TU Delft

Delft University of Technology

A Generic Framework for Prognostics of Complex Systems

Bieber, M.T.; Verhagen, W.J.C.

DOI
10.3390/aerospace9120839

Publication date
2022

Document Version
Final published version

Published in
Aerospace

Citation (APA)
Bieber, M. T., & Verhagen, W. J. C. (2022). A Generic Framework for Prognostics of Complex Systems. *Aerospace*, *9*(12), Article 839. https://doi.org/10.3390/aerospace9120839

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

aerospace

MDPI

Article

A Generic Framework for Prognostics of Complex Systems

Marie Bieber [1,*] and Wim J. C. Verhagen [2]

[1] Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands
[2] Aerospace Engineering and Aviation, RMIT University, Carlton, VIC 3053, Australia
* Correspondence: m.t.bieber@tudelft.nl

Abstract: In recent years, there has been an enormous increase in the amount of research in the field of prognostics and predictive maintenance for mechanical and electrical systems. Most of the existing approaches are tailored to one specific system. They do not provide a high degree of flexibility and often cannot be adaptively used on different systems. This can lead to years of research, knowledge, and expertise being put in the implementation of prognostics models without the capacity to estimate the remaining useful life of systems, either because of lack of data or data quality or simply because failure behaviour cannot be captured by data-driven models. To overcome this, in this paper we present an adaptive prognostic framework which can be applied to different systems while providing a way to assess whether or not it makes sense to put more time into the development of prognostic models for a system. The framework incorporates steps necessary for prognostics, including data pre-processing, feature extraction and machine learning algorithms for remaining useful life estimation. The framework 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 highlight considerations for suitability assessment of systems data towards prognostics.

Keywords: prognostics and health management; adaptive framework; remaining useful life

check for updates

Citation: Bieber, M.; Verhagen, W.J.C. A Generic Framework for Prognostics of Complex System. *Aerospace* **2022**, *9*, 839. https://doi.org/10.3390/aerospace9120839

Academic Editors: Theodoros H. Loutas and Dimitrios Zarouchas

Received: 29 October 2022
Accepted: 14 December 2022
Published: 16 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

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 [1]. Prognostics is the process of estimating a system's remaining useful life (RUL) [2], 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 costs, and increased reliability [3,4].

Prognostic approaches can be classified into three types: physics-based, data-driven, and hybrid approaches [5]. Physics-based approaches can be applied in cases in which the underlying degradation phenomenon can be mathematically modelled. There are many examples of physics-based models that were successfully applied in practical cases, such as Li-Ion batteries [6] and other structures subject to fatigue degradation [7]. Data-driven approaches are used when it is difficult to obtain a degradation model or when there is no knowledge about the system physics. Finally, hybrid approaches combine available information about underlying physical knowledge and data. Examples for such approaches are Baptista et al. [8] combining Kalman Filtering with data-driven methods, Downey et al. [9] integrating a physical model and the least square method to estimate RUL of industrial equipment, or Reddy Lyathakula et al. [10] 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 [4]. 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 datasets 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 datasets. 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 subsystems 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. [11]. 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. [12] 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. [13] 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 [14]. 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 and Kwon [15], 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 datasets and comparing its performance to other existing approaches. However, their findings are limited to simulated datasets.

This development makes sense, especially when one considers the problems and challenges arising with using real-life data: As Zio [4] points out, often collected sensor signals are collected under changing operational and environmental conditions. On top of that they are often incomplete, unlabeled, as 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 [1], they may face significant limitations when applied to 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 [16] 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 [17]. 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 [18]. Second, a HI construction approach integrating physics based and data-driven methods was applied to the same dataset to estimate the systems RUL [19].

Still, applications for generic prognostic frameworks are limited to simulated datasets. We therefore present a generic framework and apply it to both a simulated dataset as well as a 'real' dataset of operating aircraft within an airline. For both applications, the aim is to provide 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 applications 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 dataset.
- An identification of the challenges faced with using prognostic approaches on a real aircraft dataset opposed to using simulated data.

The remainder of this paper is organized as follows. Section 2 introduces the generic prognostic framework. In Section 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, we conclude by highlighting the most important findings and limitations and providing directions for further research.

2. The Generic Prognostic Framework

In essence, the Generic prognostic framework (GPF), as shown in Figure 1—originally introduced by Trinh et al. [15] 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 that contains modelling techniques covering multiple aspects of a data-driven prognostics approach and, given a system dataset, 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 dataset 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.

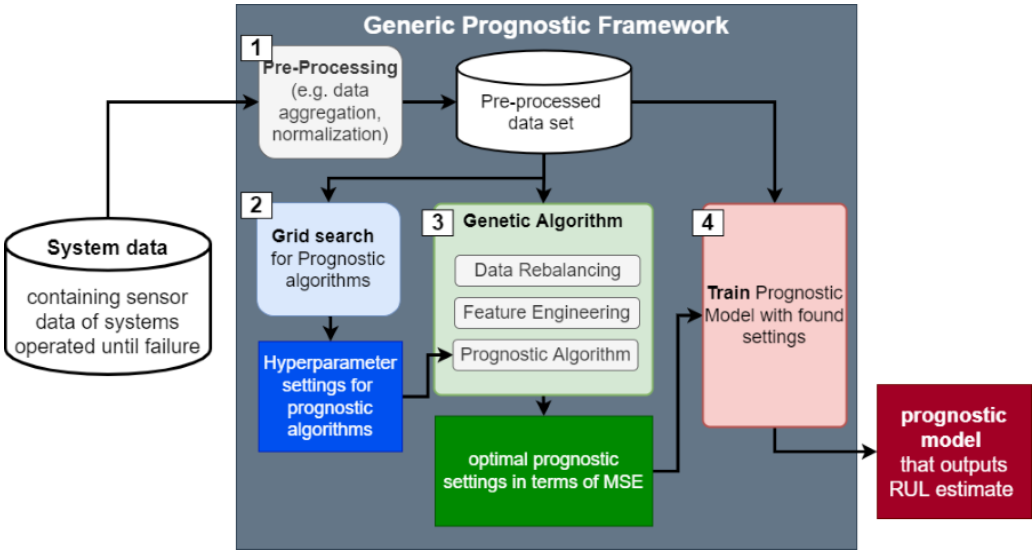

Figure 1. The elements of the generic prognostic framework.

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 (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 as shown in Figure 1. In step 1, the selected system data are pre-processed. As the GPF is a generic framework that is adaptive by nature to different datasets, the data pre-processing techniques applied are kept to a minimum. Further details about the pre-processing applied are given in Section 2.1. In step 2, the hyper parameters for prognostic algorithms are tuned by grid search, as further explained in Section 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 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 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 1 and the respective test set can be found in Table 2.

Table 1. Sample train dataset.

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	39,267	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

Table 2. Sample test dataset.

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

2.1. Step 1: Data Pre-Processing

We make the following assumptions for the system data:

- The system is operated until failure.
- System data are related to operational properties of the system, captured, e.g., through sensors and is available from the beginning 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 dataset is available.
- 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 datasets similar to those presented in Tables 1 and 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 dataset. To evaluate and validate the prognostic models, the data are split into training and test data. The splits are such that trajectories are kept in the same datasets and 10% of the trajectories (ids) are used for testing. This is demonstrated in the example datasets, 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 datasets. For those used in our case studies, we explain the steps in Section 3.

2.2. Step 2: Grid Search to Tune Prognostic Algorithms

Once the system data have 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 [20,21] 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 [22], 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 [23]. 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 been 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 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 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}

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 (1)) of the prognostic algorithm together with data re-balancing and feature engineering techniques on the pre-processed dataset. The MSE at time t is defined as

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

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

The MSE has been selected for the evaluation of the prognostics for two main reasons: first, as a score which captures accuracy, the MSE gives a good indication over how well the algorithms perform with respect to predicting the RUL. Second, despite the fact that it is important to not rely on one metric to evaluate predictions [24], 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 [25]. Due to their flexibility, Genetic algorithm (GA)s are able to solve global optimization problems and optimize several criteria at the same time, such as in our case the simultaneous selection of data re-balancing, feature engineering, and prognostic algorithm techniques [26]. This is what makes them good candidates for our optimization problem.

A GA consists of several steps as presented in Algorithm 1 and Figure 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.

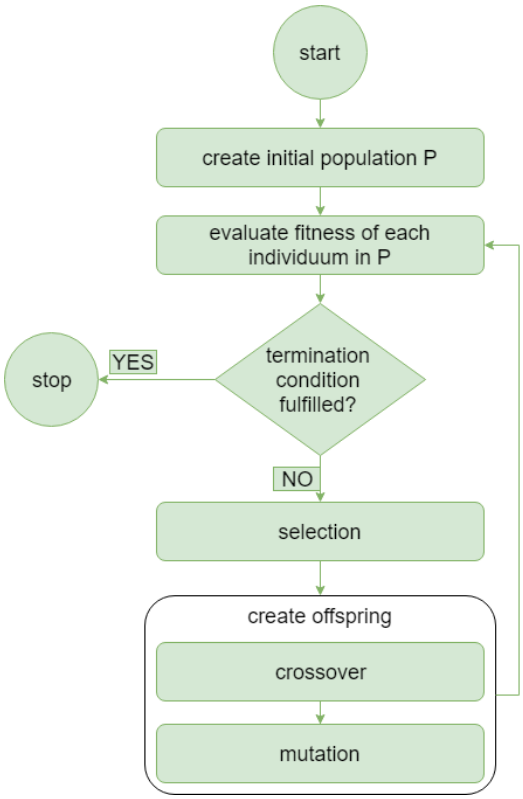

Figure 2. Genetic algorithm process.

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 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 (1)) resulting from the prognostics performed with the individual settings on the underlying dataset.

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 2.1.

Algorithm 1: 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

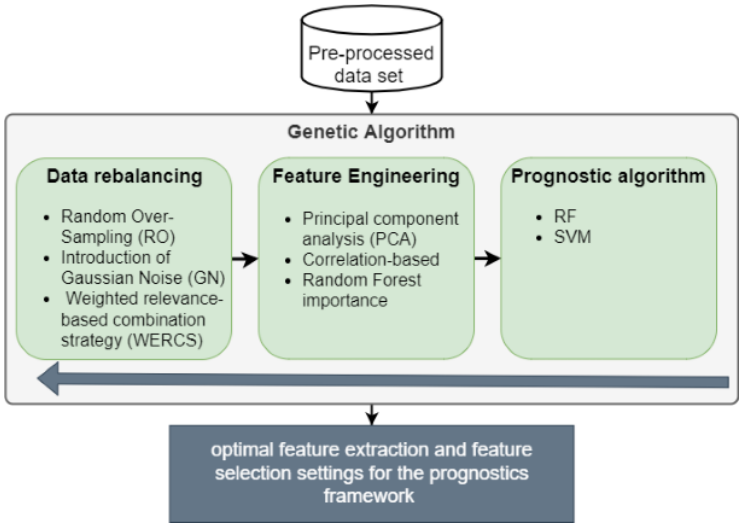

Figure 3. The prognostic steps and methodologies included in the genetic algorithm.

2.3.1. Data Re-Balancing

Data pre-processing or data manipulation is usually performed as a step prior to applying data-driven approaches for two reasons: first, to reduce the number of features in order to achieve a more efficient analysis and second, to adapt the dataset to suit the selected method [27]. Steps typically involved are data cleaning, normalization, and feature engineering [2]. In cases of imbalanced datasets, oversampling can be introduced in addition [28]. A comprehensive overview of feature engineering steps and respective methodologies is given by Jovic et al. [27]. 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 completed 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. [28] are included, namely

- Random over-sampling (RO),
- Introduction of Gaussian noise (GN),
- Weighted relevance-based combination strategy (WERCS).

The presented methodologies are suitable for regression problems, such as RUL estimation. While we do not go into details about them and refer interested readers to [28],

we introduce the underlying 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 [29] and shown in Figure 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.

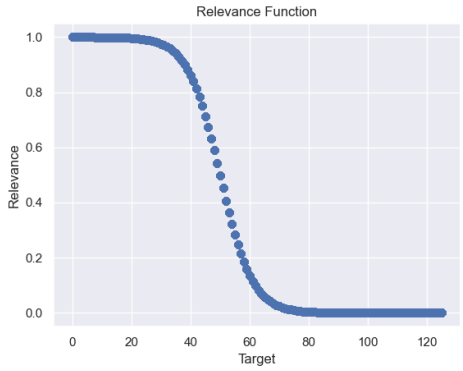

Figure 4. Example of a sigmoid relevance function similar to the one used for the rebalancing task.

- **Random oversampling:** random oversampling is often used to deal with imbalanced classification tasks. Samples from the rare class are randomly selected and replicated in a new updated dataset. In [28], 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 performed 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 5 shows the resulting dataset sizes on the demonstration dataset presented in Table 1 with the relevance threshold $cl = 1$.

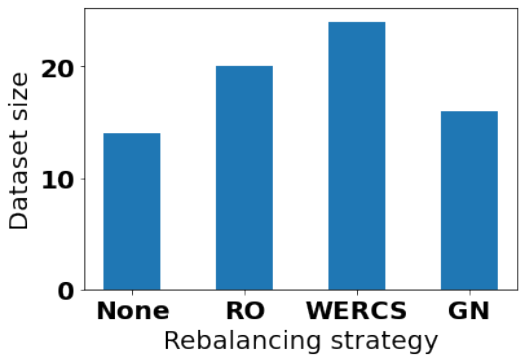

Figure 5. The dataset sizes for the different rebalancing strategies when applied to the demonstration example.

2.3.2. 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 Jovic et al. [27]. 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 Principal component analysis (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 [30]:

- 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.

2.3.3. 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 2.2.

2.3.4. Genetic Algorithm Parameters

The previous paragraphs gave an overview over the form of an individual of the GA. Of course, the GA hyper parameters also 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 [15]. 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.

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 dataset in terms of MSE. This individual is now used to build a prognostic model. As an input this model takes a new dataset 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 [31]. For the re-balancing techniques, the resreg python package is used [29].

3. Case Study and Results

In Section 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 obtain 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 analysed.

In Section 3.1, the framework is implemented and validated on a simulated turbofan engine dataset. Section 3.2 presents the results of applying the framework to an aircraft cooling unit. Finally, in Section 3.3, the results of the case studies are discussed and the generalizability and adaptivity of the GPF are assessed.

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 3.1.1. Subsequently, we explain how we applied the GPF on the dataset in Section 3.1.2, after which we go into details of how the verification and validation was conducted using this dataset in Section 3.1.3 and, finally, we present the results in Section 3.1.5.

3.1.1. Simulated Turbofan Engine Dataset

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data consist of four datasets, each containing simulated run-to-failure data for turbofan engines [32,33]. The datasets differ mainly in the number of fault modes ('modes') and operating conditions ('conditions') as listed in Table 4. 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 dataset contains the measurements corresponding to operations during one time cycle for a certain unit.

Table 4. Characteristics of the four turbofan engine datasets, note that the difference between the four datasets 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

3.1.2. Application of the GPF to the Dataset

In order to train the prognostic models we require a labelled dataset, i.e., we assume that the RUL is known at any time. In the C-MAPSS dataset 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.

3.1.3. 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 completed 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 (as shown in Figure 6).

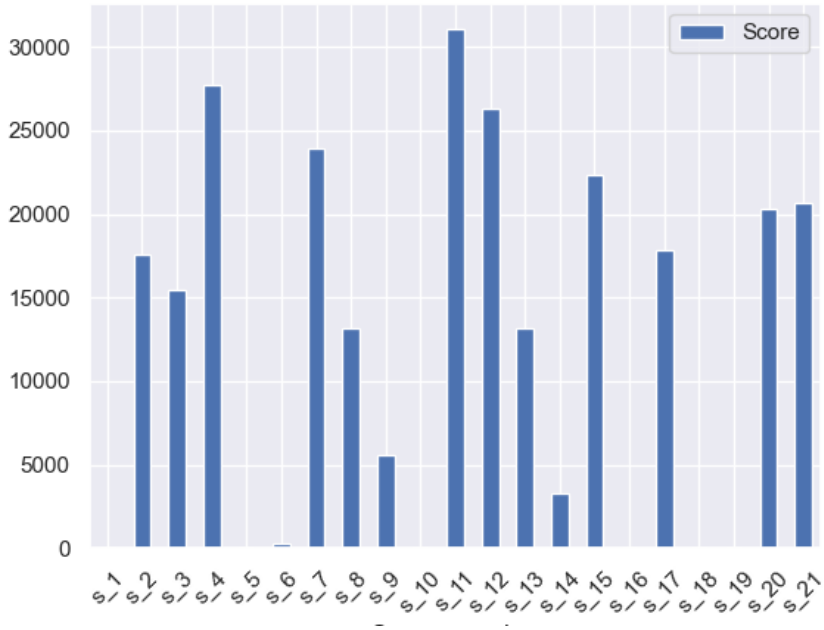

Figure 6. The features selected by the PCA and their relevance scores (the higher the more relevant).

In Section 3.1.2, we already mentioned that there are 21 features, corresponding to sensor readings. Seven of those are constant throughout the components life, leaving us with 14 features of interest, namely sensors 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 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 [34], which is mostly in alignment with what [35] 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 7. In the GPF we include three basic feature engineering methodologies as explained in Section 2.3.2: PCA, correlation-based and importance-based feature engineering. Table 5 gives an overview over the resulting 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 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 [34]**	**Paper #2 [35]**
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

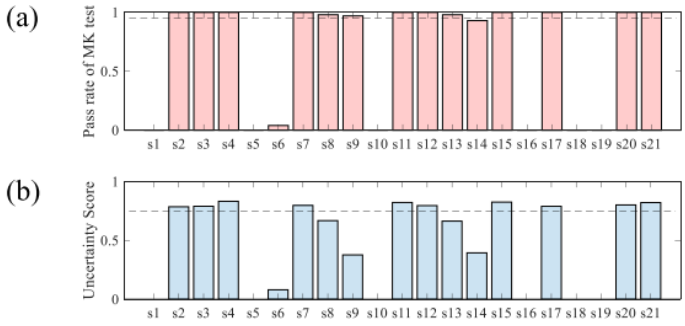

Figure 7. The most relevant features selected based on two different relevance scores by [35]. (**a**): pass rate of MK test; (**b**): Uncertainty Score.

In order to validate the outputs of the prognostic algorithms and the GPF itself, we select three papers from literature presented in Table 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 [36]), 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 3.1.5. 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 [37]. 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 7 and 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 6. Reference papers to validate the output of the prognostic algorithms in the GPF.

Paper ID	Reference
1	[38]
2	[39]
3	[35]

Table 7. Random Forest algorithm RMSE and score on the three papers of literature and using the GPF.

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	84,068	70,465	65,114
FD003	RMSE	22.34	20.27	20.76
	Score	1000.51	711.13	743.03
FD004	RMSE	29.62	31.12	30.00
	Score	22,250	46,567	26,247.53

Table 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	521,461	589,900	316,483	19,827.94
FD003	RMSE	23.3	21.05	46.32	23.69
	Score	1108.68	1598.3	22,541	2472.71
FD004	RMSE	40.77	45.35	59.96	32.24
	Score	46,611	371,140	141,122	10,248.59

3.1.4. 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 9–11 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 8a–c.

It can be seen that the rebalancing methodologies do not really affect the prognostic models in terms of MSE. As Table 9 and Figure 8a 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 10 and Figure 8b) 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 11 and Figure 8c) 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 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. In particular, 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 4).

Table 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

Table 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 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

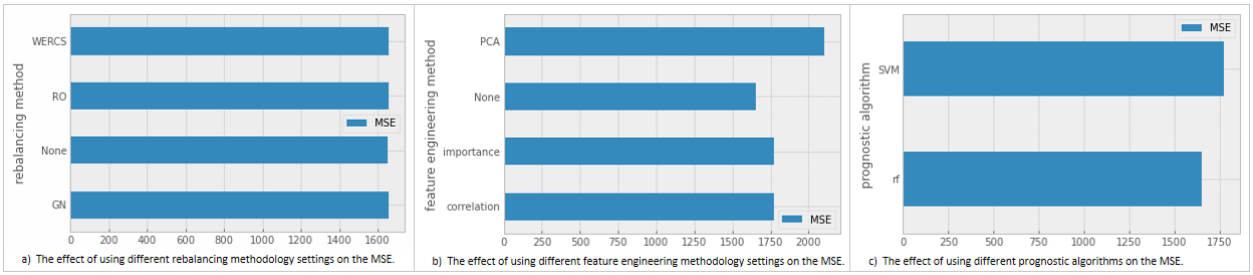

Figure 8. A comparison of applying the different prognostic settings on dataset FD001.

3.1.5. 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 2.2) directly, i.e., skipping step 3, applying the GPF. The resulting metrics are summarized in Table 12 and Figure 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/feature 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 12. The MSEs are all very close to each other.

Table 12. The resulting MSEs of using the GPF versus purely using RF or SVM.

Dataset	**Algorithm**		
	GPF (50 Individuals)	**RF**	**SVM**
FD001	1649.923528	1650.410000	1775.053164
FD002	1877.882809	1974.466387	2152.961399
FD003	4170.124626	4239.466717	4650.671887
FD004	4559.050200	4559.050200	5238.340000

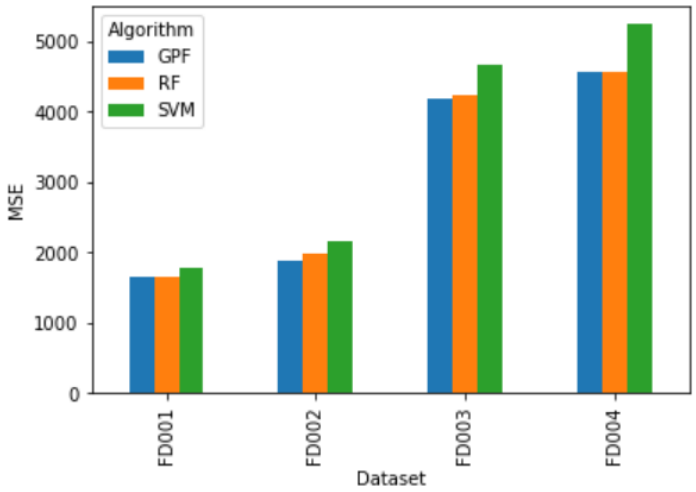

Figure 9. The MSE of GPF versus purely using RF or SVM for the four CMAPSS datasets.

More insight in the quality of predictions can be gained by observing Figures 10–13 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 10a,d,e). Although for trajectories 15 and 87, shown in Figure 10b,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 11a,b,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.

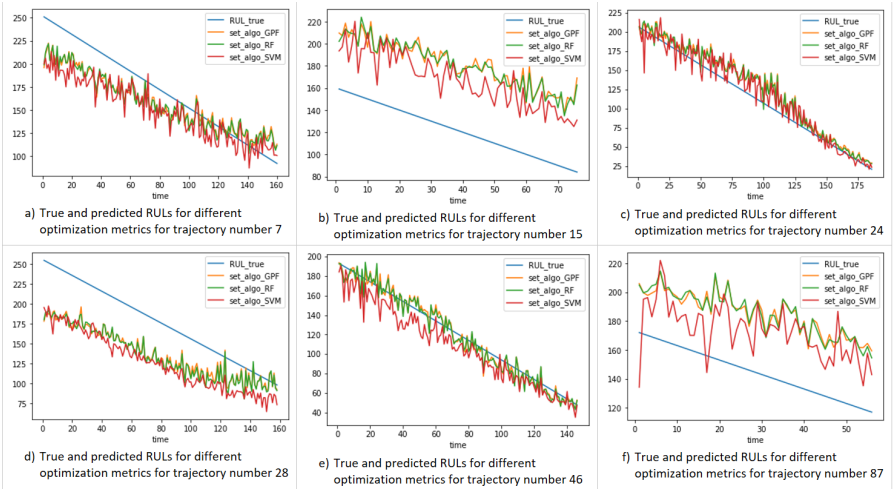

Figure 10. True and predicted value on dataset FD001 for six different trajectories when using the GPF, RF, and SVM.

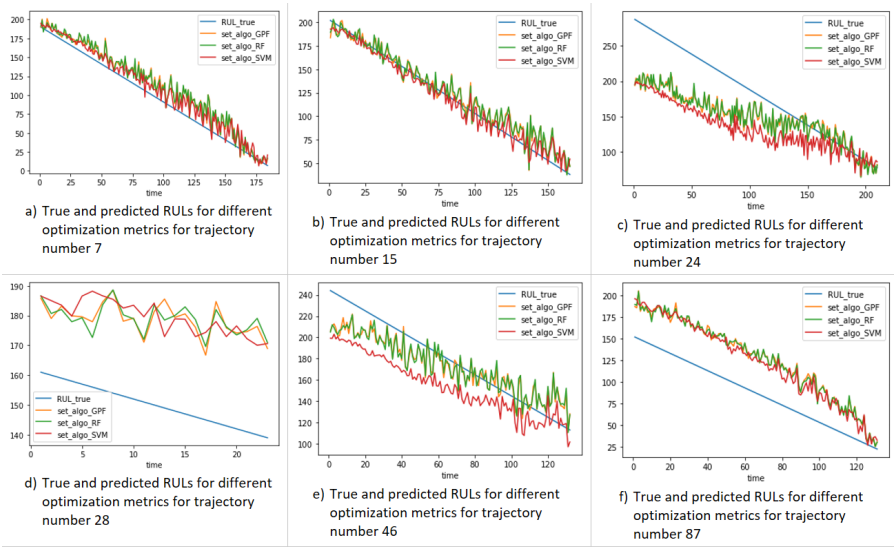

Figure 11. True and predicted value on dataset FD002 for six different trajectories when using the GPF, RF, and SVM.

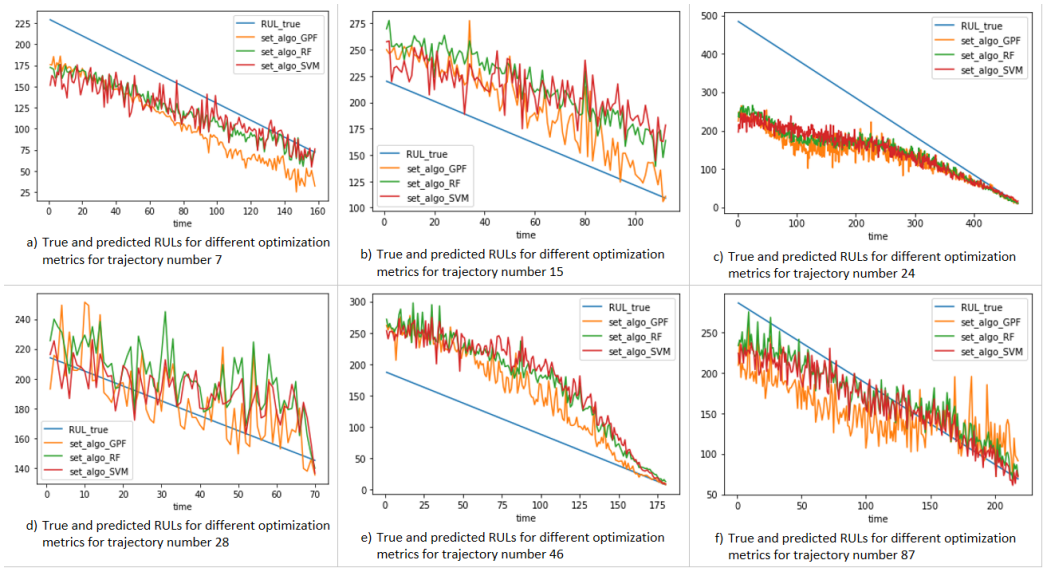

Figure 12. True and predicted value on dataset FD003 for six different trajectories when using the GPF, RF, and SVM.

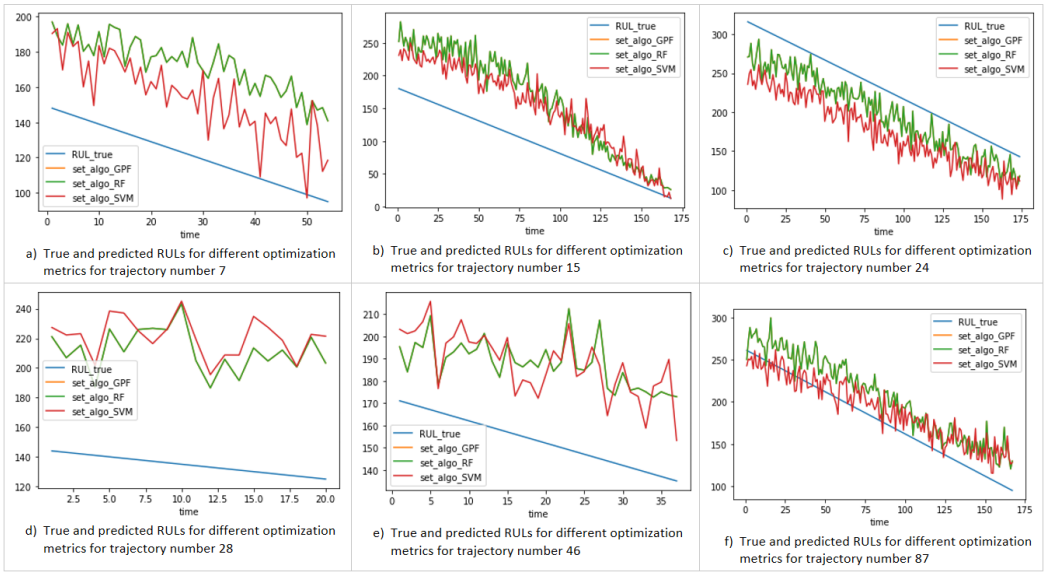

Figure 13. True and predicted value on dataset FD004 for six different trajectories when using the GPF, RF, and SVM.

Table 13 shows the chosen prognostic settings when running the GPF on the four C-MAPSS datasets 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.

Table 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

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 prediction quality.

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.

3.2.1. Cooling Units Dataset

The dataset provided by the airline contains both sensor data and contextual data. On each cooling unit, 9 sensors are installed (i.e., 36 sensors in total for the four CUs) measuring different system properties continuously during flights at a rate of 1 Hz, resulting in 26.4 GB of sensor data for two and a half years of operation corresponding to 18,295 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 [17].

3.2.2. 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 to 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 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.

Table 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	16,623	1511
114	enwslczm	12,877	1170
116	iefywfmy	11,845	1077
115	iefywfmy	11,746	1068
118	dlkzncgy	10,519	957
112	dlkzncgy	10,244	932
108	trmblwny	8998	818
109	tjyjdtaf	8921	811
113	lbhkyjhi	8836	803
105	dlkzncgy	7119	648
31	iefywfmy	6770	616
22	iilvtkok	13,440	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

Using the resulting trajectories, for each the RUL is calculated in the same way as on the C-MAPSS dataset in Section 3.1.2 as a linear function of time, in this case measured in flight cycles until failure. Figure 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 2.3, we use the following approach: for each generation 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 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.

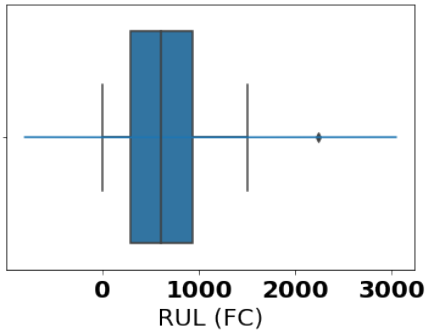

Figure 14. The mean RUL for all trajectories of the cooling unit dataset.

3.2.3. Results Cooling Unit Dataset

Table 15 and Figure 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 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 15. However, while the MSE is lower when cutting 50 Flight cycles (FC) before failure (see Table 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 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	121,133	252,559	256,327
20	100	160,608	228,725	235,180
20	200	176,610	191,486	186,678
20	500	12,818	43,626	75,002

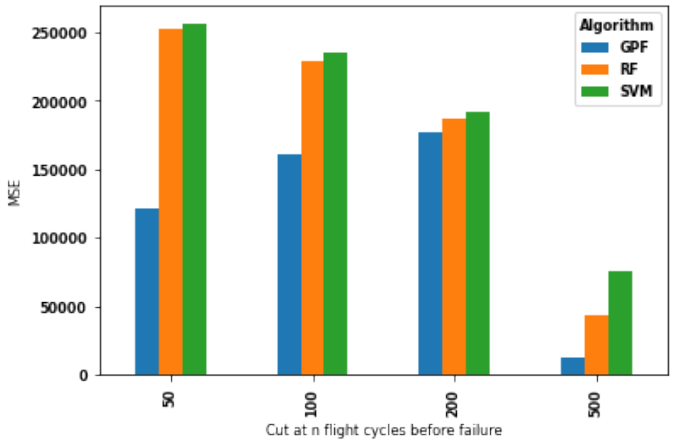

Figure 15. MSE of using GPF, only RF or SVM for different cut settings (cut 50, 100, 200, or 500 FC before failure).

Table 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.

Table 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	169,933	7,15
20	100	GN	importance	SVM	160,608	12,74
20	200	GN	PCA	SVM	119,626	27,40
20	500	WERCS	None	rf	12,818	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 16–19. 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 Figure 17). Table 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 19. In this case, the predictions become more stable again and especially the GPF captures the degradation trend quite well.

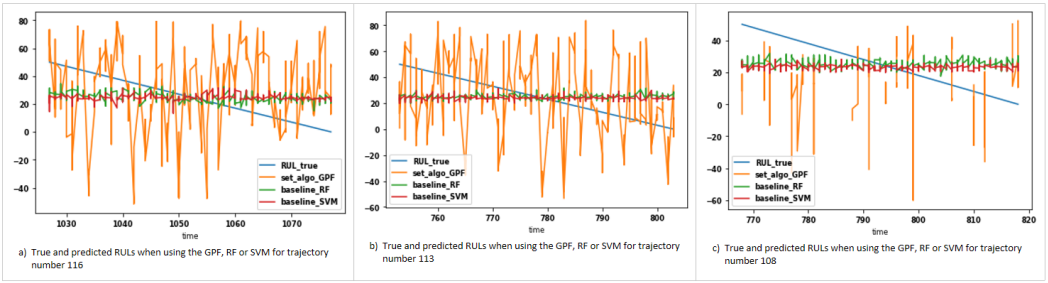

Figure 16. True and predicted values for three different trajectories of the SCU test set when using the GPF, RF, and SVM (cut 50 FC before failure).

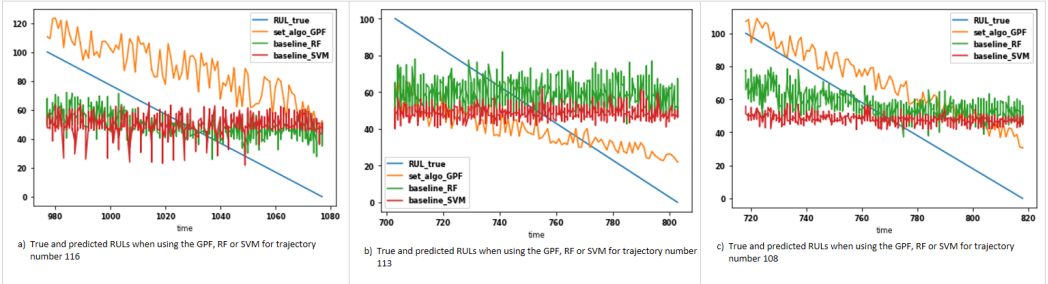

Figure 17. True and predicted values for three different trajectories of the SCU test set when using the GPF, RF, and SVM (cut 100 FC before failure).

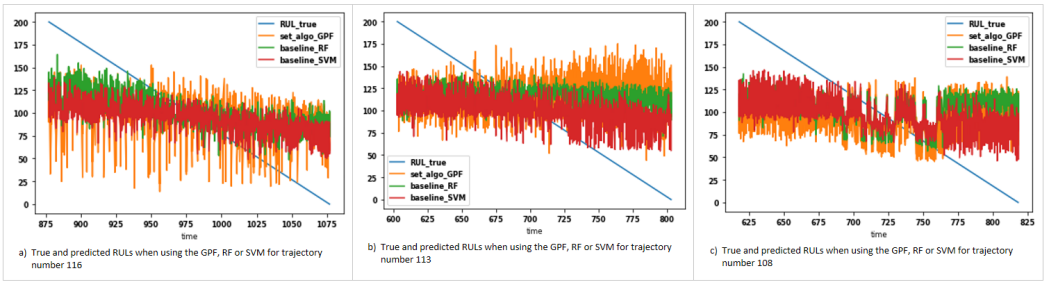

Figure 18. Trueand predicted values for three different trajectories of the SCU test set when using the GPF, RF, and SVM (cut 200 FC before failure).

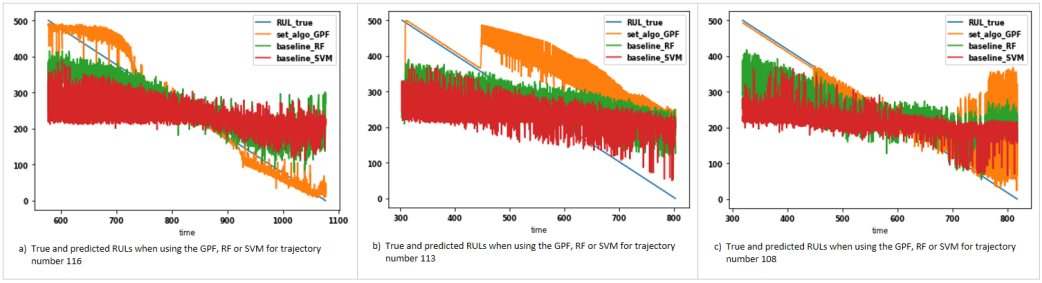

Figure 19. True and predicted values for three different trajectories of the SCU test set when using the GPF, RF, and SVM (cut 50 FC before failure).

All in all, the two main points we find when applying the GPF to the cooling unit dataset 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, such as the one presented in [40], or to use a health indicator flagging data as 'healthy' or 'faulty'.

3.3. Comparative Evaluation and Discussion of the Results

In Section 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 3.3.1, we highlight the similarities and differences between prognostics on simulated and real datasets. 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 3.3.2 and, thirdly, we discuss limitations and directions for further research in Section 3.3.3.

3.3.1. 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 3.1 and for the cooling unit dataset in Section 3.2 highlighted some of the challenges that arise when using prognostics on a real aircraft dataset as 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., Figures 11–18 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 10–13 the degradation trend is much better captured than for the cooling unit dataset (Figures 16–19). 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 16 for the cooling unit and in Table 13 for the C-MAPSS dataset. 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 12 and Figure 9, respectively, Table 15 and Figure 15), it still has much higher potential for improving the cooling unit dataset.

3.3.2. Using the GPF to Determine the Ability to Perform Prognostics on a System

As noted in the previous section, 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 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 datasets. The C-MAPSS datasets are 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 3.3.1. 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. [17] 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 [19,41]).

All in all, we would, based upon 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 3.2 and becomes especially visible in Figures 17 and 19. Based on this and our findings in the previous Sections 3.1 and 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. Table 15 shows that the GPF results in a lower MSE than the classic machine learning algorithms, but Figure 16 shows us that exactly like the RF and SVM models, it does not seem to capture any trends at all on the test dataset. For this purpose, visual representations of the predictions can be helpful.

3.3.3. 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 give 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 [4], 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. Conclusions

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; and (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 generalizability, 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 on system 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.

Author Contributions: Conceptualization, M.B. and W.J.C.V.; methodology, M.B. and W.J.C.V.; software, M.B.; validation, M.B.; writing—original draft preparation, M.B.; writing—review and

editing, W.J.C.V.; visualization, M.B.; supervision, W.J.C.V.; funding acquisition, W.J.C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 769288.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CMA	Central moving average
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CUs	Cooling units
EMA	Exponential moving average
FAG	Feature agglomeration
FC	Flight cycles
GA	Genetic algorithm
GPF	Generic prognostic framework
GRP	Gaussian random projection
MSE	Mean squared error
PCA	Principal component analysis
RF	Random forest
RUL	Remaining useful life
SMA	Simple moving average
SRP	Sparse random projection
SVM	Support vector machine
tSVD	Truncated singular value decomposition

References

1. Scott, M.; Verhagen, W.J.C.; Bieber, M.T.; Marzocca, P. A Systematic Literature Review of Predictive Maintenance for Defence Fixed-Wing Aircraft Sustainment and Operations. *Sensors* **2022**, *22*, 7070.
2. Elattar, H.M.; Elminir, H.K.; Riad, A.M. Prognostics: A literature review. *Complex Intell. Syst.* **2016**, *2*, 125–154.
3. Lei, Y.; Li, N.; Guo, L.; Li, N.; Yan, T.; Lin, J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mech. Syst. Signal Process.* **2018**, *104*, 799–834.
4. Zio, E. Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliab. Eng. Syst. Saf.* **2022**, *218*, 108119. https://doi.org/10.1016/j.ress.2021.108119.
5. Peng, Y.; Dong, M.; Zuo, M.J. Current status of machine prognostics in condition-based maintenance: A review. *Int. J. Adv. Manuf. Technol.* **2010**, *50*, 297–313.
6. Zhang, J.; Lee, J. A review on prognostics and health monitoring of Li-ion battery. *J. Power Sources* **2011**, *196*, 6007–6014. https://doi.org/10.1016/j.jpowsour.2011.03.101.
7. Brownjohn, J.; de Stefano, A.; Xu, Y.L.; Wenzel, H.; Aktan, A.E. Vibration-based monitoring of civil infrastructure: challenges and successes. *J. Civ. Struct. Health Monit.* **2011**, *1*, 79–95. https://doi.org/https://doi.org/10.1007/s13349-011-0009-5.
8. Baptista, M.; Henriques, E.P.; de Medeiros, I.P.; Malere, J.P.; Nascimento, C.L.; Prendinger, H. Remaining useful life estimation in aeronautics: Combining data-driven and Kalman filtering. *Reliab. Eng. Syst. Saf.* **2019**, *184*, 228–239. https://doi.org/10.1016/j.ress.2018.01.017.
9. Downey, A.; Lui, Y.H.; Hu, C.; Laflamme, S.; Hu, S. Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 1–12. https://doi.org/10.1016/j.ress.2018.09.018.
10. Reddy Lyathakula, K.; Yuan, F.G. Fatigue Damage Diagnostics-Prognostics Framework for Remaining Life Estimation in Adhesive Joints. *AIAA J.* **2022**, 1–19 .
11. Baruah, P.; Chinnam, R.B.; Filev, D. An autonomous diagnostics and prognostics framework for condition-based maintenance. *IEEE Int. Conf. Neural Netw.—Conf. Proc.* **2006**, 3428–3435. https://doi.org/10.1109/ijcnn.2006.247346.
12. Voisin, A.; Levrat, E.; Cocheteux, P.; Iung, B. Generic prognosis model for proactive maintenance decision support: Application to pre-industrial e-maintenance test bed. *J. Intell. Manuf.* **2010**, *21*, 177–193. https://doi.org/10.1007/s10845-008-0196-z.
13. An, D.; Kim, N.H.; Choi, J.H. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering and System Safety* **2015**, *133*, 223–236. https://doi.org/10.1016/j.ress.2014.09.014.
14. Hu, C.; Youn, B.D.; Wang, P. Ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation. *Proc. Asme Des. Eng. Tech. Conf.* **2010**, *3*, 1023–1032. https://doi.org/10.1115/DETC2010-29182.

15. Trinh, H.C.; Kwon, Y.K. A data-independent genetic algorithm framework for fault-type classification and remaining useful life prediction. *Appl. Sci.* **2020**, *10*, 368. https://doi.org/10.3390/app10010368.
16. Baptista, M.; Nascimento, C.L.; Prendinger, H.; Henriques, E. A case for the use of data-driven methods in gas turbine prognostics. In Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM, Paris, France, 2–5 May 2017, pp. 441–452.
17. Basora, L.; Bry, P.; Olive, X.; Freeman, F. Aircraft fleet health monitoring with anomaly detection techniques. *Aerospace* **2021**, *8*, 103. https://doi.org/10.3390/aerospace8040103.
18. Mitici, M.; De Pater, I. Online model-based remaining-useful-life prognostics for aircraft cooling units using time-warping degradation clustering. *Aerospace* **2021**, *8*, 168. https://doi.org/10.3390/aerospace8060168.
19. Rosero, R.L.; Silva, C.; Ribeiro, B. Remaining Useful Life Estimation of Cooling Units via Time-Frequency Health Indicators with Machine Learning. *Aerospace* **2022**, *9*, 309. https://doi.org/10.3390/aerospace9060309.
20. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140. https://doi.org/10.1007/BF00058655.
21. Breiman, L. Random forests. *Mach. Learn.* **2001**, 5–32. https://doi.org/10.1023/A:1010933404324.
22. Vapnik, V.N. The Nature of Statistical Learning Theory; *Springer Science & Business Media*: Berlin/Heidelberg, Germany, 1995; Volume 1.
23. Ward, F.R.; Habli, I. An Assurance Case Pattern for the Interpretability of Machine Learning in Safety-Critical Systems. *Lect. Notes Comput. Sci.* **2020**, *12235*, 395–407. https://doi.org/10.1007/978-3-030-55583-2_30.
24. Lewis, A.D.; Groth, K.M. Metrics for evaluating the performance of complex engineering system health monitoring models. *Reliab. Eng. Syst. Saf.* **2022**, *223*, 108473. https://doi.org/10.1016/j.ress.2022.108473.
25. Holland, J.H. *Adaptation in Natural and Artificial Systems*; MIT Press: Cambridge, MA, USA, 1992. https://doi.org/10.1016/S0376-7361(07)53015-3.
26. Stanovov, V.; Brester, C.; Kolehmainen, M.; Semenkina, O. Why don't you use Evolutionary Algorithms in Big Data? *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *173*. https://doi.org/10.1088/1757-899X/173/1/012020.
27. Jović, A.; Brkić, K.; Bogunović, N. A review of feature selection methods with applications. In Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015—Proceedings, Opatija, Croatia, 25–29 May 2015; pp. 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458.
28. Branco, P.; Torgo, L.; Ribeiro, R.P. Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing* **2019**, *343*, 76–99. https://doi.org/10.1016/j.neucom.2018.11.100.
29. Gado, J.E.; Beckham, G.T.; Payne, C.M. Improving Enzyme Optimum Temperature Prediction with Resampling Strategies and Ensemble Learning. *J. Chem. Inf. Model.* **2020**, *60*, 4098–4107. https://doi.org/10.1021/acs.jcim.0c00489.
30. Hoque, N.; Bhattacharyya, D.K.; Kalita, J.K. MIFS-ND: A mutual information-based feature selection method. *Expert Syst. Appl.* **2014**, *41*, 6371–6385. https://doi.org/10.1016/j.eswa.2014.04.019.
31. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in {P}ython. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. https://doi.org/10.1145/2786984.2786995.
32. Frederick, D.K.; DeCastro, J.A.; Litt, J.S. *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*; NASA: Washington, DC, USA, 2007.
33. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. *Damage Propagation Modeling for Aircraft Engine Prognostics*; IEEE: Piscataway, NJ, USA, 2008.
34. Wang, T.; Yu, J.; Siegel, D.; Lee, J. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In Proceedings of the 2008 International Conference on Prognostics and Health Management, PHM 2008, Denver, CO, USA, 6–9 October 2008. https://doi.org/10.1109/PHM.2008.4711421.
35. Jia, X.; Cai, H.; Hsu, Y.; Li, W.; Feng, J.; Lee, J. A novel similarity-based method for remaining useful life prediction using kernel two sample test. In Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM, Scottsdale, AZ, USA, 21–26 September 2019; Volume 11, pp. 1–9. https://doi.org/10.36001/phmconf.2019.v11i1.788.
36. Heimes, F.O. Recurrent neural networks for remaining useful life estimation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, PHM 2008, Denver, CO, USA, 6–9 October 2008. https://doi.org/10.1109/PHM.2008.4711422.
37. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, PHM 2008, Denver, CO, USA, 6–9 October 2008. https://doi.org/10.1109/PHM.2008.4711414.
38. Zhang, C.; Lim, P.; Qin, A.K.; Tan, K.C. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *28*, 2306–2318. https://doi.org/10.1109/TNNLS.2016.2582798.
39. Babu, G.S.; Zhao, P.; Li, X.L. Deep convolutional neural network based regression approach for estimation of remaining useful life. *Lect. Notes Comput. Sci.* **2016**, *9642*, 214–228. https://doi.org/10.1007/978-3-319-32025-0_14.
40. Jayasinghe, L.; Samarasinghe, T.; Yuen, C.; Chen, J.; Low, N.; Ge, S.S. Temporal Convolutional Memory Networks for Remaining Useful Life Estimation of Industrial Machinery. *arXiv* **2018**, arXiv:1810.05644.
41. de Pater, I.; Mitici, M. Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107761. https://doi.org/10.1016/j.ress.2021.107761.