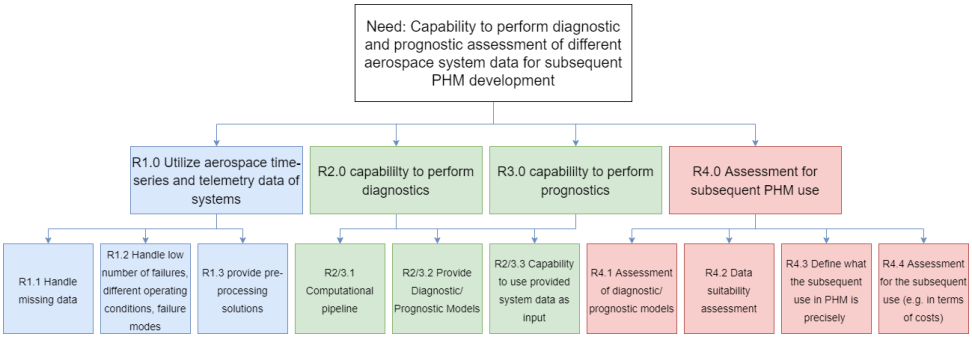


Figure 6.1: Functional Requirements for the Generic Diagnostic and Prognostic Framework.

detection methods together with data pre-processing and thresholding techniques and outputs -given system data as input- an anomaly detection model. The choice as to which methods are used to train the model is approached as a multi-objective optimization problem in which three different metrics are included to assess the models. This does not only make the framework more robust and adaptive towards multiple applications, but also enables its application for a wider range of users who can put emphasize or choose the metric which suits their use case best.

Research question 4 is very similar to RQ3 - the difference being that it puts the focus on prognostics instead on diagnostics/ anomaly detection. In Chapter 4 the GPF is introduced. It is verified and validated on simulated data sets and applied to both, simulated and commonly used data for prognostics and a real-life aircraft data set. The results showed the frameworks potential towards valid remaining useful life estimates for aircraft systems. Furthermore, two main challenges when using the GPF on real-life system data were identified: First, the much smaller number of failures leads to a smaller dataset, which makes predictions less stable and models hard to train and assess. Second including additional steps next to the prognostic algorithm, such as data pre-processing, data rebalancing or feature engineering can improve the quality of the predictions. This is true for all datasets, but we found that the impact of including such methodologies is much higher for real-life datasets compared to simulated data. We noticed that this point is closely linked to the first point and has to do with the stability of machine learning models when trained and assessed on smaller datasets. Still, or even more so, it is important to have the capability to assess system data towards prognostics. The framework provides exactly that: A means to quickly assess the ability to perform prognostics based on system data.

What we have not mentioned so far is the question of system data suitability or, as we refer to it in RQ4 "assessment of prognosability". While in Chapter 4 we presented the GPF, i.e. the tool itself, in Chapter 5 we highlighted the data suitability aspect. In a study presented in Chapter 5, we found that using only single-objective optimization for the choice of prognostic methodologies limits the ability of the framework to make proper decisions on the choice of algorithms and in addition, makes the framework less adap-

tive to different systems. Therefore, to provide a complete assessment, multiple metrics for prognostics were included in the framework, representing different aspects of predictions, namely correctness, timeliness and confidence. First, the impact of those metrics on the choice of prognostic methods (the output of the GPF) was studied. We found that the choice of metrics plays an important role in this and suiting metrics have to be chosen with care. In order to answer RQ5 (which is closely linked to RQ4), we provided a definition for a system's data suitability for prognostics based on system data. This definition was applied for a data suitability assessment of three different aircraft systems. The strength of the provided definition lies in the fact that it is embedded in the GPF and thereby is an evaluation of system prognosability within a CBM strategy. This is even more emphasized by integrating multiple metrics for the assessment of prognostic models.

6.2. REQUIREMENTS COMPLIANCE OF THE GENERIC DIAGNOSTIC AND PROGNOSTIC FRAMEWORK

Figure 6.1 gives an overview over the top-level functional requirements for the framework an the more detailed breakdown of those. In summary, the requirements cover the following aspects: System data, diagnostic capability, prognostic capability, assessment within a PHM/ CBM application.

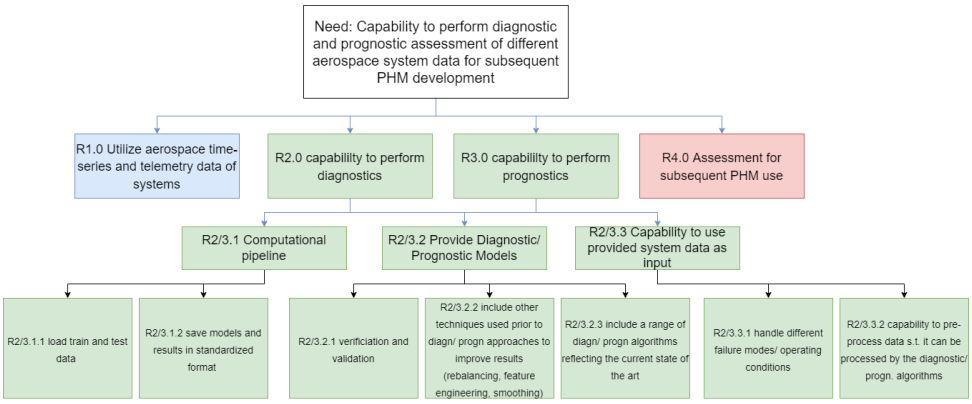


Figure 6.2: Detailed Functional Requirements for the Generic Diagnostic and Prognostic Framework.

In the following, we will look at each top level requirement and the derived sub-requirements separately, link them to the conducted studies, point out if and how they were met and how the presented framework is compliant with them.

Top level requirement R1.0 states that "The framework should be able to process time-series telemetry and sensor data related to aerospace systems health." It was broken down into three sub requirements as presented in Figure 6.3. Table 6.2 summarizes how each of the requirements was met or not and how both, the diagnostic and the prognos-

Req. #	Requirement	Diagnostics	Prognostics	Overall
R1.1	handle missing/incomplete or incorrect data	not addressed	missing data removed	no overall strategy to handle missing data
R1.2	handle imbalanced data	through anomaly detection	multiple methods included to handle imbalanced data	overall addressed in the GDPF
R1.3	handle multivariate time series data	methods included in the framework can handle multi-variate and multi-dimensional data as an input		

Table 6.2: Requirements related to top level requirement 1.0 and compliance check.

tic part of the framework are compliant with each requirement.

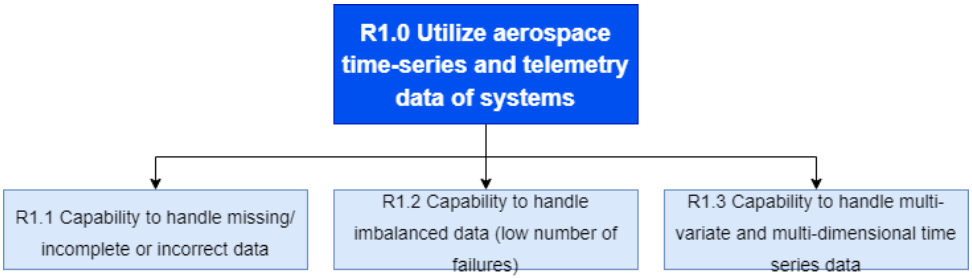


Figure 6.3: Requirements derived from top level requirement 1.0 for the Generic Diagnostic and Prognostic Framework.

Requirement R1.1 requires the framework to "include solutions to identify and pre-process missing, incomplete or incorrect data". Such solutions have overall not been included in the GDPF. However, in Chapter 4, for the case studies conducted, missing data was simply removed. The reason for this was that no solutions to handle missing data were included in the framework at that stage and apart from the in the framework included data pre-processing methods, the aim was to leave the data as untouched as possible.

Requirement R1.2 concerns the capability to handle imbalanced data, which is especially important for aerospace systems, which are designed to be reliable resulting in a relatively small number of failures or anomalies. Anomaly detection methods themselves are a way to handle imbalanced data - the input data set is usually expected to only contain a small number of failures. Accordingly, in the case studies conducted in Chapter 3, the majority of the data was healthy data and the methods are tuned to be

trained on healthy data only. For prognostics, the situation is different. Using imbalanced data to train a prognostic model can result in either an overfitted or underfitted model and lower model accuracy. Therefore, in the prognostic part of the framework multiple methods are included to handle imbalanced data. More details on this can be found in Section 4.2.3 in Chapter 4.

Finally, Requirement R1.3 declares that "the framework needs to be able to handle and pre-process time-series data containing multiple features and variables that are continuously measured over time." This requirement is met as all the machine learning algorithms used in the framework are capable of handling such data. We particularly put our focus on methods suitable for time-series and all the input data used in the case studies are multi-variate and multi-dimensional time-series data.

Top level requirements R2.0 and R3.0 state that "The framework should provide the capability to perform diagnostics/ prognostics." We will handle the two blocks in parallel since they are related to each other and the derived sub requirements are in essence the same. Again there are three sub requirements, which are further broken down to include a more detailed list of requirements as presented in Figure 6.4. Table 6.3 gives an overview over the compliance of the requirements with regards to the diagnostics and prognostics part of the GDPF and how they were overall met.

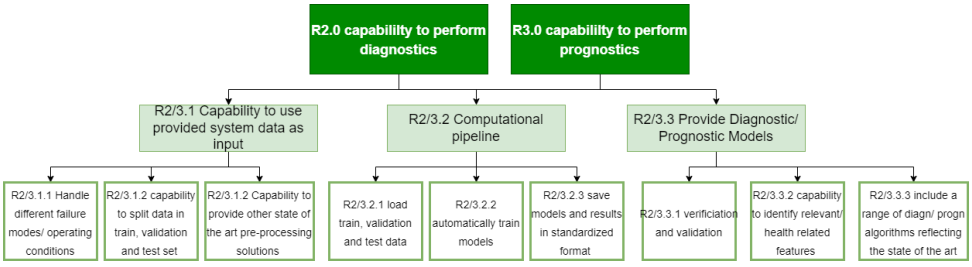


Figure 6.4: Requirements derived from top level requirements 2.0 and 3.0 for the Generic Diagnostic and Prognostic Framework.

Requirement 2/3.1 says that "The framework needs to be capable to pre-process the provided system data in a way that it can further be used by the diagnostic and prognostic models contained in the framework." It is further divided in the following requirements:

- **R2/3.1.1 The framework is able to handle different failure modes and operating conditions:** This point has not been addressed yet, the framework neither differs between failure modes nor between operating conditions. However, it indicates also a direction for further research as it is an addition to such a framework that has great potential for improvement of the diagnostic/ prognostic methods.

- **R2/3.1.2 It needs to be capable to split the data in train, validation and test set:** The framework is capable to split data in train, validation and test sets. This is

Req. #	Requirement	Diagnostics	Prognostics	Overall
R2/3.1.1	handle different failure modes and operating conditions	not addressed yet		
R2/3.1.2	split the data in train, validation and test set	framework is capable to split data in train, validation and test sets		
R2/3.1.3	provide and include other state of the art pre-processing solutions	normalization and standardization	only data rebalancing techniques included	should be addressed in further development step of the framework
R2/3.2.1	automatically load train, test and validation data	the GDPF is able to automatically load train, test and validation data		
R2/3.2.2	automatically train the diagnostic and prognostic models	the GDPF automatically trains diagnostics and prognostics models together with the chosen techniques		
R2/3.2.3	save the models and results in a standardized format	models saved in a standardized way		
R2/3.3.1	provide a verification and validation solution	unit test, several additional verification tests and validation done against existing algorithms on standard data sets		overall implemented, but new/ additional methods need to be verified and validated
R2/3.3.2	identify relevant/ health related features	choice of ML methods	feature engineering methods	overall suiting methods included in the framework
R2/3.3.3	range of diagnostic/ prognostic algorithms reflecting the state of the art	only anomaly detection methods included and there only ML	only ML based methods included	framework could be extended to include a wider range of algorithms

Table 6.3: Requirements related to top-level requirement 2/3.0 and compliance check.

done as a separate step before the optimization algorithm is applied to find the optimal set of methodologies.

- **R2/3.1.3 It needs to provide and include other state of the art pre-processing solutions:** For the anomaly detection methods, data standardization and normalization are used as standard data pre-processing techniques for machine learning. For prognostics, other than the re-balancing techniques, no pre-processing solutions for prognostics are included yet. However, this could be addressed in a further development step of the framework, considering the vast amount of existing pre-processing techniques for time series data.

Requirement R2/3.2 affirms that "the framework needs to perform the required functionalities in an automated way" and is further split into the following requirements:

- **R2/3.2.1 it needs to be able to automatically load train, test and validation data:** The framework is capable to automatically load train, test and validation data.
- **R2/3.2.2 automatically train the diagnostic and prognostic models:** The framework automatically trains anomaly detection and prognostic models. It builds a pipeline consisting of the chosen methods and than automatically performs all the steps, from loading train, test and validation data until saving the trained model.
- **R2/3.2.3 save the models and results in a standardized format, so that in a further step the evaluation metrics can be automatically calculated and extracted:** The framework saves the models as '.json' files, which is a standardized format to save machine learning models.

Requirement R2/3.3 says that "the framework needs to be able to output diagnostic/ prognostic models." The according sub requirements are:

- **R2/3.3.1 It should provide a verification and validation solution for the diagnostic and prognostic models:** For both, diagnostic and prognostic methodologies and according data pre-processing methods as well as feature engineering methodologies included in the framework, the following verification and validation procedures are included: Unit tests are implemented, several tests for the verification process are conducted and the methods are validated against existing algorithms (on standard data sets). So overall, verification and validation procedures are included, but whenever a new method should be included in the framework it has to be verified and validated separately as well as within the framework.
- **R2/3.3.2 It needs to be capable to identify relevant/ health related features:** For the anomaly detection part of the framework this is done by including suiting machine learning methods (such as PCA). In the prognostics part of the framework feature engineering methods are included for this purpose as presented in Chapter 4, Section 4.2.3.
- **R2/3.3.3 The framework should include a range of diagnostic/ prognostic algorithms reflecting the state of the art and different model types (such as machine learning based methods, or statistical algorithms):** In case of diagnostics only

anomaly detection methods included. However, anomaly detection covers only a part of diagnostics. In addition, only machine learning based methods are represented for both, diagnostics and prognostics approaches. The advantage of those is that they are easily adaptable and do not take a long time to train. All in all, it could be a step in further research to extend the framework to include a wider range of algorithms, both for diagnostics and prognostics.

Finally, in Figure 6.5, the requirements derived from top level requirement R4.0, related to the assessment within a PHM framework, are shown. Table 6.4 gives an overview over the compliance of the requirements.

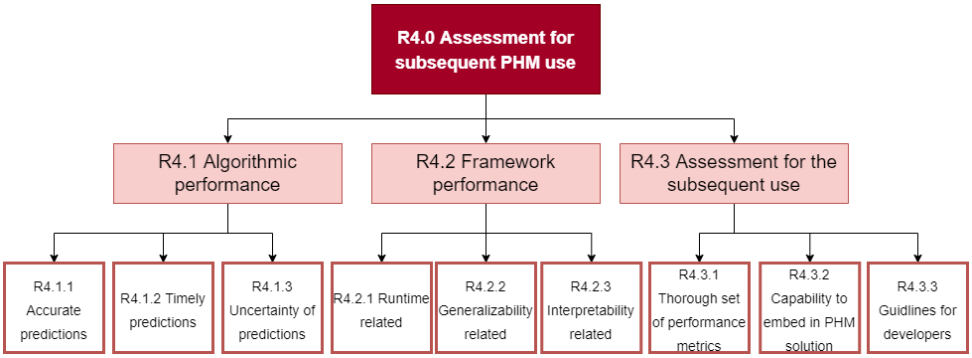


Figure 6.5: Detailed list of requirements derived from top level requirement 4.0 for the Generic Diagnostic and Prognostic Framework.

Requirement R4.1 is related to the algorithmic performance and states that "the framework needs to be able to assess the diagnostic and prognostic model performance in a comprehensive and suiting manner". To make this clearer it is split into the following requirements:

- **R4.1.1 It needs to be able to measure and output model accuracy:** The model accuracy is measured using three different metrics (F1, F1pa and FC score) in case of anomaly detection methods and using the MSE for prognostic techniques. So overall the model accuracy is captured in the according metrics.

- **R4.1.2 It needs to be able to assess whether the predictions are produced far enough in advance for subsequent use in PHM applications (e.g. for scheduling maintenance):** This point only effects the prognostics part of the framework, as diagnostic models do not predict anything. For the prognostics, the prognostic horizon is used as metric, which gives an indication of how far in advance the prediction is reliable. The metric can further be adapted within the framework to reflect the time needed to plan maintenance, i.e. it even includes a user parameter that can be set individually- depending on the underlying application.

- **R4.1.3 The framework should measure and return the prediction uncertainty:**

Req. #	Requirement	Diagnostics	Prognostics	Overall
R4.1.1	measure and output model accuracy	measured using F1 score, F1pa and FC score	measured in MSE	overall model accuracy captured in metrics
R4.1.2	predictions timely for PHM action	-	PH used as metric	PH can be adapted to reflect time needed to plan maintenance
R4.1.3	prediction uncertainty	-	$\alpha - \lambda$ metric used	uncertainty measured through $\alpha - \lambda$ metric
R4.2.1	run within specified time limit	no such check implemented so far		next step: track running times of entire framework and separate methods
R4.2.2	generalizability, adaptive, generic and robust	methods used overall adaptive and robust		generic hard to get (consider the scale of the problem)
R4.2.3	outputs interpretable to be used within PHM setting	anomaly detection outputs straightforward	figures produced	more advanced methods could be used for this purpose
R4.3.1	set of performance metrics	as discussed above, F1, F1pa and FC score	MSE, PH and alpha lambda metric	overall several performance metrics for both, diagn and progn included
R4.3.2	embedded in PHM solution	capability to both, load data from various formats and save the models in a standardized format makes this possible		
R4.3.3	guidelines for further model development and use	not done yet	done, see Chapter 5	overall, done for prognostics, could be extended for diagnostics

Table 6.4: Requirements related to top level requirement 4.0 and compliance check.

Again, this point only effects the prognostics part of the framework, for which the uncertainty is captured through the $\alpha - \lambda$ metric (see Chapter 5, Section 5.2.1).

Requirement R4.2 states that "the framework should meet performance requirements, which are mostly linked to constraints set by the stakeholder". Examples for such requirements are

- **R4.2.1 The framework needs to run within a specified time limit:** So far, no run time tracker is implemented in the framework. As a next step, the run time of the entire framework as well as run times for separate methods could be tracked and used as user input or set as requirement.
- **R4.2.2 The framework needs to be generalizable, i.e. it needs to be adaptive (applicable to various aerospace systems), generic (include a thorough set of methodologies) and robust (produce reliable and consistent results):** In general, the choice of methods included in the framework was based on this requirement: For anomaly detection as well as for prognostics machine learning methods which are well known, can easily be adapted and are relatively robust towards hyper parameter settings were chosen. Therefore, overall the framework is compliant with this requirement. However, as we pointed out in the Introduction (Chapter 1) as well as the introductions of Chapters 3 and 4, creating a truly "generic" framework is an impossible task, especially when considering the scale of the problem.
- **R4.2.3 The outputs of the framework need to be interpretable to be used within a PHM setting:** For the anomaly detection methods the interpretation of the outputs is relatively straight forward. For the prognostic methodologies, figures are produced by the framework which can help in interpreting the results. However, there is still work to be done in this regards and several more advanced methods have been developed in the past years, which could help towards the interpretability of the outputs of the framework. One example for such an approach are TrajecNets, suggested by (Shahid & Ghosh, 2019), making use of RNN based autoencoders to create trajectories, which represent the evolution of data from healthy to failure states. Baptista et al. (M. L. Baptista et al., 2022) make use of the SHAP (SHapley Additive exPlanations) model to show that prognostics metrics correlate with the SHAP model's explanations. Another commonly used technique for explainable AI, LIME (Local Interpretable Model-agnostic Explanations) is applied by Protopapadakis et al. (Protopapadakis et al., 2022) to estimate the RUL of an aeroengine and interpreted the results of DNN in an explainable way.

Requirement R2/3.3 says that "the framework should provide an assessment of the system data with regards to whether or not they are suitable for diagnostics or prognostics in PHM". The according sub-requirements are:

- **R4.3.1 It therefore needs to include a thorough set of performance metrics:** As already pointed out above, for the anomaly detection methods, three metrics are used to assess performance, namely F1, F1pa and FC score. For the prognostic methods MSE, PH and $\alpha - \lambda$ score are used. So overall, several performance metrics are included for both, diagnostics and prognostic approaches.

- **R4.3.2 It should be capable to be embedded in a PHM solution:** Because the framework is capable to load data in an automated way and saves models in a standardized format, it should be possible to embedd it within a PHM solution. However, it could be a direction for further research to actually test the framework within a PHM solution.

- **R4.3.3 It should provide guidelines for further diagnostic/ prognostic model development and deployment:** Guidelines for further model development are provided through a data suitability analysis as presented in Chapter 5. Those guidelines are obtained from the prognostics framework in particular and could as a next step be extended to include diagnostics methods as well.

6.3. LIMITATIONS AND FURTHER RESEARCH

The presented framework has three main limitations, all of those raising directions for further research. First, limitations on the range of algorithms and methodologies included in the framework. Second, data related limitations, such as the assumption that failure or anomaly related data is available. Third, limitations related to the assessment of the models within a PHM framework. The following paragraphs give an overview over each of the three limitations and point out in what way they could be addressed in further research.

The task we set ourselves of creating a truly generic framework is challenging in multiple ways. For one, the amount of available machine learning methods is big and growing, as it is still a widely researched field. Even when only considering methods used for systems diagnostics or prognostics, as we pointed out in the Introduction, Chapter 1, the scale of the problem is massive. When looking at the amount of available diagnostic algorithms (see Chapter 3) or prognostic algorithms (see Chapter 4), it becomes clear that choices as to which methods are represented in a generic framework have to be made. This is not necessarily a limitation: The aim of the framework is to provide a relatively quick first diagnostic or prognostic assessment. Including a huge amount of machine learning methods in the optimization problem the framework solves, does not help it to arrive at an assessment quickly. Furthermore, we argue that for a first assessment simple and instead more adaptive methods are more effective. Having said this, there are still limitations related to the "generic" in the GDPF. Most of those were already identified in the previous Section 6.2. Related to requirement R1.1, the framework is not capable of dealing with missing or incomplete data in an automated way. A next step in the development of the framework could therefore be to include techniques for this purpose. Requirement 2/3.1.1 showed that the framework is does not include methodologies to handle different failure modes and operating conditions. However, in aerospace applications, in which systems are operated under extreme environments and in various conditions, such a distinction is expected to improve the quality of underlying diagnostic and prognostic models. Therefore, a future research direction could be to identify suiting methodologies, to include those in the framework and to assess their impact on the resulting models. Requirement 2/3.1.3 revealed that, while there are several data preprocessing methods implemented in the GDPF, they do not cover the full range of possi-

ble methods and more techniques could be included in further framework development. Finally, the diagnostic and prognostic algorithms included in the framework do not yet- as stated in requirement R2/3.3.3 - truly "reflect the state of the art". For example, no statistical and no deep learning based methodologies, as well as no denoising, health indicator construction or elbow/knee point indication, were included in the framework. The drawback would be that such methods tend to be less tune-able and adaptive, however, it is worth looking into this direction of framework advancement. In addition, for the diagnostics part of the framework, only anomaly detection methods were considered, which -as fault detection methods are only part of a complete diagnostics solution. Techniques to represent fault isolation and identification could be included in a further step.

Next to the challenge of making the framework more generic, there remains the challenge of having an "adaptive" framework. By adaptive we refer to the capability of applying the framework to different aerospace system data sets. Again, we demonstrated the adaptivity of the GDPF in various case studies for both satellite (see Chapter 3) and aircraft systems (Chapter 4 and 5). So to a certain degree we proved it to be adjustable towards different aerospace systems. Nonetheless, there are certain requirements towards input data and assumptions made. For the diagnostic part of the framework, e.g. we assumed that even though most of the data is related to healthy system behaviour that anomalies are available - if only for validation and testing purposes. This is a strict requirement, especially when working with systems which are supposed to have an availability of around 99.9%. Furthermore, it limits the application of the framework to systems where historical data are available. For the prognostics part of the framework, the assumptions were even stricter, see Section 4.2.1. The systems are assumed to be operated until failure, the data is assumed to be labelled and the data represents all phases of operation, to give a few examples. Again, while it is challenging to meet those assumptions, this also opens the door to new research directions: For example, one could consider using federated learning, a methodology leaving training data distributed at the original location and learning a shared model through a central server by aggregating locally computed results (Konečný et al., 2016). At the moment, the value of federated learning is still being explored and infrastructures to enable it are created. For applications within the aerospace, or perhaps especially the airline industry this might be a way to both, ensure that data remain with the operator and do not have to be shared with other operators, but also train models with data from multiple operators to improve the amount of failures contained in the data set and in the end improve model quality. Another question one could ask is: Is it possible to transfer existing solutions from one system to another? Or how similar do systems have to be in order to be used as input data to train the same diagnostic or prognostic underlying models? Such a development could have the potential to enable diagnostics or prognostics for systems experiencing only a low number of failures. Over the past view years there has been a growing interest in "digital twins", a concept with the underlying idea that each aircraft or satellite has a digital twin, in which sensor values, operational data and all digitally available information is represented and used, e.g. for maintenance purposes. This raises the question of system inter dependability and how it should be represented in diagnostic or prognostic

models. Another aspect of the adaptivity of the framework is the extend to which it is extendable towards novel methods and techniques. This does not only include emerging techniques for already existing steps included in the framework but also an extension towards generalizable sub steps, such as health indicator construction or the identification of an elbow/knee point.

Finally, the central question of this thesis was the question of how to assess the application of diagnostic and prognostic methods based on system data, or in other words, how system data can be translated into a meaningful diagnostic or prognostic assessment. We already pointed out how we provided an answer to this question through the framework and a data suitability assessment as presented in Chapter 5. Still, the aim is also to integrate such an assessment within a PHM framework, i.e. to look beyond the diagnostic or prognostic models themselves. We addressed this in part by providing a data suitability assessment based on diagnostic and prognostic algorithms instead of statistical properties of the data and by integrating multiple diagnostic and prognostic metrics in the framework for the model assessment. What we did not do yet, though, is to assess models within a PHM framework. Requirement R4.3.2, which asks for a framework embedded in a PHM solution goes a bit into this direction. What impact, e.g. does a lower diagnostic or prognostic model quality have on maintenance scheduling/ planning? A first study on this has been conducted (see (Tseremoglou et al., 2022)) in which we studied the question of how uncertainty in prognostic outputs translates back to the scheduling within a PHM framework when applied to an airline. Integrating the GDPF within a PHM framework would also enable a more thorough assessment from different stakeholders perspectives, such as cost-based assessments. This is also linked to Requirement R4.2.1, which asks for the framework to run within a specified time limit. The positive thing is that, as we pointed out in the previous Section 6.2, the adaptive nature of the framework and the way it is built makes it ready to be integrated within a PHM solution.

6.4. CONTRIBUTIONS

Before we start summarizing the contributions of the conducted research, we refer back to Table 6.1 in which the research questions asked as a basis for the thesis are linked to contributions and the respective chapters in which those can be found. In the Introduction, Chapter 1 in Section 1.3.3, we listed five main contributions. In the following, we link each of those contributions with the results and main findings of the conducted research. Furthermore, we mention bigger implications of those contributions and point out their societal impacts.

As the first contribution, a set of requirements for a generic diagnostic and prognostic framework is systematically derived and presented. Such a set of requirements can be used as a formal baseline for the further development and implementation of diagnostic or prognostic frameworks.

Second, a generic diagnostic and prognostic framework is developed within the set scope and based on the defined requirements. Not only can such a framework provide guide-

lines as to where further development should go to, but it also can help in understanding if we are able to provide diagnostic or prognostic models at all. In further consequence this can help airlines and satellite operators in the application of data-driven diagnostic and prognostic methodologies and in this setting in the decision making process. It can guide in the choice as to which systems to include in a PHM/ CBM solution, which system data are suitable to train machine learning based models and for which systems we do not have sufficient or suiting data available yet. It also helps understand whether we are in principle able to capture anomalies or failures within the available data.

Third the framework is adapted and used on various system data, including simulated and real data, fault-related and component run-to-failure data, and satellite and aircraft systems case studies. The case studies not only helped in the validation of the framework but also brought to light the challenges faced with when using real-life data: Among those are the data labelling, the low number of failures/ anomalies leading to less stable predictions, the big differences of performance that can be seen when including additional steps in building diagnostic or prognostic models, such as data pre-processing techniques or the impact of the choice of underlying evaluation metrics.

The fourth contribution is closely linked to the third and lies in the comparison of results not just over different systems, but also over different applications (such as aircraft systems and satellite systems). The value and limitations of the generalizability of the framework are underlined, giving further insight into the challenges of diagnostic and prognostic methodologies, especially when applying them to real system data.

Fifth, a system data suitability assessment is presented based on the generic framework and tested on real-life aircraft data sets. It was found that the definition of whether or not data is suitable for prognostics highly depends on the chosen metrics used to evaluate the resulting models. Therefore, those metrics should be chosen with care, depending on the application, the system and especially how the model output is subsequently used within the PHM solution.

List of Figures

List of Tables

REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. http://arxiv.org/abs/1907.10902

Alam, M. R., Gerostathopoulos, I., Prehofer, C., Attanasi, A., & Bures, T. (2019). A framework for tunable anomaly detection. *Proceedings - 2019 IEEE International Conference on Software Architecture, ICSA 2019*, 201–210. https://doi.org/10.1109/ICSA.2019.00029

Amigó, E., Gonzalo, J., Artiles, J., & Verdejo, F. (2011). Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks. *Journal of Artificial Intelligence Research*, *42*, 689–718. https://doi.org/10.1613/jair.3401

An, D., Kim, N. H., & Choi, J. H. (2015). Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering and System Safety*, *133*, 223–236. https://doi.org/10.1016/j.ress.2014.09.014

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. *Principles of Data Mining and Knowledge Discovery: 6th European Conference, PKDD 2002 Helsinki, Finland, 2431 LNAI*(Proceedings 6), 15–27. https://doi.org/10.1007/3-540-45681-3{_}2

Archana, M., Pawar, M. S. S., & Prof, A. (2015). Periodicity Detection of Outlier Sequences Using Constraint Based Pattern Tree with MAD. *International Journal of Advanced Studies in Computers, Science and Engineering*, *4*(6), 34.

Atamuradov, V., Medjaher, K., Camci, F., Zerhouni, N., Dersin, P., & Lamoureux, B. (2020). Machine Health Indicator Construction Framework for Failure Diagnostics and Prognostics. *Journal of Signal Processing Systems*, *92*(6), 591–609. https://doi.org/10.1007/s11265-019-01491-4

Babu, G. S., Zhao, P., & Li, X. L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9642*, 214–228. https://doi.org/10.1007/978-3-319-32025-0{_}14

Bagdonavicius, V., & Petkevicius, L. (2019). Multiple outlier detection tests for parametric models. https://doi.org/10.3390/math8122156

Bagdonavičius, V., & Petkevičius, L. (2020). Multiple outlier detection tests for parametric models. *Mathematics*, *8*(12), 1–23. https://doi.org/10.3390/math8122156

Baptista, M., Henriques, E. M. P., de Medeiros, I. P. P., Malere, J. P. P., Nascimento, C. L. L., & Prendinger, H. (2019). Remaining useful life estimation in aeronautics: Combining data-driven and Kalman filtering. *Reliability Engineering and System Safety*, *184*(April), 228–239. https://doi.org/10.1016/j.ress.2018.01.017

Baptista, M., Nascimento, C. L., Prendinger, H., & Henriques, E. (2017). A case for the use of data-driven methods in gas turbine prognostics. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 441–452.

Baptista, M. L., Goebel, K., & Henriques, E. M. (2022). Relation between prognostics predictor evaluation metrics and local interpretability SHAP values. *Artificial Intelligence*, *306*, 103667. https://doi.org/10.1016/j.artint.2022.103667

Baruah, P., Chinnam, R. B., & Filev, D. (2006). An autonomous diagnostics and prognostics framework for condition-based maintenance. *IEEE International Conference on Neural Networks - Conference Proceedings*, 3428–3435. https://doi.org/10.1109/ijcnn.2006.247346

Basora, L., Bry, P., Olive, X., & Freeman, F. (2021a). Aircraft Fleet Health Monitoring using Anomaly Detection Techniques, 1–26.

Basora, L., Bry, P., Olive, X., & Freeman, F. (2021b). Aircraft fleet health monitoring with anomaly detection techniques. *Aerospace*, *8*(4), 1–33. https://doi.org/10.3390/aerospace8040103

Basora, L., Olive, X., & Dubot, T. (2019). Recent advances in anomaly detection methods applied to aviation. *Aerospace*, *6*(11). https://doi.org/10.3390/aerospace6110117

Bi, S., Prabhu, S., Cogan, S., & Atamturktur, S. (2017). Uncertainty quantification metrics with varying statistical information in model calibration and validation. *AIAA Journal*, *55*(10), 3570–3583. https://doi.org/10.2514/1.J055733

Bieber, M., Verhagen, W. J. C., Cosson, F., & Santos, B. F. (2023). Generic Diagnostic Framework for Anomaly Detection — Application in Satellite and Spacecraft Systems. *Aerospace*, *10*(8), 1–24.

Bieber, M., Verhagen, W. J. C., & Santos, B. F. (2021). An Adaptive Framework For Remaining Useful Life Predictions Of Aircraft Systems. *European Conference of the Prognostics and Health Management Society*, 60–70.

Bieber, M., & Verhagen, W. J. (2022). A Generic Framework for Prognostics of Complex Systems. *Aerospace*, *9*(12), 1–27. https://doi.org/10.3390/aerospace9120839

Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021). Uncertainty-aware Remaining Useful Life predictor. *arXiv preprint arXiv:2104.03613*, 1–14. http://arxiv.org/abs/2104.03613

Blanchard, B. S., Fabrycky, W. J., & Fabrycky, W. J. (1990). *Systems engineering and analysis*.

Braglia, M., Carmignani, G., Frosolini, M., & Zammori, F. (2012). Data classification and MTBF prediction with a multivariate analysis approach. *Reliability Engineering and System Safety*, *97*(1), 27–35. https://doi.org/10.1016/j.ress.2011.09.010

Branco, P., Torgo, L., & Ribeiro, R. P. (2019). Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing*, *343*(40), 76–99. https://doi.org/10.1016/j.neucom.2018.11.100

Breiman, L. (2001). Random forests. *Machine learning*, 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24*(421), 123–140. https://doi.org/10.1007/BF00058655

Broer, A. A., Benedictus, R., & Zarouchas, D. (2022). The Need for Multi-Sensor Data Fusion in Structural Health Monitoring of Composite Aircraft Structures. *Aerospace*, *9*(4), 1–26. https://doi.org/10.3390/aerospace9040183

Brown, E. R., McCollom, N. N., Moore, E. E., & Hess, A. (2007). Prognostics and health management a data-driven approach to supporting the F-35 lightning II. *IEEE Aerospace Conference Proceedings*, 1–12. https://doi.org/10.1109/AERO.2007.352833

Brownjohn, J., de Stefano, A., Xu, Y.-L., Wenzel, H., & Aktan, A. E. (2011). Vibration-based monitoring of civil infrastructure: challenges and successes. *Journal of Civil Structural Health Monitoring*, *1*(3-4), 79–95. https://doi.org/https://doi.org/10.1007/s13349-011-0009-5

Brunton, S. L., Kutz, J. N., Manohar, K., Aravkin, A. Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A. W., Hogan, T., & McDonald, D. (2021). Data-driven aerospace engineering: Reframing the industry with machine learning. *AIAA Journal*, *59*(8), 2820–2847. https://doi.org/10.2514/1.J060131

Calikus, E., Nowaczyk, S., Sant'Anna, A., & Dikmen, O. (2020). No free lunch but a cheaper supper: A general framework for streaming anomaly detection. *Expert Systems with Applications*, *155*. https://doi.org/10.1016/j.eswa.2020.113453

Challu, C., Jiang, P., Wu, Y. N., & Callot, L. (2022). Deep Generative model with Hierarchical Latent Factors for Time Series Anomaly Detection. *151*. http://arxiv.org/abs/2202.07586

Chandola, V., Banerjee, A., & Kumar, V. (2007). Outlier detection: A survey. *ACM Computing Surveys*, *14*, 15.

Chen, J., Pi, D., Wu, Z., Zhao, X., Pan, Y., & Zhang, Q. (2021). Imbalanced satellite telemetry data anomaly detection model based on Bayesian LSTM. *Acta Astronautica*, *180*(December 2020), 232–242. https://doi.org/10.1016/j.actaastro.2020.12.012

Chen, Y., Zhu, F., & Lee, J. (2013). Data quality evaluation and improvement for prognostic modeling using visual assessment based data partitioning method. *Computers in Industry*, *64*(3), 214–225. https://doi.org/10.1016/j.compind.2012.10.005

Chen, Z., Zhou, D., Zio, E., Xia, T., & Pan, E. (2023). Adaptive transfer learning for multimode process monitoring and unsupervised anomaly detection in steam turbines. *Reliability Engineering and System Safety*, *234*(May 2022), 109162. https://doi.org/10.1016/j.ress.2023.109162

Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*.

Coble, J. (2010). Merging Data Sources to Predict Remaining Useful Life – An Automated Method to Identify Prognostic Parameters. *Dissertation*. https://trace.tennessee.edu/utk_graddiss/683

Coble, J., & Wesley Hines, J. (2009). Identifying optimal prognostic parameters from data: A genetic algorithms approach. *Annual Conference of the Prognostics and Health Management Society, PHM 2009*, 1–11.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

de Pater, I., & Mitici, M. (2021). Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. *Reliability Engineering and System Safety*, *214*. https://doi.org/10.1016/j.ress.2021.107761

Diez-Olivan, A., Del Ser, J., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Information Fusion*, *50*(July 2018), 92–111. https://doi.org/10.1016/j.inffus.2018.10.005

Downey, A., Lui, Y. H., Hu, C., Laflamme, S., & Hu, S. (2019). Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds. *Reliability Engineering and System Safety*, *182*(October 2018), 1–12. https://doi.org/10.1016/j.ress.2018.09.018

Elattar, H. M., Elminir, H. K., & Riad, A. M. (2016). Prognostics: a literature review. *Complex & Intelligent Systems*, *2*(2), 125–154.

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

Frederick, D. K., DeCastro, J. A., & Litt, J. S. (2007). User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS). *No. E-16205*.

Freeman, C., Merriman, J., Beaver, I., & Mueen, A. (2022). Experimental Comparison and Survey of Twelve Time Series Anomaly Detection Algorithms (Extended Abstract). *IJCAI International Joint Conference on Artificial Intelligence*, *72*, 5737–5741. https://doi.org/10.24963/ijcai.2022/801

Fuertes, S., Picart, G., Tourneret, J. Y., Chaari, L., Ferrari, A., & Richard, C. (2016). Improving spacecraft health monitoring with automatic anomaly detection techniques. *14th International Conference on Space Operations, 2016*, (May), 1–16. https://doi.org/10.2514/6.2016-2430

Gado, J. E., Beckham, G. T., & Payne, C. M. (2020). Improving Enzyme Optimum Temperature Prediction with Resampling Strategies and Ensemble Learning. *Journal of Chemical Information and Modeling*, *60*(8), 4098–4107. https://doi.org/10.1021/acs.jcim.0c00489

Garg, A., Zhang, W., Samaran, J., Savitha, R., & Foo, C. S. (2021). An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(6), 2508–2517. https://doi.org/10.1109/TNNLS.2021.3105827

Gerdes, M., Scholz, D., & Galar, D. (2016). Effects of condition-based maintenance on costs caused by unscheduled maintenance of aircraft. *Journal of Quality in Maintenance Engineering*, *22*(4), 394–417. https://doi.org/10.1108/JQME-12-2015-0062

Goebel, K., Celaya, J., Sankararaman, S., Roychoudhury, I., Daigle, M., & Saxena, A. (2017). *Prognostics: The Science of Making Predictions*. Createspace Independent Pub.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. *2008 International Conference on Prognostics and Health Management, PHM 2008*, (November 2008). https://doi.org/10.1109/PHM.2008.4711422

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. MIT press. https://doi.org/10.1016/S0376-7361(07)53015-3

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2014). MIFS-ND: A mutual information-based feature selection method. *Expert Systems with Applications*, *41*(14), 6371–6385. https://doi.org/10.1016/j.eswa.2014.04.019

Hu, C., Youn, B. D., & Wang, P. (2010). Ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation. *Proceedings of the ASME Design Engineering Technical Conference*, *3*(PARTS A AND B), 1023–1032. https://doi.org/10.1115/DETC2010-29182

Hua, Y., Liu, Q., Hao, K., & Jin, Y. (2021). A Survey of Evolutionary Algorithms for Multi-Objective Optimization Problems with Irregular Pareto Fronts. *IEEE/CAA Journal of Automatica Sinica*, *8*(2), 303–318. https://doi.org/10.1109/JAS.2021.1003817

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–395. https://doi.org/10.1145/3219819.3219845

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012

Jayasinghe, L., Samarasinghe, T., Yuen, C., Chen, J., Low, N., & Ge, S. S. (2018). Temporal Convolutional Memory Networks for Remaining Useful Life Estimation of Industrial Machinery. *arXiv preprint arXiv:1810.05644*.

Jia, X., Cai, H., Hsu, Y., Li, W., Feng, J., & Lee, J. (2019). A novel similarity-based method for remaining useful life prediction using kernel two sample test. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, *11*(1), 1–9. https://doi.org/10.36001/phmconf.2019.v11i1.788

Jia, X., Ji, D.-Y., Minami, T., & Lee, J. (2022). Data Quality and Usability Assessment Methodology for Prognostics and Health Management: A Systematic Framework. *IFAC-PapersOnLine*, *55*(19), 55–60. https://doi.org/10.1016/j.ifacol.2022.09.183

Jiao, R., Peng, K., Dong, J., & Zhang, C. (2020). Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics. *Reliability Engineering and System Safety*, *203*(December 2019), 107028. https://doi.org/10.1016/j.ress.2020.107028

Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics : An overview of the current state-of-the-art. *European Journal of Operational Research*, *137*, 1–9.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458

Kan, M. S., Tan, A. C., & Mathew, J. (2015). A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, *62*, 1–20. https://doi.org/10.1016/j.ymssp.2015.02.016

Khan, S., Tsutsumi, S., Yairi, T., & Nakasuka, S. (2021). Robustness of AI-based prognostic and systems health management. *Annual Reviews in Control*, *51*(June), 130–152. https://doi.org/10.1016/j.arcontrol.2021.04.001

Kim, G. Y., Lim, S. M., & Euom, I. C. (2022). A Study on Performance Metrics for Anomaly Detection Based on Industrial Control System Operation Data. *Electronics (Switzerland)*, *11*(8). https://doi.org/10.3390/electronics11081213

Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S. (2022). Towards a Rigorous Evaluation of Time-Series Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(7), 7194–7201. https://doi.org/10.1609/aaai.v36i7.20680

Klawonn, F., & Rehm, F. (2011). Cluster Analysis for Outlier Detection. *Encyclopedia of Data Warehousing and Mining, Second Edition*, (100), 2006–2008. https://doi.org/10.4018/9781605660103.ch035

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, *91*(9), 992–1007. https://doi.org/10.1016/j.ress.2005.11.018

Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated Optimization: Distributed Machine Learning for On-Device Intelligence, 1–38. http://arxiv.org/abs/1610.02527

Lara, J. A., Lizcano, D., Rampérez, V., & Soriano, J. (2020). A method for outlier detection based on cluster analysis and visual expert criteria. *Expert Systems*, *37*(5). https://doi.org/10.1111/exsy.12473

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, J., & Mitici, M. (2022). Multi-objective design of aircraft maintenance using Gaussian process learning and adaptive sampling. *Reliability Engineering and System Safety*, *218*(PA), 108123. https://doi.org/10.1016/j.ress.2021.108123

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834.

Lewis, A. D., & Groth, K. M. (2022a). Metrics for evaluating the performance of complex engineering system health monitoring models. *Reliability Engineering and System Safety*, *223*(March), 108473. https://doi.org/10.1016/j.ress.2022.108473

Lewis, A. D., & Groth, K. M. (2022b). Metrics for evaluating the performance of complex engineering system health monitoring models. *Reliability Engineering and System Safety*, *223*(March), 108473. https://doi.org/10.1016/j.ress.2022.108473

Li, R., Verhagen, W. J. C., & Curran, R. (2020). A systematic methodology for Prognostic and Health Management system architecture de fi nition. *Reliability Engineering and System Safety*, *193*(August 2019), 106598. https://doi.org/10.1016/j.ress.2019.106598

Li, R., Verhagen, W. J., & Curran, R. (2020a). A systematic methodology for Prognostic and Health Management system architecture definition. *Reliability Engineering and*

System Safety, *193*(August 2019), 106598. https://doi.org/10.1016/j.ress.2019.106598

Li, R., Verhagen, W. J., & Curran, R. (2020b). Toward a methodology of requirements definition for prognostics and health management system to support aircraft predictive maintenance. *Aerospace Science and Technology*, *102*, 105877. https://doi.org/10.1016/j.ast.2020.105877

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *Eighth ieee international conference on data mining.*, 413–422.

Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, *108*, 33–47. https://doi.org/10.1016/j.ymssp.2018.02.016

Liu, Y., & Goebel, K. (2018). Information Fusion for National Airspace System Prognostics. *PHM Society Conference*, *10*(1), 1–13.

Lyathakula, R., Karthik, & Yuan, F.-G. (2022). Fatigue damage diagnostics–prognostics framework for remaining life estimation in adhesive joints. *AIAA Journal*, *60*(8), 4874–4892.

Mitici, M., & De Pater, I. (2021). Online model-based remaining-useful-life prognostics for aircraft cooling units using time-warping degradation clustering. *Aerospace*, *8*(6). https://doi.org/10.3390/aerospace8060168

Montero Jimenez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, *56*, 539–557. https://doi.org/10.1016/j.jmsy.2020.07.008

N., A., & Pawar, S. S. (2015). Periodicity Detection of Outlier Sequences Using Constraint Based Pattern Tree with MAD. *arXiv preprint arXiv:1507.01685*.

NASA Systems Engineering Handbook Rev. 2. (2018).

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access*, *9*, 78658–78700. https://doi.org/10.1109/ACCESS.2021.3083060

Ochella, S., & Shafiee, M. (2021). Performance metrics for artificial intelligence (AI) algorithms adopted in prognostics and health management (PHM) of mechanical systems. *Journal of Physics: Conference Series*, *1828*(1). https://doi.org/10.1088/1742-6596/1828/1/012005

O'meara, C., Schlag, L., Faltenbacher, L., & Wicklerz, M. (2016). ATHMoS: Automated telemetry health monitoring system at GSOC using outlier detection and supervised machine learning. *SpaceOps 2016 Conference*, 1–17. https://doi.org/10.2514/6.2016-2347

O'meara, C., Schlag, L., & Wickler, M. (2018). Applications of deep learning neural networks to satellite telemetry monitoring. *15th International Conference on Space Operations, 2018*, (June), 1–16. https://doi.org/10.2514/6.2018-2558

Omri, N., Al Masry, Z., Mairot, N., Giampiccolo, S., & Zerhouni, N. (2021). Towards an adapted PHM approach: Data quality requirements methodology for fault detection applications. *Computers in Industry*, *127*, 103414. https://doi.org/10.1016/j.compind.2021.103414

Pater, I. D., Reijns, A., & Mitici, M. (2022). Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics. *Reliability Engineering and System Safety*, *221*(January), 108341. https://doi.org/10.1016/j.ress.2022.108341

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, *12*(1), 2825–2830. https://doi.org/10.1145/2786984.2786995

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297–313.

Peng, Y., Liu, D., & Peng, X. (2010). A review: Prognostics and health management. *Journal of Electronic Measurement and Instrument*, *24*(1), 1–9. https://doi.org/10.3724/sp.j.1187.2010.00001

Protopapadakis, G., Apostolidis, A., & Kalfas, A. I. (2022). Explainable and Interpretable AI-Assisted Remaining Useful Life Estimation for Aeroengines. *Turbo Expo: Power for Land, Sea, and Air*, V002T05A002.

Ramasso, E., & Saxena, A. (2014). Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. *International Journal of Prognostics and Health Management*, *5*(2), 1–15.

Ren, K., Yang, H., Zhao, Y., Chen, W., Xue, M., Miao, H., Huang, S., & Liu, J. (2019). A Robust auc maximization framework with simultaneous outlier detection and feature selection for positive-unlabeled classification. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(10), 3072–3083. https://doi.org/10.1109/TNNLS.2018.2870666

Ren, K., Yang, H., Zhao, Y., Xue, M., Miao, H., Huang, S., & Liu, J. (2018). A Robust AUC Maximization Framework with Simultaneous Outlier Detection and Feature Selection for Positive-Unlabeled Classification. http://arxiv.org/abs/1803.06604

Rengasamy, D., Rothwell, B. C., & Figueredo, G. P. (2021). Towards a more reliable interpretation of machine learning outputs for safety-critical systems using feature importance fusion. *Applied Sciences (Switzerland)*, *11*(24). https://doi.org/10.3390/app112411854

Rosero, R. L., Silva, C., & Ribeiro, B. (2022). Remaining Useful Life Estimation of Cooling Units via Time-Frequency Health Indicators with Machine Learning. *Aerospace*, *9*(6). https://doi.org/10.3390/aerospace9060309

Sankararaman, S., Saxena, A., & Goebel, K. (2014). Are current prognostic performance evaluation practices sufficient and meaningful? *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014*, 533–545.

Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [Research Frontier]. *IEEE Computational Intelligence Magazine*, *13*(4), 59–76. https://doi.org/10.1109/MCI.2018.2866730

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. *2008 International Conference on Prognostics and Health Management, PHM 2008*. https://doi.org/10.1109/PHM.2008.4711436

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebe, K. (2009). On applying the prognostic performance metrics. *Annual Conference of the Prognostics and Health Management Society, PHM 2009*, 1–16.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On applying the prognostic performance metrics. *Annual Conference of the Prognostics and Health Management Society, PHM 2009*, 1–16.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008a). Damage Propagation Modeling for Aircraft Engine Prognostics. *2008 international conference on prognostics and health management*, 1–9.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008b). Damage propagation modeling for aircraft engine run-to-failure simulation. *2008 International Conference on Prognostics and Health Management, PHM 2008*. https://doi.org/10.1109/PHM.2008.4711414

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008c). Damage propagation modeling for aircraft engine run-to-failure simulation. *2008 International Conference on Prognostics and Health Management, PHM 2008*. https://doi.org/10.1109/PHM.2008.4711414

Saxena, A., Roychoudhury, I., Celaya, J. R., Saha, B., Saha, S., & Goebel, K. (2012). Requirements flowdown for Prognostics and Health Management. *AIAA Infotech at Aerospace Conference and Exhibit 2012*, (June), 1–13. https://doi.org/10.2514/6.2012-2554

Saxena, A., Sankararaman, S., & Goebel, K. (2014). Performance Evaluation for Fleet-based and Unit-based Prognostic Methods. *European Conference of the Prognostics and Health Management Society*, 1–12.

Scott, M., Verhagen, W. J. C., Bieber, M. T., & Marzocca, P. (2022). A Systematic Literature Review of Predictive Maintenance for Defence Fixed-Wing Aircraft Sustainment and Operations. *Sensors*, *22*(18), 7070.

Scott, M. J., Verhagen, W. J., Bieber, M. T., & Marzocca, P. (2022). A Systematic Literature Review of Predictive Maintenance for Defence Fixed-Wing Aircraft Sustainment and Operations. *Sensors*, *22*(18), 1–31. https://doi.org/10.3390/s22187070

Shahid, N., & Ghosh, A. (2019). TrajecNets: Online Failure Evolution Analysis in 2D Space. *International Journal of Prognostics and Health Management*, *10*(11).

Shao, K., He, Y., Xing, Z., & Du, B. (2023). Detecting wind turbine anomalies using nonlinear dynamic parameters-assisted machine learning with normal samples. *Reliability Engineering and System Safety*, *233*(January), 109092. https://doi.org/10.1016/j.ress.2023.109092

Sonneveld, B. (1997). *Using the mollifier method to characterize datasets and models: The case of the Universal Soil Loss Equation* (tech. rep.). https://www.researchgate.net/publication/286670128

Stanovov, V., Brester, C., Kolehmainen, M., & Semenkina, O. (2017). Why don't you use Evolutionary Algorithms in Big Data? *IOP Conf. Ser.: Mater. Sci. Eng.*, *173*(1). https://doi.org/10.1088/1757-899X/173/1/012020

Sun, W., Paiva, A. R. C., Xu, P., Sundaram, A., & Braatz, R. D. (2019). Fault Detection and Identification using Bayesian Recurrent Neural Networks. https://doi.org/10.1016/j.compchemeng.2020.106991

Swearingen, K., Majkowski, W., Bruggeman, B., Gilbertson, D., Dunsdon, J., & Sykes, B. (2007). *An open system architecture for condition based maintenance overview* (tech. rep.). http://www.mimosa.org/mimosa-osa-cbm/

Trinh, H. C., & Kwon, Y. K. (2020). A data-independent genetic algorithm framework for fault-type classification and remaining useful life prediction. *Applied Sciences (Switzerland)*, *10*(1). https://doi.org/10.3390/app10010368

Tseremoglou, I., Bieber, M., Verhagen, W. J., Santos, B. F., Freeman, F. C., & van Kessel, P. J. (2022). The Impact of Prognostic Uncertainty on Condition-Based Maintenance Scheduling: an Integrated Approach. *AIAA AVIATION 2022 Forum*. https://doi.org/10.2514/6.2022-3967

Vapnik, V. N. (1995). The nature of statistical learning theory. *Springer science & business media, 1*.

Viscio, M. A., Viola, N., Fusaro, R., & Basso, V. (2015). Methodology for requirements definition of complex space missions and systems. *Acta Astronautica*, *114*, 79–92. https://doi.org/10.1016/j.actaastro.2015.04.018

Voisin, A., Levrat, E., Cocheteux, P., & Iung, B. (2010). Generic prognosis model for proactive maintenance decision support: Application to pre-industrial e-maintenance test bed. *Journal of Intelligent Manufacturing*, *21*(2), 177–193. https://doi.org/10.1007/s10845-008-0196-z

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. *2008 International Conference on Prognostics and Health Management, PHM 2008*, (November). https://doi.org/10.1109/PHM.2008.4711421

Ward, F. R., & Habli, I. (2020). An Assurance Case Pattern for the Interpretability of Machine Learning in Safety-Critical Systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12235 LNCS*, 395–407. https://doi.org/10.1007/978-3-030-55583-2{_}30

Xiao, Z., Yan, Q., & Amit, Y. (2020). Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *Advances in Neural Information Processing Systems*, *2020-Decem*(NeurIPS).

Xiong, J., Zhou, J., Ma, Y., Zhang, F., & Lin, C. (2023). Adaptive deep learning-based remaining useful life prediction framework for systems with multiple failure patterns. *Reliability Engineering and System Safety*, *235*(March), 109244. https://doi.org/10.1016/j.ress.2023.109244

Xu, Z., Cheng, Z., & Guo, B. (2023). A hybrid data-driven framework for satellite telemetry data anomaly detection. *Acta Astronautica*, *205*, 281–294. https://doi.org/10.1016/j.actaastro.2023.02.009

Yang, L., Ma, Y., Zeng, F., Peng, X., & Liu, D. (2021). Improved deep learning based telemetry data anomaly detection to enhance spacecraft operation reliability. *Microelectronics Reliability*, *126*. https://doi.org/10.1016/j.microrel.2021.114311

Zeng, Z., Jin, G., Xu, C., Chen, S., & Zhang, L. (2022). Spacecraft Telemetry Anomaly Detection Based on Parametric Causality and Double-Criteria Drift Streaming Peaks over Threshold. *Applied Sciences (Switzerland)*, *12*(4). https://doi.org/10.3390/app12041803

Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2017). Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2306–2318. https://doi.org/10.1109/TNNLS.2016.2582798

Zhang, J., Wang, P., Yan, R., & Gao, R. X. (2018). Deep Learning for Improved System Remaining Life Prediction. *Procedia CIRP*, *72*, 1033–1038. https://doi.org/10.1016/j.procir.2018.03.262

Zhang, J., & Lee, J. (2011). A review on prognostics and health monitoring of Li-ion battery. *Journal of Power Sources*, *196*(15), 6007–6014. https://doi.org/10.1016/j.jpowsour.2011.03.101

Zhao, C., & Shen, W. (2022). Adaptive open set domain generalization network: Learning to diagnose unknown faults under unknown working conditions. *Reliability Engineering and System Safety*, *226*(April), 108672. https://doi.org/10.1016/j.ress.2022.108672

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, *20*, 1–7.

Zio, E. (2022). Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering and System Safety*, *218*(PA), 108119. https://doi.org/10.1016/j.ress.2021.108119

ABBREVIATIONS

AD Anomaly detection. 138

AI Artificial intelligence. 4, 24, 85, 138

AUCP Area Under Curve Percentage. 26, 32, 34, 41, 138

CMA Central moving average. 138

C-MAPSS Commercial Modular Aero-Propulsion System Simulation. 64, 65, 66, 67, 69, 71, 72, 73, 74, 78, 79, 80, 82, 126, 138

CBM Condition-Based Maintenance. 2, 3, 4, 5, 6, 7, 10, 11, 15, 84, 85, 86, 102, 105, 106, 138

CUs cooling units. 73, 74, 75, 126, 138

EMA Exponential moving average. 138

ESA European Space Agency. 13, 43, 138

FAG Feature agglomaration. 138

FC Flight cycles. 76, 77, 79, 138

FFBD Functional Flow Block Diagram. 13, 138

GA Genetic algorithm. 29, 30, 59, 63, 64, 65, 88, 91, 138

GDF Generic diagnostic framework. 27, 29, 35, 37, 39, 43, 44, 46, 47, 49, 138

GDPF Generic diagnostic and prognostic framework. 13, 105, 109, 110, 111, 116, 117, 118, 138

GN Gaussian Noise. 87, 138

GPF Generic prognostic framework. 54, 55, 56, 57, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 86, 87, 88, 89, 91, 92, 94, 95, 96, 97, 98, 99, 100, 101, 102, 107, 122, 123, 126, 138

GRP Gaussian random projection. 138

iF Isolation Forests. 32, 36, 40, 48, 138

KNN k-Nearest Neighbors. 32, 36, 38, 39, 40, 43, 48, 138

LSTM Long short-term memory. 26, 138

MAD Median Absolute Deviation. 26, 32, 34, 38, 43, 138

ML machine-learning. 24, 27, 138

MOGA Multi-objective Genetic Algorithm. 31, 35, 39, 94, 96, 98, 99, 122, 123, 126, 138

MROs Maintenance, repair and overhaul providers. 2, 138

MSE Mean squared error. 55, 56, 58, 59, 64, 67, 68, 69, 70, 71, 76, 77, 78, 80, 81, 84, 126, 138

MTT Modified Thompson Tau Test. 26, 32, 138

OC-SVM One Class-Support Vector Machines. 32, 36, 39, 138

OSA Open System Architecture. 2, 3, 5, 15, 138

PCA Principal component analysis. 32, 36, 40, 41, 63, 66, 68, 87, 122, 138

PHM Prognostics and Health Management. 10, 11, 13, 14, 15, 16, 18, 19, 26, 119, 138

RF Random forest. 55, 57, 63, 64, 67, 68, 69, 70, 71, 72, 76, 77, 80, 81, 126, 138

RNN Recurrent Neural Network. 26, 138

RO Random Over-Sampling. 87, 138

RUL Remaining useful life. 52, 53, 54, 55, 56, 57, 58, 59, 61, 63, 64, 65, 66, 67, 70, 71, 74, 76, 79, 80, 84, 85, 86, 89, 90, 91, 93, 95, 96, 102, 122, 138

SE Systems engineering. 11, 12, 13, 138

SMA Simple moving average. 138

SRP Sparse random projection. 138

SVM Support vector machine. 57, 63, 64, 67, 68, 69, 70, 71, 72, 76, 77, 80, 81, 122, 126, 138

tSVD truncated singular value decomposition. 138

VAE Variational Autoencoders. 26, 138

WERCS Weighted relevance-based combination strategy. 87, 98, 102, 138

ABOUT THE AUTHOR

Marie Therese Bieber was born on August 10th, 1992 in Vienna, Austria. In 2011 she started a Bachelor of Mathematics at the Swiss Federal Institute of Technology in Zürich (ETHZ) which she finished in 2015 at the Vienna University of Technology (TU Wien). After completing her B.Sc. degree, she worked for three months as an intern on coding theory at the Manipal Institute of Technology in India, before pursuing the Master of Science in Mathematics at the Vienna University of Technology from 2015 on. During her Master studies she worked as a teaching assistant for Mathematics at the TU Wien, and in summer 2017 she had the chance to spent three months as teaching assistant at the Mongolian University of Science and Technology in Ulaanbaatar. In addition to gaining experience in lecturing, Marie did an internship in operational reliability of aircraft at Airbus in Bremen, followed by an internship in simulation and modeling at DWH GmbH in Vienna. She wrote her master thesis in collaboration with Airbus on "The effect of maintenance induced availability on an airline network", resulting in her obtaining the M.Sc. degree at TU Wien in 2018. After her graduation she got a position as a Young Graduate Trainee (YGT) at ESTEC, ESA in Noordwijk, the Netherlands.

In September 2019, Marie started her PhD at the Faculty of Aerospace Engineering at Delft University of Technology (TU Delft). Her research was funded by the European Union's Horizon 2020 ReMAP project (Real-time Condition-based Maintenance for Adaptive Aircraft Maintenance Planning), as well as by ESA for a cooperation on predictive maintenance applied to satellite systems. She focused on diagnostics and prognostics within the framework of condition-based maintenance, with an emphasis on assessing data suitability for complex systems. Her work has been presented at several international conferences, leading to journal publications. Alongside her research, she coached a DSE project and supervised a Master thesis project.

In addition to her research activities, Marie is a board member of Talking Hands (https://talking-hands.nl/en/)- a foundation supporting deaf and hard of hearing people in Uganda. Her main activities are the cooperation with the partner foundation of Talking Hands in Uganda and the acquisition of money for the buildings of a new school in Uganda. Next to this fulfilling and engaging activity, Marie enjoys spending time in the mountains, is fond of climbing and is always interested in new perspectives - be it through reading a good book, great conversations or traveling.

LIST OF PUBLICATIONS

9. **M. Bieber**, W. J. Verhagen, F. Cosson, B. F. Santos, *Generic Diagnostic Framework for Anomaly Detection—Application in Satellite and Spacecraft Systems.*, Aerospace **10(8)**, 673 (2023).

8. **M. Bieber**, W. J. Verhagen, B. F. Santos, *Assessing the impact of metrics on the choice of prognostic methodologies*, [Unpublished-under review].

7. L. Herrmann, **M. Bieber**, W.J. Verhagen, F. Cosson, B.F. Santos *Unmasking Overestimation: A Re-evaluation of Deep Anomaly Detection in Spacecraft Telemetry.*, CEAS Space Journal (2023).

6. **M. Bieber**, W.J. Verhagen, *A Generic Framework for Prognostics of Complex Systems.*, Aerospace, **9(12)**, 839 (2022).

5. M. J. Scott, W. J. Verhagen, **M. T. Bieber**, P. Marzocca, *A Systematic Literature Review of Predictive Maintenance for Defence Fixed-Wing Aircraft Sustainment and Operations.*, Sensors, **22(18)**, 7070 (2022).

4. **M. Bieber**, W.J. Verhagen, B.F. Santos *The Impact of Metrics on the Choice of Prognostic Methodologies.*, AIAA AVIATION 2022 Forum, p. 3966 (2022).

3. I. Tseremoglou, **M. Bieber**, B. F. Santos, W.J. Verhagen, F.C. Freeman, P. van Kessel, *The Impact of Prognostic Uncertainty on Condition-Based Maintenance Scheduling: An Integrated Approach.*, AIAA AVIATION 2022 Forum, p. 3967 (2022).

2. **M. Bieber**, W.J. Verhagen, B.F. Santos *An adaptive framework for remaining useful life predictions of aircraft systems.*, PHM Society European Conference, Vol. 6, No. 1, pp. 11-11 (2021).

1. **M. Bieber**, W.J. Verhagen, B.F. Santos *Data-Driven Prognostics Incorporating Environmental Factors for Aircraft Maintenance.*, 2021 Annual Reliability and Maintainability Symposium (RAMS), pp. 1-6 (2021).