Uncertainty modelling in remaining useful life estimations of aircraft engines

MSc thesis

A. A. Francken

TU Delft Delft University of Technology

Challenge the future

Uncertainty modelling in remaining useful life estimations of aircraft engines

MSc thesis

by

Arjan Francken

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday, June 21, 2022.

Student number:	4939409
Project duration:	1-09-2021 – 18-05-2022
Thesis committee:	Dr. M.A. Mitici
	Dr. ir. B.F. Santos
	Dr. S. Teixeira de Freitas

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Cover image by Mark Wagtendonk

Acknowledgements

"Ambition is the path to success. Persistence is the vehicle you arrive in."
— Bill Bradley

This quote characterizes my personal educational path that I have taken throughout the last 12 years. I have started my educational path by studying for four years to become an aircraft mechanic. Thereafter, I had the desire to expand my knowledge regarding aviation even further and studied for an additional four years to obtain my BSc in aviation engineering. Even though this study substantially expanded my knowledge regarding aviation as well as mathematics, I had the desire to broaden my knowledge even more. As a result, for the past four years (including a two years bridging program), I have been studying aerospace engineering to obtain the degree of MSc in aerospace engineering. Throughout the past four years and especially during the last year when writing this thesis, I have acquired numerous skills and learned so many things for which I am really grateful. Nevertheless, this thesis marks the formal end of these last four years of studying, but more importantly, a formal end to a 12 year educational journey.

I would like to thank and express my gratitude to numerous people. First of all, I would like to thank Mihaela Mitici and Ingeborg de Pater for their guidance, support and valuable feedback throughout the thesis. Furthermore, I would like to thank my parents and sister for their support and kind words throughout these 12 years, and especially this last year. Moreover, I would like to thank Rik for listening to my theories and providing feedback, but also for those numerous long nights in which we played guitar. Those nights really helped me in balancing between working very hard on this thesis and on the other hand, taking the necessary time to relax.

Lastly, I would like to thank and express my gratitude to Madelon for all your love, support and faith in me throughout this past year.

Thank you all, Yours truly,

Arjan

Delft, The Netherlands
May 18, 2022

A. A. Francken BSc.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

Abs	Absolute
AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
CMAPSS	Commercial ModularAero-Propulsion System Simulation
CV	Cross Validation
EoL	End of Life
FC	Flight Cycle(s)
FFNN	Feed-forward Neural Network
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
FPT	First Predicting Time
FTA	Fault Tree Analysis
GPR	Gaussian Process Regression
GRRMD-CUMSUM	Growth Rate of Real-time Mahalanobis Distance Cumulative Sum
HI	Health Indicator
HMM	Hidden Markov Model
HPC	High Pressure Compressor
HPT	High Pressure Turbine
HS	Health Stage
HSMM	Hidden Semi-Markov model
IATA	International Air Transport Association
ICAO	International Civil Aviation Organisation

IG	Inverse Gaussian
KPCA	Kernel Principle Component Analysis
KUR	Kurtosis
LOWESS	Locally Weighted Scatterplot Smoothing
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
Max	Maximum
MD	Mahalanobis Distance
MDI	Mean Decrease Impurity
Min	Minimum
MRO	Maintenance Repair and Overhaul
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
NF	Neuro-fuzzy
PCA	Principal Component Analysis
PDF	Probability Density Function
PF	Particle Filter
PH	Proportional Hazard
PHI	Physical Health Indicator
PHM	Prognostics and Health Management
RF	Random Forest
RFE	Recursive Feature Elimination
RFECV	Recursive Feature Elimination with cross validation
RFR	Random Forest Regression

RMS	Root Mean Square
RMS-CUMSUM	Root Mean Square Cumulative Sum
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
RVM	Relevant Vector Machine
SOM	Self Organising Map
SVM	Support Vector Machine
SVR	Support Vector Regression
VHI	Virtual Health Indicator

List of Symbols

Symbol	Description	Units
$\boldsymbol{A}$	Matrix containing actual x-values	-
a	Index number	-
alt	Altitude	ft
B()	Brownian motion	-
b	Index number	-
b	Bias of ANN model	-
$\boldsymbol{b}$	Matrix containing actual y-values	-
C	Bisquare weighting function	-
C	Impurity measure	FC^2
C()	Covariate function	-
D	Failure threshold	-
d	Degree of polynomial	-
E	Efficiency	-
E[]	Expected value of random variables in GPR model	-
$\boldsymbol{e_i}$	Vector containing residuals between y-values and smoothed y-value belonging to x_i	-
e_{i_j}	jth element of $\boldsymbol{e_i}$	-
F	Flow	-
F	Activation function of ANN model	-
fan_eff_mod	Fan efficiency modifier	-
fan_flow_mod	Fan flow modifier	-
f	Random quantity following a gamma distribution	-
f	Transition function	-
f^{cur}	Current time	s
flt-class	Flight Class	-
frac	Fraction of data to consider for LOWESS	-
Ga()	PDF of gamma process	-
GP()	Gaussian process	-
H	Decision tree	-
HPC_eff_mod	HPC efficiency modifier	-
HPC_flow_mod	HPC flow modifier	-
HPT_eff_mod	HPT efficiency modifier	-
HPT_flow_mod	HPT flow modifier	-
Hz	Hertz	s^{-1}
H_b	The bth decision tree	-
h	Measurement function	-
h_s	Health state	-

IG()	Inverse Gaussian distribution	-
i	Index number	-
inf()	Limit inferior	-
it	Number of iterations for LOWESS	-
j	Index number	-
k	K-nearest neighbour point	-
k	Feature	-
LPC_eff_mod	LPC efficiency modifier	-
LPC_flow_mod	LPC modifier	-
LPT_eff_mod	LPT efficiency modifier	-
LPT_flow_mod	LPT modifier	-
$L(\boldsymbol{z}\|\boldsymbol{\Theta})$	PDF value of $\mathbf{z}$ which is conditional on $\boldsymbol{\Theta}$	-
l	Index number	-
M	Window size	FC
Mach	Mach number	-
MDI	Mean Decrease of Impurity function	FC^2
MDI_{global}	Global Mean Decrease of Impurity function	FC^2
m	node	-
max_depth	Threshold for maximum tree depth	-
min_samples_leaf	Threshold value for minimum samples in a leaf node	-
min_samples_split	Threshold value for minimum samples at a split node	-
N	Number of samples	-
N	Number of bootstrapped samples	-
N	Number of RUL point estimations	-
Nc	Physical core speed	RPM
Nf	Physical fan speed	RPM
N_m	Number of samples in node m	-
n	Index number	-
$n_{estimators}$	Number of decision trees in a RF	-
n_{splits}	Parameter for total number of splits in Time Series Split	-
opt	Nodes for which maximum impurity decrease is obtained for a certain feature	-
P2	Total pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P21	Total pressure at fan outlet	psia
P24	Total pressure at LPC outlet	psia
P30	Total pressure at HPC outlet	psia
P40	Total pressure at burner outlet	psia
P45	Total pressure at HPT outlet	psia
P50	Total pressure at LPT outlet	psia
Ps30	Static pressure at HPC outlet	psia

$P_{(d)}$	dth degree polynomial function	-
$p(\boldsymbol{\Theta})$	Prior PDF of $\boldsymbol{\Theta}$	-
$p(\boldsymbol{\Theta}\|\mathbf{z})$	Posterior PDF of $\boldsymbol{\Theta}$ which is conditional on $\mathbf{z}$	-
q	Median of $\|\boldsymbol{e}\|$	-
q	Transformed variable via linear mapping	-
r	Value to evaluate weighting function	-
r	Index number	-
S	Sum to minimize to obtain $\hat{\beta}$	-
Smfan	Fan stall margin	-
SmHPC	HPC stall margin	-
SmLPC	LPC stall margin	-
TRA	Throttle-resolver angle	%
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T40	Total temperature at burner outlet	°R
T48	Total temperature at HPT outlet	°R
T50	Total temperature at LPT outlet	°R
t	Time	s
t_{EoL}	Moment in time when EoL starts	s
t_{FPT}	Moment in time when FPT starts	s
unit	Unit number	-
v_k	Process noise	-
W	Tricube weighting function	-
$\boldsymbol{W}$	Diagonal matrix containing the weights for weighted least squares	-
Wf	Fuel flow in combustor	pps
W21	Fan flow	pps
W22	Flow out of LPC	lbm/s
W25	Fan into HPC	lbm/s
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s
W48	Flow out of HPT	lbm/s
W50	Flow out of LPT	lbm/s
w	Scenario descriptors	-
w	Feature	-
$w_i()$	Weight for an input node of ANN model	-
w_j	Weight of weighted least squares belonging to point x_j of a sensor signal	-
$X_{f^{cur}+t}$	Component health state at $f^{cur}+t$	-
X_t	Stochastic process	-
x	Original variable in a dataset	-

x	Covariate	-
x	Random variable of GPR model	-
x'	Random variable of GPR model	-
$x[]$	Input of sensor data	-
x_i	X-value of point i of a sensor signal	FC
$x_i()$	Input value at input node of ANN model	-
x_j	X-value of point j of a sensor signal	FC
x_k	System's degradation state	-
x_s	Physical measured signals	-
x_v	Virtual engine health sensors	-
Y	Remaining Useful Life	FC
Y_n	Observable stochastic process	-
Y_s	General degradation process	-
y	Dependent variable	-
$y()$	ANN model output	-
$y[]$	Average value of data points in M	-
y_i	Y-value of point i of a sensor signal	-
$\hat{y}_i$	Smoothed y-value of point i of a sensor signal	-
y_j	Unsmoothed y-value of point x_j,y_j of a sensor signal	-
y_j	True RUL value at jth RUL point estimation	FC
y_j	Sample value of jth sample	-
$\bar{y}_j$	Average value of Nm samples	-
$\hat{y}_j$	Estimated RUL value at jth RUL point estimation	FC
Z_n	Hidden Markov chain	-
z	Threshold value of node in RFR	-
z	Covariate y	-
z_k	Measured data	-
$\boldsymbol{z}$	Vector with observed data	-
α	Shape parameter	-
	Varying random slope	-
	Regression coefficient	-
$\hat{\boldsymbol{\beta}}$	Polynomial coefficients	-
$\Gamma()$	Gamma function	-
$\gamma()$	Wiener process	-
$\gamma()$	Total hazard function	-
$\gamma_0()$	Baseline hazard function	-
Δ_I	Decrease in impurity	FC^2
δ	Residual	-
δ_j	Weight of weighted least squares belonging to point x_j, y_j of a sensor signal	-
ϵ	Residual	-

ϵ_t	Random error (white noise)	-
$\boldsymbol{\Theta}$	Vector with unknown parameters	-
θ	Unit health parameters	-
θ	Function of k and z	-
θ_p	AR model parameter at time t - p	-
$\boldsymbol{\theta_k}$	Vector containing model parameters	-
$\Lambda()$	Monotone increasing function	-
Λ	Function	-
λ	Scale parameter	-
μ	Mean	-
$\mu()$	Mean function	-
σ	Standard deviation	-
σ	Diffusion coefficient	-
phi	Ratio of Wf to Ps30	-
ω_k	Measurement noise	-
#	Item number within a table	-

Introduction

Throughout the past decades, air traffic numbers have been growing rapidly. Despite a short fallback due to COVID-19, air traffic numbers are expected to recover in the next few years and continue to increase afterwards. These high air traffic numbers result into heavy aircraft usage, hence high aircraft maintenance costs. Airlines are desperately searching for methods to reduce these aircraft maintenance costs (especially maintenance costs related to aircraft engines, because aircraft engines contribute to a large part of aircraft maintenance costs). A possible way to reduce these costs is to have the ability to predict when aircraft engines are expected to fail, instead of simply replacing the engines after a predefined number of flight hours or flight cycles.

This thesis focused on the development of a model capable of estimating the Remaining Useful Life (RUL) of aircraft engines whilst also incorporating and modelling the corresponding uncertainty of these estimations. In particular the latter part (modelling the uncertainty of the estimations) contributes to the novelty of this thesis, as most studies in the literature do not take into account the uncertainty of the RUL estimations. In stead, most of the studies seem only interested in the accuracy of the RUL estimations. As explained, the main contribution of this thesis is to provide airlines with a model to estimate the RUL of aircraft engines, whilst providing the uncertainty of these estimations. Such a model allows airlines to decrease their aircraft maintenance costs as it is no longer required to have engine components in stock. Instead, they can be ordered only when required which saves storage space, hence decreases costs.

The structure of this thesis report is the following: in Part I, the scientific paper is presented. Thereafter in Part II, the literature study report is presented which contains the relevant literature that supports the research performed during this thesis. Finally, Part III contains the appendices from the scientific paper.

Part I
Scientific Paper

Uncertainty Modelling in Remaining Useful Life Estimations of Aircraft Engines

A. A. Francken

Under supervision of Dr. M. A. Mitici

Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology
Kluyverweg 1, HS 2926, Delft, The Netherlands

***Abstract*—The largely increasing number of flights throughout the past decades leads to high aircraft (engine) maintenance costs. As a means to decrease these costs, models are required which are capable of estimating the moment an aircraft engine is expected to fail. That is, knowing the moment an aircraft engine will fail in advance, allows airlines to decrease their maintenance costs by being able to schedule the required maintenance personnel and hangars more efficiently. Various models regarding aircraft engine Remaining Useful Life (RUL) estimation have already been proposed in the literature. However, the majority of these models only focused on generating so-called RUL point estimations without taking into account and modelling the uncertainty of these estimations. Therefore, this study focused on the development of a model capable of generating RUL point estimations of aircraft engines as well as taking into account and modelling the accompanying uncertainty of these estimations. In order to do so, the data-driven Random Forest Regression algorithm has been applied which generated RUL point estimations as well as the corresponding probability distributions to show the uncertainty of the estimations. The results show that the proposed model competes well with existing models in the literature in terms of accuracy, and even outperforms some of the existing models. Moreover, the results show that the accuracy of the RUL point estimations increases over time, whilst the uncertainty of these estimations decreases as the aircraft engines approach their end of life.**

***Index Terms*—Aircraft, Engines, RUL, Random Forest Regression, Uncertainty modelling**

1 Introduction

The vastly increasing number of flights throughout the past decades causes severe increases in aircraft Maintenance, Repair and Overhaul (MRO) costs. According to the International Air Transport Association (IATA), these MRO costs are estimated to be 10.3% of an airline's operational costs, with aircraft engines accounting for the largest part (43%) of these total MRO costs [1]. This illustrates the necessity for the development of methods to reduce MRO costs as much as possible, especially the MRO costs associated with aircraft engines. Generally, aircraft maintenance can be divided into reactive and proactive maintenance, where proactive maintenance can be further divided into preventive and predictive (sometimes called condition-based) maintenance [2], [3]. Whereas reactive maintenance would only be performed after an engine failed (which is highly undesirable), preventive maintenance is performed after the engine operated for a predetermined number of flight hours or flight cycles (FC). This latter method has the drawback of "wasting" some of the engine's Remaining Useful Life (RUL), as maintenance is performed before the engine has actually failed [2]. Nevertheless, preventive maintenance is currently widely applied in the field of aircraft maintenance as it ensures air safety [3]. On the other hand, predictive maintenance has the potential to drastically reduce MRO costs as it is a prognostic strategy concerned with the estimation of an engine's RUL before failure occurs. Using predictive maintenance allows for engine maintenance at "the right time", thus minimizing the waste of an engine's RUL and thereby reducing MRO costs. As a result, various models regarding RUL estimation have already been proposed, both model-based as well as data-driven. However, the majority of these proposed models only focuses on single RUL estimations (so-called point estimations), without taking into account and modelling the uncertainty of those estimations. This study proposes a model for RUL estimations of aircraft engines which also takes into account and models the uncertainty of these estimations. It uses a data-driven approach which, besides providing the RUL point estimations, additionally generates probability distributions of the estimated RUL values. These probability distributions allow for determining the uncertainty of the aircraft engine RUL estimations.

The remainder of this paper is organized as follows. In section 2, the related work regarding currently existing RUL estimation models is described. Section 3 thoroughly describes the methodology that has been followed in order to construct the proposed model. In Section section 4, the results including RUL points estimations as well as their probability distributions are presented. Section 5 contains a discussion on the constructed model and its broader implications on the aviation industry, whereafter the conclusions and recommendations are provided in section 6 and section 7, respectively.

2 Related Work

This section presents related work, obtained during the literature study, in which previously conducted studies on RUL estimations were explored. First, a general division of RUL methodologies into different categories is presented. Thereafter, the most commonly used methodologies per category are described.

Over the years, various model types have been proposed in the literature regarding RUL estimation of industrial systems. In most studies, these model types are divided into two main categories: model-based and data-driven [4], [5]. However, Lei et al. [6] made a more thorough distinction by dividing the RUL estimation model types into four different categories: physical models, statistical models, artificial intelligence models and hybrid models.

(a) Physical models

Physical models are usually based on an underlying mathematical model that represents a machine's physical degradation process [7]. According to the literature, these physical models are capable of generating accurate RUL estimations in case the parameters of the underlying mathematical models are well estimated [6]. However, this is not considered to be a straightforward task, as these parameters are usually derived from laboratory experiments [8]. Two of the most popular physical models used for RUL estimation (mainly applied for systems experiencing crack growth) are the so-called Paris-Erdogan model and the Norton model. For example, Zhao et al. [9] used the Paris-Erdogan model to estimate the RUL of gears having cracked teeth from which the cracks grow over time. Likewise, Baraldi et al. [4] employed the Norton model to simulate creep evolution of gas turbine blades and estimate the turbine's RUL. Despite the fact that physical models could create fairly accurate RUL estimations, their usage is limited since accurately describing the physical degradation process of complex systems is considered to be challenging [6].

(b) Statistical models

Statistical models (sometimes called empirical models) use statistical methods which are based on empirical data to estimate a system's RUL [6]. Moreover, RUL estimation models using statistical models are created without depending on any physics or principles. Instead, they are usually constructed using underlying stochastic process models [6]. According to the literature, some commonly used statistical models for RUL estimation are autoregressive models, Wiener process models and Proportional hazard models. Autoregressive models are applied by several studies such as conducted by Liu et al. [10] who used the autoregressive model to estimate the RUL of lithium-ion batteries, or Qian et al. [11] who used it for the modelling of the degradation process of bearings and to estimate the failure of bearings 50 minutes ahead. Wiener process models (and variations of it) are for example used by Lin and Lin [12] which applied it to model the degradation process of aircraft engines and used it for RUL estimation.

Another example is the study conducted by Huang et al. [13] who used a variation of the Wiener process (skew-Wiener process) to estimate the RUL of roller ball bearings, as they found the degradation speed of industrial machines to follow a skew-normal distribution. Proportional hazard models are also applied for the RUL estimation of bearings by Liao et al. [14] as well as for the RUL estimation of low methane compressors by Tran et al. [15]. Although many applications of statistical models for RUL estimation exist in the literature, their usage for RUL estimations of mechanical systems experiencing complex deterioration patterns remains a challenge [6].

(c) Artificial intelligence models

The main difference of artificial intelligence (AI) models compared to physical and statistical models is their ability to recognize and learn patterns in deterioration data [6]. Consequently, AI models have become very popular for degradation modelling and RUL estimations of mechanical systems with complex degradation processes, as these processes are usually too difficult to be accurately described by physical and/or statistical models [6]. However, their main disadvantage lies in the fact that it is generally difficult to explain how AI models obtain their results due to a lack of transparency. Therefore, AI models are sometimes referred to as "black-boxes" [5]. Nevertheless, AI models are known for many successful implementations. For example, the use of a Recurrent Neural Network (RNN) model as applied by Heimes [16] to estimate the RUL of aircraft engines or the use of a Long Short Term Memory (LSTM) model as applied by Yuan et al. [17] to predict the RUL of aircraft engines operating in complicated operational conditions. Another successful implementation of an AI model is the application of a Random Forest Regression (RFR) model by Chen et al. [18] who used it to also estimate the RUL of aircraft engines. Furthermore, Tanwar and Raghavan [19] successfully used an AI model called Gaussian Process Regression (GPR) model for the RUL estimation of lubricating oil of an oil lubrication system. In this study, the RUL is estimated by predicting the remaining number of hours before the oil is contaminated with a certain predetermined level of wear debris [19]. Additionally, Biggio et al. [20] used three different Gaussian Process models (Stochastic Variational Gaussian Process model, Deep Gaussian Process model and Deep Sigma Point Process model) to estimate the uncertainty of aircraft engine RUL estimations. This is an exceptional study as it is one of the very few studies who attempts to model the uncertainty of aircraft engine RUL estimations.

(d) Hybrid models

Hybrid models are combinations of the previously described models and try to create an even better model in terms of prediction accuracy [6]. However, hybrid models are currently used the least in the literature. Perhaps because merging several models into a single hybrid model might be more challenging compared to using the models individually. Nevertheless, there are some successful implementations of hybrid models. For example, the study conducted by Tamssaouet et al. [21] who successfully proposed a hybrid model for the RUL estimations of aircraft engines (including a probability density function to model uncertainty) by combining a LSTM model with a particle filter model. Again, this is considered exceptional as very few studies attempt to model the uncertainty of aircraft engine RUL estimations. Note that since there exist numerous RUL estimation models in the literature, the overview of commonly used RUL estimation models as provided above in this section, is not exhaustive.

3 METHODOLOGY

Throughout this section, the entire methodology from the data acquisition step to the hyperparameter tuning step is extensively described. First, data acquisition, data description and data processing are described in paragraph 3.1, paragraph 3.2 and paragraph 3.3, respectively. Thereafter, the process of feature selection, RUL model implementation and hyperparameter tuning are extensively discussed in paragraph 3.4, paragraph 3.5 and paragraph 3.6, respectively.

3.1 Data acquisition

This paragraph describes the data acquisition process. First, the difficulties of obtaining adequate deterioration data are discussed. Thereafter, a brief explanation regarding the dataset that will be used is provided. In order to create any predictive model, appropriate deterioration data needs to be available not only in terms of quality, but also in terms of quantity. However, it appears to be difficult to obtain such data because of several reasons. For example, data acquisition is a very expensive and time consuming process due to the fact that the vast majority of industrial machines is characterized by long-term degradation processes [6]. Another important reason is concerned with commercial competition. In case a commercial company is able to obtain deterioration data, it will most likely not be made publicly available due to commercial competition [6], [22].

Therefore, in 2008, the Prognostics Center of Excellence at the National Aeronautics and Space Administration (NASA) created a large aircraft turbofan engine deterioration dataset, in order to allow for the development of RUL predictive models. This dataset contains an extensive amount of so-called run-to-failure time series data from turbofan engines and can be downloaded from the NASA Ames Prognostics Data Repository [23]. It has been constructed using NASA's high fidelity Commercial Modular Aero-Propulsion System Simulation (CMAPSS) computer model, which is especially developed to simulate realistic run-to-failure behaviour of large turbofan engines [24]. More recently, in 2021, the Prognostics Center of Excellence at NASA published a new, even more realistic run-to-failure aircraft turbofan engine time series deterioration dataset (also using CMAPSS), which can likewise be downloaded from the NASA Ames Prognostics Data Repository [25]. This dataset is an improvement of the old dataset from 2008, as it consists of more data and contains additional sensors monitoring the health condition of the turbofans. Furthermore, by relating the turbofan's engine degradation process to its operational history, its degradation modelling is improved [24]. Note that the new dataset published in 2021 is the only dataset that will be considered and described in this paper.

3.2 Data description

This paragraph provides a thorough description of the data that will be used for the development of the RUL estimation model. This includes, among others, the specific subsets that will be used, the number of available turbofan engines per subset as well as the available sensors for deterioration monitoring. First, a general description for all subsets is provided, whereafter a more specific description is given for the first data subset (subparagraph 3.2.1) as well as the second and third data subsets (subparagraph 3.2.2).

As can been seen from Table 1, the aircraft turbofan engine deterioration dataset is divided into eight distinct subsets. Each of these subsets contains a different number of data entries as well as a certain number of turbofan engines (called units by Chao et al. [24]). Within each subset, the data is further divided into a development set and a test set. The development set will be used to train, optimize model parameters and cross validate the model, whereas the test set will be used to assess the model's performance on unseen data and obtain the model's RUL estimation accuracy and corresponding uncertainty.

Table 1
OVERVIEW OF THE NASA AIRCRAFT TURBOFAN ENGINE DETERIORATION DATASET [24].

Data subset name	Number of units	Data subset size
DS01	10	7.6 M
DS02	9	6.5 M
DS03	15	9.8 M
DS04	10	10.0 M
DS05	10	6.9 M
DS06	10	6.8 M
DS07	10	7.2 M
DS08	54	35.6 M

The engines (units) in the various data subsets could be affected by single or multiple failure modes. The definition of a failure mode is given by Matthews [26] as follows: *A failure mode is a possible way a component can fail.* An example of a failure mode is a deformed turbine blade prohibiting the flow to obtain its optimal velocity. This failure mode could be caused by a variety of factors such as eroded or corroded turbine blades (which are called failure mechanisms [27]). Similar to real turbofan engines, various failure modes can occur in the different components of the CMAPSS turbofan engine. These different engine components consist of a fan, Low Pressure Compressor (LPC), High Pressure Compressor (HPC), High Pressure Turbine (HPT) and Low Pressure Turbine (LPT) and are depicted in Figure 1. Additionally, Figure 1 also shows the combustor, Nf (a sensor measuring the physical fan speed) as well as Nc (a sensor measuring the physical core speed).

The failure modes (as defined and stated by Chao et al. [24]) corresponding to the various engine components are listed per data subset in Table 2. Note that engines are referred to as units from now on. Table 2 shows that (apart from the components of units in DS01) each component could be affected by two main types of failure modes: a disrupted flow F or a decreased efficiency E. However, as mentioned previously, some units could be affected by a combination of failure modes.

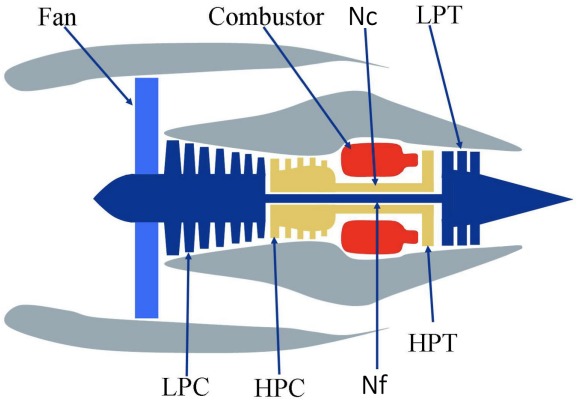


Figure 1. Schematic representation of the CMAPSS turbofan engine [28].

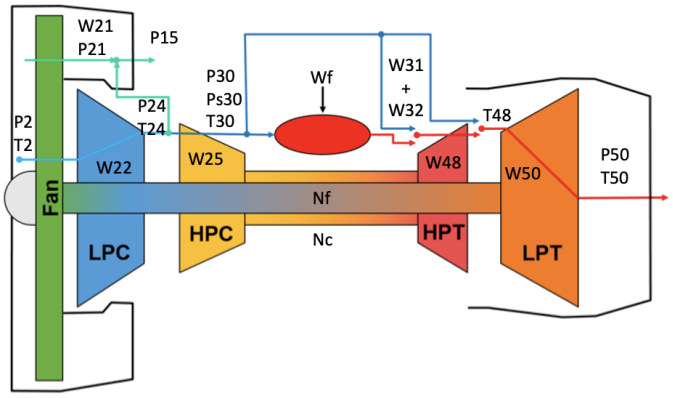


Figure 2. Sensor locations of the CMAPSS turbofan engine [24].

Table 2
OVERVIEW OF FAILURE MODES PER DATA SUBSET [24].

Data subset name	Fan	LPC	HPC	HPT	LPT
DS01				E	
DS02				E	E, F
DS03				E	E, F
DS04	E, F				
DS05		E, F			
DS06		E,F	E, F		
DS07					E, F
DS08	E, F	E, F	E, F	E, F	E, F

For example, a number of units in DS02 could be affected by a decreased HPT efficiency, whilst some of the other units in DS02 could be affected by all three failure modes simultaneously (being a decreased HPT and LPT efficiency plus a disrupted LPT flow). Given that only units in DS01 are affected by the exact same failure mode (a decreased HPT efficiency), makes it the most straightforward subset. Consequently, DS01 has been selected as the dataset for model development (training and parameter optimization) as well as model testing to obtain its accuracy.

Furthermore, the considerably more complex DS03 has been selected for model verification, because the units in DS03 contain multiple simultaneous failure modes including a decreased HPT efficiency. Therefore, it can be assessed if the developed model is capable of handling more complex data whilst producing (accurate) RUL estimations. Finally, DS02 has been chosen to validate the model because the results can be compared to the study conducted by Biggio et al. [20], who also used DS02 for the development of a RUL estimation model.

3.2.1 Data subset DS01

As described previously, DS01 consists of ten units and is divided into a development set (with six units) and a test set (with four units). Both the development and the test set comprises five types of main variables: scenario descriptors w (Table 3), physical measured signals x_s (Table 4), virtual unit health sensors x_v (Table 5), unit health parameters θ (Table 6) and so-called auxiliary data (Table 7) [24]. Note that all other data subsets consist of the exact same variables as listed in Table 3 - Table 7. Apart from these variables, each data subset also contains the actual true RUL values per unit, denoted by Y and provided in FC. A clear view of the locations of the various sensors listed in Table 4 and Table 5 is depicted in Figure 2.

Table 3
SCENARIO DESCRIPTORS - w [24].

#	Symbol	Description	Units
1	alt	Altitude	ft
2	Mach	Mach number	-
3	TRA	Throttle-resolver angle	%
4	T2	Total temp. fan inlet	°R

According to Chao et al. [24], the units are assigned to a specific flight class depending on the flight length. Furthermore, each flight class represents a certain flight condition (low altitude and speed, medium altitude and speed as well as high altitude and speed for flight classes 1,2 and 3, respectively). The official definition of the flight classes can be found in Table 8. The authors in [24] assumed that all flights corresponding to a certain unit are assigned to the same flight class. Consequently, it has been decided to develop one model per flight class due to the substantial differences in flight conditions. As a second assumption, the authors in [24] assumed a unit to reach its End of Life (EoL) when a certain (unknown) health index reaches a value of zero or whenever the unit has flown for 100 FC, whichever comes first [24]. However, this means that units having flown for 100 FC must be discarded from the development set, as those units did not run to failure. Instead, their degradation simulations were simply terminated after 100 FC. As a result, units 1 and 3 have been discarded from the development set of DS01.

Table 4
PHYSICAL MEASUREMENTS - x_s [24].

#	Symbol	Description	Units
1	Wf	Fuel flow in combustor	pps
2	Nf	Physical fan speed	rpm
3	Nc	Physical core speed	rpm
4	T24	Total temp. LPC outlet	°R
5	T30	Total temp. HPC outlet	°R
6	T48	Total temp. HPT outlet	°R
7	T50	Total temp. LPT outlet	lbm/s
8	P15	Total pressure in bypass-duct	psia
9	P2	Total pressure fan inlet	psia
10	P21	Total pressure fan outlet	psia
11	P24	Total pressure LPC outlet	psia
12	Ps30	Static pressure HPC outlet	psia
13	P40	Total pressure burner outlet	psia
14	P50	Total pressure LPT outlet	psia

Table 5
VIRTUAL HEALTH SENSORS - x_v [24].

#	Symbol	Description	Units
1	T40	Total temp. burner outlet	°R
2	P30	Total pressure HPC outlet	psia
3	P45	Total pressure HPT outlet	psia
4	W21	Fan flow	pps
5	W22	Flow out of LPC	lbm/s
6	W25	Flow into HPC	lbm/s
7	W31	HPT coolant bleed	lbm/s
8	W32	LPT coolant bleed	lbm/s
9	W48	Flow out of HPT	lbm/s
10	W50	Flow out of LPT	lbm/s
11	SmFan	Fan stall margin	-
12	SmLPC	LPC stall margin	-
13	SmHPC	HPC stall margin	-
14	phi	Ratio of Wf to Ps30	pps/psi

An overview of all units in the development and test sets in DS01 (including the discarded units 1 and 3), with their corresponding flight classes and total life from start to EoL (expressed as total FC till EoL), can be found in Table 9. More information regarding the variables and the full details of the complete dataset, with all its subsets, can be found in the paper by Chao et al. [24].

3.2.2 Data subsets DS02 and DS03

As explained earlier, DS02 and DS03 will be used for validation and verification, respectively. An overview of the units in DS02 and DS03 together with their corresponding flight classes and total number of FC till EoL, can be found in Table 10 and Table 11, respectively. Investigation of DS02 revealed that all training units in the development set belong to flight class 3. Additionally, it has been found that training units 2, 5 and 10 are affected by a single failure mode (deteriorated HPT efficiency), whilst the remaining training units (as well as all test units) have three different simultaneous failure modes (deteriorated HPT and LPT efficiency, and disrupted a LPT flow). As a consequence, only training units 16, 18 and 20 are suitable for model development in DS02.

3.3 Data processing

This paragraph describes two methods which are applied for the process of noise removal, data smoothing and data downsampling in order to increase model accuracy and reduce computation time. Noise removal and data smoothing are necessary as the raw data from the sensors listed in Table 4 and Table 5 is greatly affected by noise. An example of this noisy raw sensor data can be seen in Figure 3, which shows the raw data of the $SmLPC$ sensor for unit 6 of flight class 2 (DS01) over its entire lifetime (from start to EoL). The noise can be clearly observed and must be reduced and smoothed as much as possible, since noisy data can have a

Table 6
ENGINE HEALTH PARAMETERS - θ [24].

#	Symbol	Description	Units
1	fan_eff_mod	Fan efficiency modifier	-
2	fan_flow_mod	Fan flow modifier	-
3	LPC_eff_mod	LPC efficiency modifier	-
4	LPC_flow_mod	LPC flow modifier	-
5	HPC_eff_mod	HPC efficiency modifier	-
6	HPC_flow_mod	HPC flow modifier	-
7	HPT_eff_mod	HPT efficiency modifier	-
8	HPT_flow_mod	HPT flow modifier	-
9	LPT_eff_mod	LPT efficiency modifier	-
10	LPT_flow_mod	LPT flow modifier	-

Table 7
AUXILIARY DATA [24].

#	Symbol	Description	Units
1	unit	Unit number	-
2	flt-class	Flight class	-
3	h_s	Health state	-

Table 8
OVERVIEW OF FLIGHT CLASSES [24].

Flight class	Flight length [hrs]
1	1 - 3
2	3 - 5
3	> 5

Table 9
OVERVIEW OF UNITS IN THE DEVELOPMENT AND TEST SETS OF DS01, WITH THEIR TOTAL NUMBER OF FC TILL EOL AND CORRESPONDING FLIGHT CLASSES.

Unit	Flight class	Total FC till EoL	Development set	Test set
1	1	100	✓	-
2	3	75	✓	-
3	2	100	✓	-
4	1	95	✓	-
5	3	89	✓	-
6	2	94	✓	-
7	1	90	-	✓
8	2	89	-	✓
9	1	80	-	✓
10	3	82	-	✓

severely negative influence on prediction results [29]. Apart from noise reduction and data smoothing, the degradation data requires downsampling (to a single sensor value per FC) in order to drastically decrease computation time, as the amount of recorded degradation data is enormous due to the high $1Hz$ sampling rate. Moreover, this high sampling rate is redundant since degradation occurs gradually over time. Hence, suitable methods being capable of effective noise removal, data smoothing and data downsampling, to a single sensor value per FC, are required. The first applied method is the so-called FC average smoothing method, which working principle as well as an example showing a smoothed sensor with this method, are described in subparagraph 3.3.1. The working principle of the second applied smoothing method, Locally Weighted Scatterplot Smoothing (LOWESS), is shown in subparagraph 3.3.2 also together with an example showing a smoothed sensor using this method.

Table 10
OVERVIEW OF UNITS IN THE DEVELOPMENT AND TEST SETS OF DS02, WITH THEIR TOTAL NUMBER OF FC TILL EOL AND CORRESPONDING FLIGHT CLASSES.

Unit	Flight class	Total FC till EoL	Development set	Test set
2	3	75	✓	-
5	3	89	✓	-
10	3	82	✓	-
11	3	59	-	✓
14	1	76	-	✓
15	2	67	-	✓
16	3	63	✓	-
18	3	71	✓	-
20	3	66	✓	-

Table 11
OVERVIEW OF UNITS IN THE DEVELOPMENT AND TEST SETS OF DS03, WITH THEIR TOTAL NUMBER OF FC TILL EOL AND CORRESPONDING FLIGHT CLASSES.

Unit	Flight class	Total FC till EoL	Development set	Test set
1	1	72	✓	-
2	2	73	✓	-
3	2	67	✓	-
4	2	60	✓	-
5	1	93	✓	-
6	3	63	✓	-
7	2	80	✓	-
8	3	71	✓	-
9	1	84	✓	-
10	3	66	-	✓
11	3	59	-	✓
12	1	93	-	✓
13	3	77	-	✓
14	1	76	-	✓
15	2	67	-	✓

3.3.1 FC average smoothing method

The first applied method for noise reduction, data smoothing and data downsampling whilst preserving the original data characteristics, i.e. data deterioration trends, is called the FC average smoothing method. As the name implies, it consists of computing the average value for a given sensor per FC. Hence, this method produces a single smoothed sensor value per FC. The main benefit of using this method is the little required run time as well as the fact that the same 'amount' of data will be used to smooth a sensor value, regardless a unit's lifetime.

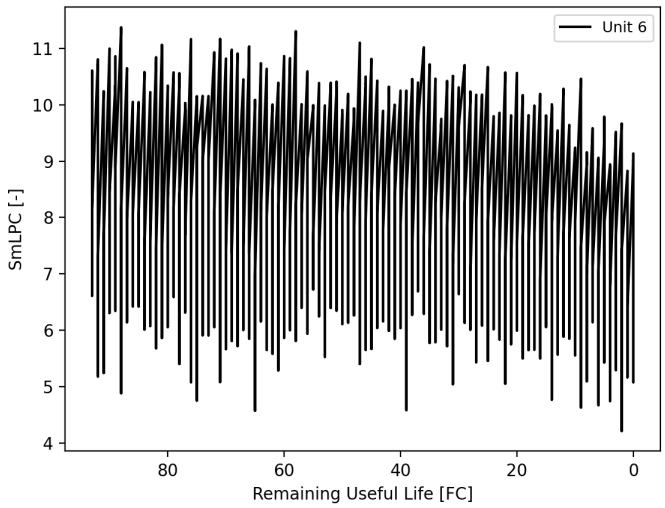


Figure 3. Raw data of SmLPC sensor for unit 6 of flight class 2 (DS01) in the development set.

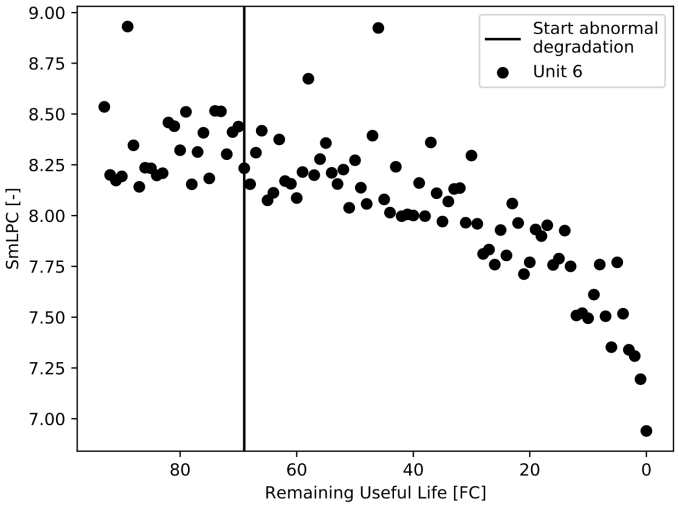


Figure 4. Smoothed data of SmLPC sensor for unit 6 of flight class 2 (DS01) after applying the FC average smoothing method.

To be precise, when smoothing a unit's sensor values for a specific FC, only the data corresponding to this specific FC is required (not data from a previous or future FC). This is considered beneficial as data from a future FC would obviously be not available when smoothing sensor data from a test unit in reality. The reason the FC average method can be used is because the vertical lines in the raw data shown in Figure 3 (the range of sensor values obtained during a FC is shown as a vertical line per FC), have approximately the same vertical length throughout a unit's lifetime. In case the length of the vertical lines would have increased or decreased considerably over a unit's lifetime, it would not have been possible to compute a sensor average per FC (as that would result into unfair comparisons).

Naturally, the FC average method has been applied for all 28 sensors listed in Table 4 and Table 5. An example of applying this method on the exact same $SmLPC$ sensor (of unit 6, flight class 2 (DS01)) as used in Figure 3, is depicted in a scatter plot and shown in Figure 4. The thin vertical black line denotes the moment a particular failure mode becomes active and causes a unit's degradation state to transform from normal degradation, caused by regular wear and tear, to abnormal degradation. Figure 4 reveals a clear decreasing trend as the unit degrades over time. During its early lifetime, the smoothed sensor value indicates a shallow deterioration trend (normal degradation), whereas this decreasing trend exponentially grows after the abnormal degradation has started. The few distant smoothed values, in the beginning of the unit's lifetime, are expected to be flights which behaved slightly different in comparison to the other 'regular' flights, hence the slightly different smoothed values.

3.3.2 LOWESS smoothing method

The second applied method for noise reduction, data smoothing and data downsampling is the commonly used Locally Weighted Scatterplot Smoothing (LOWESS) method [30], [31], [32]. LOWESS is a novel nonparametric data smoothing method which uses locally weighted least squares to obtain for each point ((x_i, y_i), $i = 1, ..., n$), a single smoothed y-value ($\hat{y}_i$) per corresponding x-value (x_i) [33]. Here, the y-values are the sensor values belonging to the corresponding FC (x-values). Its main benefit is obtaining highly smoothed sensor values per FC.

In essence, its working principle can be described as follows. For each x_i, certain weights $w_j(x_i)$ are defined for all x_j, with $j = 1, ..., k$ (k-nearest neighbor data points) using a weight function W [33]. The weights are large in case x_j is close to x_i and small in case it is not. The primary fitted y-value, $\hat{y}_i$, per corresponding x_i-value is obtained by fitting a dth degree polynomial function to the k-nearest neighbor points using weighted least squares with weights $w_j(x_i)$ [33]. This initial fitting of smoothed $\hat{y}_i$-values is called locally weighted regression [33]. In order to prevent outliers from distorting the initially smoothed points, an additional weight δ_j is assigned per (x_j, y_j) based on the residual $y_j - \hat{y}_i$ [33]. The weights are large if the residual is small, and small in case the residual is large. The new smoothed y-values are fitted the same way as previously described, however, the weights $w_j(x_i)$ are replaced by $\delta_j w_j(x_i)$ [33]. This process is called robust locally weighted regression [33]. For more information regarding the mathematics of LOWESS, a detailed LOWESS calculation procedure is provided in Appendix A.

The LOWESS method has been applied for all 28 sensors listed in Table 4 and Table 5. Figure 5 depicts a scatter plot with the results of applying LOWESS on the same $SmLPC$ sensor as shown previously in Figure 3 for unit 6 of flight class 2 (DS01). Again, the thin vertical black line denotes the moment a particular failure mode becomes active and causes a unit's degradation state to transform from normal degradation, caused by regular wear and tear, to abnormal degradation [24]. As can be seen from Figure 5, an even clearer deterioration trend becomes visible (compared to Figure 4) after the severe

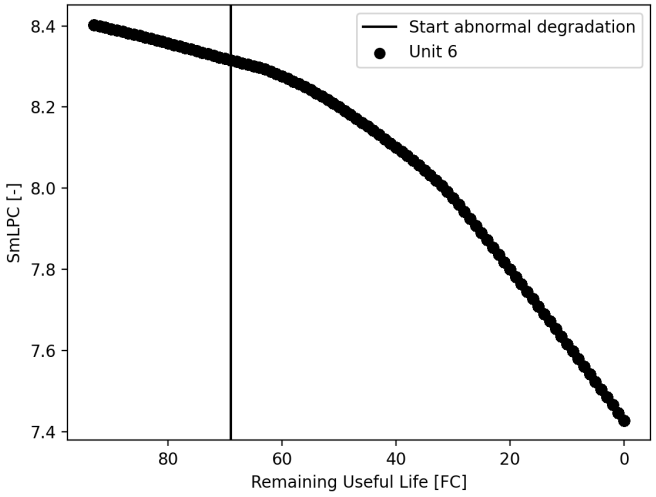


Figure 5. Smoothed data of SmLPC sensor for unit 6 of flight class 2 (DS01) after applying LOWESS smoothing method.

noise has been removed and a single smoothed SmLPC-value per FC has been obtained. Besides the clearly visible deterioration trend, Figure 5 also shows the linear normal degradation trend transitioning into an abnormal degradation trend as the impact of the failure mode develops over time.

Nevertheless, this clearer deterioration trend does not automatically imply more accurate RUL point estimations on the test units. This is explained by the fact that during model training, a unit's entire lifetime can be used for LOWESS to obtain smoothed sensor values per FC. However, when obtaining smoothed sensor values per FC for test units during their initial life, the entire amount of data (from its start to EoL) is obviously not available. Therefore, such a smoothed sensor value per FC is likely to be slightly different from the value it would have had in case the entire amount of data would have been available for data smoothing. This slight difference in smoothed sensor values obtained at FC during a unit's initial life, might influence the accuracy of the RUL point estimations.

Due to this reason as well as to prevent an overwhelming number of tables and graphs, the results of feature selection (paragraph 3.4) and hyperparameter tuning (paragraph 3.6) obtained for sensor data that has been smoothed with LOWESS, will be shown in Appendix B and Appendix C, respectively. Hence, only the results of feature selection and hyperparameter tuning obtained for sensor data that has been smoothed using the FC average method will be shown in paragraph 3.4 and paragraph 3.6, respectively.

3.4 Feature selection

Throughout this paragraph, the process of feature selection using Python's Recursive Feature Elimination with cross validation (RFECV) method (from the Scikit-learn package [34]) is described together with the selected features (sensors) that will be used for the algorithm performing the RUL estimations. Note that this paragraph only contains the results for data that has been smoothed using the FC average method described in subparagraph 3.3.1, as explained above. Besides discussing the selected features throughout this paragraph, an additional brief physical meaning is provided for the selected features of DS01 in subparagraph 3.4.1.

Feature selection is defined as the process of selecting the most relevant features required to predict the target variable (a unit's RUL) [35]. It is an essential step required to increase the model's prediction accuracy and decrease its training time by eliminating redundant and irrelevant features [35]. RFECV is a so-called wrapper method as it determines the most relevant features based on the performance of an underlying algorithm [36]. It has been decided to use RFR as the underlying algorithm, which working principle is explained in paragraph 3.5.

Generally, given an initial set of features, RFECV recursively performs the following steps until one feature remains [37]:

1) For every fold of data (generated by a cross validation (CV) method), train an untuned RFR algorithm on the set of features and compute the feature importance (concept explained in paragraph 3.5) for every feature in this set.
2) Test the trained RFR algorithm on a test fold and repeat this process of training and testing on folds (with the features in step 1) to calculate an average CV score (expressed as the Root Mean Squared Error, RMSE shown in Equation 1).
3) Store this average CV score (obtained in step 2) belonging to this specific number of features on which the RFR algorithm has been trained and tested. Additionally, remove the least important feature in every train fold of the data.

After recursively performing the steps above until one feature is left in the initial given set of features, the number of features having the lowest CV score (computed as RMSE according to Equation 1 [34]) is selected by RFECV. In Equation 1, N denotes the number of samples, y_j the true (RUL) value of the j-th sample and $\hat{y}_j$ the corresponding estimated (RUL) value. Therefore, RMSE is expressed in FC.

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=0}^{N-1}(y_j - \hat{y}_j)^2} \tag{1}$$

Note that RFECV does not "know" the names of the actual features, but only the number of features having the lowest CV score. Thus, a Recursive Feature Elimination (RFE) method (also from the Scikit-learn package [34]) is run to obtain the actual feature names that belong to the number of features having the lowest CV score. The RFE method is very similar to RFECV, except this method does not perform CV and runs on the entire development set. It performs the following steps until the number of features remaining in the set is equal to the number of features obtained by RFECV (which has the lowest average CV score):

1) Train an untuned RFR algorithm with all features from the feature set.

2) Compute the feature importance of each feature.
3) Discard the least important feature from the feature set.
4) Perform steps 1 - 3 until the number of features in the feature set equals the number of features obtained by RFECV.

After performing these steps, the most relevant features to estimate a unit's RUL are obtained. Therefore, RFECV is able to automatically select the optimal number of relevant features and obtain their accompanying feature importances.

The average CV score is obtained using Python's Time Series Split (also from the Scikit-learn package) which is a variation of the commonly used K-fold CV [34]. Although K-fold CV is considered to be one of the most frequently used methods for CV and often applied to time series data such as done in [38], [39] and [40], it is inappropriate for time series data [41]. K-fold CV assumes the data to be independent and identically distributed, which is an invalid assumption for time series data as future observations might depend on past observations [41].

Time Series Split however, does take into account time-dependence of observations, as it splits a given time series data into so-called train and test subsets, where in each test subset, the test indices must be higher than in the training subset [34]. Figure 6 depicts a representation of the Time Series Split with a total of four iterations.

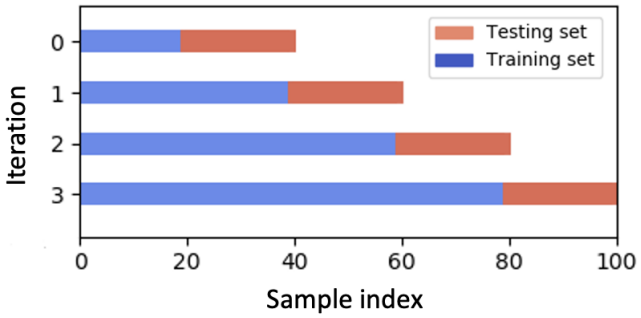


Figure 6. Representation of Time Series Split with four iterations [34].

As can be seen in Figure 6, in order to have four iterations, the total dataset must be split into five equal parts. Then, during the kth iteration, k-folds are used for training whereas the (k+1)th fold is used as the test set to evaluate the model and obtain a RMSE score [34]. It has been decided to use $n_{splits} = 5$ (parameter for total number of splits) for the Time Series CV as this results into the same number of equally sized subsets in case the commonly used K-fold CV with $k = 5$ would have been applied as done in [37], and [39].

The selected features for DS01, after performing RFECV, with their corresponding feature importances (concept explained in paragraph 3.5) can be seen per flight class RFR model in Table 12. Table 13 and Table 14 contain the selected features with corresponding feature importances per flight class RFR model for DS02 and DS03, respectively. Furthermore, the associated CV scores per flight class RFR model in DS01, DS02 and DS03 can be seen in Table 15. These CV scores are considered as the baseline errors that need to be reduced through hyperparameter tuning (explained in paragraph 3.6). Note that these tables (showing the selected features, feature importances and CV scores) obtained for data that has been smoothed using LOWESS can be found in Appendix B. Also note that Table 13 and Table 15 only contain a single flight class RFR model for DS02 because the development set of DS02 only consists of units from flight class 3.

Table 12
SELECTED FEATURES WITH CORRESPONDING FEATURE IMPORTANCES OF DS01 FOR FLIGHT CLASS RFR MODELS 1,2 AND 3.

Flight class RFR model	Selected features	Feature importance [-]
1	T50	0.3156
	SmLPC	0.2802
	SmHPC	0.2698
	Nc	0.0844
	SmFan	0.0500
2	SmLPC	0.8399
	SmHPC	0.0935
	SmFan	0.0666
3	SmLPC	0.6838
	SmHPC	0.1792
	SmFan	0.1370

Table 13
SELECTED FEATURES WITH CORRESPONDING FEATURE IMPORTANCES OF DS02 FOR FLIGHT CLASS RFR MODEL 3.

Flight class RFR model	Selected features	Feature importance [-]
3	phi	0.2707
	SmLPC	0.2652
	SmHPC	0.2224
	Nc	0.0856
	Nf	0.0629
	T50	0.0594
	T24	0.0338

3.4.1 Physical meaning of selected features

Apart from the fact that RFECV selected features that yielded the lowest CV scores, it can also be explained why the selected features in DS01 are relevant from a physical perspective. Given that all units from DS01 are suffering from HPT efficiency deterioration, it is likely that it could result into the HPT being incapable of efficiently transforming the hot gas into energy, thus generating less power [42]. As a consequence, the fuel flow Wf must be increased in order to still generate the same amount of power, hence causing the

Table 14
SELECTED FEATURES WITH CORRESPONDING FEATURE IMPORTANCES OF DS03 FOR FLIGHT CLASS RFR MODELS 1,2 AND 3.

Flight class RFR model	Selected features	Feature importance [-]
1	SmLPC	0.4550
	T50	0.2530
	Nc	0.1057
	SmHPC	0.1040
	W25	0.0822
2	SmLPC	0.5742
	SmHPC	0.1067
	T50	0.0787
	SmFan	0.0641
	Nc	0.0480
	phi	0.0380
	Nf	0.0318
	T48	0.0230
	T24	0.0285
3	SmLPC	0.5352
	SmHPC	0.1528
	SmFan	0.0860
	Nc	0.0749
	T50	0.0546
	Nf	0.0347
	phi	0.0325
	T48	0.0292

Table 15
BASELINE CV SCORES FROM RFECV OF DS01, DS02 AND DS03 FOR FLIGHT CLASS RFR MODELS 1,2 AND 3.

Data subset	Flight class RFR model	Baseline CV score (RMSE) [FC]
DS01	1	22.1953
	2	20.6365
	3	20.1936
DS02	3	16.2246
DS03	1	17.9665
	2	17.9310
	3	16.4581

temperature $T50$ to increase [42]. As a result, the entire unit will rotate faster (Nc will eventually increase). An effect of a faster rotating unit is a decrease in stall margins $SmLPC$, $SmHPC$ and $SmFan$ [42]. Note that this explanation can only be provided for units in DS01 as those units have only one single failure mode. For units in DS02 and DS03, this can not be explained as they are affected by multiple simultaneous failure modes.

These multiple simultaneous failure modes make it too complex to explain why the selected features are relevant from a physical perspective.

3.5 RUL estimation algorithm

This paragraph explains the working principle of the RFR algorithm that is used to generate the RUL estimations. Furthermore, a brief explanation on how the RFR algorithm is used to model the uncertainty of the RUL estimations is provided in subparagraph 3.5.1.

RFR is a decision tree-based ensemble machine learning algorithm which grows individual trees using bootstrapping and aggregates the outcomes of the trees into a single estimation [43], [44]. Bootstrapping ensures randomized trees since individual trees are grown using a bootstrap sample of the dataset [44]. In order to randomize the growing of decision trees even further, random subsets of a given set of features are used when creating splits in the trees [44]. During the creation of such a split, RFR attempts to maximize the decrease in impurity for that split [44]. Generally for regression, this decrease in impurity is measured using the Mean Squared Error (MSE) (with the MSE being the impurity value of a certain node m) as given in Equation 2 [45].

$$MSE(m) = \frac{1}{N_m} \sum_{j=1}^{N_m} (y_j - \bar{y}_j)^2 \tag{2}$$

With N_m being the number of samples in the considered node m, y_j denotes the sample value of the jth sample, and $\bar{y}_j$ representing the average value of the N_m samples in node m. Sometimes, the MSE is referred to as the variance of the samples in a node [45]. Since RFR aims to maximize impurity decrease when creating splits (growing a tree), it is also capable to provide a measure of feature importance [46]. Feature importance describes the contribution of an individual feature to the target (i.e. RUL estimation) when splitting a node [47]. This measure of feature importance is usually given as the Mean Decrease of Impurity (MDI) [44].

In order to calculate the MDI, the general working principle of RFR to generate RUL estimations (which is implemented using Python's Scikit-learn package [34]) must be explained.

Let N represent the bootstrapped samples available to build a decision tree H. Also, let m be a node having N_m samples and let y_j denote the jth sample of N_m. Furthermore, let $\theta = (k, z)$, with feature k and threshold z, be candidate splits for node m. Moreover, let node m be written as $m(\theta)$ when considering a certain candidate split.

Each decision tree H is grown by recursively performing the following steps for every node m. At first, a subset of randomly selected features (from the bootstrapped data available to build tree H) is created [48]. Then, for each candidate split of node m ($m(\theta)$), this subset of randomly selected features is split into a left child node ($m(\theta)^{left}$) if $y_j \leq z$ and a right child node ($m(\theta)^{right}$) if $y_j > z$, respectively [48].

Hereafter, the optimal split for node m ($opt(m)$) is obtained by maximizing the impurity decrease which given by Equation 3:

$$\Delta_I(m(\theta)) = C(m(\theta)) - \frac{N^{left}_{m(\theta)}}{N_{m(\theta)}} C(m(\theta)^{left}) - \frac{N^{right}_{m(\theta)}}{N_{m(\theta)}} C(m(\theta)^{right}) \quad (3)$$

Where $C(m(\theta))$ denotes the impurity measure (given by Equation 2) of node m for a specific θ, whilst $C(m(\theta)^{left})$ and $C(m(\theta)^{right})$ represent the same for the left and right child nodes of node m, respectively. Furthermore, $N_m(\theta)$ denotes the samples in node m, whereas $N_m(\theta)^{left}$ and $N_m(\theta)^{right}$ are the samples for the left and right child nodes of node m, respectively. Finally, the decrease in impurity of node m for a specific θ is given by $\Delta_I(m(\theta))$.

The growing of trees stops in case one of the following situations occurs: 1) all nodes are pure (impurity does not decrease anymore), 2) the child nodes of node m contain less samples than a certain threshold called *min_samples_leaf*, 3) the number of samples in node m (N_m) is less than a certain threshold called *min_samples_split*, 4) the maximum depth of a tree reached a certain threshold called max_depth [34], [48].

Then, the MDI (feature importance) for a certain feature w, obtained for a single decision tree H, is given by Equation 4 as [48]:

$$MDI(w, H) = \sum_{m \in H, opt(m)=w} \frac{N_{m(\theta)}}{N} \Delta_I(m(\theta)) \quad (4)$$

In which $opt(m) = w$ means that only nodes m for which the maximum impurity decrease is obtained when splitting on feature w, will be considered. Therefore, Equation 4 sums the maximum impurity decreases for all nodes in a single decision tree that have been split using a particular feature, and multiplies that by a ratio [44]. This ratio decreases for splits 'deeper' in the tree, causing those splits to be of less importance compared to splits made in the beginning of the tree.

Finally, the global feature importance of feature w on the entire RF ($MDI_{global}(w)$), having a number of trees equal to $n_{estimators}$, can be calculated by Equation 5:

$$MDI_{global}(w) = \frac{1}{n_{estimators}} \sum_{b=1}^{n_{estimators}} MDI(w, H_b) \quad (5)$$

Where H_b denotes the bth decision tree. After creating an entire RF (i.e. growing a number trees equal to $n_{estimators}$ with each generating an estimated RUL value) and determining the feature importance for all given features, the RUL estimations will be aggregated into a single RUL point estimation by computing the average of all estimations.

3.5.1 Modelling RUL estimation uncertainty

In order to obtain the uncertainty of the RUL point estimations generated with RFR, the general working principle of RFR must be slightly modified. Instead of aggregating the RUL estimations of all individual decision trees together into a single point estimation, the RUL estimations of all individual trees must be obtained separately. Hereafter, these individual tree RUL estimations will be visualized in a histogram in order to display the probability distribution with the probabilities of these individual tree estimations and hence, the uncertainty of the RUL point estimations.

3.6 Hyperparameter tuning

This paragraph discusses the need for hyperparameter tuning as well as the tuning method that has been used. Furthermore, it describes the specific hyperparameters that have been tuned and the results of the tuning (hyperparameter values and CV score). Note that this paragraph only contains the results for data that has been smoothed using the FC average method described in subparagraph 3.3.1, as explained previously. The results of the hyperparameter tuning for data that has been smoothed using LOWESS is provided in Appendix C.

Hyperparameter tuning is an important step as it allows for better overall performance by revealing the optimal hyperparameter values of the RFR model [49]. A hyperparameter is a parameter which value must be set before training the model. This is in contrast with a general model parameter, where its value is learned during model training [50]. The hyperparameter tuning is performed using the popular random grid search with CV as well as the exhaustive grid search with CV (both from the Scikit-learn package [34]) [49]. Given a defined search grid for certain hyperparameters, random grid search will randomly pick a value per hyperparameter from the search grid, whereafter it will train a RFR model [34].

Then, CV is performed using Time Series Split to obtain a cross validated RMSE score for the specific parameter values. The number of random searches is determined to be 10% of the total number of combinations that can be obtained in the search grid. Hereafter, the hyperparameter values that yield the lowest CV score are used to define a new search grid around those values. With this new search grid, exhaustive grid search with CV is performed in which every combination of hyperparameter values is used and a CV score obtained for each combination [34]. Finally, the combination of hyperparameter values yielding the lowest CV score are considered to be the optimal values for the RFR model.

Table 16 shows the hyperparameters that have been tuned (as well as their tuned values) for the RFR of flight class RFR models 1, 2 and 3 of DS01, DS02 and DS03. The tuned RFR hyperparameters of the flight class RFR models of DS01, DS02 and DS03 obtained for data that has been smoothed with LOWESS is shown in Appendix C. The ranges stated in the caption of Table 16 are defined based on the ranges used in [39], [43] and [51]. The remaining RFR parameters are left at default in accordance with [39], [43] and [51].

Table 16
RFR HYPERPARAMETERS FOR FLIGHT CLASS RFR MODELS 1, 2 AND 3 OF DS01, DS02 AND DS03. THE (INITIAL) RANGES OF THE SEARCH GRID ARE THE FOLLOWING: NUMBER OF TREES [100, 300, 500, 800, 1000, 1200], SPLIT CRITERION [MSE, POISSON], MAXIMUM TREE DEPTH [4, 6, 8, 10, 15, 25, 40, 100, NONE], MINIMUM SAMPLES PER SPLIT [2, 4, 8, 10], MINIMUM SAMPLES PER LEAF NODE [1, 3, 5, 7, 9] AND FRACTION OF FEATURES PER SPLIT [0.25, 0.50, 0.75, AUTO].

Data subset	Flight class RFR model	Number of trees	Split criterion	Maximum tree depth	Minimum samples per split	Minimum samples per leaf node	Fraction of features per split
DS01	1	100	MSE	100	2	1	0.8
	2	290	MSE	98	3	1	Auto
	3	90	MSE	100	2	1	Auto
DS02	3	90	MSE	100	3	1	Auto
DS03	1	320	MSE	98	2	1	Auto
	2	280	MSE	23	3	1	Auto
	3	90	MSE	100	2	1	Auto

Table 17
FINAL CV SCORES AFTER HYPERPARAMETER TUNING FOR FLIGHT CLASS RFR MODELS 1,2 AND 3 OF DS01, DS02 AND DS03.

Data subset	Flight class RFR model	Final CV score (RMSE) [FC]
DS01	1	22.1067
	2	20.4486
	3	20.1148
DS02	3	16.0536
DS03	1	17.8219
	2	17.8616
	3	16.4225

Note that for the hyperparameter "Fraction of features per split" the value "Auto" means that the RFR model will consider all available features when determining the best feature to split on. When examining the tuned values in Table 16, it can be seen that some of these are different compared to the values listed in the (initial) ranges in the caption. This is explained by the fact that the ranges have been refined for the exhaustive grid search based on the results of the random grid search, as explained previously. The new (final) CV scores obtained on the development sets of DS01, DS02 and DS03 (with the optimal hyperparameter values) for all flight class RFR models are listed in Table 17. It can be seen that the final CV scores improved for all flight class RFR models in all data subsets, which implies hyperparameter tuning to be an important step with respect to increasing model performance. However, the CV scores only slightly improved, which indicates that hyperparameter tuning does not have a substantial effect on these specific data subsets.

4 RESULTS

Throughout this section, the results of the various flight class RFR models for DS01, DS02 (validation data subset) and DS03 (verification data subset) are discussed. These results consist of the RUL point estimations and accompanying probability distributions (depicting the uncertainty of the RUL point estimations), and are discussed in paragraph 4.1 for data smoothed with the FC average smoothing method. The results for data smoothed with LOWESS can be found in Appendix D. Furthermore, in paragraph 4.2, various evaluation metrics such as the RMSE and Mean Absolute Error (MAE, shown in Equation 6 [34]) are used to compare the results of the different flight class RFR models in DS01, DS02 and DS03 (both for data smoothed with the FC average method as well as with LOWESS). In Equation 6, N denotes a unit's total number of RUL point estimations (since a RUL point estimation per FC is generated for a given unit), y_j the true RUL value of the jth RUL point estimation and $\hat{y}_j$ the corresponding RUL point estimation value itself. Consequently, MAE is expressed in FC.

$$MAE = \frac{1}{N} \sum_{j=0}^{N-1} |y_j - \hat{y}_j| \tag{6}$$

In order to obtain the RUL point estimations and probability distributions, the exact same steps with respect to noise removal, data smoothing and data cleaning (as described in paragraph 3.3) have been performed on the test datasets of DS01, DS02 and DS03. Thereafter, the RFR flight class models of DS01, DS02 and DS03 were trained on the entire corresponding development sets (using the hyperparameter values in paragraph 3.6). With these tuned models, RUL point estimations have been obtained for all available test units per flight class RFR model in DS01, DS02 and DS03. Given that the total number of test units equals 11 (four test units in DS01, one test unit in DS02 and six test units in DS03), it has been decided to present the graphs of the RUL point estimations and accompanying probability distributions for a selection of units per flight class RFR model for DS01 and DS03. That is, per flight class RFR model, these graphs are shown for a particular test unit from DS01 as well as a particular test unit from DS03. The remaining RUL point estimation graphs and corresponding probability distributions of test units in DS01,

DS02 and DS03, together with a brief analysis, can be found in Appendix E.

4.1 RUL point estimations and corresponding uncertainty (DS01 and DS03)

This paragraph thoroughly discusses the RUL point estimations and accompanying probability distributions obtained with the flight class RFR models in DS01 and DS03.

For a selection of test units in DS01 and DS03, a graph is generated depicting the RUL point estimations (which are generated per FC and start from a unit's 10th FC) as well as three histograms showing the probability distributions of the first, middle and last RUL point estimations, respectively. The RUL point estimations for all test units are started at the 10th FC due to two reasons. Firstly, starting estimations at the 10th FC allows the test units to accumulate some data before starting the estimations. Secondly, starting estimations at a unit's very first flight is meaningless as there would be an insufficient amount of available data for data smoothing with LOWESS. The reason to produce the probability distributions of the first, middle and last RUL point estimations of each test unit is to provide an accurate overview of the uncertainty of the point estimations, without creating an overwhelming number of histograms. Figure 7 and Figure 8 depict the RUL point estimations and associated probability distributions of test units 7 and 14 (both flight class RFR model 1), respectively. The RUL point estimations and associated probability distributions of test units 8 and 15 (both flight class RFR model 2) as well as test units 10 and 11 (both flight class RFR model 3) are depicted in Figure 9, Figure 10, Figure 11 and Figure 12, respectively. When examining these figures, numerous interesting results can be observed.

First of all, the overall results of all test units are considered to be satisfactory, both in terms of RUL point estimations as well as the shapes of the probability distributions. Nevertheless, the overall results (in terms of RUL point estimations) of test units of DS01 are found to be less accurate compared to the results of test units of DS03, especially during the first half of the RUL point estimations. The main reason for these overall less accurate results is believed to be the exceptional low number of training units. As a matter of fact, test units 7 and 8 in DS01 (Figure 7 and Figure 9, respectively) had only a single training unit available, whereas test unit 10 in DS01 had two training units available during training of their corresponding flight class RFR models. Moreover, it is understandable that especially the first half of the RUL point estimations is less accurate because the test units might follow a slightly different deterioration trend compared to the deterioration trends on which the units have been trained. These possible slightly different deterioration trends during the test units initial lives could result into largely different RUL point estimations, due to the way in which thresholds were created by RFR when constructing the forests.

Furthermore, it is generally found to be more difficult to obtain accurate estimations during the test units initial lives, as their estimation windows (time from a certain estimation to a unit's EoL) are still large. This is supported by the observation that the RUL point estimations of all test units in DS01 converge to the true RUL values as the units approach their EoL. Additionally, it is supported by the shape of the probability distributions corresponding to the last RUL point estimations (true RUL of zero FC) as shown in Figure 7, Figure 9 and Figure 11. As can be observed, the main peaks of these probability distributions are located at, or close to a RUL value of zero FC which is in line with the expected peak locations of these distributions. Furthermore, when comparing the individual test units in DS01, it can be seen that test unit 7 (Figure 7) has the most accurate overall results in terms RUL point estimations, despite the fact it has been trained on a single training unit. This is a remarkable result and expected to be caused by the test unit fairly accurately following the training unit.

Another remarkable outcome is found for the results of test units in DS03 (units 11, 14 and 15), as they are noticeably better (especially the RUL point estimation graphs) compared to those of the test units in DS01 (units 7, 8 and 10). This is remarkable given the complexity of DS03 compared to DS01 (three different simultaneous failure modes affecting the units versus one failure mode affecting the units, respectively). The likely explanation for these more accurate RUL point estimations is the fact that the test units of DS03 have been trained on a larger number of training units compared to the test units of DS01 (test units of DS03 were trained on a total of nine units, whereas test units of DS01 were trained on a total of four units). Nevertheless, these generally more accurate results for test units in DS03 verify that the developed model as a whole is capable of being used on simple as well as considerably more complex datasets. Here, complexity refers to the number of different simultaneously failure modes present in the units of a dataset. Moreover, comparing the overall RMSE and MAE scores (obtained using the FC average method as the data smoothing method) that are computed by producing RUL point estimations for every FC starting from a unit's 10th FC (these scores are shown in the FC average entries of Table 18 that is discussed in paragraph 4.2), shows that these scores are approximately equal to or sometimes lower than the scores obtained for test units in DS01. This is considered as an additional way of verifying that the developed model is capable of correctly handling various types of datasets.

However, further analysis of the RUL point estimation graphs of test units in DS03 shows that, especially test units 15 (Figure 10) and 11 (Figure 12), also have fairly large deviations from the corresponding true RUL values during the first half of the estimations (and tend to converge to the true RUL values as the units approach their EoL). As explained, this might be caused by the test units initially following a slightly different deterioration trend, which could lead to largely different RUL point estimations due to the way in which thresholds were created by RFR. Additionally, in general, obtaining accurate estimations during test units initial lives is more difficult because of the large estimation windows.

Besides this aspect of test units having fairly large deviations compared to their true RUL values during the first half of the estimations, two other interesting aspects can be noticed when comparing the results of test units in DS03. Firstly, the RUL point estimations of test unit 14 (Figure 8) are the closest to the true RUL values. Although a specific extensive explanation of the behaviour of this single unit cannot be given due to the complex nature of RFR, it is likely that the test unit is adequately following the degradation patterns as learned by the RFR flight class model during training. Secondly, the RUL point estimations of test unit 15 (Figure 10) show relatively large deviations between a true RUL of 30 FC and 20 FC. Again, a specific extensive explanation can unfortunately not be provided. However, it is likely that those specific flights behaved slightly different (in terms of throttle setting) compared to the other flights of test unit 15. This different throttle setting might have been required due to for example different wind speeds (jet streams) during these specific flights. In case tail jet streams were present because of a certain storm, it could have been required to set a lower throttle setting, hence resulting into different smoothed values (and ultimately different estimated RUL values).

When considering the histograms of the test units from DS01 as well as DS03, it can be seen that the majority of the histograms has a wide spread, but with clear peaks centered at certain RUL values (this wide spread is especially visible in histograms corresponding to the first and middle RUL point estimations). This behaviour is considered to be natural for RFR as its power lies in using "wisdom of the crowd". As explained, in RFR, single uncorrelated decision trees are built which might be under or overestimating the true RUL. However, by averaging the estimations of all trees (using "wisdom of the crowd") RFR is generally able to perform fairly accurate RUL point estimations. This is also supported by closely examining the probability distributions of all test units, as it is found that the probabilities of correctly estimating the true RUL values are generally found to be low. Nevertheless, inspecting the accuracy of the corresponding RUL point estimations shows that combining the RUL estimations of all trees into a single RUL point estimation results into more accurate RUL point estimations, which converge towards the true RUL values as the units approach their EoL. Additionally, it can be observed that the wide spread of the probability distributions decreases over time (hence the uncertainty decreases) as the units reach their EoL. This shows that the flight class RFR models are capable of converging to the true RUL values, not only in terms of accuracy, but also in terms of uncertainty.

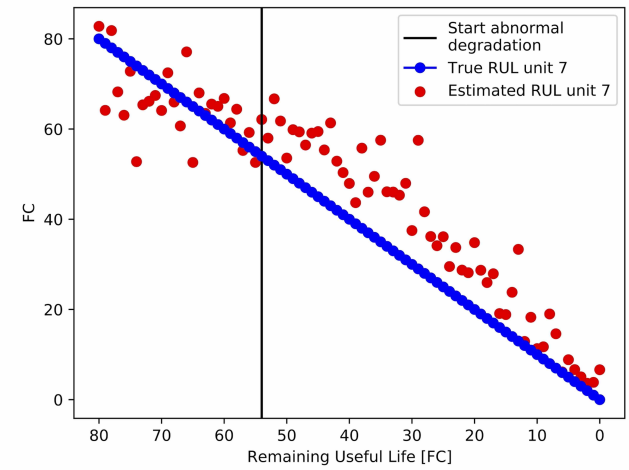
4.2 RUL point estimation errors (DS01, DS02 and DS03)

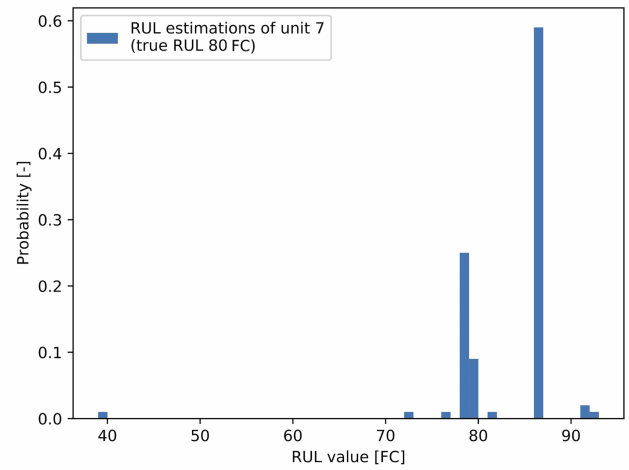
This paragraph discusses the results of the RUL point estimations (in terms of errors) obtained with the flight class RFR models in DS01, DS02 and DS03. Note again that these RUL point estimations are obtained for every FC (starting from a unit's 10th FC). Also, note again that the results for DS02 are only obtained for test unit 11 from flight class RFR model 3, as this is the only present flight class RFR model in DS02 (with solely this single corresponding test unit). Finally, note that this paragraph contains both the results for test units which used the FC average method for data smoothing as well as units which used LOWESS for data smoothing.

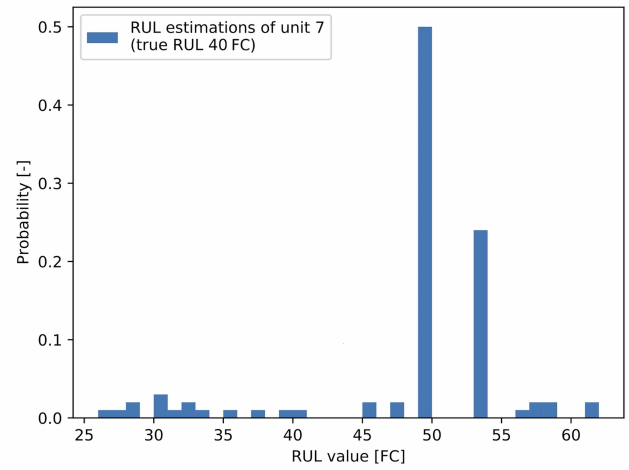
For each test unit per flight class RFR model in the data subsets, the RMSE, MAE, Minimum Absolute Error (Min. Abs. Error) as well as Maximum Absolute error (Max. Abs. Error) are computed based on the RUL point estimations. This is clearly shown for the FC average as well as LOWESS entries in Table 18. The utmost remarkable result shown in Table 18 is the fact that the errors (especially the RMSE, MAE and Minimum Absolute Error) for the far majority of the test units, from which their data has been smoothed using the FC average method, is considerably lower compared to the errors of the test units of the LOWESS entries. Consequently, it can be concluded that using the FC average as a smoothing method, has a large positive effect on the developed model as a whole.

This is also supported by the fact that the Maximum Absolute Errors, obtained when using LOWESS, are generally substantially larger, compared to the Maximum Absolute Errors obtained when using the FC average method. Thorough analysis reveals that only for test unit 8 of DS01, the Maximum Absolute Error for which the data has been smoothed with the FC average method is slightly higher (0.3 FC), compared to the Maximum Absolute Error of this unit for which the data has been smoothed with LOWESS, as can be seen in Table 18. Although a precise explanation for this minor difference can not be provided due to the complex nature of RFR, it has been found that these particular Maximum Absolute Error values for test unit 8 of DS01 (for the FC average entry and the LOWESS entry, respectively), are obtained for estimations corresponding to different true RUL values. To be precise, the Maximum Absolute Error corresponding to the FC average entry is obtained at a true RUL of 31 FC, whilst the Maximum Absolute Error corresponding to the LOWESS entry is obtained at a true RUL of zero FC.

Hence, it additionally indicates that using LOWESS yields less accurate results since it is expected to provide better results as the unit approaches its EoL, due to the fact that more data becomes available for smoothing with LOWESS. As a result, the remainder of this paragraph will exclusively focus on the errors corresponding to the FC average entries in Table 18. Thorough examination of the FC average entries in Table 18 shows that test unit 8 of DS01 has the overall highest RMSE score (where it also has the overall highest Maximum Absolute Error, as mentioned previously). Having this highest overall RMSE score is expected to be caused by the fairly large difference between the RUL point estimations and corresponding true RUL values in the interval between a true RUL of 50 FC and 30 FC (as can be seen in Figure 9). These fairly large deviations are believed to be caused by the test unit following an initially less steep deterioration pattern, in comparison to the pattern learned from its single training unit.







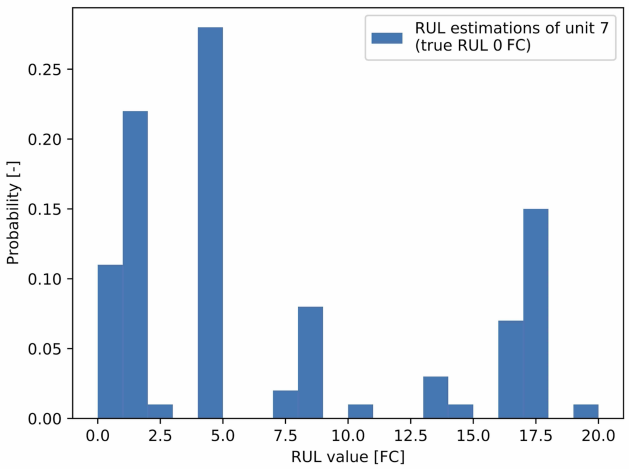
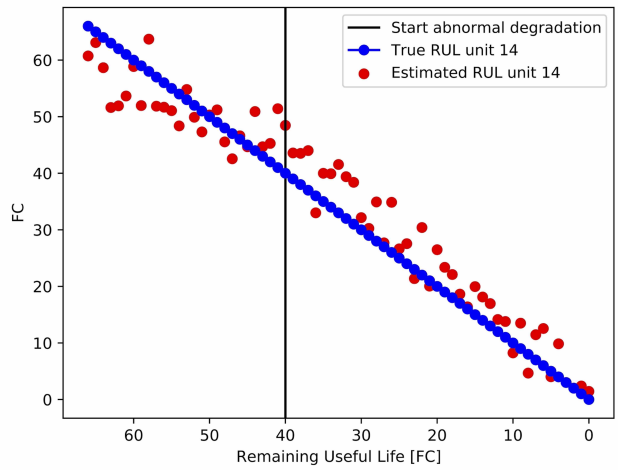
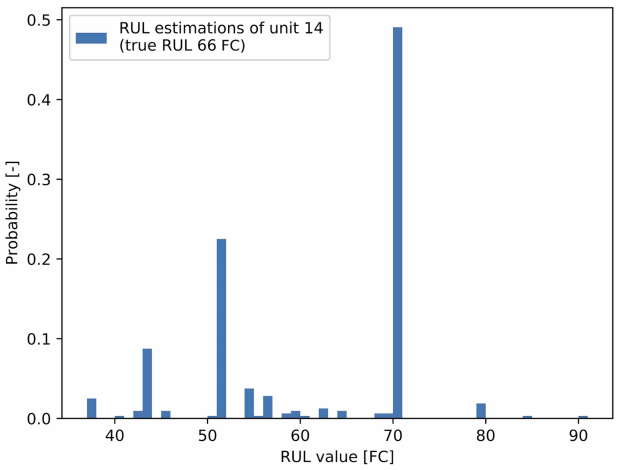
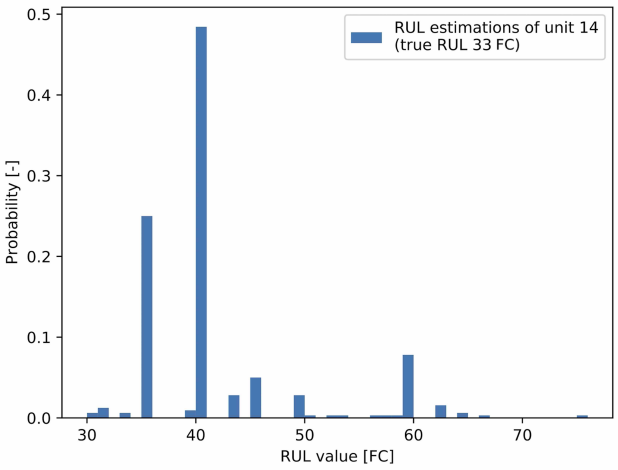


Figure 7. RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 7, DS01.







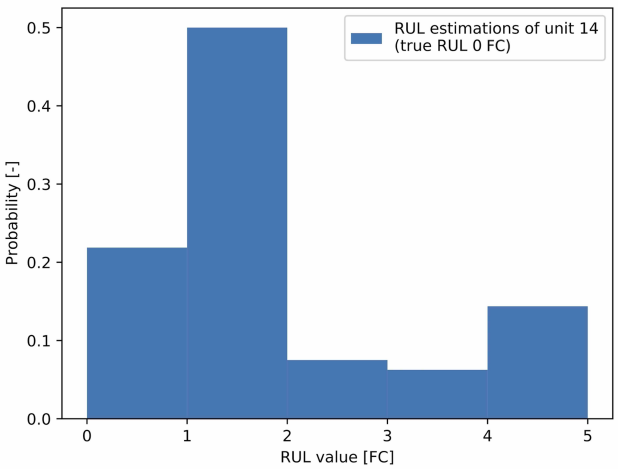


Figure 8. RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 14, DS03.

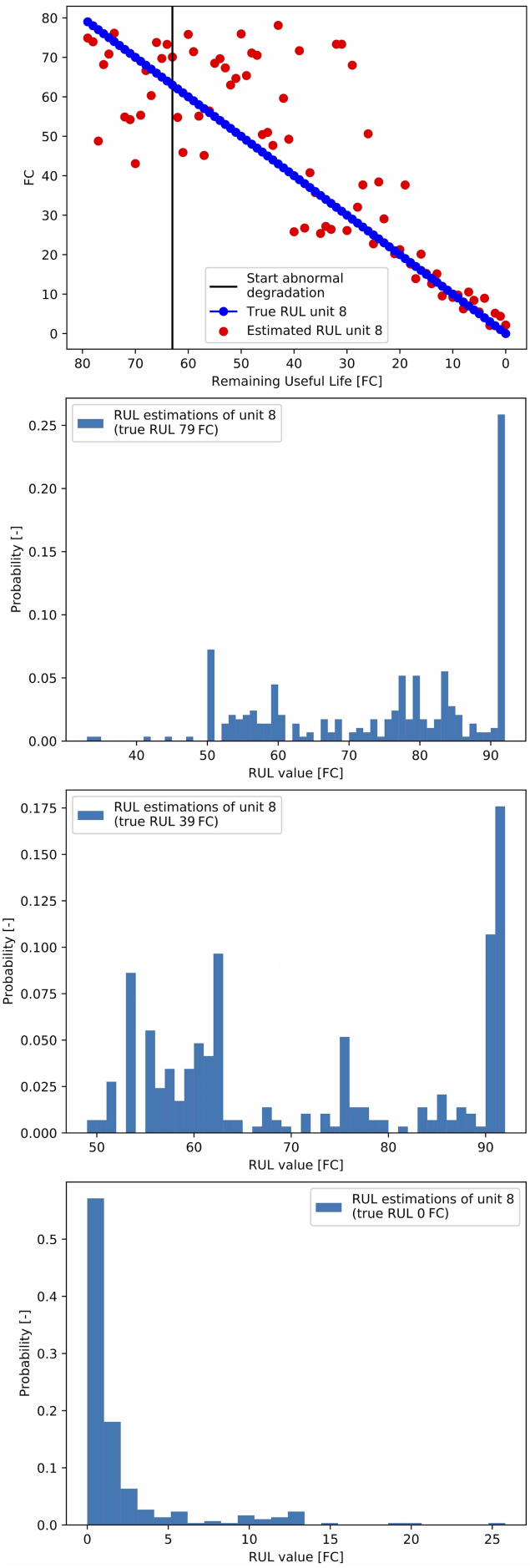


Figure 9. RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 8, DS01.

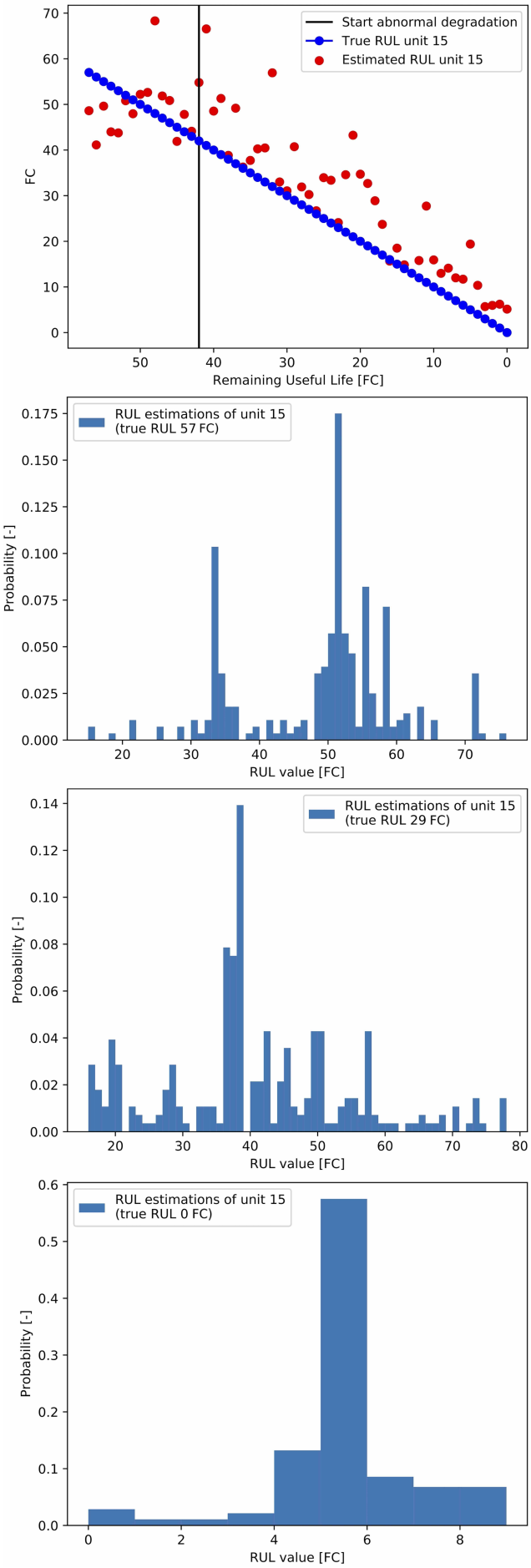
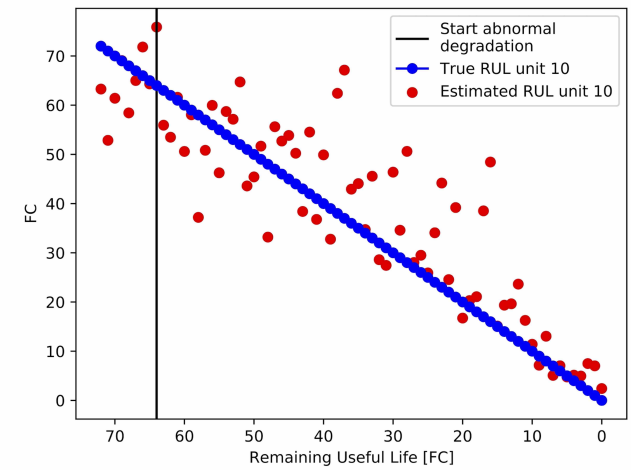
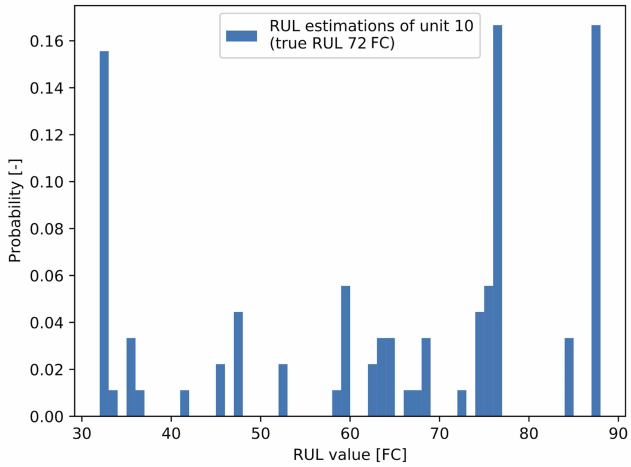
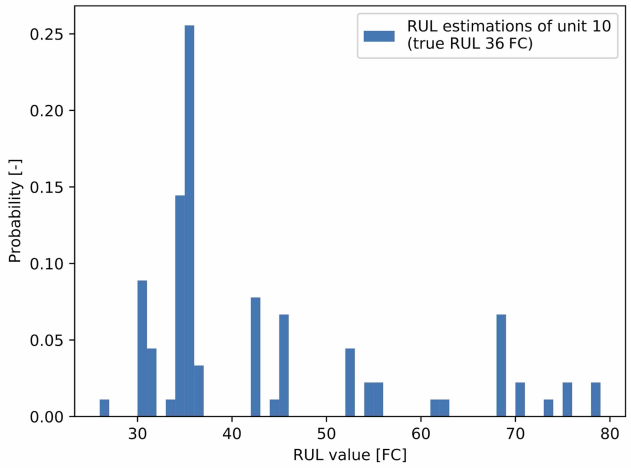


Figure 10. RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 15, DS03.







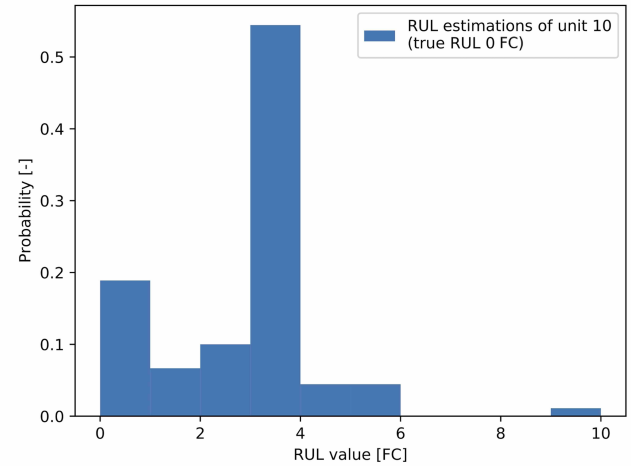
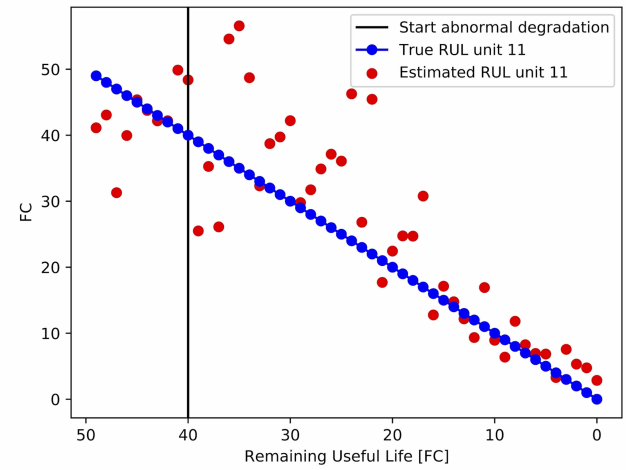
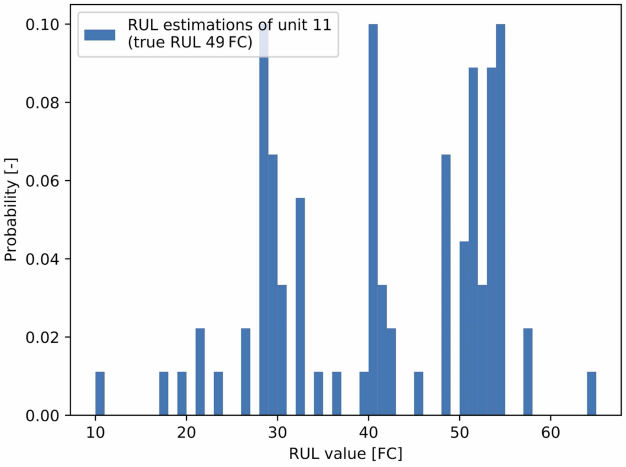
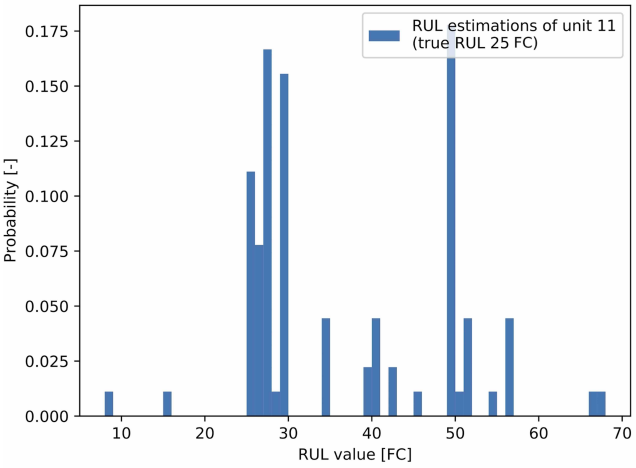


Figure 11. RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 10, DS01.







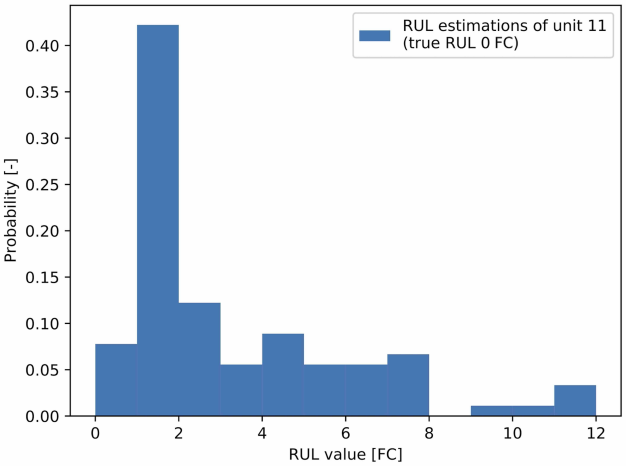


Figure 12. RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 11, DS03.

Table 18
RESULTS OF RUL POINT ESTIMATIONS (FOR EACH FC STARTING FROM 10TH FC) OF TEST UNITS FOR FLIGHT CLASS RFR MODELS 1, 2 AND 3 IN DS01, DS02 AND DS03.

Data subset	FC average / LOWESS	Flight class RFR model	Test unit number	RMSE [FC]	MAE [FC]	Min. Abs. Error [FC]	Max. Abs. Error [FC]
DS01	FC average	1	7	10.1090	8.4136	0.1500	28.5100
		1	9	10.9003	9.0736	0.0700	32.7900
		2	8	14.2437	9.9091	0.0123	42.3261
		3	10	10.4777	7.7062	0.0667	32.4444
DS01	LOWESSS	1	7	21.2952	17.3712	0.4545	55.1882
		1	9	24.4653	20.5472	0.6818	46.3455
		2	8	27.9983	25.2246	2.2667	42.0167
		3	10	22.2084	18.5335	1.2226	54.9677
DS02	FC average	3	11	7.4630	5.5295	0.0427	18.8648
DS02	LOWESS	3	11	10.9586	8.9191	0.0633	24.3300
DS03	FC average	1	12	13.5063	10.3939	0.0406	34.8312
		1	14	10.6102	7.6266	0.0125	11.3813
		2	15	9.5315	7.2841	0.0004	25.5427
		3	10	13.0331	10.4942	0.2556	31.0556
		3	11	11.2808	8.6007	0.1889	23.4444
		3	13	12.5605	9.5952	0.4333	36.1333
DS03	LOWESSS	1	12	18.0241	11.9780	0.0750	46.0875
		1	14	17.0300	12.2588	0.2875	36.7250
		2	15	16.2781	13.7128	1.0667	37.9556
		3	10	14.0664	11.2966	0.5316	43.5569
		3	11	15.4393	12.5263	0.2108	33.0392
		3	13	14.5313	11.5381	0.4438	42.2990

Moreover, another interesting result is obtained when considering the FC average entries in Table 18. It can be seen that test unit 11 of DS02 has the lowest overall errors in terms of RMSE and MAE. This is interesting given the fact that this test unit (as well as the test units in DS03) is affected by three different simultaneous failure modes, whilst the test units in DS01 are only affected by a single failure mode. Having a unit being affected by a larger number of different simultaneous failure modes is expected to have a decreased RUL point estimation accuracy, as this yields a more complex deterioration pattern. Nevertheless, test unit 11 of DS02 has been trained on more training units (three units) in comparison to the test units in DS01. As a matter of fact, test unit 10 of DS01 is trained on only two training units, whereas units 7 and 9 have been trained together on solely a single training unit. Moreover, also unit 8 of DS01 has been trained on solely a single training unit. Another explanation for unit 11 of DS02 having these lowest RMSE and MAE scores is that this test unit happened to adequately match the degradation pattern of its training units. Furthermore, Table 18 shows that also the errors (in terms of RMSE, MAE, Minimum Absolute Error and Maximum Absolute Error) for test units in DS03 are approximately equal to or sometimes even smaller than the errors obtained for test units in DS01. Moreover, the overall lowest Minimum and Maximum Absolute Errors are obtained for test units 15 and 14 in DS03, respectively. Additionally, besides test unit 15 of DS03 having the overall lowest Minimum Absolute Error, it also has the second lowest overall RMSE as well as MAE scores. The above are considered to be equally interesting results, given that the test units in DS03 are also affected by three simultaneous different failure modes (which is expected to decrease the estimation accuracy of the RUL point estimations). Again, these accurate results are believed to occur due to the overall higher number of available training units compared to the number of available training units in DS01. As a result, it verifies that the developed model is capable of correctly handling various types of datasets in terms of complexity as well as in terms of the total number of units they contain. Furthermore, it has been demonstrated that using the FC average method for data smoothing, generally results into the lowest errors in terms of RMSE, MAE, Minimum Absolute Error and Maximum Absolute Error.

5 Discussion

In reality, the implications of an airline being able to estimate a unit's (engine's) RUL with accompanying uncertainty are twofold. Firstly, it allows for optimized planning of the required maintenance personnel to replace the specific unit that is estimated to fail once it has been affected by a certain failure mode. Secondly, it allows for an optimized planning and usage of the available hangers as it will be known in advance how many units are estimated to fail. Hence, the total required time to replace the unit(s) can be estimated more accurately, compared to preventively replacing all of an aircraft's units after a predefined number of FC or flight hours.

Apart from the implications of this study, it has been found that developing three separate flight class RFR models (one per flight class) was necessary due to the fundamentally different operational flight conditions. In case a single model would have been developed to estimate the RUL of units from all three different flight classes, the accuracy would have been considerably lower (and corresponding uncertainty considerably higher) compared to the results obtained in section 4. This also illustrates the limitation of the proposed models in this study. That is, the proposed flight class RFR models are only suitable to handle units that exclusively operate in the flight class for which the models are developed. In case a unit would have been used for short as well as medium or long-haul flights, it would be difficult to produce accurate RUL estimations with low uncertainty.

Besides this, one of the most remarkable aspects that has been found when considering all steps of the applied methodology (as discussed in section 3) is concerned with the use of the K-fold CV method on time series data by other studies. As described, K-fold CV is highly unsuitable for time series data. That is, K-fold CV assumes data to be independent and identically distributed, whilst this assumption is clearly violated for this type of data. Time series data cannot be considered to be independent, as future observations (feature values) are dependent on past observations. In case one does use K-fold CV for time series data, it will almost certainly give 'biased' results in terms of RMSE scores on the test dataset and especially CV scores on the development dataset.

Finally, comparing the results obtained for the test unit of the validation data subset (DS02) to existing literature (that also used DS02), leads to the conclusion that the proposed developed model competes well with the proposed models by Biggio et al. [20]. As a matter of fact, out of the five RUL estimation models proposed by Biggio et al. [20], the proposed model of this study outperforms two of those models. However, given that the study of Biggio et al. [20] does not provide RUL point estimation graphs, only the RMSE scores can be compared. The proposed model of this study has a RMSE score of 7.4630 FC (which is obtained by performing a RUL estimation for every FC, starting from the 10th FC of test unit 11). This RMSE score is lower than the two highest RMSE scores of the five models proposed by Biggio et al. [20], which are 8.7000 FC and 7.7100 FC, respectively. Furthermore, the RMSE score of 7.4630 FC is only slightly higher (0.1530 FC) than the lowest RMSE score of the five models proposed by Biggio et al. [20], which is 7.3100 FC. However, it is unknown how the RMSE scores of the study by Biggio et al. [20] have been calculated. More precisely, it is unknown how many training units were used, which test unit has been used to obtain these RMSE scores as well as how many RUL point estimations were used to obtain the scores. Nevertheless, it has been demonstrated by comparing the RMSE scores that the proposed model of this study performs well compared to the proposed models of the study by Biggio et al. [20], especially because those models have a higher complexity as they are based on Gaussian Process models as well as Deep Neural Network models.

6 Conclusion

Whilst the majority of studies regarding Remaining Useful Life (RUL) estimation of aircraft units (engines) solely focused on the accuracy of so-called RUL point estimations, this study additionally focused on modelling the corresponding uncertainty of those RUL point estimations. The data-driven Random Forest Regression (RFR) algorithm has been applied in order to obtain the RUL point estimations as well as the probability distributions for a selection of those RUL point estimations. Before the application of RFR, the data has been processed using the flight cycle (FC) average smoothing method as well as the Locally Weighted Scatterplot Smoothing (LOWESS) method. Hereafter, for both individual smoothing methods, Recursive Feature Elimination with cross validation (RFECV) has been applied to determine the required features that yielded the most accurate RUL point estimations. The above mentioned steps (data processing, the use of RFECV and the use of RFR) have been performed per flight class due to the substantially different operational flight conditions. In other words, a model has been developed per flight class. Therefore, the models are referred to as flight class RFR models. As a consequence, it can be concluded that the main limitation of this study is the fact that the flight class RFR models are only capable of accurately estimating the RUL of units that exclusively operate in the flight class for which the models are developed.

Besides this conclusion, it has been found and can be concluded for all flight class RFR models that using the FC average smoothing method for data smoothing, yielded considerably more accurate RUL point estimations for all test units (compared to using LOWESS as data smoothing method). The main reason for this substantial difference is caused by the amount of available data for LOWESS to provide accurately smoothed data. During training of the flight class RFR models, data corresponding to the entire lifetimes of training units has been available for smoothing with LOWESS, whilst this was obviously not available for test units. In practice, only past data will be available for test units instead of the data corresponding to their entire lifetimes. Hence, this lack of available data for smoothing with LOWESS resulted into different smoothed data compared to the smoothed training data.

This different smoothed data eventually caused largely different RUL point estimations in comparison to the RUL point estimations obtained with the FC average data smoothing method. Furthermore, a conclusion can be obtained regarding the uncertainty of the RUL point estimations. It has been found that the probability of accurately estimating a single RUL point estimation is generally relatively low, as most probability distributions show rather large tails (hence the uncertainty is relatively high). However, the true power of RFR is found in combining all estimations into a single average estimation, as this compensates for single trees under or overestimating the true RUL value. Moreover, it has been found that the RUL point estimations of the majority of the test units, especially from which the data has been smoothed with the FC average smoothing method, converged towards the true RUL over time as the units approached their end of life (hence the accuracy improved). As a matter of fact, not only the RUL estimation accuracy improved over time, the corresponding uncertainty decreased as well. This latter part is concluded due to the fact that the width of the tails of the majority of the probability distributions decreased over time.

Furthermore, it has been found that despite data subset DS03 is considerably more complex as its units were affected by three different simultaneous failure modes, the results (in terms of errors) of the test units were approximately equal to or even lower compared to the results obtained for test units in data subset DS01 (which is substantially less complex as those units were affected by only a single failure mode). This is explained by the fact that DS03 contained more training units than DS01, hence the flight class RFR models were able to learn enhanced degradation patterns and eventually obtain these approximately equal or even lower errors compared to DS01. Therefore, it can be concluded that increasing the number of training units even more, is expected to improve the accuracy of the RUL estimations even further. Finally when considering data subset DS02 (which corresponding test unit has also been affected by three different simultaneous failure modes, but has been trained on more training units than the test units in DS01), it has been shown that the proposed model even outperformed two out of the five models from existing literature. As a result, it can be concluded that the proposed model effectively competes with the currently proposed models in the literature.

7 Recommendations

In order to further investigate the dataset with its subsets, as well as to further improve Remaining Useful Life (RUL) estimation accuracy and decrease the corresponding uncertainty, it is recommended to investigate the following aspects in future work. First of all, it would be interesting to explore the possibility of using a more advanced algorithm for RUL estimations, as it might be able to further improve the RUL estimations and decrease the accompanying uncertainty. Therefore, it is recommended to investigate the use of an advanced algorithm such as a neural network for the RUL estimations and their corresponding uncertainty. Furthermore, it is recommended to explore the use of data subset DS08 whenever it becomes fully available (all five parts within DS08), as this data subset contains the most training units. Consequently, it is expected to improve the accuracy of the RUL point estimations as it has been found that an increased number of available training units seems to yield more accurate RUL point estimations. Another recommendation regarding the number of available training units is concerned with the development of methods which are capable of generating accurate RUL estimations, despite having a very limited number of training units available. This is useful as in general, a limited number of training units is expected to be available for RUL estimation in practice.

Finally, this study focused on estimating the RUL of a unit as a whole (without considering which specific component would fail). Therefore, it is recommended to conduct research into the development of a method which is also capable of identifying which specific component within the unit will fail. This is considered to be valuable information since knowing in advance which specific component will fail, allows one to decrease the number of spare components in stock. That is, the specific component could then be ordered only when required, thereby further reducing the necessary storage space, hence further reducing costs.

References

[1] International Air Transport Association, "Airline maintenance cost executive commentary," 1 2021, Accessed on: 2022-04-7. [Online]. Available: https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2019-mctg-report_public.pdf

[2] S. de Bree, "Big data in aviation - reduce costs thru predictive maintenance," 7 2019, Accessed on: 2022-04-7. [Online]. Available: https://www.exsyn.com/blog/big-data-in-aviation-predictive-maintenance

[3] N. Hölzel and V. Gollnick, "Cost-benefit analysis of prognostics and condition-based maintenance concepts for commercial aircraft considering prognostic errors," in *Annual Conference of the PHM Society*, vol. 7, no. 1, 10 2015. [Online]. Available: https://doi.org/10.36001/phmconf.2015.v7i1.2716

[4] P. Baraldi, F. Mangili, and E. Zio, "Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data," *Reliability Engineering System Safety*, vol. 112, pp. 94–108, 2013. [Online]. Available: https://doi.org/10.1016/j.ress.2012.12.004

[5] M. Mitici and I. de Pater, "Online model-based remaining-useful-life prognostics for aircraft cooling units using time-warping degradation clustering," *Aerospace*, vol. 8, no. 6, 2021. [Online]. Available: https://www.doi.org/10.3390/aerospace8060168

[6] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018. [Online]. Available: https://doi.org/10.1016/j.ymssp.2017.11.016

[7] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1803–1836, 2011. [Online]. Available: https://doi.org/10.1016/j.ymssp.2010.11.018

[8] D. An, N. H. Kim, and J.-H. Choi, "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliability Engineering System Safety*, vol. 133, pp. 223–236, 2015. [Online]. Available: https://doi.org/10.1016/j.ress.2014.09.014

[9] F. Zhao, Z. Tian, and Y. Zeng, "Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 146–159, 2013. [Online]. Available: http://doi.org/10.1109/TR.2013.2241216

[10] D. Liu, J. Zhou, D. Pan, Y. Peng, and X. Peng, "Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning," *Measurement*, vol. 63, pp. 143–151, 2015. [Online]. Available: https://doi.org/10.1016/j.measurement.2014.11.031

[11] Y. Qian, R. Yan, and H. Shijie, "Bearing degradation evaluation using recurrence quantification analysis and kalman filter," *Instrumentation and Measurement, IEEE Transactions*, vol. 63, pp. 2599–2610, 11 2014. [Online]. Available: http://doi.org/10.1109/TIM.2014.2313034

[12] J. Lin and Z. Lin, "Research on remaining useful life prediction for aircraft engine with a fault point," *Journal of Physics: Conference Series*, vol. 1549, no. 5, p. 052065, jun 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1549/5/052065

[13] Z. Huang, Z. Xu, X. Ke, W. Wang, and Y. Sun, "Remaining useful life prediction for an adaptive skew-wiener process model," *Mechanical Systems and Signal Processing*, vol. 87, pp. 294–306, 2017. [Online]. Available: https://doi.org/10.1016/j.ymssp.2016.10.027

[14] H. Liao, W. Zhao, and H. Guo, "Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model," in *RAMS '06. Annual Reliability and Maintainability Symposium, 2006.*, 2006, pp. 127–132. [Online]. Available: http://doi.org/10.1109/RAMS.2006.1677362

[15] V. T. Tran, H. Thom Pham, B.-S. Yang, and T. Tien Nguyen, "Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine," *Mechanical Systems and Signal Processing*, vol. 32, pp. 320–330, 2012, uncertainties in Structural Dynamics. [Online]. Available: https://doi.org/10.1016/j.ymssp.2012.02.015

[16] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 International Conference on Prognostics and Health Management*, 2008, pp. 1–6. [Online]. Available: http://doi.org/10.1109/PHM.2008.4711422

[17] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, 2016, pp. 135–140. [Online]. Available: http://doi.org/10.1109/AUS.2016.7748035

[18] X. Chen, G. Jin, S. Qiu, M. Lu, and D. Yu, "Direct remaining useful life estimation based on random forest regression," in *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai)*, 2020, pp. 1–7. [Online]. Available: http://doi.org/10.1109/PHM-Shanghai49105.2020.9281004

[19] M. Tanwar and N. Raghavan, "Lubricating oil remaining useful life prediction using multi-output gaussian process regression," *IEEE Access*, vol. 8, pp. 128 897–128 907, 2020. [Online]. Available: http://doi.org/10.1109/ACCESS.2020.3008328

[20] L. Biggio, A. Wieland, M. A. Chao, I. Kastanis, and O. Fink, "Uncertainty-aware prognosis via deep gaussian process," *IEEE Access*, vol. 9, pp. 123 517–123 527, 2021. [Online]. Available: http://doi.org/10.1109/ACCESS.2021.3110049

[21] F. Tamssaouet, N. Khanh, K. Medjaher, and M. Orchard, "Combination of long short-term memory and particle filtering for future uncertainty characterization in failure prognostic," 01 2021, pp. 275–281. [Online]. Available: http://doi.org/10.3850/978-981-18-2016-8_152-cd

[22] H. Richter and P. R. Slowinski, "The data sharing economy: On the emergence of new intermediaries," *IIC - International Review of Intellectual Property and Competition Law*, vol. 50, no. 1, pp. 4–29, Jan 2019. [Online]. Available: https://doi.org/10.1007/s40319-018-00777-7

[23] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," NASA Ames Prognostics Data Repository, 2008. Accessed on: 2022-01-24. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

[24] M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics," *Data*, vol. 6, no. 1, 2021. [Online]. Available: https://doi.org/10.3390/data6010005

[25] M. Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions," NASA Ames Prognostics Data Repository, 2021. Accessed on: 2022-01-24. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

[26] C. Matthews, "12 - mechanical seals – improving design reliability," in *Case Studies in Engineering Design*, C. Matthews, Ed. London: Butterworth-Heinemann, 1998, pp. 113–126. [Online]. Available: https://doi.org/10.1016/B978-034069135-9/50015-1

[27] M. Rajabinezhad, A. Bahrami, M. Mousavinia, S. J. Seyedi, and P. Taheri, "Corrosion-fatigue failure of gas-turbine blades in an oil and gas production plant," *Materials*, vol. 13, no. 4, 2020. [Online]. Available: https://doi.org/10.3390/ma13040900

[28] D. K. Frederick, J. A. Decastro, and J. S. Litt, "User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS)," Tech. Rep., NASA: Washington, DC, USA, 2007.

[29] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Procedia Computer Science*, vol. 161, pp. 466–474, 2019, the Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia. [Online]. Available: https://doi.org/10.1016/j.procs.2019.11.146

[30] J. Chen, D. Zhou, C. Lyu, and C. Lu, "A novel health indicator for pemfc state of health estimation and remaining useful life prediction," *International Journal of Hydrogen Energy*, vol. 42, no. 31, pp. 20 230–20 238, 2017. [Online]. Available: https://doi.org/10.1016/j.ijhydene.2017.05.241

[31] J. M. Bai, G. S. Zhao, and H. J. Rong, "Novel direct remaining useful life estimation of aero-engines with randomly assigned hidden nodes," *Neural Computing and Applications*, vol. 32, no. 18, pp. 14 347–14 358, Sep 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04478-1

[32] K. Chen, S. Laghrouche, and A. Djerdir, "Health state prognostic of fuel cell based on wavelet neural network and cuckoo search algorithm," *ISA Transactions*, vol. 113, pp. 175–184, 2021. [Online]. Available: https://doi.org/10.1016/j.isatra.2020.03.012

[33] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979. [Online]. Available: https://doi.org/10.1080/01621459.1979.10481038

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed on: 2022-03-4. [Online]. Available: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[35] S. Wang and S. Chen, "Insights to fracture stimulation design in unconventional reservoirs based on machine learning modeling," *Journal of Petroleum Science and Engineering*, vol. 174, pp. 682–695, 2019. [Online]. Available: https://doi.org/10.1016/j.petrol.2018.11.076

[36] P. Lu, Z. Zhuo, W. Zhang, J. Tang, Y. Wang, H. Zhou, X. Huang, T. Sun, and J. Lu, "A hybrid feature selection combining wavelet transform for quantitative analysis of heat value of coal using laser-induced breakdown spectroscopy," *Applied Physics B*, vol. 127, no. 2, p. 19, Jan 2021. [Online]. Available: https://doi.org/10.1007/s00340-020-07556-8

[37] M. Lambelho, M. Mitici, S. Pickup, and A. Marsden, "Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions," *Journal of Air Transport Management*, vol. 82, p. 101737, 2020. [Online]. Available: https://doi.org/10.1016/j.jairtraman.2019.101737

[38] Y. Fan, S. Nowaczyk, and T. Rögnvaldsson, "Transfer learning for remaining useful life prediction based on consensus self-organizing models," *Reliability Engineering System Safety*, vol. 203, p. 107098, 2020. [Online]. Available: https://doi.org/10.1016/j.ress.2020.107098

[39] Z. Kang, C. Catal, and B. Tekinerdogan, "Remaining useful life (rul) prediction of equipment in production lines using artificial neural networks," *Sensors*, vol. 21, no. 3, 2021. [Online]. Available: https://doi.org/10.3390/s21030932

[40] K. T. Chui, B. B. Gupta, and P. Vasant, "A genetic algorithm optimized rnn-lstm model for remaining useful life prediction of turbofan engine," *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: https://www.doi.org/10.3390/electronics10030285

[41] M. Schnaubelt, "A comparison of machine learning model validation schemes for non-stationary time series data," Nürnberg, FAU Discussion Papers in Economics 11/2019, 2019, Accessed on 2022-03-21. [Online]. Available: http://hdl.handle.net/10419/209136

[42] A. Razak, *Industrial gas turbines: performance and operability*. Elsevier, 2007.

[43] M. Zoutendijk and M. Mitici, "Probabilistic flight delay predictions using machine learning and applications to the flight-to-gate assignment problem," *Aerospace*, vol. 8, no. 6, 2021. [Online]. Available: https://doi.org/10.3390/aerospace8060152

[44] S. Nembrini, I. R. König, and M. N. Wright, "The revival of the Gini importance?" *Bioinformatics*, vol. 34, no. 21, pp. 3711–3718, 05 2018. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty373

[45] P. Zhou, Z. Li, S. Snowling, B. W. Baetz, D. Na, and G. Boyd, "A random forest model for inflow prediction at wastewater treatment plants," *Stochastic Environmental Research and Risk Assessment*, vol. 33, no. 10, pp. 1781–1792, Oct 2019. [Online]. Available: https://doi.org/10.1007/s00477-019-01732-9

[46] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[47] Z. Wang, C. Lai, X. Chen, B. Yang, S. Zhao, and X. Bai, "Flood hazard risk assessment model based on random forest," *Journal of Hydrology*, vol. 527, pp. 1130–1141, 2015. [Online]. Available: https://doi.org/10.1016/j.jhydrol.2015.06.008

[48] X. Li, Y. Wang, S. Basu, K. Kumbier, and B. Yu, "A debiased mdi feature importance measure for random forests," *Advances in neural information processing systems*, 06 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1906.10845

[49] I. D. Raji, H. Bello-Salau, I. J. Umoh, A. J. Onumanyi, M. A. Adegboye, and A. T. Salawudeen, "Simple deterministic selection-based genetic algorithm for hyperparameter tuning of machine learning models," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://doi.org/10.3390/app12031186

[50] D. Sun, J. Xu, H. Wen, and D. Wang, "Assessment of landslide susceptibility mapping based on bayesian hyperparameter optimization: A comparison between logistic regression and random forest," *Engineering Geology*, vol. 281, p. 105972, 2021. [Online]. Available: https://doi.org/10.1016/j.enggeo.2020.105972

[51] A. Thakur, *Approaching (Almost) Any Machine Learning Problem*. Abhishek Thakur, 2020.

Part II
Literature Study

Graded previously under course AE4020

1 Introduction

Throughout the past decades, the aviation industry has been growing rapidly. In 2019, the International Civil Aviation Organisation (ICAO) announced that a total of 4.5 billion passengers have been transported by scheduled services, which is a 3.6% increase compared to 2018 [1]. In that same year, the increase of transported passengers is accompanied by a 1.7% increase of departed aircraft to 38.3 million, compared to 2018 [1]. Despite the fact that passenger demand currently experiences an extraordinary decrease due to the COVID-19 pandemic [2], it is estimated that global passenger demand will recover within 2.4 years to numbers equal to the demand in 2019 [3], and will continue to grow afterwards to approximately 5.6 billion passengers by 2030 according to the International Air Transport Association (IATA) [4]. In order to transport these passengers whilst ensuring safe operation of the aircraft, aircraft maintenance is essential. However, Maintenance, Repair and Overhaul (MRO) costs correspond to 10.3% of the airline's operational costs on average, according to IATA [5]. Within these MRO costs, aircraft engines account for the largest maintenance cost being equal to 43% of the total maintenance costs [5]. Therefore, airlines try to decrease the MRO costs as much as possible, especially the maintenance costs associated with aircraft engines.

In general, maintenance can be divided into reactive and proactive maintenance (see Figure 1.1), in which the latter can be further divided into so-called preventive maintenance and predictive maintenance [6]. Reactive maintenance is maintenance only carried out after a component has failed. A preventive maintenance strategy (currently widely applied in the aircraft maintenance sector [7]) focuses on scheduled replacement of specific components after a predetermined number of flight hours or flight cycles [6], thereby 'wasting' some of the component's Remaining Useful Life (RUL) since the component will be replaced before failure occurs. Predictive maintenance is a prognostic strategy concerned with the prediction of component failure (taking into account it's health condition) and therefore allows one to predict when maintenance should be performed and thus estimates the component's RUL [6]. Note, the terms predictive maintenance and condition-based maintenance are regularly used as synonyms in literature [7],[8] and [9]. Furthermore, predictive maintenance is considered as being part of the so-called Prognostics and Health Management (PHM) process [10].

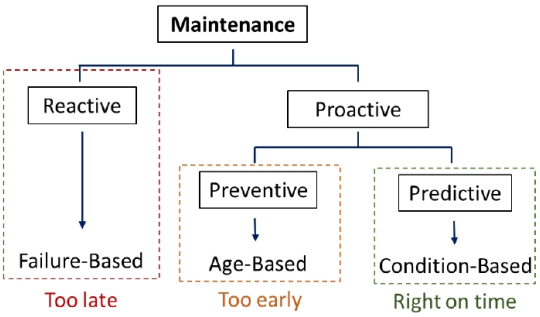


Figure 1.1: Different types of maintenance [6].

The use of predictive maintenance allows for component removal at 'the right time' thus minimizing the waste of the component's RUL and thereby reducing costs. As a result, one of the possibilities to decrease the maintenance costs associated with aircraft engines whilst preserving flight safety, as components are replaced before the occurrence of actual failure [11], is to implement predictive maintenance [7]. In fact, using predictive maintenance does not only allow one to estimate a component's RUL, it also allows for the identification of a component's current health state as well as to predict its future health state [12]. This is achieved by firstly examining component deterioration data in order to estimate its current health state and reveal deterioration trends [13]. Thereafter, these deterioration trends will be provided as input for algorithms suitable for the prediction of the component's future health state and estimate its corresponding RUL [13].

1.1 Motivation

Since aircraft engines consist of multiple components which could fail separately and thus require maintenance, it is useful for an airline to know in advance which engine components are going to fail. To be more specific, it is useful for an airline to know which engine components are going to fail together with the engine's associated RUL as well as the corresponding uncertainty of this RUL estimation. Having a model capable of achieving these requirements is believed to be a solution to seriously decrease aircraft maintenance costs. However, such a complete model does not exist yet and thus reveals the need for the development of such a model. Therefore, this research focuses on the development a model being able to analyse deterioration data of multiple aircraft engines, identify individual engine components that will fail (so-called faulty components), determine degradation trends in this deterioration data and produce aircraft engine RUL estimations with the probability distributions of these estimations in order to model the uncertainty.

1.2 Report outline

The remainder of this literature study report starts with an extensive literature review in Chapter 2. This chapter thoroughly describes the general phases of a PHM process, along with the most commonly used methods and techniques as described in the literature in each phase. Furthermore, proposed techniques to identify faulty components have been addressed in this chapter as well as the literature gap that has been identified. Finally, Chapter 3 describes the research proposal which follows from the previously described problem description and the identified literature gap. This chapter contains the research's objective, main research questions, sub-questions as well as the research scope.

2 State-of-the-art literature review

This chapter thoroughly describes the extensive literature review which has been conducted regarding RUL estimation as well as faulty component identification. It describes the most widely used methods and techniques proposed in previous research studies along with the cutting edge techniques i.e. the state-of-the-art. The chapter starts with the division of the four main phases within a general health prognostic process for machinery components as described in Paragraph 2.1. From this division, the various techniques and methods as described in literature will be discussed per process phase (Paragraph 2.2, Paragraph 2.3, Paragraph 2.4 and Paragraph 2.5). Thereafter, Paragraph 2.6 addresses techniques described in literature to identify faulty components within aircraft engines. Finally, Paragraph 2.7 contains a discussion regarding the extensively described literature review and describes the identified literature gap.

2.1 General phases of a PHM process

According to the literature, a PHM process can generally be divided into four main phases (assuming one has access to sensor data from the system or component of interest): 1) Data cleaning phase, 2) Health indicator construction phase, 3) Health stage division phase and 4) RUL estimation phase [14]. An elaborate explanation on the purpose of each individual phase together with their corresponding most widely used techniques and methods as described in the literature, is provided in Paragraph 2.2, Paragraph 2.3, Paragraph 2.4 and Paragraph 2.5, respectively.

2.2 Data cleaning phase

Assuming one has access to deterioration data from sensors which are installed on the system or component of interest, the first step in the PHM process is to clean this data as sensor data is usually contaminated with noise [15]. Noisy data can have a severely negative influence on prediction results [16]. In order to reduce this noise as much as possible, several techniques have been proposed throughout the years. In the literature, one of the most commonly used techniques for noise reduction is the moving average filter algorithm [17], [18], [19] and [20], because of its easy implementation and satisfying results. The moving average filter algorithm is an optimal algorithm for the reduction of random noise in data whilst maintaining the original data characteristics, i.e. data deterioration trends [21]. The mathematical form of the moving average filter algorithm as described by Smith [21] can be seen in Equation 1, where $x[\,]$ contains the input sensor data, M denotes the window size (number of points considered when calculating the average) and $y[\,]$ contains the average value of the data points considered in the window size. In principle, the algorithm calculates the average of a certain number of data points (specified by the window size M) from the sensor data and stores this average value as one output data point for the points included in the window size [21].

$$y[a] = \frac{1}{M} \sum_{b=0}^{M-1} x[a+b] \tag{1}$$

As an example, suppose one wants to use the moving average filter starting from sensor data entry 25, with a window size of 4 . Then, the output of the algorithm for data entry 25 is given by Equation 2.

$$y[25] = \frac{x[25] + x[26] + x[27] + x[28]}{4} \tag{2}$$

Besides using algorithms to reduce noise in data, sometimes a very simple, yet effective method which does not require the use of any specific noise reduction algorithm is applied. For example, a study conducted by Ghorbani and Salahshoor states that they simply cleaned the dataset (consisting of deterioration data from multiple sensors) from noise by discarding sensor data which showed a constant value over time [22]. However, it can be argued that this method of 'noise removal' actually should be described as a method belonging to sensor selection.

2.3 Health indicator construction phase

After the deterioration data has been cleaned sufficiently, Health Indicators (HIs) must be constructed to be able to identify the degradation trend of the component of interest [14] as well as to obtain its current health status [23]. According to Lei et al., the construction of suitable HIs is important as it is expected to improve the prediction accuracy of the RUL estimations [14].

Before one can actually construct HIs, it must be determined which sensors will be used in case the dataset contains data from multiple sensors. In the literature, a commonly used method for relevant sensor selection is to include only sensors which data shows non-constant values over time, as applied by Zhang et al. [24], Yang et al. [25] and Coble and Hines [26] (which also considered the correlation between sensors as a measure of a sensor's suitability). Another widely used technique in literature is Principal Component Analysis (PCA). PCA is a dimension reduction technique used for data which has multiple dimensions (e.g. multiple different sensor measurements per time step) and is also suitable to handle white noise [27]. As described by Lasheras et al. [28], PCA transforms the original variables in a dataset (the sensors) $x_1, x_2, ..., x_n$ into a different dataset $q_1, q_2, ..., q_r$, by linear mapping, in which $r \leq n$. By applying PCA, one is able to reduce the number of sensors in the dataset whilst preserving the data characteristics [14]. According to Ghorbani and Salahshoor [22], PCA is only considered to be an appropriate method in case the data is linearly separable. In case of nonlinear data, Kernel PCA (KPCA) might be used to overcome the limitations of PCA [22]. KPCA uses nonlinear mapping for converting the high dimension input space to a low dimension output space [22].

After selecting the relevant sensor data, HIs can be constructed. Hu et al. found that two types of HIs can be distinguished: the physical HIs (PHIs) and the virtual HIs (VHIs) [23]. The PHIs are HIs which are constructed directly from sensor data which is related to the physics of specific failures [23]. On the other hand, VHIs are created by combining several PHIs or the use of multi-sensor data [14]. A variety of PHIs as well as VHIs were constructed and described in literature, from which the most important ones are described in Subparagraph 2.3.1 and Subparagraph 2.3.2, respectively.

2.3.1 Physical health indicators

Root Mean Square PHIs
Throughout the years, one of the most commonly used PHIs is the Root Mean Square (RMS) value of a signal [14]. It appears to be especially applied for the analysis of bearings. In 2011, Malhi et al. calculated the RMS value from vibration signals of bearings and used it as a PHI for the prediction of bearing defect progression [29]. A study conducted by Huang et al. in 2017 used the RMS value of vibration signals of ball bearings as a PHI to reveal the bearing's degradation trajectory [30]. Liao and Tian decided to take the logarithm of the RMS value of the vibration signals of bearings as a means to measure the degradation behaviour [31]. A slightly different version of RMS has been proposed by Meng et al., which is the RMS with cumulative sum (RMS-CUMSUM) [32]. It has been found that it is able to show the tendency of the bearing's degradation process [32]. Besides these bearing application, the RMS value is also used as a PHI for health state monitoring of turbine cutting tools (which are used to manufacture turbine blades) in a study conducted by Yingchao et al. [33]. In 2021, Mitici and de Pater employed the moving average of the maximum RMS as a PHI for the health status monitoring of aircraft cooling units [34]. Večeř et al. used the RMS value to observe and track the health status of gearboxes [35].

Kurtosis PHIs
According to the literature, another popular conventional PHI is the kurtosis (KUR) value [36]. Kurtosis can be best explained as a means to describe a distribution's 'tailedness' (e.g. it's peakness or flatness) [37]. In the literature, a KUR value higher than three indicates a sharp peak in the signal [38]. Imagine one is monitoring vibrations of gearboxes. At some point in time, a fault may arise in a specific component which in turn could cause a peak in the amplitude distribution of the gearbox's vibration signal. As a result, this would cause an increasing KUR value which can be used to detect this fault at an early stage [39]. Elasha et al. employed the KUR value as a PHI to monitor the health status of wind turbine gearboxes [38]. However, it has been found that KUR is only able to identify an initial fault at an early stage in time [39]. Barszcz and Randall used Spectral Kurtosis as a PHI for the identification of strong peaks in the frequency signal caused by planetary gear moving over cracked tooth in a wind turbine [40]. Unlike the KUR value, Spectral Kurtosis is able to identify non-Gaussian components (e.g., a different frequency in the gearbox due to suddenly cracked gear teeth) in a vibration signal [40].

Additional and new PHIs
Other commonly used conventional PHIs as described in the literature to monitor the health status of machines are the skewness, peak-to-peak value and Crest Factor [41]. Apart from these conventional PHIs, some researchers constructed new PHIs. For example, Hananchi et al. [42] constructed two new PHIs in order to monitor the performance deterioration of gas turbine engines. The first PHI is the heat loss index which is a measure for the loss of thermal power in comparison to the engine's healthy condition [42]. Their second PHI is the power deficit index which has been defined by Hananchi et al. as: *"the deficiency ratio of the gas turbine engine output power due to the performance deterioration"* [42].

2.3.2 Virtual health indicators

PCA and SOM-based VHIs
As explained previously, VHIs can be constructed using multi-sensor data or by combining PHIs. When examining the literature, sometimes PCA is used when construction VHIs [14]. For example, Widodo et al. [43] first employed PCA to reduce the dimensionality of the features in the dataset. Then, quantization errors (deviation between unknown health condition and healthy state condition) were computed and used as a VHI. Li et al. applied PCA to reduce the data dimensionality and constructed a VHI from multi-sensor data (four sensors) in a dataset from gas turbine engines [44]. Besides using PCA to construct VHIs, some researchers applied the Self Organising Map (SOM) for VHI construction. The SOM is a non-supervised neural network that organises the data into two-dimensional map units, thereby reducing the number of initial features [45]. Hong et al. proposed a VHI being the so-called Confidence Value, which is obtained from the SOM to represent the health state of a bearing [45].

Sensor-fused VHIs
However, not all studies employed PCA or SOM to construct VHIs. For example, Guo et al. proposed a recurrent neural network based VHI which fuses multiple features in order to monitor the health state of bearings [46]. Zhao et al. [47] merged 21 sensors from gas turbine engine data together into one single VHI using linear regression. The authors stated that using linear regression, the weight of each sensor (indicating if it contains relevant deterioration data) would be automatically determined [47]. Using a linear regression model to construct a VHI out of an gas turbine engine degradation dataset is also employed by Xinxin et al. [48]. However, the authors chose to only select sensors showing monotonously degradation behaviour, which lead to the selection of five sensors as input for the linear regression model to construct the VHI [48]. Bechhoefer created a VHI by merging six different features of gears which are based on heavily tailed Probability Density Functions (PDFs) or so-called Rayleigh PDFs [49].

Mahalanobis Distance VHIs
The Mahalanobis Distance (MD) is another frequently employed VHI. It is a distance measure which takes into account correlation between variables in data [50]. Therefore, it is able to identify abnormal data in a dataset [51]. Meng et al. used the so-called Growth Rate of Real-time Mahalanobis Distance with Cumulative Sum (GRRMD-CUMSUM) as a VHI to monitor the performance degradation of bearings [32]. Li et al. used a similar VHI based on the MD and CUMSUM for the estimation of several bearing performance degradation health stages [52]. In 2016, Wang et al. employed the MD to fuse 14 statistical features into one VHI representing the degradation process of bearings [53]. Using this VHI, the authors found the degradation process to be approximately linear, which in turn contributes to being able to more accurately determine the moment at which the bearing transfers from a healthy state to an unhealthy state.

Nevertheless, it should be noted that neither of the aforementioned studies selected sensors and/or HIs which identifies the failure of a specific component (a faulty component) in a system consisting of multiple components. The studies seemed to be only interested in selecting suitable sensors and HIs which accurately represent the degradation process of a system as a whole, without taking into account failure of specific components.

2.4 Health stage division phase

During a machine's degradation process, the constructed HIs might show changing degradation trends as a certain failure develops over time [14]. Therefore, Health Stage (HS) division (two-stage or multi-stage) is required as it divides the degradation process into HSs related to the obtained degradation trends from the HIs [14].

Two-stage HS division
A two-stage HS division distinguishes between a healthy state and an unhealthy state [54]. Such a HS division from a bearing degradation process using vibration data is depicted in Figure 2.1, where t_{FPT} denotes the First Prediction Time (FPT) and t_{EoL} the End of Life (EoL) time. Note, t_{FPT} starts as soon as the degradation process is considered to enter the unhealthy state.
Multiple studies used the two-stage HS division approach. Often in the literature, the FPT (the moment the component is considered to transition from a healthy to an unhealthy state) is determined by defining a predefined threshold at which the unhealthy stage starts. However, no consensus is reached yet on a general method to determine these thresholds. As a matter of fact, the determination of the thresholds is considered to be a huge challenge as these thresholds should not be set to soon (will raise false alarms) or too late (will result in not enough time to take precautionary actions [54]) [55]. Researchers proposed different approaches as a means to define thresholds in a two-stage HS division process. For example, Wang et al. adopted the two-stage HS division in a bearing degradation process and selected the 3σ interval of the MD as the threshold for the FPT [53]. In 2016, a study conducted by Jin et al. first used Box-Cox transformation to transform the HIs from bearings into Gaussian distributed data and

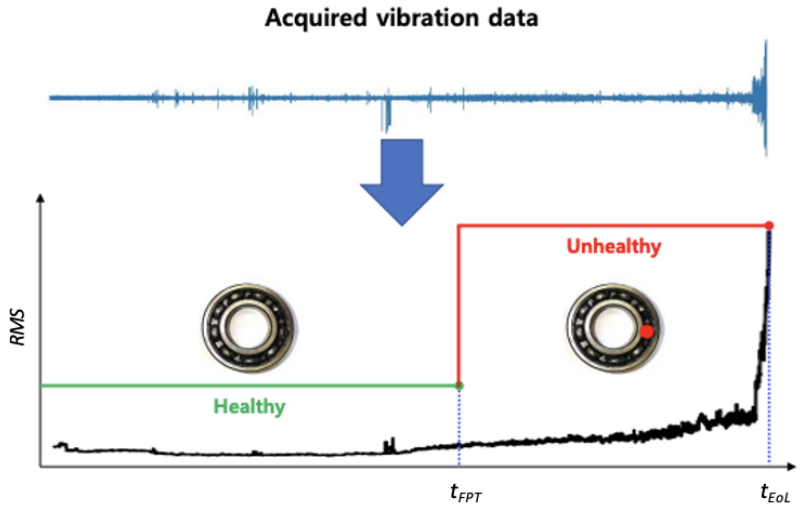


Figure 2.1: Example of two-stage Health Stage division of bearing degradation process [54].

then also employed 3σ interval as the threshold to distinguish between the healthy and unhealthy state [56]. Zhang et al. also focused on the degradation process of bearings in combination with a two-stage HS division and proposed the use of a predefined confidence threshold value of 90% as a means to determine that a bearing contains a certain fault (and is therefore considered to be in the unhealthy state), in order to decrease the number of false alarms [57]. Mitici and de Pater used the two-stage HS division as well and set a 2σ interval as an alarm threshold to define the start of the unhealthy state of aircraft cooling units [34].

Multi-stage HS division

Besides a two-stage HS division, there exists the so-called multi-stage HS division. A multi-stage HS division, as the name already implies, distinguishes between multiple HSs in case the degradation trend of the component of interest does not have a consistent unhealthy stage (e.g. the unhealthy state could be split into a degradation and a critical stage), and thereby cannot be expressed by a single degradation model [14]. An example of a degradation process with a non-consistent unhealthy stage is shown in Figure 2.2.

Multi-stage HS division through changing points

Some researchers chose to divide the degradation process into multiple substages using certain changes in HIs as a means to define a change from a HS to another [14]. For example, a study conducted by Kimotoho et al. proposed to utilize the peak amplitude shifts in frequency obtained from the power spectral density analysis of ball bearings to divide the degradation process into five different stages [58]. Another approach has been adopted by Hu et al. which used changing points of confidence levels as a means to split the degradation process of wind turbine generator bearings into four separate HSs [59].

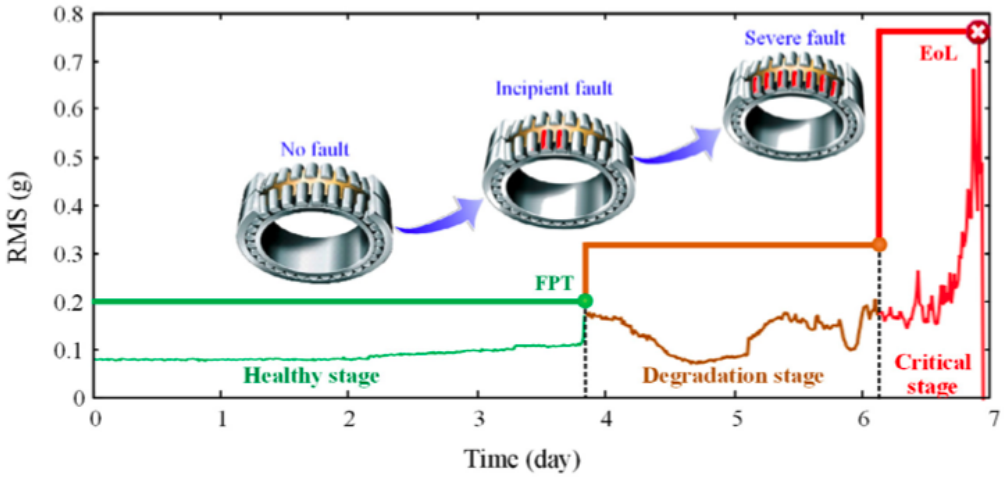


Figure 2.2: Example of multi-stage Health Stage division of bearing degradation process [14].

Multi-stage HS division through clustering

Apart from using changes in HIs to distinguish between multiple HSs, clustering has also been applied to categorize different HSs. In 2019 and more recently in 2021, Wang et al. applied the Fuzzy C-means clustering algorithm to obtain four different HSs for aircraft engines [20], [18]. The Fuzzy C-means algorithm is popular because it has the ability to assign data points to more than one cluster together with its corresponding probability score [60]. However, according to the literature, a drawback of the Fuzzy C-means algorithm is that its performance heavily depends on the initial selection of the cluster center at initialisation of the algorithm [61]. Ren et al. also used clustering for the division of HSs and applied the widely known k-means clustering algorithm which also resulted in the distinction of four different HSs of aircraft engines [62]. Liu et al. employed the k-means clustering algorithm to obtain three, four and even five different HSs in the degradation process of multiple aircraft engines [63]. A large difference with the Fuzzy C-means algorithm is that the k-means algorithm belongs to the so-called hard clustering group in which a data point will either be considered to fully belong to a certain cluster or not at all [60]. As a result, this algorithm will only assign data points to a cluster if it fully belongs to that cluster. Nevertheless, at initialisation, the k-means algorithm as well as the Fuzzy C-means algorithm both require a specific number of clusters that it should produce which is generally unknown [64], [60].

Multi-stage HS division through Autoencoders

It can be observed that for all the aforementioned techniques, the construction of HIs is required to be able to perform HS division. Therefore, a different approach for HS division is by means of an Autoencoder. An Autoencoder is an unsupervised artificial neural network which is able to learn and extract useful features from a high-dimensional dataset [65] and is also capable of transforming this into a low-dimensional representation of the original dataset [66]. Therefore, Autoencoders seem to be promising since

their main advantage is the fact that they are able to extract relevant features from high-dimensional data via an unsupervised manner, thereby circumventing the need for manual construction of suitable HIs [66], [54]. However, Autoencoders are still subjected to the previously described pre-specified threshold problem as they require a predefined threshold to function properly [54].

2.5 Remaining Useful Life estimation phase

The RUL estimation phase involves the actual models which are capable of performing an estimation of the RUL of a system or component based on the previously obtained degradation trends from the HIs before it reaches its 'failed phase'. In literature, it is generally assumed that a component degrades according to a stochastic process and it is considered to be failed once it exceeds a certain failure threshold [67]. For example, Mitici and de Pater [34] assumed that a component degrades over time following a stochastic process $\{X_t, t \geq 0\}$ and defined component failure as follows:

Definition: *A component following a stochastic degradation process* $\{X_t, t \geq 0\}$ *is considered to be failed if* $X_t \geq D, t > 0$ *(where* $D > 0$ *denotes the failure threshold)*

Furthermore, RUL is generally defined in literature as the remaining time before a machine's health state exceeds a certain failure threshold [34], [67], [14]. A mathematical definition is given by Mitici and de Pater and is expressed in Equation 3 as:

$$RUL = inf\{t : X_{f^{cur}+t} \geq D | X_0, ..., X_{f^{cur}}\} \tag{3}$$

In which f^{cur} denotes the current time, $inf()$ describes the limit inferior and $X_{f^{cur}+t}$ describes the component's health state at $f^{cur} + t$ [34]. A variety of methodologies have been proposed for RUL estimation over the years and most studies divide them into two main categories: model-based approaches and data-driven approaches [68]. However, a more thorough distinction has been made by Lei et al. who distinguished between four different categories of RUL estimation models: physical models (Subparagraph 2.5.1), statistical models (Subparagraph 2.5.2), artificial intelligence models (Subparagraph 2.5.3) and hybrid models (Subparagraph 2.5.4) [14].

2.5.1 Physical models

Physical models usually employ a mathematical model which represents the machine's physical degradation process [69]. The parameters of these models are generally obtained from laboratory experiments or sometimes derived from conducted historical measurements of the physical degradation process [70]. Physical models are able to create accurate RUL estimations if the parameters of the underlying mathematical models are well estimated [14]. According to the literature, the most applied physical model is the Paris-Erdogan model which is mostly used to estimate the RUL of systems experiencing crack growth [14]. This crack growth model has been proposed already in 1963 by Paris and Erdogan [71]. Thereafter, several studies adopted this Paris-Erdogan model for the

RUL estimation of systems subjected to crack growth. For example in 2012, Coppe et al. employed this Paris-Erdogan model for a system which has fatigue crack growth, estimated its RUL and showed it can even be used for complex geometries [72]. Besides Coppe et al., the Paris-Erdogan model is utilized for RUL estimation of systems which experience crack growth in several other studies [73], [74] , [75].

Besides the Paris-Erdogan model, another common physical model utilizes the Norton law as the mathematical underlying model. In 2013, Baraldi et al. used the Norton law model to simulate the creep evolution of turbine blades [68]. More recently in 2016, Hu et al. applied the Norton law model as well for the simulation of creep evolution of turbine blades [76]. Even though physical models are able to create accurate RUL estimations, their applications are limited since it is usually a challenge to accurately describe the physics of a degradation process of complex systems [14].

2.5.2 Statistical models

Statistical models or sometimes called empirical models use statistical methods which are in turn based on empirical data to estimate the RUL of systems or components [14]. As one can imagine, there exist numerous statistical models. Therefore, only the most common ones will be described and discussed.

(a) Autoregressive models
Autoregressive models (AR) are widely used for the forecasting of near-future values of a variable in a time series [77]. Its a linear model which obtains this forecasting using previous signal values to forecast the future value [78]. To be more specific, the variable of interest is forecasted with the use of a linear combination of past values of this specific variable of interest [79]. The number of previous values which are taken into account for the forecasting is referred to as the model's order [78]. A general p-order AR model is given by Equation 4 [80]:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t \tag{4}$$

In which subscript t denotes the current time, ϵ_t represents a random error (white noise), y_{t-1} corresponds to the value of time series at time $t-1$, and $\phi_0, .., \phi_p$ are AR model parameters [80], [79]. These model parameters must be estimated using for example the well-known least-squares method [78]. Among AR models used for forecasting of time series data, the most popular ones are the Autoregressive Moving Average (ARMA) and the Autoregressive Integrated Moving Average (ARIMA) [81]. According to Sikorska et al., the ARMA model should only be applied for stationary data (in which the mean and variance should are constant over time) as this model might remove a temporary trend [69]. An ARIMA model on the other hand, is not subjected to this problem and is therefore suitable for data which posses trends [69].

For all AR models, its development generally consists of three different phases as described by Skikorska et al. [69]:

1. *Model identification:* values for the order of the model as well as the moving average parts of the model's equation are hypothesized.

2. *Parameter estimation:* the parameters of the model's equation are estimated with non-linear optimization techniques such as the least-squares method.

3. *Model validation:* in order to validate the adequacy of the developed AR model, new unseen data is fed to the model and it's performance is analysed.

AR models are used by several studies such as Liu et al. which adopted the AR model for the RUL estimation of lithium-ion batteries [82]. In 2014, Qian et al. used the AR model for the modelling of the degradation process of bearings and the prediction of bearing failure 50 minutes in advance [83]. Another example of a study who employed the AR model is conducted by Escobet et al. which used it for the RUL estimation of an AC electric motor of a conveyor belt system [84].

(b) Wiener process models

The Wiener process is another frequently used statistical model for the modelling of degradation processes. A general Wiener process $(\gamma(t), t \geq 0)$ can be mathematically written as seen in Equation 5, where $B(.)$ denotes a standard Brownian motion, $\sigma > 0$ is the diffusion coefficient, $\Lambda(.)$ denotes a monotone increasing function which represents a time scale and λ denotes the drift parameter [85].

$$\gamma(t) = \lambda\Lambda(t) + \sigma B(\Lambda(t)) \tag{5}$$

The first passage time (required time for a variable to obtain a certain threshold value) of the degradation path to a predetermined failure threshold described by the Wiener process is given by an inverse Gaussian distribution [85]. The Wiener process has many applications in RUL estimations. For example, Wang used the Wiener process model for the degradation modelling of bridge beams due to chloride ion ingression [86]. Another application of the Wiener process model is the optimal scheduling of burn-in policies for LED lights which are based on the continuous degradation path that is in turn modelled by the Wiener process model [87]. Huang et al. used a variation of the Wiener process which is the so-called skew-Wiener process to estimate the RUL of roller ball bearings, as they found that the degradation speed of industrial machines can be described by a skew-normal distribution [30]. Lin and Lin adopted the Wiener process model for the modelling of the degradation process of aircraft engines and used it for RUL estimation [17].

Despite these positive applications, the Wiener process can only be used for degradation processes which are time homogeneous. For example, the degradation process of crack growth due to fatigue is inhomogeneous as it can grow faster or slower throughout the

crack propagation process [88]. Another drawback of using the Wiener process for degradation modelling is the fact that it possess the Markov property. This Markov property means that its future state only depends on the current state. As a result, this implies that given a current state, the future state of a degradation process only depends on this current state and is independent of the past [88].

(c) Random coefficient models

Random coefficient models are used to describe degradation processes which posses a certain degree of stochasticity [14]. This stochasticity is modelled by including random coefficients to degradation models and are typically assumed to be normally distributed [14]. The parameters of these models are usually estimated by utilizing a historical database of degradation data [89]. An example of a random coefficient model in which a random slope is estimated through a covariate z which in turn controls the effect of a covariate x on the dependent variable y is described by Muthén et al. [90] and given by Equation 6:

$$y_i = \alpha + \beta_0 x_i + \beta_1 x_i z_i + \delta_i x_i + \beta_2 z_i + \epsilon_i \tag{6}$$

In which β denotes varying slopes (called random slopes), x and z are covariates of the dependent variable y, ϵ and δ are residuals and allowed to covary [90]. Random coefficient models are often used for RUL estimation. Hu and Tse combined two exponential functions into a random coefficient model for the RUL prediction of field pump impellers [91]. Lu and Meeker proposed a random coefficient model for RUL estimation of machines using nonlinear mixed-effects to represent the degradation processes [92]. Additionally, this model also estimated the PDF of the RUL [92].

Given a discrete random variable, a PDF shows the probability distribution (that is, the probability of this discrete random variable having a specific value) [93]. Therefore, the fact that random coefficient models are able to estimate the PDF is very useful as it allows one to see the complete distribution of RUL values with corresponding probabilities instead of just a single RUL value without a probability (and thus thereby reducing uncertainty). On the other hand, their applications might be limited since the random coefficient model uses the assumption that the random coefficients follow a Gaussian distribution [14].

(d) Gamma process models

Another type of statistical models for RUL estimation which is commonly used and described in the literature is the gamma process model. The main assumption in a gamma process model is that the increments of the degradation process at disjoint time intervals are independent random variables following a gamma distribution [14]. Gamma process models are found to be especially useful when a certain degradation process develops gradually (monotonic) over time whilst having positive increments [88]. The RUL of systems or a component can be estimated using the gamma process probability distribution in combination with a certain predefined threshold value (to denote a component

to be failed) [94]. The PDF of a gamma process is given by Equation 7 [95]:

$$Ga(f|\alpha,\lambda) = \frac{\lambda^{(\alpha)}}{\Gamma(\alpha)} f^{(\alpha-1)} e^{-\lambda f} \tag{7}$$

Where f denotes a random quantity following a gamma distribution with scale parameter $\lambda \geq 0$ and shape parameter $\alpha > 0$. Furthermore, $\Gamma(\alpha)$ represents the gamma function of α and $\Gamma(\alpha) = (\alpha - 1)!$ [95]. According to the literature, Gamma process models are found to be effective when determining maintenance decisions and for RUL prediction [88]. This is mainly due to the relatively easy mathematical calculations as well as the fact that the underlying physics are easy to understand [88]. In 2010, Wang et al. used the gamma process model in order to provide better maintenance schedules of machinery [96]. A gamma process with random effects has been employed by Peng et al. as a degradation model for a lubrication system of a machine tool's spindle system [97]. Furthermore, a gamma process has been used to represent the degradation process in a model to predict the RUL of batteries [98]. Yan et al. applied it to predict the degradation PDF of a milling machine [99]. However, the applicability of gamma process models is limited as the noise in such a model is assumed to follow a gamma distribution [14]. Furthermore, it can only be used for monotonic processes. Moreover, likewise to the Wiener process model, it posses the Markov property [88].

(e) Inverse Gaussian process models

The Inverse Gaussian (IG) process model assumes a degradation process to have monotone independent increments which follow an IG distribution [14]. It has been first proposed by Wang and Xu and found to be a flexible model to represent degradation processes [100]. A simple IG process model (as defined by Peng et al. [101]) having a function $\Lambda(t)$ with parameter λ (shape parameter) and μ (mean) has been defined for a general degradation process $\{Y_s(t), t \geq 0\}$, if for any $t \geq s \geq 0$ and $Y_s(0) = 0$ the following holds:

1. The degradation process $Y_s(t)$ has independent increments. In other words, $Y_s(t_2) - Y_s(t_1)$ and $Y_s(s_2) - Y_s(s_1)$ are completely independent for $\forall\, t_2 > t_1 \geq s_2 > s_1$ [101].
2. $Y_s(t) - Y_s(s)$ is following an IG distribution $IG(\mu, \lambda)$ [101].

The IG process model is widely used in literature. Qin et al. employed the IG process model for the modelling of the development of metal-loss corrosion defects on energy pipelines [102]. Pan et al. applied the IG process model with random effect to describe the degradation process of a deteriorating laser system [103]. More recently, Huang et al. applied the IG process model with random effects for the wear modelling of cutting tools which is in turn used for the RUL estimation of these cutting tools [104]. One of the main advantages of using IG process models lies in the fact that they are capable of handling different kind of random effects [14]. Nevertheless, these models also possess the Markov property and can only be applied to monotonic degradation processes [14].

(f) Markov models
In 2003, Kharoufeh introduced Markov models to perform RUL estimations [105]. Kharoufeh described that a Markov model assumes a certain degradation process of a system or component to develop over time, whilst adhering to the Markov property within a finite state space [105]. Besides the assumption that a Markov model adheres to the Markov property, it is also generally assumed that one can directly reveal a system's HS using its sensor data [88], [14]. However, it is usually a challenge or impossible to observe a system's HS directly [14].

In order to overcome this obstacle and thus be able to reveal the degradation process of a hidden HS of a system, a Hidden Markov Model (HMM) has been introduced [14]. As described by Si et al. [88], a HMM consists of two stochastic processes:

1. A hidden Markov chain $\{Z_n, n \geq 0\}$. This hidden Markov chain describes the real deterioration state and is unobservable. It is assumed that this deterioration process transforms according to a hidden Markov chain on a finite state space [88].

2. An observable stochastic process $\{Y_n, n \geq 0\}$. This represents the obtained sensor data of the system or component of interest.

Applications of this HMM in the literature are for example to identify the HSs of cutting tools together with an estimation of their RULs [106]. Zhou et al. also employed the HMM to reveal relationships between hidden failures and analysed observation data of a continuous stirred tank reactor [107]. Bunks et al. constructed a HMM and applied it to a Westland helicopter gearbox dataset and were able to classify torque-levels of the gearbox as well as different defect types (such as planetary bearing corrosion), since each specific defect relates to a specific vibration frequency [108].

However, as mentioned before, a HMM assumes that the state transformation adheres to the Markov property which might not be a valid assumption for all degradation processes in general. Additionally, since the true state transformation is unknown, models which assume the Markov property should be considered to be an approximation [88]. In order to generalize the HMM, a Hidden Semi-Markov model (HSMM) has been proposed[88]. In light of this generalization, the unobserved state is allowed to be a semi-Markov chain instead of a full Markov chain as with the HMM [88]. An example of an application of a HSMM is the use of the HSMM for online degradation state estimation as well as the evolution of the RUL distribution [109]. A general drawback of all Markov models is that for the estimation of the transition probabilities between HSs a large number of training data is usually required [14].

(g) Proportional Hazard models
Another type of statistical models are Proportional Hazard (PH) models. They were introduced in paper by Cox [110] in 1972. Over the years, this paper became one of the most cited papers in statistical sciences because of its simplicity and generality [88].

Cox assumed a system's hazard rate to consist of two factors: a covariate function and a certain baseline hazard function [14]. The general Cox proportional hazard model as described by Hahshim and Weiderpass [111] is given by Equation 8:

$$\gamma(t) = \gamma_0(t)\ \exp(\beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k) \tag{8}$$

Where $\gamma_0(t)$ denotes the baseline hazard function (with all covariates being equal to zero), β being the regression coefficient of the covariate x and $\gamma(t)$ being the total hazard function that gives the probability of occurrence that a certain event (e.g. component failure) happens before time t [111].

PH models are used in many applications, for example for the RUL prediction of bearings [112], [113]. Another application of the PH model is the RUL estimation of low methane compressors by Tran et al [114]. Besides using the PH models for RUL estimations of machinery, other applications are for example in the medical field in which PM models are used to model survival probabilities of patients having a certain disease [111]. PH models are believed to produce increasingly accurate RUL estimations in case large amounts of event data (actual failure data) as well as condition monitoring data becomes available, since PH models use both data types [14]. Nevertheless, in practice, it is generally a challenge to obtain both types of data simultaneously [14]. A second drawback relates to the PH model's covariate function which must be described by a different type of stochastic model (for example a Markov model). This adds to the computational complexity of the model as a whole [14].

(h) Particle filter models

Particle Filter (PF) models have recently been developed for PHM applications, i.e., RUL estimations and are considered to be the state-of-the-art models for RUL estimations [115]. PF models are based on a statistical method so-called Bayesian inference [116]. This Bayesian inference is based on the Bayes' theorem [117] as described in Equation 9:

$$p(\boldsymbol{\Theta}|\boldsymbol{z}) \propto L(\boldsymbol{z}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}) \tag{9}$$

In which $\mathbf{z}$ denotes a vector containing observed data, $\boldsymbol{\Theta}$ represents a vector containing unknown parameters, $p(\boldsymbol{\Theta})$ denotes the prior PDF of $\boldsymbol{\Theta}$, $p(\boldsymbol{\Theta}|\mathbf{z})$ represents the posterior PDF of $\boldsymbol{\Theta}$ which is conditional on $\mathbf{z}$ and $L(\boldsymbol{z}|\boldsymbol{\Theta})$ denotes the PDF value of $\mathbf{z}$ which is conditional on the provided $\boldsymbol{\Theta}$ [116]. A general PF model consists of a measurement function h (Equation 11) and a state transition function f (Equation 10), a.k.a. the degradation model [116], [118]:

$$x_k = f(x_{k-1}, \boldsymbol{\theta_k}, v_k) \tag{10}$$

$$z_k = h(x_k, \omega_k) \tag{11}$$

In which x_k represents the degradation state of the system, k denotes the time step, z_k denotes the measured data and $\boldsymbol{\theta_k}$ represents a vector containing model parameters.

Furthermore, ω_k and v_k denote the measurement and process noise, respectively [116]. A graphic representation of the general PF model given by An et al. is depicted in Figure 2.3 [116].

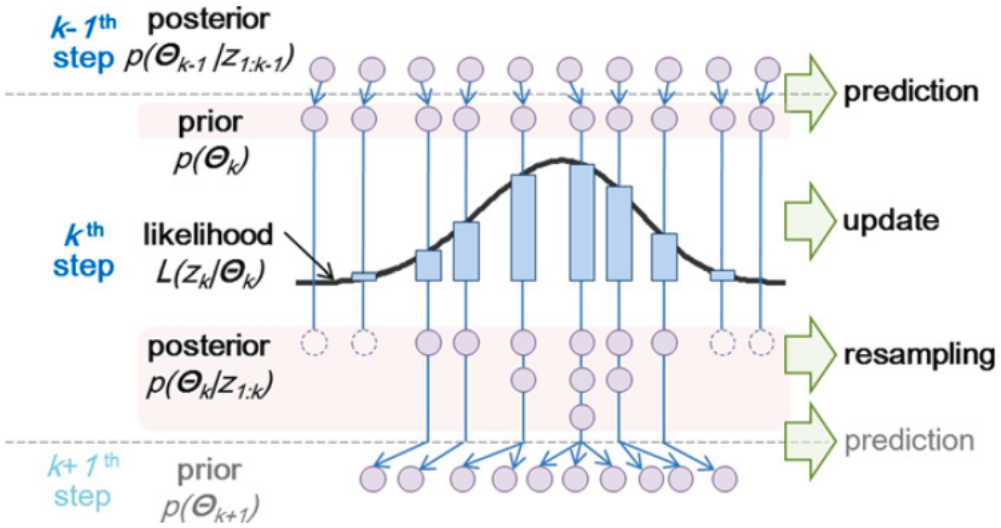


Figure 2.3: Graphic representation of a general PF model [116].

Imagine, at time $k = 1$, one wants to draw n samples (particles) of the parameters of a degradation model from an initial (prior) distribution [116]. In order to be able to this, the following four steps will be performed:

1. **Initialisation**
   N samples are taken from the system's initial state (i.e., parameters values of the degradation model) at time step $k - 1$ [115]. This is represented by the posterior distribution.

2. **Prediction**
   The posterior distribution of the degradation model parameters (at the $k-1^{th}$ step) is utilized as the prior distribution at the k^{th} step [116]. Then, the degradation state of the system (x_k) at current time step k, is predicted by filling in the parameter values (the samples) of the previous time step ($k - 1$) in the degradation model. These samples are denoted by $p(\boldsymbol{\Theta})$ [116].

3. **Updating**
   The update step is associated with likelihood function which evaluates if the particles are getting close to the measured data point at time step k and assigns weights to each particle [116]. The likelihood function ($L(\boldsymbol{z}|\boldsymbol{\Theta})$) takes as arguments: z_k and $\boldsymbol{\theta_k^i}$ (which also contains the system's degradation state x_k^i, given by the degradation model) with $i = 1, ..., n$ [116]. Particles with a high likelihood will receive a higher weight than particles with a relatively low likelihood.

4. **Resampling**
   During the resampling step, particles with low weights are removed whereas particles with high weights will be duplicated [116]. Thereafter, n new samples will be selected with a resampling method. Finally, these n resampled particles represent the posterior distribution at time step k, $p(\boldsymbol{\theta_k}|\boldsymbol{z_{1:k}})$, and also represent the prior distribution at the next time step $(k+1)$ [116].

This process is generally repeated until satisfactory parameter values are obtained for the degradation model. Thereafter, RUL estimation can be performed by inserting the model parameters into the degradation model and evaluating the model for future time steps until a certain failure threshold is reached [116]. Besides obtaining a single RUL estimation value when using PF models, it is possible to obtain a RUL probability distribution.

Several studies already successfully used PF models such as for RUL estimation of aircraft cooling units [34], for the RUL estimation of turbine blades being affected by creep growth [119], but also for the RUL estimation of Lithium-ion batteries [120], [121] as well as to estimate the RUL of LED light sources [122]. One of the main advantages of PF models is that it can be used for nonlinear systems with non-Gaussian characteristics [121]. Furthermore, PF models can be used for the modelling of multivariate dynamic processes. On the other hand, a drawback is that PF models are computationally intensive [69]. Another drawback is that PF models require a large number of samples in order to avoid the degeneracy problem [69]. The degeneracy problem can be described as the case that after a while, only very few particles will still remain with a significant weight, whilst all other particles will have small weights [123].

2.5.3 Artificial intelligence models

Artificial Intelligence (AI) models differ from the physical and statistical models as they are able to learn patterns in data (in this case degradation patterns from sensor data) [14]. Over the years, AI models have become more attractive to model degradation processes and perform RUL estimations of machines, because these models are able to handle mechanical systems with complex deterioration process, which are usually too difficult to be accurately described by physical and/or statistical models [14]. However, a general drawback of AI models is that, due to the lack of transparency, it is difficult to explain their results which is why AI models are referred to as "black-boxes" [14], [34]. Since a lot of different AI models are described in literature for degradation modelling and RUL estimation, only the most commonly used models will be described and discussed.

(a) Artificial Neural Network models

An Artificial Neural Network (ANN) model is a machine-learning model which mimics the working process of the human brain through the connection of a large number of nodes [124], [14].

It is used for several purposes such as pattern recognition, clustering and prediction using a large amount of training data [124]. Therefore, ANN models are the most widely used models within AI models for the estimation of RUL [14]. ANN models typically consist of an input layer, a hidden layer and an output layer [125]. First, the input layer uses raw input data from the sensors, after which the hidden layer receives and processes this raw data [125]. Thereafter, the output layer will receive the obtained value from the hidden layer and will finally produce an output (RUL value) [125].

A graphical representation of a simple node from an ANN model is depicted in Figure 2.4. In this figure, the inputs of the model are denoted by $A(N)$, which are multiplied with their corresponding weights $W(N)$ representing the 'impact' of this input node [125]. Thereafter, the so-called weighted inputs and a bias are sent to a summation function. This bias (has a value of 1 in the figure) is usually required to create a better fit to the data (means better RUL estimation) by being able to shift the activation function [126]. Finally, the value obtained by the summation function is fed into the activation function (which determines if this node should be activated depending on the value of the summation function) to eventually produce the output value [127], [125]. This activation function (sometimes called transfer function) can be defined as any mathematical function, but the most commonly used ones are the Linear function, Nonlinear (Sigmoid) function and the Step function [127].

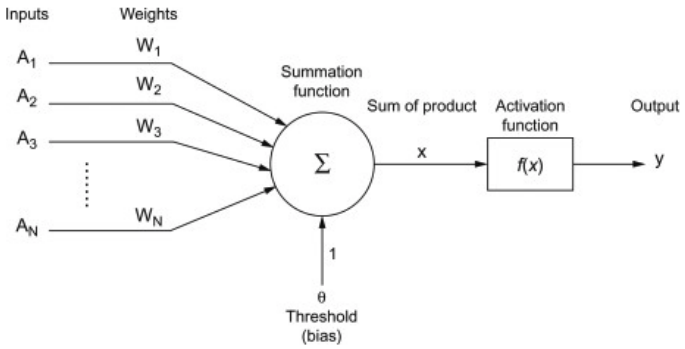


Figure 2.4: Simple node from ANN [125]

A general mathematical representation of an ANN model is provided by Krenker et al. [127] and is given by Equation 12:

$$y(k) = F(\sum_{i=0}^{m} w_i(k) \cdot x_i(k) + b) \tag{12}$$

In which $x_i(k)$ represents the input values at the input nodes at discrete time step k, $w_i(k)$ are the weights corresponding to each input node, b represents a bias, F denotes the activation function and $y(k)$ denotes the output of the ANN model [127]. Among the ANN models, the Feed-forward Neural Network (FFNN) model and the Recurrent Neural Network (RNN) model are the most applied for RUL estimations [14].

FFNN models
In a FFNN model, data always 'flows' in one direction from the input layer to the output layer without any back-loops [127]. The list of studies who employed FFNN is almost endless, therefore only a few studies and their applications will be mentioned. Examples of studies who used FFNN are for the RUL estimation of aluminium plates being affected by fatigue cracks [128], the RUL estimation of Lithium-ion batteries [129], but also to learn a relationship between the RUL and constructed HIs of electrical motors [130].

RNN models and improved RNN models
As already mentioned, RNN models are also popular for RUL estimations. This is mainly due to its capability of being able to handle explicit time-series data [14]. A RNN model differs from a FFNN model as data can 'flow' forwards as well as backwards between different nodes and layers [127]. Some examples of studies who employed RNN are Guo et al. who proposed a RNN based on a HI for the RUL prediction of bearings [46] and Heimes who applied RNN to estimate the RUL of aircraft engines [131]. Besides the standard RNN, several improvements are proposed such as by Song et al. who used a modified version of the RNN (RNN with Gated Recurrent Unit) for the RUL estimation of Lithium-ion batteries [132]. Another improvement of the RNN model is the Long Short Term Memory (LSTM) model which is for example applied to predict the RUL of aircraft engines having complicated operational conditions [133], [134].

Generally, ANN models are capable of learning nonlinear complex relationships in data which allows for accurate RUL estimations [14]. Nevertheless, as explained previously, ANN models are considered as "black-boxes" since they are not transparent [14]. Furthermore, in order for ANN models to function properly, large amounts of training data must be available (which is generally not available in practice) [14].

(b) Neuro-fuzzy models
Neuro-fuzzy (NF) models are fuzzy logic systems and use ANN optimized membership functions as well as expert knowledge based membership functions to determine their inference structures [135]. Figure 2.5 [136] shows a schematic representation of a general NF model. In this figure, the ANN creates neural outputs which are the fuzzy inputs for the fuzzy inference. During the training, the weights within the ANN are changed in order to match the output of the ANN with its target [136]. These weights represent the parameters for the membership functions as well as for the fuzzy if-then rules [136], [135].
NF systems are frequently used for RUL estimation applications as described in the literature. A NF model has been adopted by Wang et al. to predict the RUL of gears in a gearbox being affected by wear as well as cracks [137]. NF models have also been applied for the prediction of the health condition and RUL estimation of bearings [138]. Also slightly different versions of the NF models are employed such as the adaptive NF inference system to predict the trend of a vibration based HI of gearboxes of wind turbines [139].

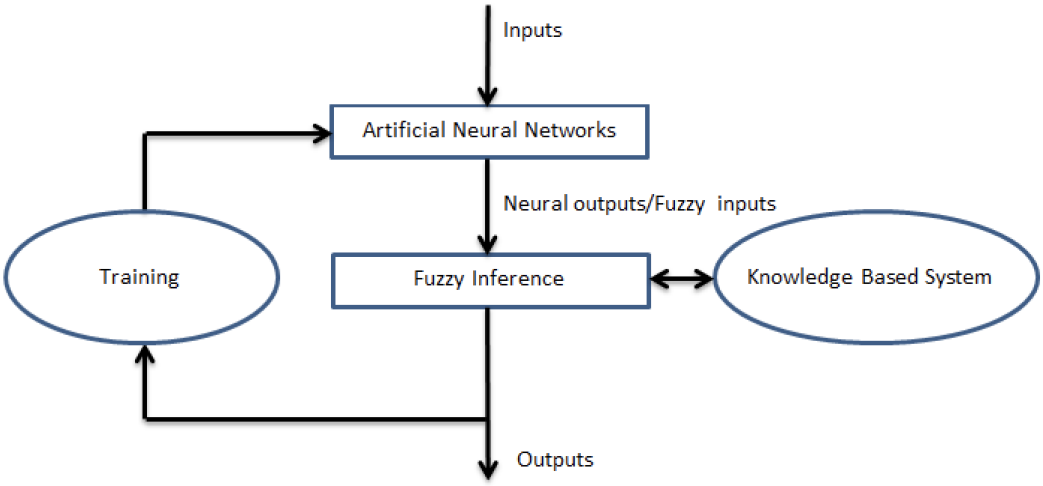


Figure 2.5: Schematic representation of a general NF system [136]

The main advantage of NF systems is that they combine knowledge from experts with ANN intelligence which is believed to be a powerful tool according to the literature [14]. On the other hand, it also requires a large amount of training data in order to create accurate predictions [14].

(c) Random Forest Regression models

Random Forest Regression (RFR) models are part of decision trees, which in turn are machine learning algorithms [140]. A RFR model uses an ensemble method which means that the model consists of many individual trees whose corresponding predictions are combined into an average point estimate for the target variable (e.g. RUL value) [141], [140]. A graphical representation of the structure of a general random forest model is depicted in Figure 2.6.

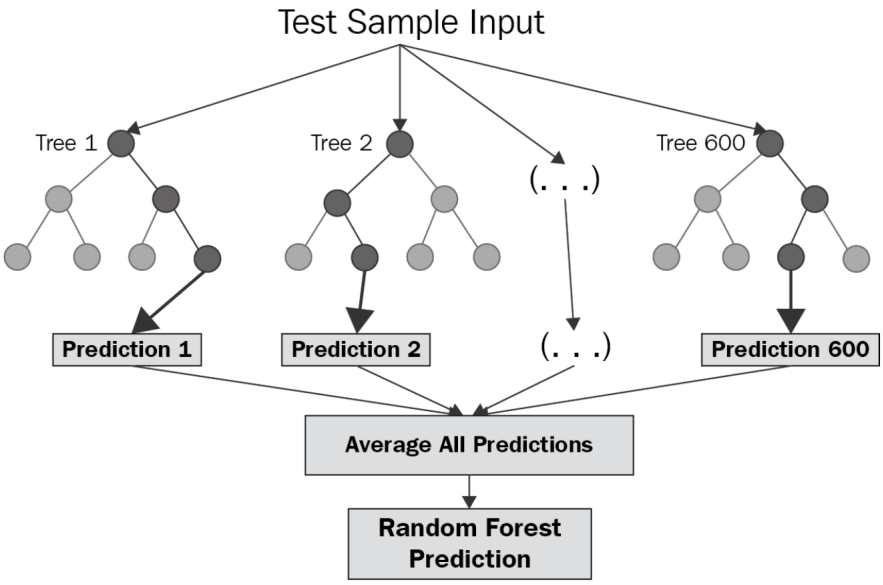


Figure 2.6: Structure of a general random forest model [140].

RFR models have so-called hyperparameters such as for example the number of estimators considered in a tree when splitting a node, or the maximum number of trees which will be built in the overall forest [141]. An example of a study who employed a RFR-based approach is conducted by Chen et al. who used RFR to predict the RUL of aircraft engines using engine deterioration data [142]. However, this RUL prediction is a single value, i.e. a single point estimate. Nevertheless, besides producing single RUL point estimates with the RFR model by computing the average value of all decision trees (as is usually done when computing a single RUL value), some studies collected all of them instead and applied a kernel density estimation to obtain a normalized PDF [141], [143]. The study conducted by Zoutendijk and Mitici used the RFR model plus kernel density estimation to estimate the PDF of flight delays using deterministic values of the considered features, thereby circumventing the need for stochastic variables [141]. Whilst other studies such as Schlosser et al. employed RFR to estimate the probability distribution of perception forecasts in a mountainous region, and used stochastic variables for the construction of the decision trees [144], [141].

The main advantage of RFR models is that, given a set of variables, RFR computes the relative importance of each variable on the prediction created in the decision trees [145]. Therefore, it provides a useful inside in the importance of variables to the prediction (RUL value). On the other hand, the main disadvantage of RFR models is that the runtime of the algorithm becomes large as an extensive number of decision trees is required to create accurate predictions, thereby making it less suitable for real-time forecasting applications [145].

(d) Gaussian Process Regression models
Gaussian Process Regression (GPR) models are considered as effective models for estimating predictions and the development of forecasting models according to the literature [146]. A GPR model performs regression based on a Gaussian process [14], [146]. A Gaussian process has been described by Lei et al. as "*a cumulative damage process consisting of random variables having joint multivariate Gaussian distributions*" [14]. Only a covariance function and a mean function are required to be able to completely describe a Gaussian process [147]. The mathematical representation of the Gaussian process, mean function and covariate function, as described by Hong and Zhou, is given in Equation 13, Equation 14 and Equation 15, respectively [147]:

$$f(x) \sim GP\Big(\mu(x), C(x, x')\Big) \tag{13}$$

$$\mu(x) = E[f(x)] \tag{14}$$

$$C(x, x') = E[\Big(f(x) - \mu(x)\Big)\Big(f(x') - \mu(x')\Big)] \tag{15}$$

Where $x, x' \in X$ represent random variables and $E(.)$ denotes the expected value of the random variables [147]. The mean value that will be predicted in a Gaussian process consists of a linear combination of the covariance function (Equation 15) [147]. As explained, GPR models are based on the Gaussian process and are described as a probabilistic method suitable for nonlinear regression [147]. It is capable of calculating estimations of the posterior degradation through means of constraining the prior distribution in order to fit the training data [147]. To be able to use the GPR model, the covariance function needs to be specified by the user, whilst the training data will be used to learn the required hyperparameters [147].

Examples of applications of the GPR model are for example to estimate the RUL of bearings as performed by Hong et al. in 2014 [45]. A GPR based model for RUL estimation of batteries has been applied by Saha et al. [148]. In 2020, Tanwar and Raghavan even used the GPR model to estimate the RUL of lubricating oil of a general oil lubrication system by predicting how many hours are remaining before the oil is contaminated with a certain predetermined level of wear debris [149]. The main advantages of using GPR models is that they are generally found to be suitable in case of high dimensional data, but also for small sample sizes and nonlinear dynamic systems [45], [150]. The largest disadvantage of GPR models is their high computational demand [14], [150].

(e) Support Vector Machine and Relevant Vector Machine models
Support Vector Machine (SVM) models, also called maximum margin classifiers [150], are a type of AI technique which use the statistical learning theory that is introduced by Vapnik [151]. In general, a SVM model has samples which have two different classes (a positive and a negative class) [150]. A SVM model tries to optimize a boundary curve through maximization of the distance of the closest point to this boundary curve [150].

The number of support vectors (this is a subset of the available training data for the two different classes) define this boundary curve [150]. SVM models are able to produce predictions in terms of a linear combination of kernel functions (the support vectors) which are centred on a particular subset of the training data [152].

In the beginning, SVM models were only used for pattern recognition applications [153]. Only after Vapnik introduced an insensitive loss function, SVM models were also applied to time series forecasting, nonlinear regression applications and RUL estimations [14], [150]. As described in the literature, several types of SVM models have been used for RUL estimation such as the one-class SVM model [154], multi-class SVM model [155], [156] and the least squares SVM model [157]. However, according to Benkedjouh et al., Support Vector Regression (SVR) models are the most widely used form of SVM models for RUL estimations [158]. SVR models are suitable to build empirical models using available training data [159]. An application of the SVR model is a study conducted Wang et al. who applied the SVR model for the estimation of the RUL of Lithium-ion batteries [160]. Another application is the study conducted by Liu and Zio who employed the SVR model for the RUL estimation of components in a nuclear power generation system [159].

Besides the large number of applications of SVM models, its main disadvantage is that they can only generate point predictions instead of a complete probability distribution [152]. In order to overcome this main disadvantage, the Relevant Vector Machine (RVM) model has been proposed, which function form is the same as that of the SVM [152]. However, RVM models are capable of producing a complete predictive distribution [152]. Examples of applications of the RVM model in literature are mostly for RUL estimation of Lithium-ion batteries [161], [162] and RUL estimation of bearings [43]. According to Lei et al., SVM models and RVM models might be more applicable for RUL estimations than other AI models in cases where a limited amount of data is available [14]. Nevertheless, the selected kernel functions highly influence their performance, whereas a standard method for selecting kernel functions does not exist yet [14]. In addition, estimation of the parameter values is considered to be a challenge for both SVM and RVM models [14].

2.5.4 Hybrid models

Hybrid models try to combine the advantages of several RUL estimation models into a single model, thereby trying to create an even better model in terms of prediction accuracy [14]. However, hybrid models are the least used and reported models for RUL estimation throughout the literature. An explanation for this might be that it is more difficult and complex to fuse multiple methods together into one single hybrid model, than to use the single models individually. An example of a study who did apply a hybrid model is a study conducted by Tamssaouet et al. who proposed the combination of a LSTM model with a PF model to perform RUL estimations of aircraft engines and also produced the PDF associated with the RUL estimation [163].

Some other applications of a hybrid model to estimate the RUL of aircraft engines is the study conducted by Khan et al. [164] who proposed a hybrid model through the integration of a neural network with a PF model for aircraft engine RUL estimation and the study conducted by Ji et al. [165]. who fused PCA with a bidirectional LSTM to learn the relationship between RUL and the HS monitoring data. Besides using hybrid models for RUL estimation of aircraft engines, some papers proposed hybrid models for other applications such as the prediction of the HS of bearings by fusion of a HMM with a NF model [166] or the RUL estimation of a milling machine through combining a LSTM model with a HMM into a hybrid model [167].

2.6 Faulty component identification

Faulty component identification is concerned with the identification of components that will fail within a system or machine, but does not include information on what causes the component to fail. Whereas the previous sections (Paragraph 2.2, Paragraph 2.3, Paragraph 2.4 and Paragraph 2.5) focused on studies who proposed methods for the PHM process, this paragraph will address studies who explicitly tried to identify faulty components, as that appeared to be of little interest to the researchers who focused on RUL estimations in the previous sections.

Faulty component identification should not be confused with failure mode identification. A failure mode is defined by Matthews as [168]:

Definition: *A failure mode is a possible way the component can fail.*

An example of failure mode of the compressor in an aircraft gas turbine engine could be that at a certain point in time, the gas flow through the compressor is unable to reach the initially designed velocity. Such as failure mode could be caused by a variety of factors such as wear or corrosion of the compressor blades (which are called failure mechanisms [169]). Common methods as described in the literature to identify failure modes in systems or machinery components are: Fault Tree Analysis (FTA), Failure Modes, Effects and Criticality Analysis (FMECA), Failure Modes and Effects Analysis (FMEA) as well as expert knowledge [170].

However, suitable data must be available to be able to apply these failure mode identification methods. Nevertheless, another possibility is to identify faulty components within a system, rather than the specific failure modes of a particular component. An example of this approach is proposed by Schwartz et al. who used a SOM model together with a kernel density estimator on a dataset consisting of deteriorating aircraft engines operating under different operational conditions (such as a varying Mach number and altitude) [171].

First, the SOM model has been applied to create clusters which correspond to the different operational conditions of the engines. Then, the kernel density estimator has been applied to produce PDFs that relate to the number of faulty components per operational condition [171]. The kernel density estimator was capable of correctly identifying the number of faulty components in the considered different datasets (some datasets had only one faulty component, whilst others had two faulty components) [171].

2.7 Discussion and literature gap

It becomes evident from the extensive literature review from the previous sections, that most research has focused on techniques to construct HIs which can be used to divide HSs and eventually perform accurate RUL estimations. However, these RUL estimations usually just provide a single value (such as 50 remaining hours before the system of interest will fail), with a consequently unknown probability of this RUL value. Very few studies have given the complete probability distribution of the RUL estimation, thereby including the corresponding probabilities to the provided RUL values and thus taking into account uncertainty. Additionally, little to no research has been conducted into methods which reveal which specific component(s) will fail within a system. Let alone a model that first identifies which component will fail in a system and additionally provides a complete probability distribution of the RUL of this system, thus taking into account RUL estimation uncertainty.

Since this thesis will focus on aircraft gas turbine engines and given the fact that aircraft engines consist of multiple components which could fail separately and thus require maintenance, it is useful for an airline to know which components inside the engine are going to fail along with the engine's associated RUL and the corresponding certainty of this RUL estimation. This is valuable information as it allows airlines to order specific engine parts only when required (thereby reducing storage costs) instead of being forced to have these parts in stock continuously because the engine parts have to replaced preventively after a predefined number of flight hours or flight cycles. Furthermore, knowing in advance which component will fail along with the engine's RUL and its corresponding probability will also allow for more efficient maintenance operations, as the required personnel to remove and install the component as well as the hanger (which is required to perform the maintenance) can be allocated and planned in advance. Finally, it can be argued that the most valuable part might even be that the uncertainty of the RUL estimations will be provided. One can understand its importance since knowing the corresponding probability of a certain estimated RUL value is really beneficial when scheduling aircraft engine maintenance. Therefore, the need arises for the development of a model that is capable to identify which components of an aircraft gas turbine engine will fail, along with its corresponding RUL and associated probability.

3 Research proposal

This chapter thoroughly describes the research proposal of this thesis project. It uses the extensively discussed state-of-the-art literature review together with the described literature gap as well as the problem statement in the introduction to establish a well defined research objective (Paragraph 3.1) together with its associated research questions (Paragraph 3.2). Furthermore, the research scope will be defined in Paragraph 3.3 since a clearly predefined scope is required in order to be able to finish this thesis within the nominal time.

3.1 Research objective

After performing the extensive literature review, obtaining the literature gap and using the problem definition as described in the introduction, the research objective has been established and defined. Taking into account the RUL estimation of aircraft engines having one or multiple faulty engine components and considering the accompanied uncertainty of this RUL estimation results in the following research objective:

> ***The research objective of this thesis is to develop a model which is capable of identifying faulty components of aircraft engines and estimate their remaining useful life, whilst taking into account and modelling the corresponding uncertainty of this remaining useful life estimate by using the NASA engine degradation data set.***

3.2 Research questions

From this research objective, the following main research question has been formulated:

> ***What type of model must be developed to identify faulty components of aircraft engines and estimate their corresponding remaining useful life, whilst taking into account and modelling the uncertainty of this remaining useful life estimate using the NASA engine degradation data set?***

In order to be able to answer this main research question, the following sub-research questions have been defined:

> ***1) Which sensor data is relevant for identifying faulty aircraft engine components in the NASA engine degradation data set?***

> ***2) How to identify these faulty components of aircraft engines in the NASA engine degradation data set and when are they considered to be in a faulty state?***

3) How to estimate the remaining useful life of aircraft engines having an identified faulty component along with the corresponding uncertainty of this estimate?

4) How to verify and validate the model's identification of faulty engine components as well as its remaining useful life estimations of these aircraft engines and corresponding uncertainties of these estimations?

5) Does considering an aircraft engine's faulty component when estimating the remaining useful life lead to a statistically significant improvement compared to not taking into account an engine's faulty component?

3.3 Research scope

It is important to clearly define the research scope of this thesis and thereby describe what aspects will be included in the thesis and what aspects will be excluded. First of all, only the first and third data set (FD001 and FD003, respectively) within the NASA engine degradation data set [172] will be used. These data sets both have the same operational condition, that is, all engines are simulated to operate at sea level. The reason to select these two data sets is that it is easier to compare different aircraft engines with each other if they all operate in the same conditions. Hence, it is believed to produce more accurate RUL estimations. The second reason to select only these data sets is that these data sets contain one and two faulty components, respectively.

Another part of the scope of this thesis is that only the faulty components within the engine will be identified. As a result, no analysis will be performed on the failure modes or failure mechanisms that cause these components to become classified as faulty. Lastly, when estimating the RUL the aircraft engines, not only a single RUL value will be provided without any associated probability. Instead, a complete RUL probability distribution will be provided, therefore also providing the probability that a certain predicted RUL value will actually occur.

References

[1] International Civil Aviation Organisation. *The World of Air Transport in 2019*. 2019. URL: https://www.icao.int/annual-report-2019/Pages/the-world-of-air-transport-in-2019.aspx (visited on 09/27/2021).

[2] Xiaoqian Sun et al. "COVID-19 pandemic and air transportation: Successfully navigating the paper hurricane". In: *Journal of Air Transport Management* 94 (2021), p. 102062. ISSN: 0969-6997. DOI: https://doi.org/10.1016/j.jairtraman.2021.102062.

[3] S.V. Gudmundsson, M. Cattaneo, and R. Redondi. "Forecasting temporal world recovery in air transport markets in the presence of large economic shocks: The case of COVID-19". In: *Journal of Air Transport Management* 91 (2021), p. 102007. ISSN: 0969-6997. DOI: https://doi.org/10.1016/j.jairtraman.2020.102007.

[4] International Air Transport Association. *An almost full recovery of air travel in prospect*. May 2021. URL: https://www.iata.org/en/iata-repository/publications/economic-reports/an-almost-full-recovery-of-air-travel-in-prospect/ (visited on 09/28/2021).

[5] International Air Transport Association. *Airline Maintenance Cost Executive Commentary*. Jan. 2021. URL: https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2019-mctg-report_public.pdf (visited on 09/28/2021).

[6] Sander de Bree. *BIG DATA IN AVIATION - REDUCE COSTS THRU PREDICTIVE MAINTENANCE*. URL: https://www.exsyn.com/blog/big-data-in-aviation-predictive-maintenance (visited on 09/28/2021).

[7] Nico Hölzel and Volker Gollnick. "Cost-benefit Analysis of Prognostics and Condition-based Maintenance Concepts for Commercial Aircraft Considering Prognostic Errors". In: Proceedings of the Annual Conference of the PHM Society 2015, vol. 7, no. 1. Oct. 2015. DOI: https://doi.org/10.36001/phmconf.2015.v7i1.2716.

[8] Rui Li, Wim J.C. Verhagen, and Richard Curran. "Toward a methodology of requirements definition for prognostics and health management system to support aircraft predictive maintenance". In: *Aerospace Science and Technology* 102 (2020), p. 105877. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2020.105877.

[9] H. M. Hashemian. "State-of-the-Art Predictive Maintenance Techniques". In: *IEEE Transactions on Instrumentation and Measurement* 60.1 (2011), pp. 226–236. DOI: https://doi.org/10.1109/TIM.2010.2047662.

[10] Jimmy Iskandar et al. "Predictive Maintenance in semiconductor manufacturing". In: *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. 2015, pp. 384–389. DOI: https://doi.org/10.1109/ASMC.2015.7164425.

[11] Saeid Mokhatab, William A. Poe, and John Y. Mak. "Chapter 17 - Gas Processing Plant Operations". In: *Handbook of Natural Gas Transmission and Processing (Fourth Edition)*. Ed. by Saeid Mokhatab, William A. Poe, and John Y. Mak. Fourth Edition. Gulf Professional Publishing, 2019, pp. 509–535. ISBN: 978-0-12-815817-3. DOI: https://doi.org/10.1016/B978-0-12-815817-3.00017-4.

[12] Francesca Calabrese et al. "Predictive Maintenance: A Novel Framework for a Data-Driven, Semi-Supervised, and Partially Online Prognostic Health Management Application in Industries". In: *Applied Sciences* 11.8 (2021). ISSN: 2076-3417. DOI: https://doi.org/10.3390/app11083380.

[13] Gian Antonio Susto et al. "Machine learning for predictive maintenance: A multiple classifier approach". In: *IEEE Transactions on Industrial Informatics* 11 (3 June 2015), pp. 812–820. ISSN: 15513203. DOI: https://doi.org/10.1109/TII.2014.2349359.

[14] Yaguo Lei et al. *Machinery health prognostics: A systematic review from data acquisition to RUL prediction*. May 2018. DOI: https://doi.org/10.1016/j.ymssp.2017.11.016.

[15] E. Elnahrawy and B. Nath. "Cleaning and Querying Noisy Sensors". In: *Proceedings of the Second ACM International Workshop on Wireless Sensor networks and Applications, WSNA 2003*. 2003, pp. 78–87. DOI: https://doi.org/10.1145/941350.941362.

[16] Shivani Gupta and Atul Gupta. "Dealing with Noise Problem in Machine Learning Data-sets: A Systematic Review". In: *Procedia Computer Science* 161 (2019). The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia, pp. 466–474. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.11.146.

[17] Jingdong Lin and Zheng Lin. "Research on Remaining Useful Life Prediction for Aircraft Engine with a Fault Point". In: *Journal of Physics: Conference Series* 1549 (June 2020), p. 052065. DOI: https://doi.org/10.1088/1742-6596/1549/5/052065.

[18] Cunsong Wang et al. "A data-driven degradation prognostic strategy for aero-engine under various operational conditions". In: *Neurocomputing* 462 (2021), pp. 195–207. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.07.080.

[19] Zhao Liu et al. "Empirical mode decomposition based hybrid ensemble model for electrical energy consumption forecasting of the cement grinding process". In: *Measurement* 138 (2019), pp. 314–324. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2019.02.062.

[20] Cunsong Wang et al. "A Data-Driven Aero-Engine Degradation Prognostic Strategy". In: *IEEE Transactions on Cybernetics* 51.3 (2021), pp. 1531–1541. DOI: https://doi.org/10.1109/TCYB.2019.2938244.

[21] Steven W. Smith. *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1997.

[22] Sheida Ghorbani and Karim Salahshoor. "Estimating Remaining Useful Life of Turbofan Engine Using Data-Level Fusion and Feature-Level Fusion". In: *Journal of Failure Analysis and Prevention 2020 20:1* 20 (1 Jan. 2020), pp. 323–332. ISSN: 1864-1245. DOI: https://doi.org/10.1007/S11668-020-00832-X.

[23] Chao Hu et al. "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life". In: *Reliability Engineering & System Safety* 103 (2012), pp. 120–135. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2012.03.008.

[24] Bin Zhang et al. "Aircraft Engine Prognostics Based on Informative Sensor Selection and Adaptive Degradation Modeling with Functional Principal Component Analysis". In: *Sensors* 20 (Feb. 2020), p. 920. DOI: https://doi.org/10.3390/s20030920.

[25] Yang Ge, Jian Wu, and Xiao-Mei Jiang. "A Prediction Method Using Bayesian Theory for Remaining Useful Life". In: *2019 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*. 2019, pp. 856–862. DOI: https://doi.org/10.1109/QR2MSE46217.2019.9021252.

[26] Jamie Coble and J. Hines. "Applying the General Path Model to Estimation of Remaining Useful Life". In: *International Journal of Prognostics and Health Management* 2 (Jan. 2011), pp. 2153–2648. DOI: https://doi.org/10.36001/ijphm.2011.v2i1.1352.

[27] Jakob Spiegelberg and Ján Rusz. "Can we use PCA to detect small signals in noisy data?" In: *Ultramicroscopy* 172 (2017), pp. 40–46. ISSN: 0304-3991. DOI: https://doi.org/10.1016/j.ultramic.2016.10.008.

[28] Fernando Sánchez Lasheras et al. "A Hybrid PCA-CART-MARS-Based Prognostic Approach of the Remaining Useful Life for Aircraft Engines". In: *Sensors* 15.3 (2015), pp. 7062–7083. ISSN: 1424-8220. DOI: https://doi.org/10.3390/s150307062.

[29] Arnaz Malhi, Ruqiang Yan, and Robert X. Gao. "Prognosis of Defect Propagation Based on Recurrent Neural Networks". In: *IEEE Transactions on Instrumentation and Measurement* 60.3 (2011), pp. 703–711. DOI: https://doi.org/10.1109/TIM.2010.2078296.

[30] Zeyi Huang et al. "Remaining useful life prediction for an adaptive skew-Wiener process model". In: *Mechanical Systems and Signal Processing* 87 (2017), pp. 294–306. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2016.10.027.

[31] Haitao Liao and Zhigang Tian. "A framework for predicting the remaining useful life of a single unit under time-varying operating conditions". In: *IIE Transactions* 45.9 (2013), pp. 964–980. DOI: https://doi.org/10.1080/0740817X.2012.705451.

[32] Jiadong Meng et al. "Health Indicator of Bearing Constructed by rms-CUMSUM and GRRMD-CUMSUM With Multifeatures of Envelope Spectrum". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–16. DOI: https://doi.org/10.1109/TIM.2021.3054000.

[33] Yingchao Liu, Xiaofeng Hu, and Wenjuan Zhang. "Remaining useful life prediction based on health index similarity". In: *Reliability Engineering & System Safety* 185 (2019), pp. 502–510. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.02.002.

[34] Mihaela Mitici and Ingeborg de Pater. "Online Model-Based Remaining-Useful-Life Prognostics for Aircraft Cooling Units Using Time-Warping Degradation Clustering". In: *Aerospace* 8.6 (2021). ISSN: 2226-4310. DOI: https://doi.org/10.3390/aerospace8060168.

[35] P. Večeř, M. Kreidl, and R. Šmíd. "Condition Indicators for Gearbox Condition Monitoring Systems". In: *Acta Polytechnica* 45 (6 Jan. 2005). ISSN: 1805-2363. DOI: https://doi.org/10.14311/782.

[36] Suliman Shanbr et al. "Detection of natural crack in wind turbine gearbox". In: *Renewable Energy* 118 (2018), pp. 172–179. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2017.10.104.

[37] Anders Kallner. "Formulas". In: *Laboratory Statistics (Second Edition)*. Ed. by Anders Kallner. Second Edition. Elsevier, 2018, pp. 1–140. ISBN: 978-0-12-814348-3. DOI: https://doi.org/10.1016/B978-0-12-814348-3.00001-0.

[38] Faris Elasha et al. "Prognosis of a Wind Turbine Gearbox Bearing Using Supervised Machine Learning". In: *Sensors* 19.14 (2019). ISSN: 1424-8220. DOI: https://doi.org/10.3390/s19143092.

[39] Luana Batista et al. "A classifier fusion system for bearing fault diagnosis". In: *Expert Systems with Applications: An International Journal* 40 (Dec. 2013), pp. 6788–6797. DOI: https://doi.org/10.1016/j.eswa.2013.06.033.

[40] Tomasz Barszcz and Robert B. Randall. "Application of spectral kurtosis for detection of a tooth crack in the planetary gear of a wind turbine". In: *Mechanical Systems and Signal Processing* 23.4 (2009), pp. 1352–1365. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2008.07.019.

[41] Junda Zhu et al. "Survey of condition indicators for condition monitoring systems". In: *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014* (Jan. 2014), pp. 635–647. DOI: https://doi.org/10.36001/phmconf.2014.v6i1.2514.

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

[43] Achmad Widodo and Bo-Suk Yang. "Application of relevance vector machine and survival probability to machine degradation assessment". In: *Expert Systems with Applications* 38.3 (2011), pp. 2592–2599. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.08.049.

[44] Yongxiang Li et al. "A LS-SVM based Approach for Turbine Engines Prognostics Using Sensor Data". In: *2019 IEEE International Conference on Industrial Technology (ICIT)*. 2019, pp. 983–987. DOI: https://doi.org/10.1109/ICIT.2019.8755209.

[45] Sheng Hong et al. "Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method". In: *Digital Signal Processing* 27 (2014), pp. 159–166. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2013.12.010.

[46] Liang Guo et al. "A recurrent neural network based health indicator for remaining useful life prediction of bearings". In: *Neurocomputing* 240 (2017), pp. 98–109. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.02.045.

[47] Zeqi Zhao et al. "Remaining useful life prediction of aircraft engine based on degradation pattern learning". In: *Reliability Engineering & System Safety* 164 (2017), pp. 74–83. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2017.02.007.

[48] Xiong Xinxin, Li Qing, and Cheng Nong. "Remaining useful life prognostics of aircraft engine based on fusion algorithm". In: *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*. 2016, pp. 628–633. DOI: https://doi.org/10.1109/CGNCC.2016.7828859.

[49] Eric Bechhoefer. "Data Driven Prognostics for Rotating Machinery". In: *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques* (Jan. 2012). DOI: https://doi.org/10.4018/978-1-4666-2095-7.ch006.

[50] R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. "The Mahalanobis distance". In: *Chemometrics and Intelligent Laboratory Systems* 50.1 (2000), pp. 1–18. ISSN: 0169-7439. DOI: https://doi.org/10.1016/S0169-7439(99)00047-7.

[51] Genichi Taguchi et al. *Taguchi's quality engineering handbook*. John Wiley & Sons, 2005. ISBN: 9780471413349. DOI: 10.1002/9780470258354.

[52] Qiang Li et al. "Health Indicator Construction Based on MD-CUMSUM With Multi-Domain Features Selection for Rolling Element Bearing Fault Diagnosis". In: *IEEE Access* 7 (2019), pp. 138528–138540. DOI: https://doi.org/10.1109/ACCESS.2019.2942371.

[53] Y. Wang et al. "A Two-Stage Data-Driven-Based Prognostic Approach for Bearing Degradation Problem". In: *IEEE Transactions on Industrial Informatics* 12.3 (2016), pp. 924–932. DOI: https://doi.org/10.1109/TII.2016.2535368.

[54] Sungho Suh et al. "Supervised Health Stage Prediction Using Convolutional Neural Networks for Bearing Wear". In: *Sensors* 20.20 (2020). ISSN: 1424-8220. DOI: https://doi.org/10.3390/s20205846.

[55] Vepa Atamuradov et al. "Prognostics and Health Management for Maintenance Practitioners-Review, Implementation and Tools Evaluation". In: *International Journal of Prognostics and Health Management* 8 (Dec. 2017), p. 31. ISSN: 21532648. DOI: https://doi.org/10.36001/ijphm.2017.v8i3.2667.

[56] Xiaohang Jin et al. "Anomaly Detection and Fault Prognosis for Bearings". In: *IEEE Transactions on Instrumentation and Measurement* 65.9 (2016), pp. 2046–2054. DOI: https://doi.org/10.1109/TIM.2016.2570398.

[57] Bin Zhang et al. "A Probabilistic Fault Detection Approach: Application to Bearing Fault Detection". In: *IEEE Transactions on Industrial Electronics* 58.5 (2011), pp. 2011–2018. DOI: https://doi.org/10.1109/TIE.2010.2058072.

[58] J. K. Kimotho et al. "Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution - Particle swarm optimization". In: *Chemical Engineering Transactions* 33 (2013), pp. 619–624. DOI: https://doi.org/10.3303/CET1333104.

[59] Yaogang Hu et al. "A probability evaluation method of early deterioration condition for the critical components of wind turbine generator systems". In: *Mechanical Systems and Signal Processing* 76-77 (2016), pp. 729–741. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2016.02.001.

[60] Satyam Kumar. *Fuzzy C-Means Clustering —Is it Better than K-Means Clustering?* URL: https://towardsdatascience.com/fuzzy-c-means-clustering-is-it-better-than-k-means-clustering-448a0aba1ee7 (visited on 10/07/2021).

[61] Ming-Chuan Hung and Don-Lin Yang. "An efficient Fuzzy C-Means clustering algorithm". In: *Proceedings 2001 IEEE International Conference on Data Mining*. 2001, pp. 225–232. DOI: https://doi.org/10.1109/ICDM.2001.989523.

[62] Chenhui Ren et al. “A Clustering-Based Method for Health Conditions Evaluation of Aero-Engines”. In: *2019 Prognostics and System Health Management Conference (PHM-Paris)*. 2019, pp. 72–78. DOI: https://doi.org/10.1109/PHM-Paris.2019.00020.

[63] Junqiang Liu et al. “Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion”. In: *Reliability Engineering & System Safety* 214 (2021), p. 107807. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107807.

[64] Kristina P. Sinaga and Miin-Shen Yang. “Unsupervised K-Means Clustering Algorithm”. In: *IEEE Access* 8 (2020), pp. 80716–80727. DOI: https://doi.org/10.1109/ACCESS.2020.2988796.

[65] Will Badr. *Auto-Encoder: What Is It? And What Is It Used For? (Part 1) | by Will Badr | Towards Data Science*. Apr. 2019. URL: https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726 (visited on 10/06/2021).

[66] Jicong Fan, Wei Wang, and Haijun Zhang. “AutoEncoder based high-dimensional data fault detection system”. In: *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. 2017, pp. 1001–1006. DOI: https://doi.org/10.1109/INDIN.2017.8104910.

[67] Xiao-Sheng Si et al. “A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation”. In: *Mechanical Systems and Signal Processing* 35.1 (2013), pp. 219–237. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2012.08.016.

[68] Piero Baraldi, Francesca Mangili, and Enrico Zio. “Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data”. In: *Reliability Engineering & System Safety* 112 (2013), pp. 94–108. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2012.12.004.

[69] J.Z. Sikorska, M. Hodkiewicz, and L. Ma. “Prognostic modelling options for remaining useful life estimation by industry”. In: *Mechanical Systems and Signal Processing* 25.5 (2011), pp. 1803–1836. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2010.11.018.

[70] Dawn An, Nam H. Kim, and Joo-Ho Choi. “Practical options for selecting data-driven or physics-based prognostics algorithms with reviews”. In: *Reliability Engineering & System Safety* 133 (2015), pp. 223–236. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2014.09.014.

[71] P. Paris and F. Erdogan. “A Critical Analysis of Crack Propagation Laws”. In: *Journal of Basic Engineering* 85.4 (Dec. 1963), pp. 528–533. ISSN: 0021-9223. DOI: https://doi.org/10.1115/1.3656900.

[72] Alexandra Coppe et al. “Using a Simple Crack Growth Model in Predicting Remaining Useful Life”. In: *Journal of Aircraft* 49.6 (2012), pp. 1965–1973. DOI: https://doi.org/10.2514/1.C031808.

[73] D. Xu et al. “Residual fatigue life prediction of ball bearings based on Paris law and RMS”. In: *Chinese Journal of Mechanical Engineering (English Edition)* 25.2 (2012), pp. 320–327. DOI: https://doi.org/10.3901/CJME.2012.02.320.

[74] Fuqiong Zhao, Zhigang Tian, and Yong Zeng. "Uncertainty Quantification in Gear Remaining Useful Life Prediction Through an Integrated Prognostics Method". In: *IEEE Transactions on Reliability* 62.1 (2013), pp. 146–159. DOI: https://doi.org/10.1109/TR.2013.2241216.

[75] Yuning Qian, Ruqiang Yan, and Robert X. Gao. "A multi-time scale approach to remaining useful life prediction in rolling bearing". In: *Mechanical Systems and Signal Processing* 83 (2017), pp. 549–567. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2016.06.031.

[76] Yang Hu et al. "Online Performance Assessment Method for a Model-Based Prognostic Approach". In: *IEEE Transactions on Reliability* 65.2 (2016), pp. 718–735. DOI: https://doi.org/10.1109/TR.2015.2500681.

[77] Bing Long et al. "An improved autoregressive model by particle swarm optimization for prognostics of lithium-ion batteries". In: *Microelectronics Reliability* 53.6 (2013), pp. 821–831. ISSN: 0026-2714. DOI: https://doi.org/10.1016/j.microrel.2013.01.006.

[78] Vaibhav Gandhi. "Chapter 2 - Interfacing Brain and Machine". In: *Brain-Computer Interfacing for Assistive Robotics*. Ed. by Vaibhav Gandhi. San Diego: Academic Press, 2015, pp. 7–63. ISBN: 978-0-12-801543-8. DOI: https://doi.org/10.1016/B978-0-12-801543-8.00002-8.

[79] Rob J. Hyndman and George Athanasopoulos. *Forecasting : principles and practice, 2nd edition*. OTexts: Melbourne, Australia. 2018. URL: OTexts.com/fpp2 (visited on 10/12/2021).

[80] Charles Zaiontz. *Autoregressive Processes Basic Concepts*. URL: https://www.real-statistics.com/time-series-analysis/autoregressive-processes/autoregressive-processes-basic-concepts/ (visited on 10/12/2021).

[81] George.E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

[82] D. Liu et al. "Lithium-ion battery remaining useful life estimation based on nonlinear AR model combined with degradation feature". In: *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2012, PHM 2012*. 2012, pp. 336–342. DOI: https://doi.org/10.36001/phmconf.2012.v4i1.2165.

[83] Y. Qian, R. Yan, and S. Hu. "Bearing degradation evaluation using recurrence quantification analysis and kalman filter". In: *IEEE Transactions on Instrumentation and Measurement* 63.11 (2014), pp. 2599–2610. DOI: https://doi.org/10.1109/TIM.2014.2313034.

[84] Teresa Escobet, Joseba Quevedo, and Vicenç Puig. "A fault/anomaly system prognosis using a data-driven approach considering uncertainty". In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 2012, pp. 1–7. DOI: https://doi.org/10.1109/IJCNN.2012.6252688.

[85] Zhi-Sheng Ye, Nan Chen, and Yan Shen. "A new class of Wiener process models for degradation analysis". In: *Reliability Engineering & System Safety* 139 (2015), pp. 58–67. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2015.02.005.

[86] Xiao Wang. "Wiener processes with random effects for degradation data". In: *Journal of Multivariate Analysis* 101.2 (2010). Statistical Methods and Problems in Infinite-dimensional Spaces, pp. 340–351. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2008.12.007.

[87] Sheng-Tsaing Tseng, Jen Tang, and In-Hong Ku. "Determination of burn-in parameters and residual life for highly reliable products". In: *Naval Research Logistics (NRL)* 50.1 (2003), pp. 1–14. DOI: https://doi.org/10.1002/nav.10042.

[88] Xiao-Sheng Si et al. "Remaining useful life estimation – A review on the statistical data driven approaches". In: *European Journal of Operational Research* 213.1 (2011), pp. 1–14. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2010.11.018.

[89] Nagi Gebraeel, Alaa Elwany, and Jing Pan. "Residual Life Predictions in the Absence of Prior Degradation Knowledge". In: *IEEE Transactions on Reliability* 58.1 (2009), pp. 106–117. DOI: https://doi.org/10.1109/TR.2008.2011659.

[90] Bengt Muthén, Linda Muthén, and Tihomir Asparouhov. *Random Coefficient Regression*. 2015. URL: https://www.statmodel.com/download/Random_coefficient_regression.pdf (visited on 10/12/2021).

[91] Jinfei Hu and Peter W. Tse. "A Relevance Vector Machine-Based Approach with Application to Oil Sand Pump Prognostics". In: *Sensors* 13.9 (2013), pp. 12663–12686. ISSN: 1424-8220. DOI: https://doi.org/10.3390/s130912663.

[92] C. Joseph Lu and William Q. Meeker. "Using Degradation Measures to Estimate a Time-to-Failure Distribution". In: *Technometrics* 35.2 (1993), pp. 161–174. ISSN: 00401706. DOI: https://doi.org/10.2307/1269661.

[93] Roberto Pedace. *Econometrics and the Probability Density Function (PDF)*. URL: https://www.dummies.com/education/economics/econometrics/econometrics-and-the-probability-density-function-pdf/ (visited on 10/13/2021).

[94] Yang Zhang et al. "Remaining Useful Life Estimation Using Long Short-Term Memory Neural Networks and Deep Fusion". In: *IEEE Access* 8 (2020), pp. 19033–19045. DOI: https://doi.org/10.1109/ACCESS.2020.2966827.

[95] Mojtaba Mahmoodian and Chun Q. Li. "Chapter 11 - Stochastic failure analysis of defected oil and gas pipelines". In: *Handbook of Materials Failure Analysis with Case Studies from the Oil and Gas Industry*. Ed. by Abdel Salam Hamdy Makhlouf and Mahmood Aliofkhazraei. Butterworth-Heinemann, 2016, pp. 235–255. ISBN: 978-0-08-100117-2. DOI: https://doi.org/10.1016/B978-0-08-100117-2.00014-5.

[96] Xiaolin Wang et al. "Residual life estimation based on bivariate non-stationary gamma degradation process". In: *Journal of Statistical Computation and Simulation* 85.2 (2015), pp. 405–421. DOI: https://doi.org/10.1080/00949655.2013.824448.

[97] Weiwen Peng et al. "Leveraging Degradation Testing and Condition Monitoring for Field Reliability Analysis With Time-Varying Operating Missions". In: *IEEE Transactions on Reliability* 64.4 (2015), pp. 1367–1382. DOI: https://doi.org/10.1109/TR.2015.2443858.

[98] Qidong Wei and Dan Xu. "Remaining useful life estimation based on gamma process considered with measurement error". In: *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*. 2014, pp. 645–649. DOI: https://doi.org/10.1109/ICRMS.2014.7107275.

[99] Heng-Chao Yan, Jun-Hong Zhou, and Chee Khiang Pang. "Gamma process with recursive MLE for wear PDF prediction in precognitive maintenance under aperiodic monitoring". In: *Mechatronics* 31 (2015), pp. 68–77. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2015.05.009.

[100] Xiao Wang and Dihua Xu. "An Inverse Gaussian Process Model for Degradation Data". In: *Technometrics* 52.2 (2010), pp. 188–197. DOI: https://doi.org/10.1198/TECH.2009.08197.

[101] Peng Weiwen et al. "Degradation Analysis Using Inverse Gaussian Process Model With Random Effects: A Bayesian Perspective". In: Proceedings of the ASME Design Engineering Technical Conference, vol 8. Aug. 2013. DOI: https://doi.org/10.1115/DETC2013-12884.

[102] Hao Qin, Shenwei Zhang, and Wenxing Zhou. "Inverse Gaussian process-based corrosion growth modeling and its application in the reliability analysis for energy pipelines". In: *Frontiers of Structural and Civil Engineering* 7 (Sept. 2013). DOI: https://doi.org/10.1007/s11709-013-0207-9.

[103] Donghui Pan, Jia-Bao Liu, and Jinde Cao. "Remaining useful life estimation using an inverse Gaussian degradation model". In: *Neurocomputing* 185 (2016), pp. 64–72. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2015.12.041.

[104] Yuanxing Huang et al. "Remaining Useful Life Prediction of Cutting Tools Using an Inverse Gaussian Process Model". In: *Applied Sciences* 11.11 (2021). ISSN: 2076-3417. DOI: https://doi.org/10.3390/app11115011.

[105] Jeffrey P. Kharoufeh. "Explicit results for wear processes in a Markovian environment". In: *Operations Research Letters* 31.3 (2003), pp. 237–244. ISSN: 0167-6377. DOI: https://doi.org/10.1016/S0167-6377(02)00229-8.

[106] P. Baruah and R. B. Chinnam. "HMMs for diagnostics and prognostics in machining processes". In: *International Journal of Production Research* 43.6 (2005), pp. 1275–1293. DOI: https://doi.org/10.1080/00207540412331327727.

[107] Zhi-Jie Zhou et al. "A model for real-time failure prognosis based on hidden Markov model and belief rule base". In: *European Journal of Operational Research* 207.1 (2010), pp. 269–283. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2010.03.032.

[108] Carey Bunks, Dan McCarthy, and Tarik Al-Ani. "CONDITION-BASED MAINTENANCE OF MACHINES USING HIDDEN MARKOV MODELS". In: *Mechanical Systems and Signal Processing* 14.4 (2000), pp. 597–612. ISSN: 0888-3270. DOI: https://doi.org/10.1006/mssp.2000.1309.

[109] Tongshun Liu, Kunpeng Zhu, and Liangcai Zeng. "Diagnosis and Prognosis of Degradation Process via Hidden Semi-Markov Model". In: *IEEE/ASME Transactions on Mechatronics* 23.3 (2018), pp. 1456–1466. DOI: https://doi.org/10.1109/TMECH.2018.2823320.

[110] D. R. Cox. "Regression Models and Life-Tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972), pp. 187–202. DOI: https://doi.org/10.1111/j.2517-6161.1972.tb00899.x.

[111] Dana Hashim and Elisabete Weiderpass. "Cancer Survival and Survivorship". In: *Encyclopedia of Cancer (Third Edition)*. Ed. by Paolo Boffetta and Pierre Hainaut. Third Edition. Oxford: Academic Press, 2019, pp. 250–259. ISBN: 978-0-12-812485-7. DOI: https://doi.org/10.1016/B978-0-12-801238-3.65102-4.

[112] H. Liao, W. Zhao, and H. Guo. "Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model". In: *Proceedings - Annual Reliability and Maintainability Symposium*. 2006, pp. 127–132. DOI: https://doi.org/10.1109/RAMS.2006.1677362.

[113] Lu Wang, Li Zhang, and Xue-zhi Wang. "Reliability estimation and remaining useful lifetime prediction for bearing based on proportional hazard model". In: *Journal of Central South University* 22 (Dec. 2015), pp. 4625–4633. DOI: https://doi.org/10.1007/s11771-015-3013-9.

[114] Van Tung Tran et al. "Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine". In: *Mechanical Systems and Signal Processing* 32 (2012). Uncertainties in Structural Dynamics, pp. 320–330. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2012.02.015.

[115] Marine Jouin et al. "Particle filter-based prognostics: Review, discussion and perspectives". In: *Mechanical Systems and Signal Processing* 72-73 (2016), pp. 2–31. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2015.11.008.

[116] Dawn An, Joo-Ho Choi, and Nam Ho Kim. "Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab". In: *Reliability Engineering & System Safety* 115 (2013), pp. 161–169. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2013.02.019.

[117] Thomas Bayes and null Price. "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S". In: *Philosophical Transactions of the Royal Society of London* 53 (1763), pp. 370–418. DOI: https://doi.org/10.1098/rstl.1763.0053.

[118] Enrico Zio and Giovanni Peloni. "Particle filtering prognostic estimation of the remaining useful life of nonlinear components". In: *Reliability Engineering & System Safety* 96.3 (2011), pp. 403–409. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2010.08.009.

[119] Piero Baraldi et al. "Model-based and data-driven prognostics under different available information". In: *Probabilistic Engineering Mechanics* 32 (2013), pp. 66–79. ISSN: 0266-8920. DOI: https://doi.org/10.1016/j.probengmech.2013.01.003.

[120] M Dalal, J Ma, and David He. "Lithium-ion battery life prognostic health management system using particle filtering framework". In: *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 225 (Mar. 2011), pp. 81–90. DOI: https://doi.org/10.1177/1748006XJRR342.

[121] Yinjiao Xing et al. "An ensemble model for predicting the remaining useful performance of lithium-ion batteries". In: *Microelectronics Reliability* 53.6 (2013), pp. 811–820. ISSN: 0026-2714. DOI: https://doi.org/10.1016/j.microrel.2012.12.003.

[122] Jiajie Fan, Kam-Chuen Yung, and Michael Pecht. "Predicting long-term lumen maintenance life of LED light sources using a particle filter-based prognostic approach". In: *Expert Systems with Applications* 42.5 (2015), pp. 2411–2420. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.10.021.

[123] Emin Orhan. *Bayesian Inference: Particle Filtering*. Aug. 2012. URL: https://www.cns.nyu.edu/~eorhan/notes/particle-filtering.pdf.

[124] Seunghui Lee, Sungwon Jung, and Jaewook Lee. "Prediction Model Based on an Artificial Neural Network for User-Based Building Energy Consumption in South Korea". In: *Energies* 12.4 (2019). ISSN: 1996-1073. DOI: https://doi.org/10.3390/en12040608.

[125] Mohammed Imran and Sarah A. Alsuhaibani. "Chapter 7 - A Neuro-Fuzzy Inference Model for Diabetic Retinopathy Classification". In: *Intelligent Data Analysis for Biomedical Applications*. Ed. by D. Jude Hemanth, Deepak Gupta, and Valentina Emilia Balas. Intelligent Data-Centric Systems. Academic Press, 2019, pp. 147–172. ISBN: 978-0-12-815553-0. DOI: https://doi.org/10.1016/B978-0-12-815553-0.00007-0.

[126] Gavril Ognjanovski. *Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun*. 2019. URL: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a (visited on 10/19/2021).

[127] Andrej Krenker, Janez Bešter, and Andrej Kos. "Introduction to the Artificial Neural Networks". In: *Artificial Neural Networks*. Ed. by Kenji Suzuki. Rijeka: IntechOpen, 2011. Chap. 1. DOI: https://doi.org/10.5772/15751.

[128] C. Sbarufatti et al. "Sequential Monte-Carlo sampling based on a committee of artificial neural networks for posterior state estimation and residual lifetime prediction". In: *International Journal of Fatigue* 83 (2016). From Microstructure to Design: Advances in Fatigue of Metals, pp. 10–23. ISSN: 0142-1123. DOI: https://doi.org/10.1016/j.ijfatigue.2015.05.017.

[129] Ji Wu, Chenbin Zhang, and Zonghai Chen. "An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks". In: *Applied Energy* 173 (2016), pp. 134–140. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.04.057.

[130] Feng Yang et al. "Health Index-Based Prognostics for Remaining Useful Life Predictions in Electrical Machines". In: *IEEE Transactions on Industrial Electronics* 63.4 (2016), pp. 2633–2644. DOI: https://doi.org/10.1109/TIE.2016.2515054.

[131] Felix O. Heimes. "Recurrent neural networks for remaining useful life estimation". In: *2008 International Conference on Prognostics and Health Management*. 2008, pp. 1–6. DOI: https://doi.org/10.1109/PHM.2008.4711422.

[132] Yuchen Song et al. "Lithium-Ion Battery Remaining Useful Life Prediction Based on GRU-RNN". In: *2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS)*. 2018, pp. 317–322. DOI: https://doi.org/10.1109/ICRMS.2018.00067.

[133] Mei Yuan, Yuting Wu, and Li Lin. "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network". In: *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*. 2016, pp. 135–140. DOI: https://doi.org/10.1109/AUS.2016.7748035.

[134] Yuting Wu et al. "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks". In: *Neurocomputing* 275 (2018), pp. 167–179. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.05.063.

[135] Yaguo Lei et al. "Fault diagnosis of rotating machinery based on multiple ANFIS combination with GAs". In: *Mechanical Systems and Signal Processing* 21.5 (2007), pp. 2280–2294. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2006.11.003.

[136] Tarik Rashid and Haval Ahmed. "Building an Adaptive Neural Fuzzy Inference System for Revised General Test/Gross Point Average Problem". In: *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)* (Jan. 2012), pp. 56–65.

[137] Wilson Q. Wang, M.Farid Golnaraghi, and Fathy Ismail. "Prognosis of machine health condition using neuro-fuzzy systems". In: *Mechanical Systems and Signal Processing* 18.4 (2004), pp. 813–831. ISSN: 0888-3270. DOI: https://doi.org/10.1016/S0888-3270(03)00079-7.

[138] Fagang Zhao et al. "Neuro-fuzzy Based Condition Prediction of Bearing Health". In: *Journal of Vibration and Control* (2009). vol. 15, no. 7, pp. 1079–1091. DOI: https://doi.org/10.1177/1077546309102665.

[139] Sajid Hussain and Hossam A.Gabbar. "Vibration Analysis and Time Series Prediction for Wind Turbine Gearbox Prognostics". In: *International Journal of Prognostics and Health Management* 4.3 (Jan. 2013). DOI: https://doi.org/10.36001/ijphm.2013.v4i3.2144.

[140] Afroz Chakure. *Random Forest Regression*. June 2019. URL: https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f (visited on 10/20/2021).

[141] Micha Zoutendijk and Mihaela Mitici. "Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem". In: *Aerospace* 8.6 (2021). ISSN: 2226-4310. DOI: https://doi.org/10.3390/aerospace8060152.

[142] Xin Chen et al. "Direct Remaining Useful Life Estimation Based on Random Forest Regression". In: *2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai)*. 2020, pp. 1–7. DOI: https://doi.org/10.1109/PHM-Shanghai49105.2020.9281004.

[143] Lei Zhang et al. "Probability Density Forecasting of Wind Speed Based on Quantile Regression and Kernel Density Estimation". In: *Energies* 13.22 (2020). ISSN: 1996-1073. DOI: https://doi.org/10.3390/en13226125.

[144] Lisa Schlosser et al. "Distributional regression forests for probabilistic precipitation forecasting in complex terrain". In: *The Annals of Applied Statistics* 13.3 (2019), pp. 1564–1589. DOI: https://doi.org/10.1214/19-AOAS1247.

[145] Niklas Donges. *A complete Guide to the Random Forest algorithm*. July 2021. URL: https://builtin.com/data-science/random-forest-algorithm (visited on 10/20/2021).

[146] Abbas Abdullah Baiz et al. "A Gaussian process regression model to predict energy contents of corn for poultry". In: *Poultry Science* (2020). vol. 99, no. 11, pp. 5838–5843. ISSN: 0032-5791. DOI: https://doi.org/10.1016/j.psj.2020.07.044.

[147] Sheng Hong and Zheng Zhou. "Application of Gaussian Process Regression for bearing degradation assessment". In: *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)*. Jan. 2012, pp. 644–648. ISBN: 978-1-4673-0876-2.

[148] Sankalita Saha et al. "Distributed prognostic health management with gaussian process regression". In: *2010 IEEE Aerospace Conference*. 2010, pp. 1–8. DOI: https://doi.org/10.1109/AERO.2010.5446841.

[149] Monika Tanwar and Nagarajan Raghavan. "Lubricating Oil Remaining Useful Life Prediction Using Multi-Output Gaussian Process Regression". In: *IEEE Access* 8 (2020), pp. 128897–128907. DOI: https://doi.org/10.1109/ACCESS.2020.3008328.

[150] Man Shan Kan, Andy C.C. Tan, and Joseph Mathew. "A review on prognostic techniques for non-stationary and non-linear rotating systems". In: *Mechanical Systems and Signal Processing* 62-63 (2015), pp. 1–20. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2015.02.016.

[151] V.N. Vapnik. "An overview of statistical learning theory". In: *IEEE Transactions on Neural Networks* 10.5 (1999), pp. 988–999. DOI: https://doi.org/10.1109/72.788640.

[152] Christopher M. Bishop and Michael E. Tipping. "Variational Relevance Vector Machines". In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. UAI '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 46–53. ISBN: 1558607099.

[153] Xu Xiang-min et al. "Classification Performance Comparison between RVM and SVM". In: *2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID)*. 2007, pp. 208–211. DOI: https://doi.org/10.1109/IWASID.2007.373728.

[154] J. A. Carino et al. "Remaining useful life estimation of ball bearings by means of monotonic score calibration". In: *2015 IEEE International Conference on Industrial Technology (ICIT)*. 2015, pp. 1752–1758. DOI: https://doi.org/10.1109/ICIT.2015.7125351.

[155] James Kuria Kimotho et al. "Machinery Prognostic Method Based on Multi-Class Support Vector Machines and Hybrid Differential Evolution – Particle Swarm Optimization". In: *Chemical engineering transactions* 33 (2013), pp. 619–624. DOI: https://doi.org/10.3303/CET1333104.

[156] F. Sloukia et al. "Bearings prognostic using Mixture of Gaussians Hidden Markov Model and Support Vector Machine". In: *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*. 2013, pp. 1–4. DOI: https://doi.org/10.1109/AICCSA.2013.6616438.

[157] Shaojiang Dong and Tianhong Luo. "Bearing degradation process prediction based on the PCA and optimized LS-SVM model". In: *Measurement* (2013). vol. 46, no. 9, pp. 3143–3152. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2013.06.038.

[158] T. Benkedjouh et al. “Health assessment and life prediction of cutting tools based on support vector regression”. In: *Journal of Intelligent Manufacturing* (Apr. 2015). vol. 26, no. 2, pp. 213–223. ISSN: 1572-8145. DOI: https://doi.org/10.1007/s10845-013-0774-6.

[159] Jie Liu and Enrico Zio. “An adaptive online learning approach for Support Vector Regression: Online-SVR-FID”. In: *Mechanical Systems and Signal Processing* 76-77 (2016), pp. 796–809. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2016.02.056.

[160] Shuai Wang et al. “Prognostics of Lithium-Ion Batteries Based on Battery Performance Analysis and Flexible Support Vector Regression”. In: *Energies* (Oct. 2014). vol. 7, pp. 6492–6508. DOI: https://doi.org/10.3390/en7106492.

[161] Yuchen Song et al. “Satellite lithium-ion battery remaining useful life estimation with an iterative updated RVM fused with the KF algorithm”. In: *Chinese Journal of Aeronautics* (2018). vol. 31, no. 1, pp. 31–40. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2017.11.010.

[162] Datong Liu et al. “Lithium-ion battery remaining useful life estimation with an optimized Relevance Vector Machine algorithm with incremental learning”. In: *Measurement* (2015). vol. 63, pp. 143–151. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2014.11.031.

[163] Ferhat Tamssaouet et al. “Combination of Long Short-Term Memory and Particle Filtering for Future Uncertainty Characterization in Failure Prognostic”. In: *Proceedings of the 31st European Safety and Reliability Conference* (). DOI: https://doi.org/10.3850/978-981-18-2016-8152-cd.

[164] Faisal Khan et al. “Adaptive Degradation Prognostic Reasoning by Particle Filter with a Neural Network Degradation Model for Turbofan Jet Engine”. In: *Data* (2018). vol. 3, no. 4. ISSN: 2306-5729. DOI: https://doi.org/10.3390/data3040049.

[165] Shixin Ji et al. “Remaining Useful Life Prediction of Airplane Engine Based on PCA–BLSTM”. In: *Sensors* (2020). vol. 20, no. 16. ISSN: 1424-8220. DOI: https://doi.org/10.3390/s20164537.

[166] Abdenour Soualhi et al. “Prognosis of Bearing Failures Using Hidden Markov Models and the Adaptive Neuro-Fuzzy Inference System”. In: *IEEE Transactions on Industrial Electronics* (2014). vol. 61, no. 6, pp. 2864–2874. DOI: https://doi.org/10.1109/TIE.2013.2274415.

[167] Zhengrui Tao et al. “A novel method for tool condition monitoring based on long short-term memory and hidden Markov model hybrid framework in high-speed milling Ti-6Al-4V”. In: *The International Journal of Advanced Manufacturing Technology* (Dec. 2019). vol. 105, no. 7, pp. 3165–3182. ISSN: 1433-3015. DOI: http://doi.org/10.1007/s00170-019-04464-w.

[168] Clifford Matthews. “12 - Mechanical seals – improving design reliability”. In: *Case Studies in Engineering Design*. Ed. by Clifford Matthews. London: Butterworth-Heinemann, 1998, pp. 113–126. ISBN: 978-0-340-69135-9. DOI: https://doi.org/10.1016/B978-034069135-9/50015-1.

[169] M. L. Baptista. *Lecture 1 - Maintenance in the aerospace context + RAMS (2021).* 2021. URL: https://brightspace.tudelft.nl/d2l/le/content/278324/viewContent/2067970/View (visited on 10/25/2021).

[170] Srikesh Arunajadai et al. "A Framework for Creating a Function-Based Design Tool for Failure Mode Identification". In: *14th International Conference on Design Theory and Methodology, Integrated Systems Design, and Engineering Design and Culture* 4 (Feb. 2002), pp. 195–206. DOI: http://doi.org/10.1115/DETC2002/DTM-34018.

[171] Sébastien Schwartz et al. "A fault mode identification methodology based on self-organizing map". In: *Neural Computing and Applications* (Sept. 2020). vol. 32, no. 17, pp. 13405–13423. ISSN: 1433-3058. DOI: https://doi.org/10.1007/s00521-019-04692-x.

[172] NASA Ames Research Center. *NASA Ames Prognostics Data Repository.* URL: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/ (visited on 10/26/2021).

[173] William S. Cleveland. "Robust Locally Weighted Regression and Smoothing Scatterplots". In: *Journal of the American Statistical Association* 74.368 (1979), pp. 829–836. DOI: https://doi.org/10.1080/01621459.1979.10481038.

[174] Michal Smolik, Václav Skala, and Ondrej Nedved. "A Comparative Study of LOWESS and RBF Approximations for Visualization". In: *Computational Science and Its Applications - ICCSA 2016 - 16th International Conference, Beijing, China, July 4-7, 2016, Proceedings, Part II.* Vol. 9787. Lecture Notes in Computer Science. Springer, 2016, pp. 405–419. DOI: https://doi.org/10.1007/978-3-319-42108-7_31.

[175] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference.* 2010.

[176] A.M.Y. Razak. *Industrial gas turbines: performance and operability.* Elsevier, 2007.

Part III
Appendices

Appendix A LOWESS detailed calculation procedure

This appendix contains a detailed calculation procedure of the LOWESS method used for noise removal, data smoothing and data downsampling.

The calculation procedure to perform noise removal, data smoothing and data downsampling with LOWESS can be explained in the following way.
Let $i = 1, ..., n$ and $j = 1, ..., k$. The goal of LOWESS is to obtain a smoothed $\hat{y}_i$-value per x_i-value which is mathematically formulated by Cleveland [173] as shown in Equation A.16:

$$\hat{y}_i = \sum_{l=0}^{d} \hat{\beta}_l(x_i) x_i^l \tag{A.16}$$

Where d denotes the degree of the polynomial function, $\hat{\beta}_l(x_i)$ represents the unknown coefficients of the dth degree polynomial function and x_i is the x-value for which a smoothed y-value, $\hat{y}_i$, must be obtained.

For each i, calculate estimates of these unknown coefficients $\hat{\beta}_l(x_i)$ of the dth degree polynomial regression function for y_j on x_j [173]. To be precise, these unknown coefficients $\hat{\beta}_l(x_i)$ are obtained through minimizing a sum S (Equation A.17) which is given by Smolik et al. [174] as:

$$S = \sum_{j=1}^{k} \omega_j(x_i) \cdot (y_j - P_{(d)}(x_j))^2 \tag{A.17}$$

In which $P_{(d)}(x_j) = \hat{\beta}_0(x_i) + \hat{\beta}_1(x_i)x_j + ... + \hat{\beta}_d(x_i)x_j^d$ is the dth degree polynomial function with the unknown coefficients $\hat{\boldsymbol{\beta}} = [\hat{\beta}_0(x_i), \hat{\beta}_1(x_i), ..., \hat{\beta}_d(x_i)]^T$. Furthermore, y_j denotes the original (unsmoothed) y-value at point (x_j, y_j) and $w_j(x_i)$ denotes the weight [173]. Equation A.17 can be rewritten in matrix form as given in Equation A.18:

$$S = (\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{\beta}})^T \cdot \boldsymbol{W} \cdot (\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{\beta}}) \tag{A.18}$$

With $\boldsymbol{b} = [y_1, y_2, ..., y_k]^T$ being a vector containing the actual y-values [174], matrix $\boldsymbol{A}$ being a matrix consisting of the actual x-values which is defined as:

$$\boldsymbol{A} = \begin{bmatrix} 1 & x_1 & & x_1^d \\ 1 & x_2 & \cdots & x_2^d \\ & \vdots & \ddots & \vdots \\ 1 & x_k & \cdots & x_k^d \end{bmatrix}$$

and matrix $\boldsymbol{W}$ being a diagonal matrix containing the weights which is defined as:

$$\boldsymbol{W} = \begin{bmatrix} w_1(x_i) & & & 0 \\ & w_2(x_i) & & \\ & & \ddots & \\ 0 & & & w_k(x_i) \end{bmatrix}$$

The weights, $w_j(x_i)$, are scaled from zero to one (see Equation A.19) by, for each x_j, dividing the distance between x_j and x_i by the maximum observed absolute distance between x_i and the furthest x_j (where the furthest x_j is the most distance k-nearest neighbor with respect to x_i). Therefore, the closest point to x_i gets a scaled value (r) of zero, whilst the furthest point to x_i gets a scaled value (r) of one.

$$w_j(x_i) = W\left(\left\|\frac{x_j - x_i}{max(|x_i - x_j|)}\right\|\right) \tag{A.19}$$

After scaling the distances, the actual weights are calculated using a tricube weighting function $W(r)$ which is defined by Cleveland [173] and expressed in Equation A.20 as:

$$W(r) = \begin{cases} (1 - r^3)^3 & r \in \langle 0; 1\rangle \\ 0 & r > 1 \end{cases} \tag{A.20}$$

In order to minimize Equation A.18, it must first be expanded which leads to the expression as given in Equation A.21 [174]:

$$\begin{aligned} S &= \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} - (\boldsymbol{A}\hat{\boldsymbol{\beta}})^T\boldsymbol{W}\boldsymbol{b} + (\boldsymbol{A}\hat{\boldsymbol{\beta}})^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^T\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{b} + \hat{\boldsymbol{\beta}}^T\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} \end{aligned} \tag{A.21}$$

Equation A.21 can be minimized by taking the partial derivative of S with respect to $\hat{\boldsymbol{\beta}}$ and equating the outcome equal to zero. This leads to the result as given in Equation A.22 since $\boldsymbol{W} = \boldsymbol{W}^T$ [174]:

$$\frac{\partial S}{\partial \hat{\boldsymbol{\beta}}} = -\left(\boldsymbol{b}^T\boldsymbol{W}\boldsymbol{A}\right)^T - \boldsymbol{A}^T\boldsymbol{W}\boldsymbol{b} + 2\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} = \boldsymbol{0} \tag{A.22}$$

Hence, the coefficients $\hat{\beta}_l(x_i)$ can be calculated by expanding Equation A.22 and solving for $\hat{\boldsymbol{\beta}}$ as shown in Equation A.23 [174]:

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{\beta}} &= \boldsymbol{A}^T\boldsymbol{W}\boldsymbol{b} \\ \hat{\boldsymbol{\beta}} &= \left(\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{b} \end{aligned} \tag{A.23}$$

Note that these equations result in explicit coefficients per x_i. However, these coefficients might change since as an additional weight δ_j is assigned per (x_j, y_j), based on the residual $y_j - \hat{y}_i$, to account for outliers.

Next, let $\boldsymbol{e_i} = [y_j - \hat{y}_i]$ denote a vector containing the residuals between the 'original' y-values (of the k-nearest neighbor points) compared to the currently fitted $\hat{y}_i$-value.

Furthermore, let q denote the median of $|\boldsymbol{e_i}|$. Then, the additional weight is defined by Cleveland [173] as expressed in Equation A.24:

$$\delta_j = C(\frac{e_{i_j}}{6q}) \tag{A.24}$$

With e_{i_j} denoting the jth element of the $\boldsymbol{e_i}$ vector and $C(r)$ being the bisquare weighting function (Equation A.25) defined by Cleveland [173] as:

$$C(r) = \begin{cases} (1-r^2)^2 & r \in \langle 0;1 \rangle \\ 0 & r > 1 \end{cases} \tag{A.25}$$

Finally, the new smoothed y-values, $\hat{y}_i$, are calculated for each i in the same way as described previously. However, the weights for the weighted least squares are replaced by $\delta_j w_j(x_i)$ for (x_j, y_j). More information regarding the full details of the LOWESS method can be found in the paper by Cleveland [173].

The LOWESS method as described above has been implemented using the Python package called *statsmodels.nonparametric.smoothers_lowess.lowess* [175]. This package uses a first degree polynomial function ($d = 1$), which is in accordance with the findings obtained in [173]. It has been found that using a first degree polynomial yields adequately smoothed data whilst maintaining computational ease [173]. Furthermore, the package's two most important parameters, except from the x-values (flight cycles) and y-values (sensor values), are the number of iterations it and the window $frac$ (which determines the number of k-nearest neighbor points that will be considered when smoothing the data). The number of iterations used to reweigh weights and recompute smoothed y-values has been left at the default value of $it = 3$, as it provides satisfactory results. The value of the window $frac$ (with $0 < frac \leq 1$) has been left at the default value of $frac = 2/3$, as it yields satisfying smoothed y-values. Choosing this relatively high $frac$ default value is desired, as it is required to select a value as large as possible without altering the underlying data pattern in order to obtain the finest smoothing [173]. Consequently, these parameter settings have been used when applying the LOWESS method for all 28 sensors listed in Table 4 and Table 5 of the paper.

Appendix B Feature selection results (obtained using LOWESS for data smoothing)

This appendix contains the results of feature selection using RFECV for data that has been smoothed with LOWESS. It comprises the selected features (with corresponding feature importances) of the flight class RFR models in DS01, DS02 and DS03. Additionally, the accompanying CV scores are provided as well.

Table B2 shows the selected features with their accompanying feature importances of the flight class RFR models for DS01 and DS02. Note that for DS02, only flight class RFR model 3 is listed in this table since the development set of DS02 only consists of units from flight class 3. The selected features and corresponding feature importances of flight class RFR models for DS03 are shown in Table B3.

When considering the selected features for flight class RFR models in DS01, it can be explained why the selected features in DS01 are also relevant from a physical perspective. Given that all units from DS01 are suffering from HPT efficiency deterioration, it is likely that it could result into the HPT being incapable of efficiently transforming the hot gas into energy, thus generating less power [176]. As a consequence, Wf is increased in order to still generate the same amount of power, thus causing the temperatures $T50$ and $T48$ to rise and phi to increase [176]. As a result, the entire unit will rotate faster (Nf will increase) causing the LPC to compress more air, hence produce a higher $Ps30$. A side effect of compressing extra air is an increased $T24$ [176]. Furthermore, a faster rotating unit results into decreased stall margins $SmLPC$ and $SmHPC$ [176]. Note that this explanation can only be provided for units in DS01 as those units are experiencing the effect of only a single failure mode. For units in DS02 and DS03, this can not be explained as they are affected by multiple simultaneous failure modes. These multiple simultaneous failure modes make it too complex to explain why the selected features are relevant from a physical perspective.

Furthermore, apart from the tables showing the selected features for the data subsets, Table B4 is showing the CV scores obtained for the selected features per flight class RFR model for DS01, DS02 and DS03. These CV scores are considered as the baseline errors that need to be reduced through hyperparameter tuning (from which the results are shown in Appendix C). When comparing the CV scores in Table B4 to the CV scores obtained for data that has been smoothed using the FC average smoothing method, it becomes evident that the scores in Table B4 are considerably lower. This is expected to be caused due to the better smoothing results of LOWESS.

Table B2: Selected features with corresponding feature importances of DS01 (for flight class RFR models 1,2 and 3) and DS02 (for flight class RFR model 3).

Data subset	Flight class RFR model	Selected features	Feature importance [-]
DS01	1	T50	0.2136
		T48	0.2098
		Ps30	0.2012
		Phi	0.1899
		T24	0.1855
DS01	2	SmLPC	0.6312
		Nf	0.3688
DS01	3	T50	0.7791
		SmHPC	0.1256
		Wf	0.0953
DS02	3	SmLPC	0.3466
		phi	0.2979
		SmHPC	0.1408
		T50	0.0900
		T48	0.0633
		SmFan	0.0194
		T40	0.0100
		Nf	0.0082
		Wf	0.0050
		Nc	0.0049
		T30	0.0029
		P45	0.0023
		W50	0.0023
		W25	0.0019
		W31	0.0018
		Ps30	0.0014
		P2	0.0011

Table B3: Selected features with corresponding feature importances of DS03 for flight class RFR models 1,2 and 3.

Data subset	Flight class RFR model	Selected features	Feature importance [-]
DS03	1	SmLPC	0.7327
		SmHPC	0.1414
		T24	0.1259
DS03	2	T48	0.3695
		T40	0.3197
		SmLPC	0.1098
		Nf	0.0503
		SmHPC	0.0424
		P2	0.0384
		phi	0.0309
		SmFan	0.0197
		P15	0.0079
		T24	0.0045
		W21	0.0043
		P50	0.0026
DS03	3	SmHPC	0.3386
		phi	0.3361
		Nc	0.1465
		SmLPC	0.0880
		T48	0.0213
		SmFan	0.0174
		T50	0.1380
		Nf	0.0097
		T40	0.0088
		T30	0.0069
		W21	0.0033
		T24	0.0026
		W25	0.0025
		W22	0.0025
		Ps30	0.0020

Table B4: Baseline CV scores from RFECV of DS01, DS02 and DS03 for flight class RFR models 1,2 and 3.

Data subset	**Flight class RFR model**	**Baseline CV score (RMSE) [FC]**
DS01	1	12.2754
	2	11.2924
	3	10.2639
DS02	3	10.5136
DS03	1	13.1764
	2	11.1559
	3	11.0989

Appendix C Hyperparameter tuning results (obtained using LOWESS for data smoothing)

This appendix contains the results of the hyperparameter tuning for all flight class RFR models in DS01, DS02 and DS03 for which the data has been smoothed with LOWESS. To be precise, it shows which specific RFR hyperparameters are tuned as well as their tuned values in Table C5. Additionally, the corresponding new (final) CV scores obtained on the development sets of DS01, DS02 and DS03 (with the optimal hyperparameter values) for all flight class RFR models are listed in Table C6. It can be seen that the final CV scores improved for all flight class RFR models in all data subsets compared to Table B4. However, the CV scores only slightly improved, thereby indicating that hyperparameter tuning does not have a substantial effect on these specific data subsets.

Table C5: RFR hyperparameters for flight class RFR models 1, 2 and 3 of DS01, DS02 and DS03. The (initial) ranges of the search grid are the following: Number of trees [100, 300, 500, 800, 1000, 1200], Split criterion [MSE, Poisson], Maximum tree depth [4, 6, 8, 10, 15, 25, 40, 100, None], Minimum samples per split [2, 4, 8, 10], Minimum samples per leaf node [1, 3, 5, 7, 9] and Fraction of features per split [0.25, 0.50, 0.75, Auto].

Data subset	**Flight class RFR model**	**Number of trees**	**Split criterion**	**Max. tree depth**	**Min. samples per split**	**Min. samples per leaf node**	**Fraction of features per split**
DS01	1	110	MSE	13	2	1	0.4
	2	120	MSE	8	2	1	0.5
	3	310	MSE	98	2	1	Auto
DS02	3	300	MSE	98	2	1	Auto
DS03	1	80	MSE	100	2	1	Auto
	2	90	MSE	100	2	1	Auto
	3	1020	MSE	8	2	1	Auto

Table C6: Final CV scores after hyperparameter tuning of DS01, DS02 and DS03 for flight class RFR models 1,2 and 3.

Data subset	**Flight class RFR model**	**Final CV score (RMSE) [FC]**
DS01	1	12.2432
	2	11.2861
	3	10.2110
DS02	3	10.4192
DS03	1	13.1562
	2	11.1465
	3	11.0170

Appendix D RUL point estimations and corresponding uncertainty (obtained using LOWESS for data smoothing)

This appendix discusses the RUL point estimations and corresponding uncertainty of these estimations, displayed via probability distributions, of all test units in DS01 (units 7, 8, 9 and 10), DS02 (unit 11) and DS03 (units 10, 11, 12, 13, 14 and 15).

Figure D.1 and Figure D.2 depict the RUL point estimations and associated probability distributions (of the first, middle and last RUL point estimations) of test units 7 and 9 (both flight class RFR model 1), respectively. Figure D.3 and Figure D.4 depict equivalent information for test units 12 and 14 (both flight class RFR model 1 as well), respectively. In Figure D.5 and Figure D.6, the RUL point estimations and associated probability distributions are shown for test units 8 and 15 (both flight class RFR model 2), respectively. Finally, Figure D.7, Figure D.8, Figure D.9, Figure D.10 and Figure D.11 show the same for flight class RFR model 3 test units 10 (DS01), 11 (DS02), 10 (DS03), 11 (DS03) and test unit 13, respectively. When analyzing all figures, it becomes apparent that the results (RUL point estimations as well as the accompanying probability distributions) are considerably less satisfactory compared to those of test units for which their data have been smoothed with the FC average data smoothing method. These considerably less satisfactory results are believed to be a result of unpreventable smoothing issues with LOWESS. As explained, a scarce amount of test data is available for smoothing with LOWESS when generating the RUL point estimations, especially for RUL point estimations during a unit's initial life.

Due to this scarce amount of available smoothing data, the eventual smoothed data is expected to be slightly different from the smoothed (training) data on which the RFR flight class models have been trained. Because the training data on which these RFR flight class models were trained, comprised of a unit's entire lifetime. Consequently, a larger fraction of data has been available for LOWESS when smoothing the data in the development set, possibly leading to slightly different smoothed values. Unfortunately in reality, having a test unit's entire lifetime of data available during its early life is impossible (otherwise one would have already known when it failed, thereby eliminating the necessity for RUL estimation models in the first place). As a result, it is likely that this slightly different smoothed data during a unit's early life causes the RFR flight class models to generate considerably different RUL estimations due to the way in which thresholds were created by the RFR when constructing the forests.

Additionally, it can be observed that the RUL point estimations for most test units do not converge towards the true RUL values, as the units approach their end of life. A possible explanation for this behaviour is concerned with the deterioration patterns of the test units. Because in case the deterioration patterns differ slightly from the patterns learned during training of the flight class RFR models, their smoothed data will be

different. Hence, it can yield wildly different RUL point estimations, as explained previously. Only the RUL point estimations of test units 12 (Figure D.3), 14 (Figure D.4) and 11 (Figure D.8) converge towards the true RUL values as they approach their end of life. This is expected to happen due to the test units (eventually) following the degradation patterns of their training units fairly well. Furthermore, it can be seen that test unit 12 (Figure D.3) has the highest accuracy in terms of RUL point estimations (that is, the estimations are close to the corresponding true RUL values, especially after 40 FC).

Overall, it can be concluded that the test units of DS03 generally yield the best results is terms of RUL point estimations as well as probability distributions. This is believed to be a direct effect of having more training units available for the test units in DS03 compared to the test units in DS01, even though DS03 has a higher complexity than DS01. Finally, when considering the probability distributions of all test units, it is found that the probabilities of correctly estimating the true RUL values are generally found to be low. However, it can be also observed that for five out of the 11 test units, the uncertainty of the estimations decreases as the units approach their end of life, because the spread of the probability distributions decreases. Also, inspecting the accuracy of the corresponding RUL point estimations shows that combining the individual tree RUL estimations into a single RUL point estimation is generally found to increase RUL point estimation accuracy.

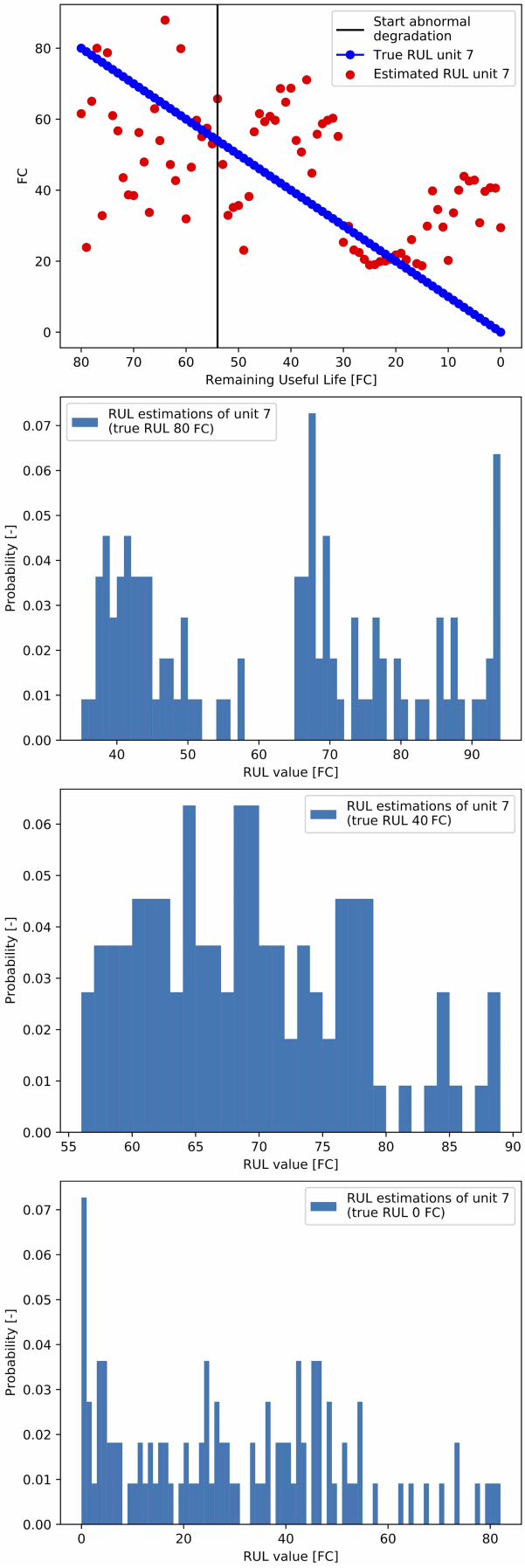


Figure D.1: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 7, DS01.

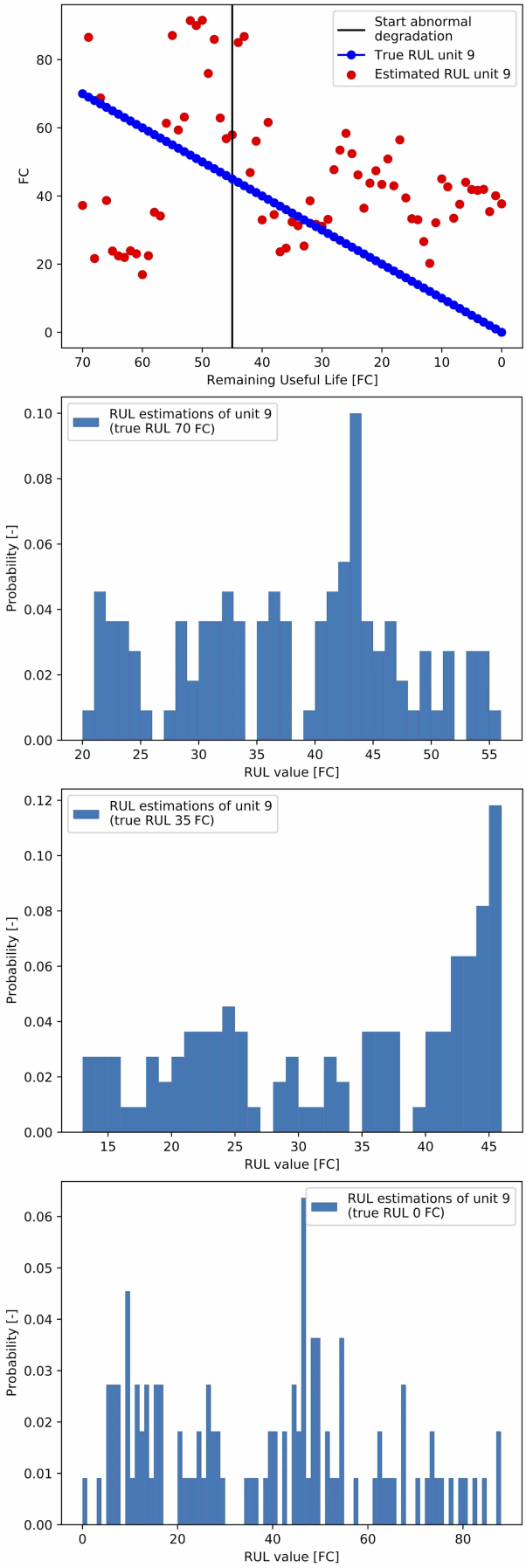


Figure D.2: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 9, DS01.

L

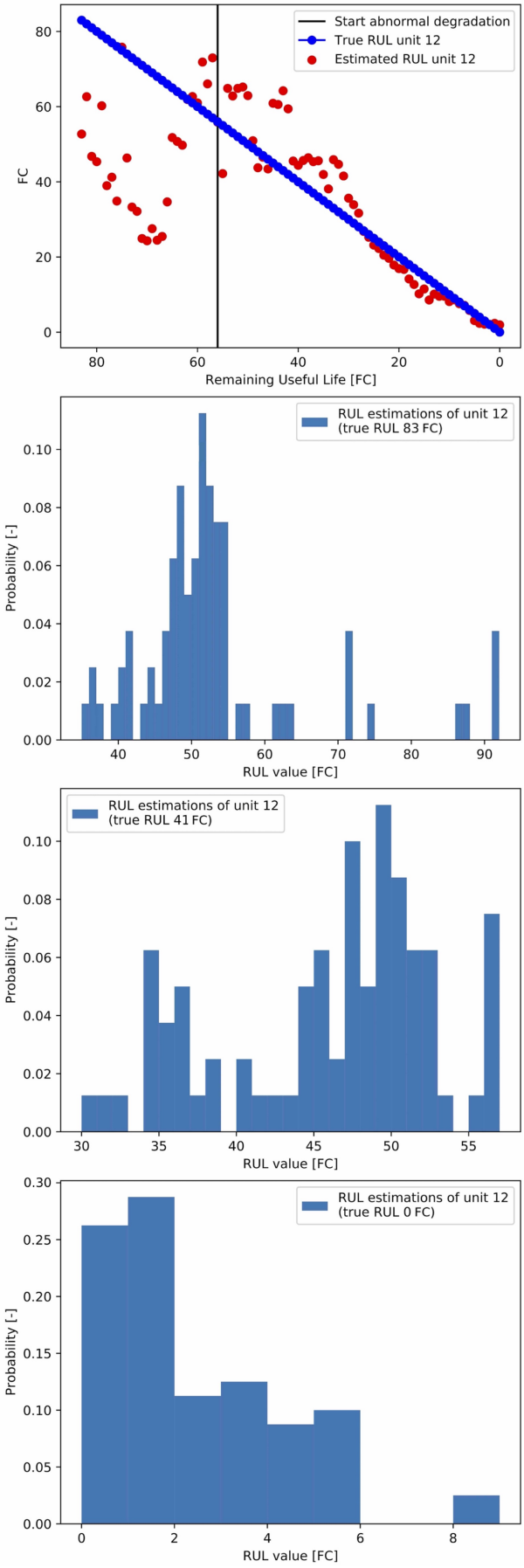


Figure D.3: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 12, DS03.

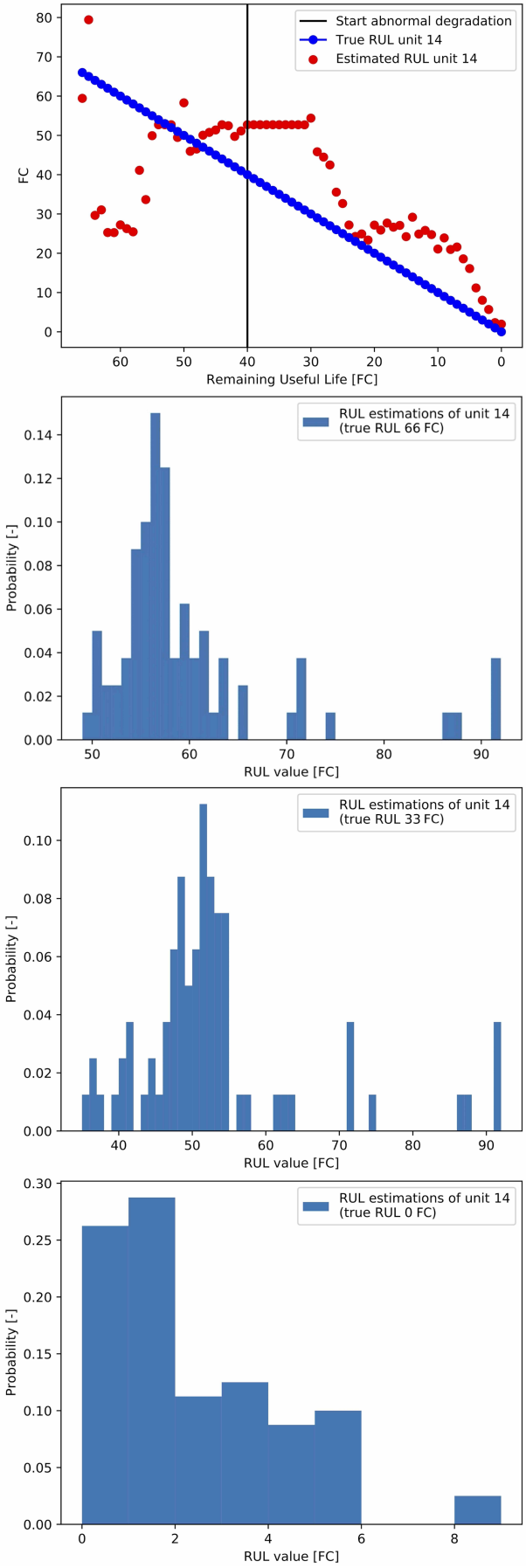


Figure D.4: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 14, DS03.

M

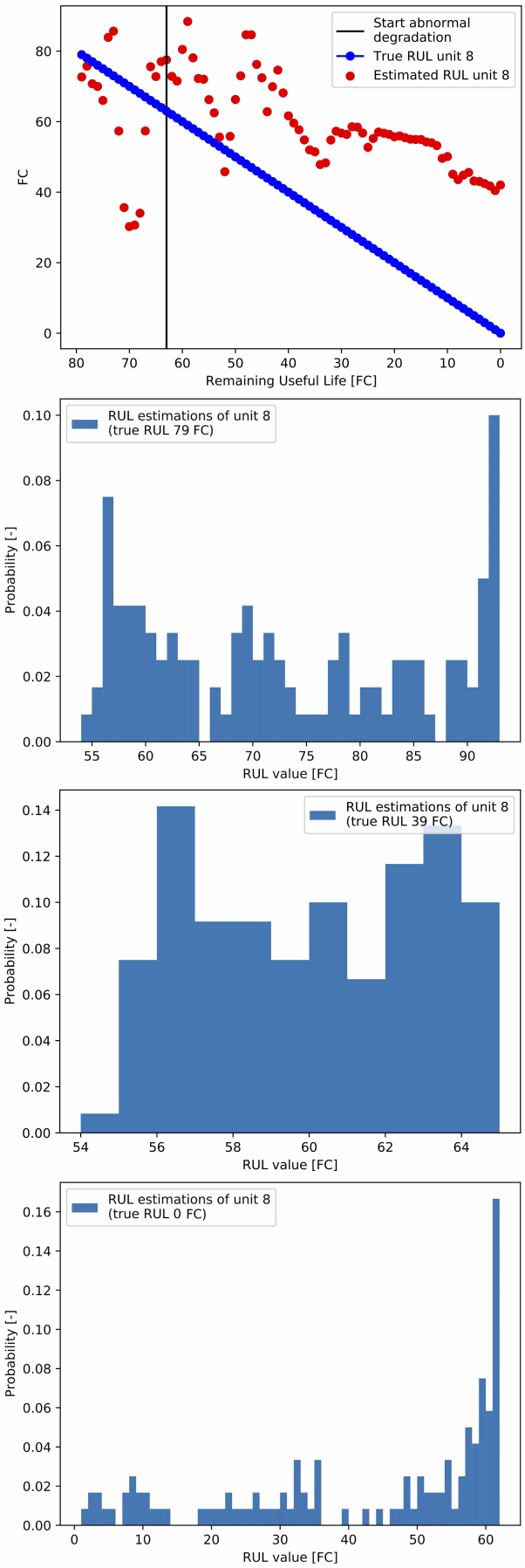


Figure D.5: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 8, DS01.

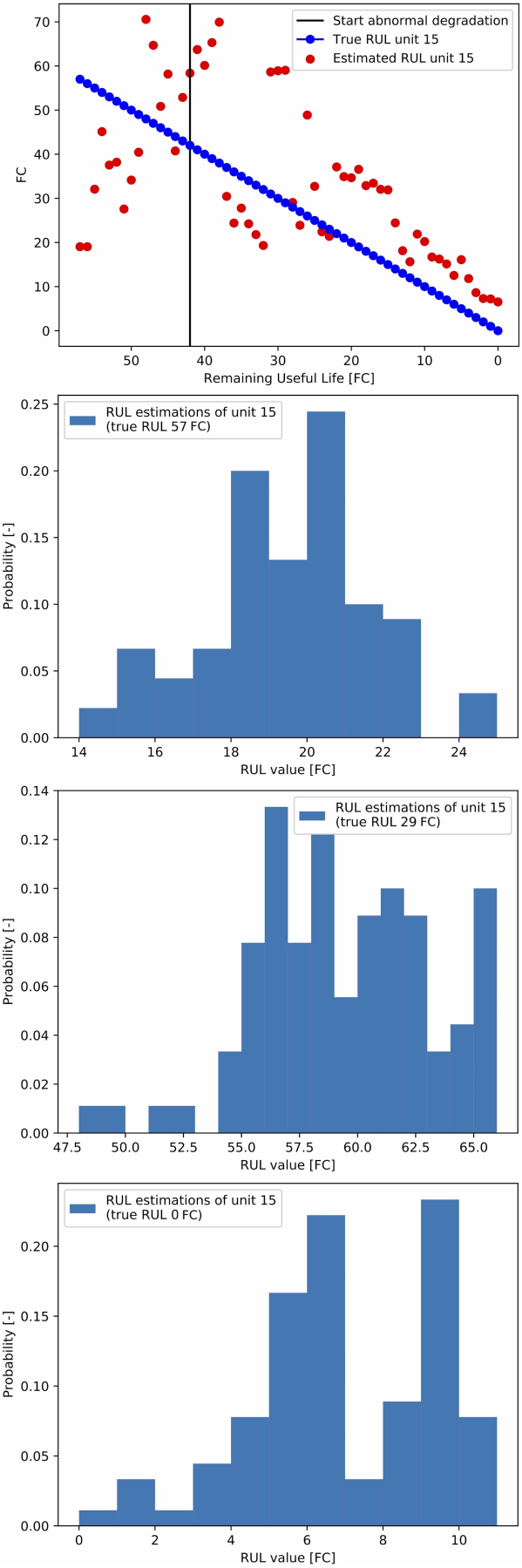


Figure D.6: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 15, DS03.

N

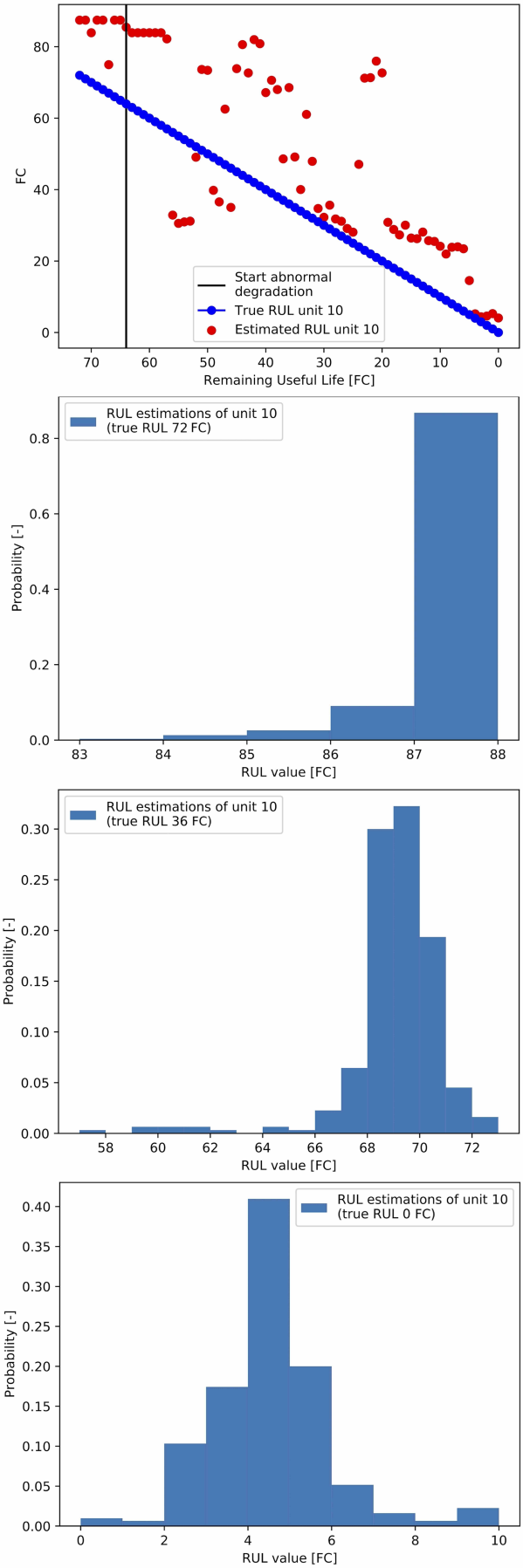


Figure D.7: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 10, DS01.

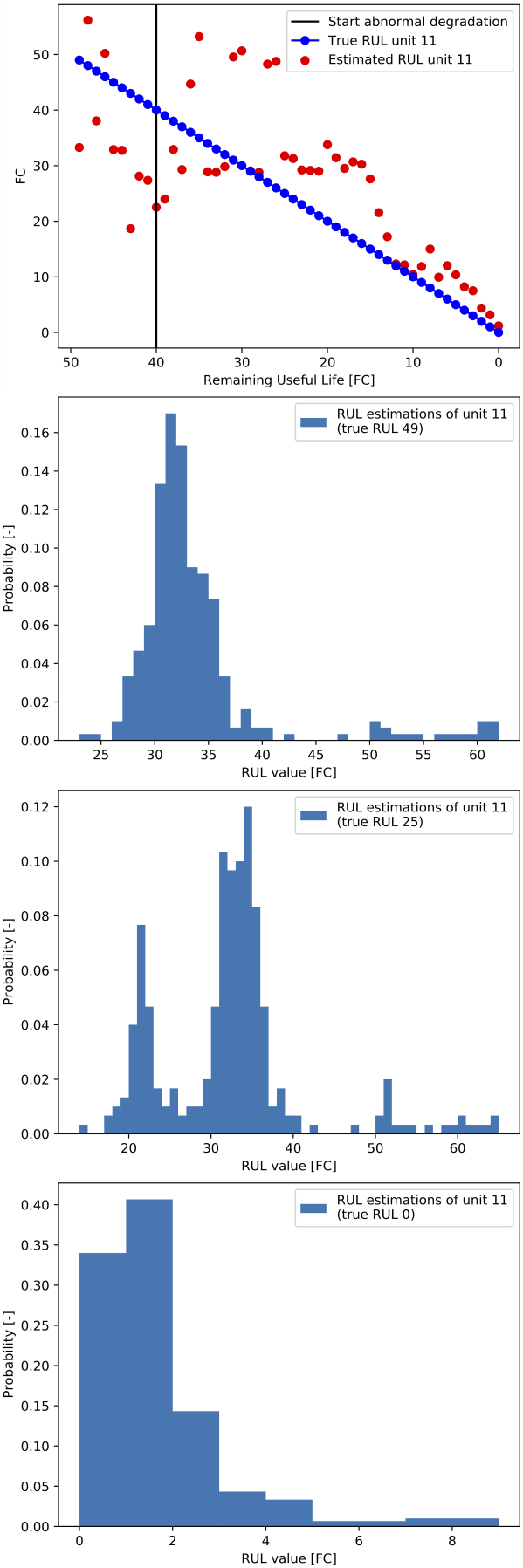
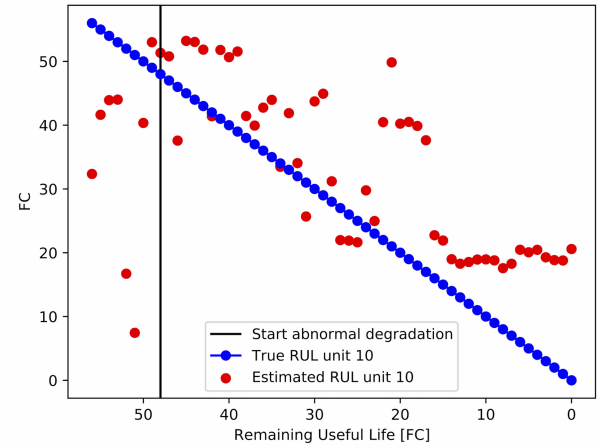
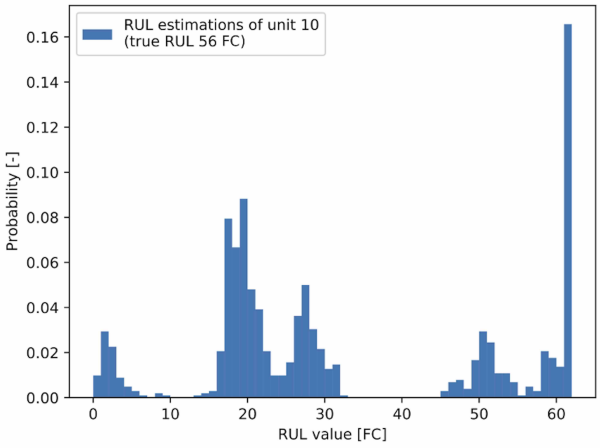
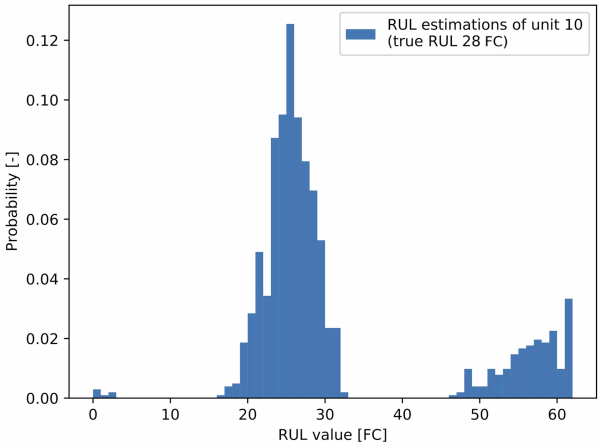


Figure D.8: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 11, DS02.

O







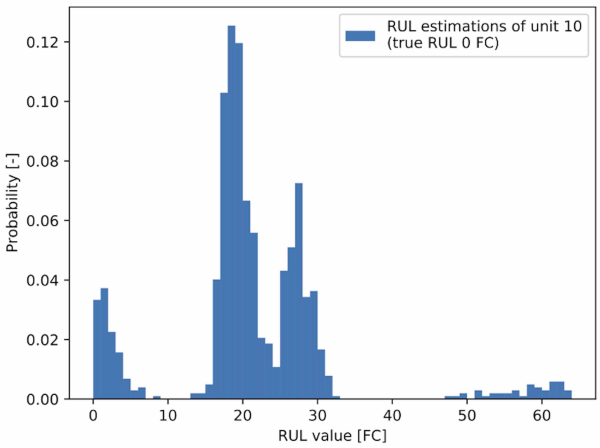
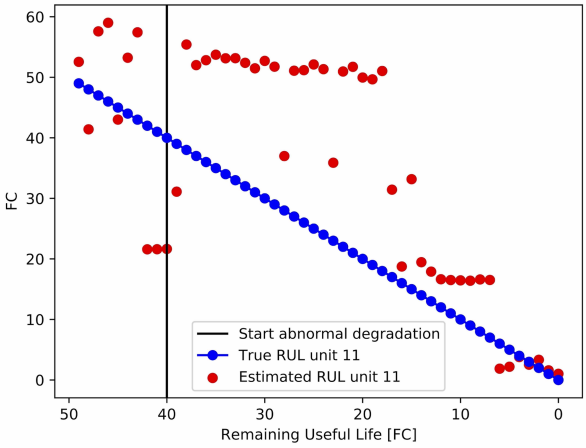
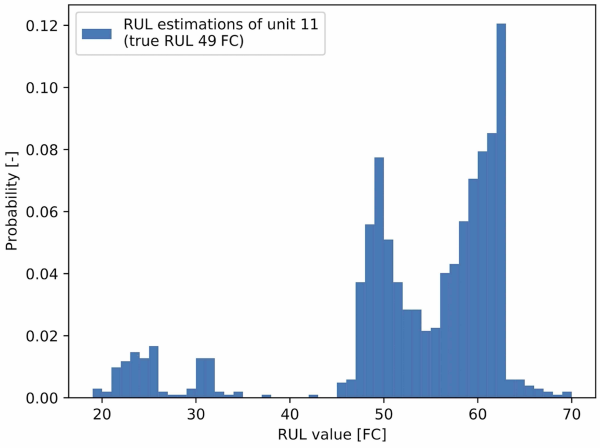
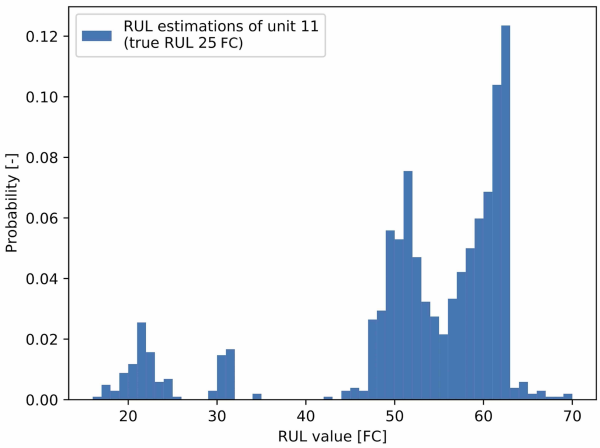


Figure D.9: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 10, DS03.







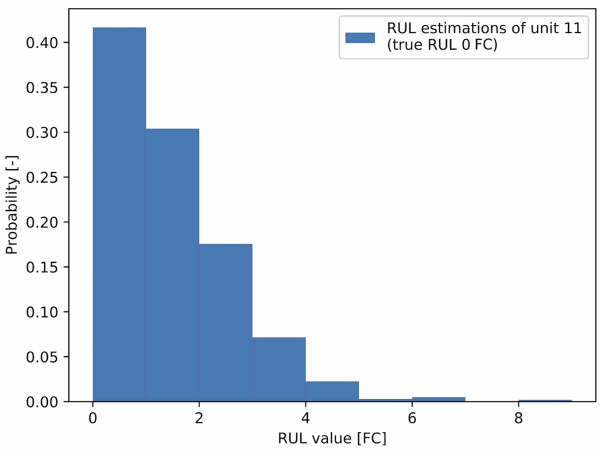


Figure D.10: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 11, DS03.

P

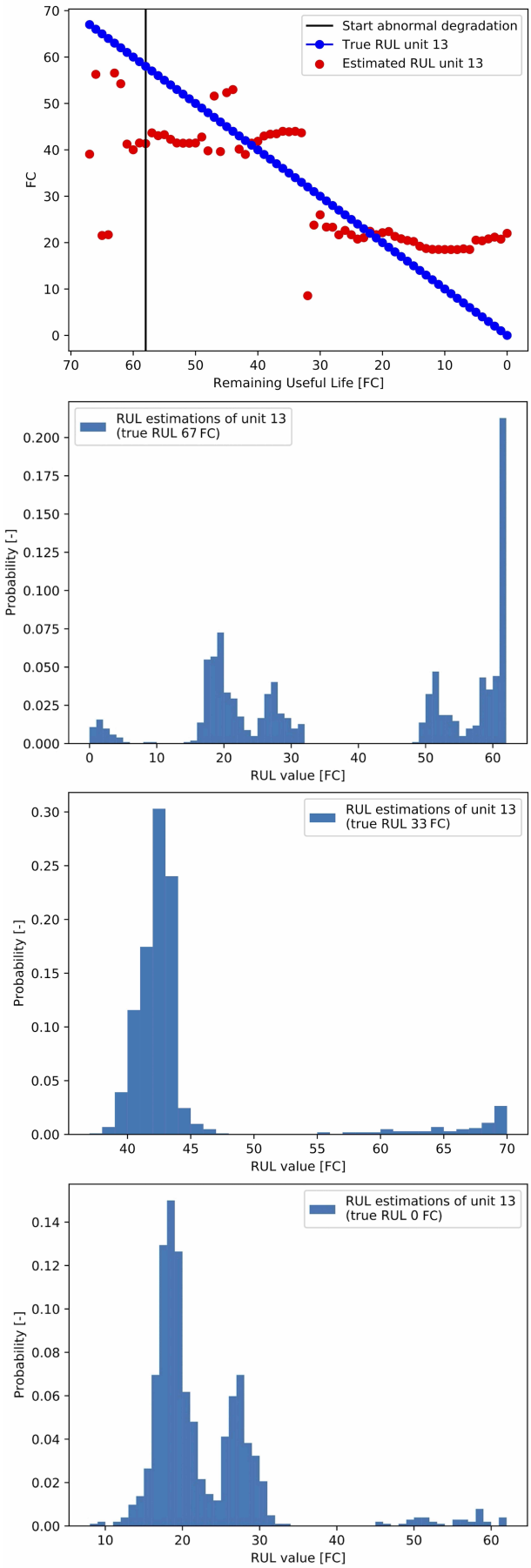


Figure D.11: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 13, DS03.

Q

Appendix E RUL point estimations and corresponding uncertainty (obtained using FC average for data smoothing)

This appendix discusses the RUL point estimations and probability distributions (which show the corresponding uncertainty of these estimations) of the remaining test units in DS01 (unit 9), DS02 (unit 11) and DS03 (units 10, 12 and 13).

Figure E.1 and Figure E.2 depict the RUL point estimations and corresponding probability distributions (of the first, middle and last RUL point estimations) of test unit 9 and test unit 12 (both flight class RFR model 1), respectively. The RUL point estimations and associated probability distributions of test units 10 and 13 (flight class RFR model 3) are depicted in Figure E.3 and Figure E.4, respectively. Figure E.5 presents this same information for test unit 11 (flight class RFR model 3). Approximately the same type of results as discussed in paragraph 4.1 of the paper can be observed in these graphs. First of all, it can be seen that the results in terms of RUL point estimations of test units 9 and 12 (Figure E.1 and Figure E.2, respectively) appear to be the closest to the true RUL. Furthermore, for these test units as well as test unit 11 (Figure E.5), a converging trend towards zero RUL can be identified. This is expected as it generally becomes easier to produce estimations as the estimation window decreases over time. However, this converging trend is not identified for test units 10 and 13 (Figure E.3 and Figure E.4, respectively). As a matter of fact, the estimated RUL values of test unit 10 (and test unit 13 to a lesser extend) seem to be parallel to the true RUL values, except for some individual estimations. The expected reason for this parallel behaviour is that the training units for these test units (both test units were trained on the exact same training units) followed a less steep deterioration pattern, compared to these two test units. The expected reason for the individual estimations of test units 10 and 13 which do not follow the overall parallel trends (especially visible during the last 20 FC), is that the flights of these estimations operated slightly different (in terms of throttle setting) compared to the overall flights. This possibly slightly different throttle setting could result into slightly different smoothed data values, which in turn can lead to largely different RUL point estimations due to the way thresholds were created by RFR when constructing the trees.

Finally, when considering the probability distributions of all test units, it can be observed that for the majority of the test units, the uncertainty of the estimations decreases over time because the width of the probability distributions decreases (especially for the distributions corresponding to a true RUL value of zero FC). This is very useful as it is especially important to have a low uncertainty for the RUL estimations corresponding to a unit's end of life, because a high uncertainty of those estimations would endanger flight safety.

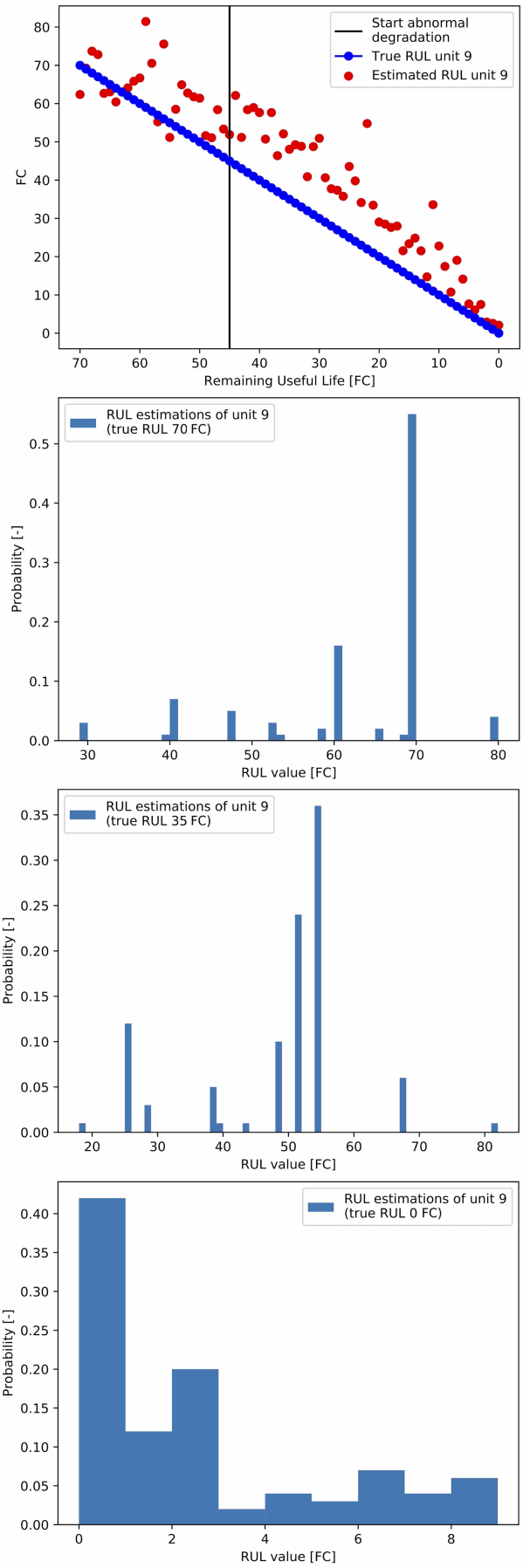


Figure E.1: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 9, DS01.

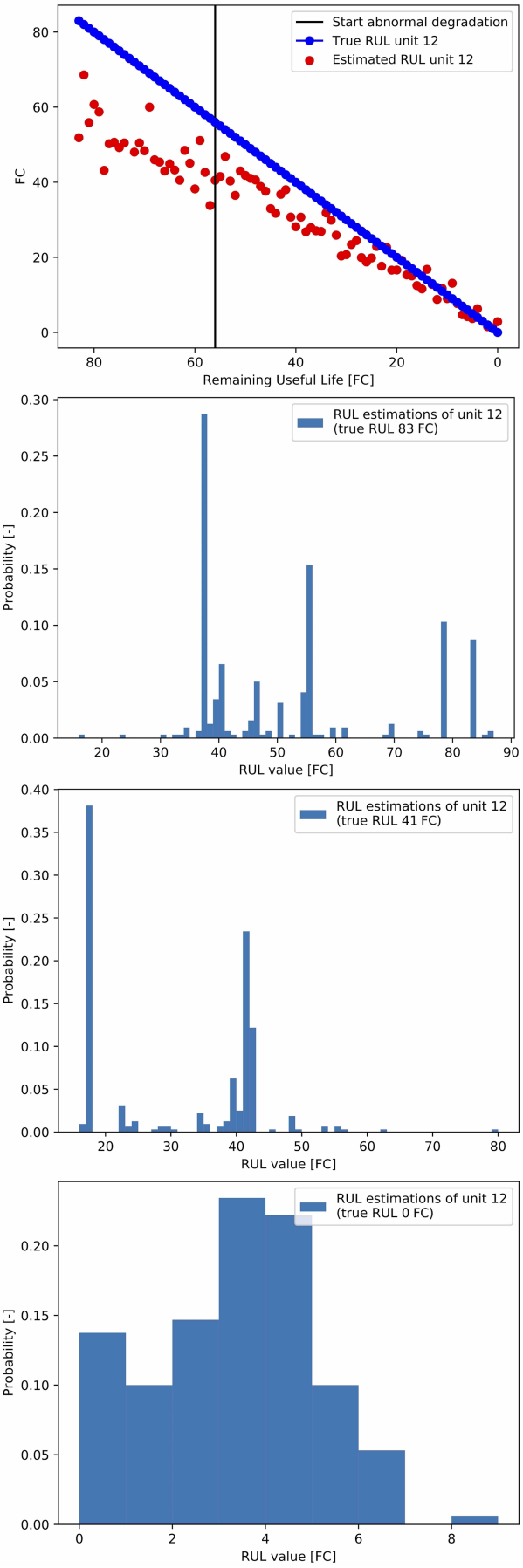


Figure E.2: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 12, DS03.

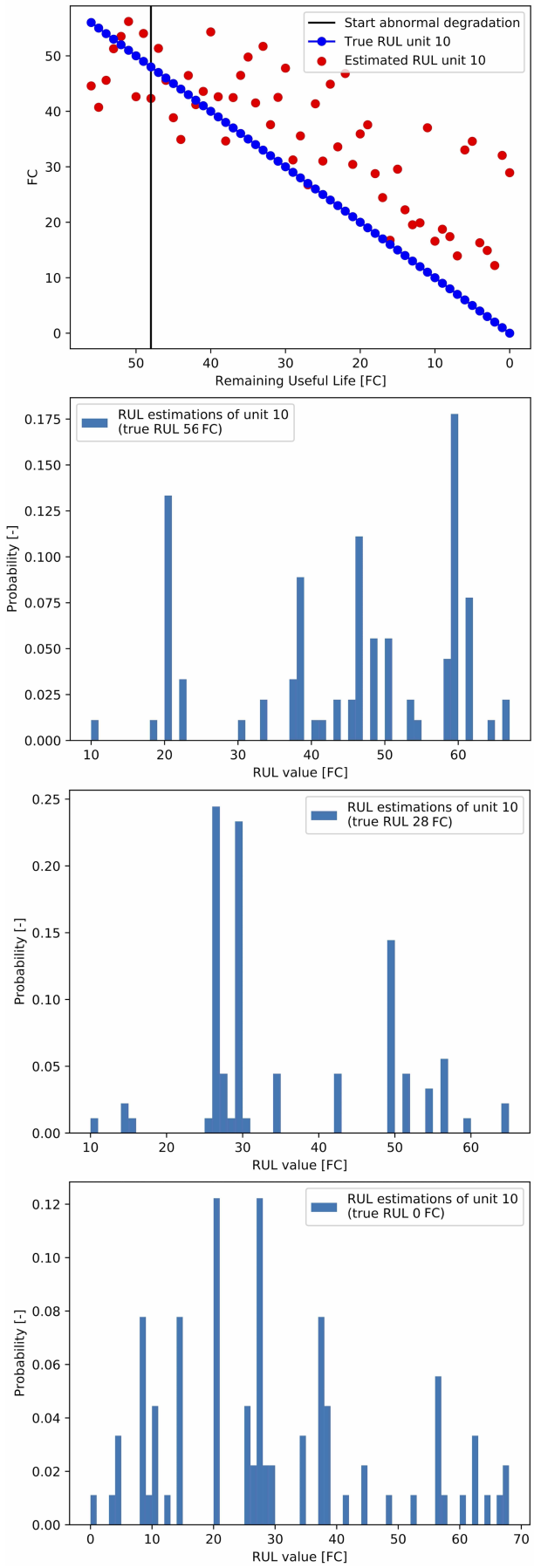


Figure E.3: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 10, DS03.

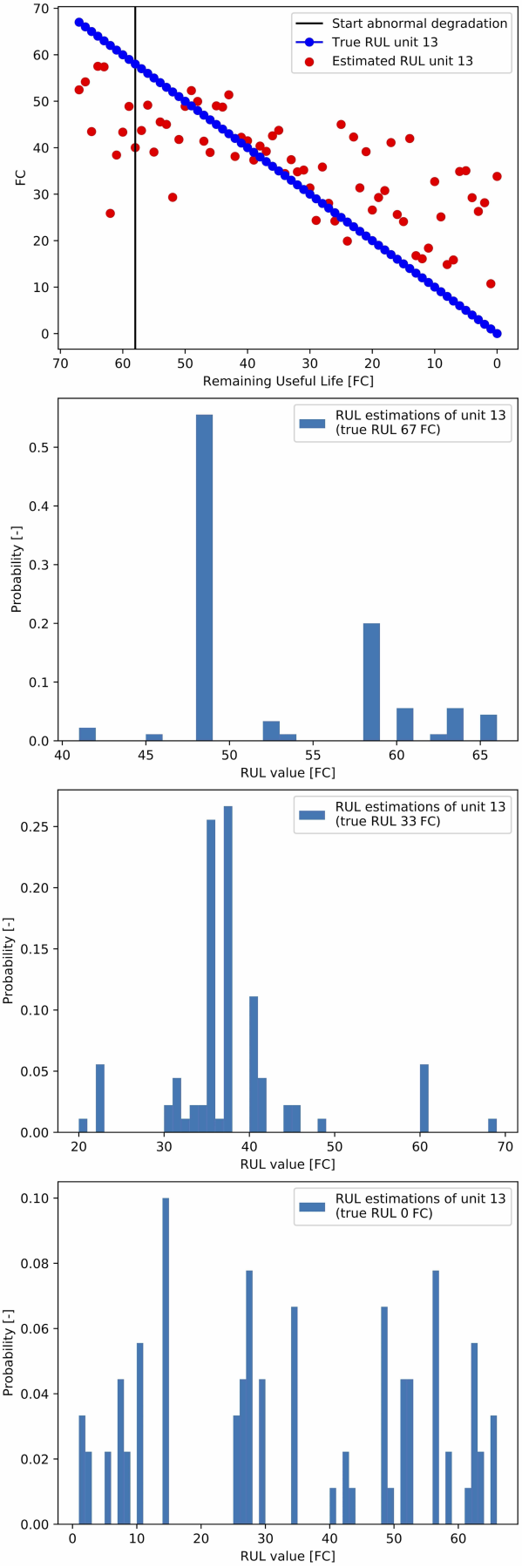


Figure E.4: RUL point estimations and probability distributions of the first, middle and last RUL point estimations of unit 13, DS03.

T

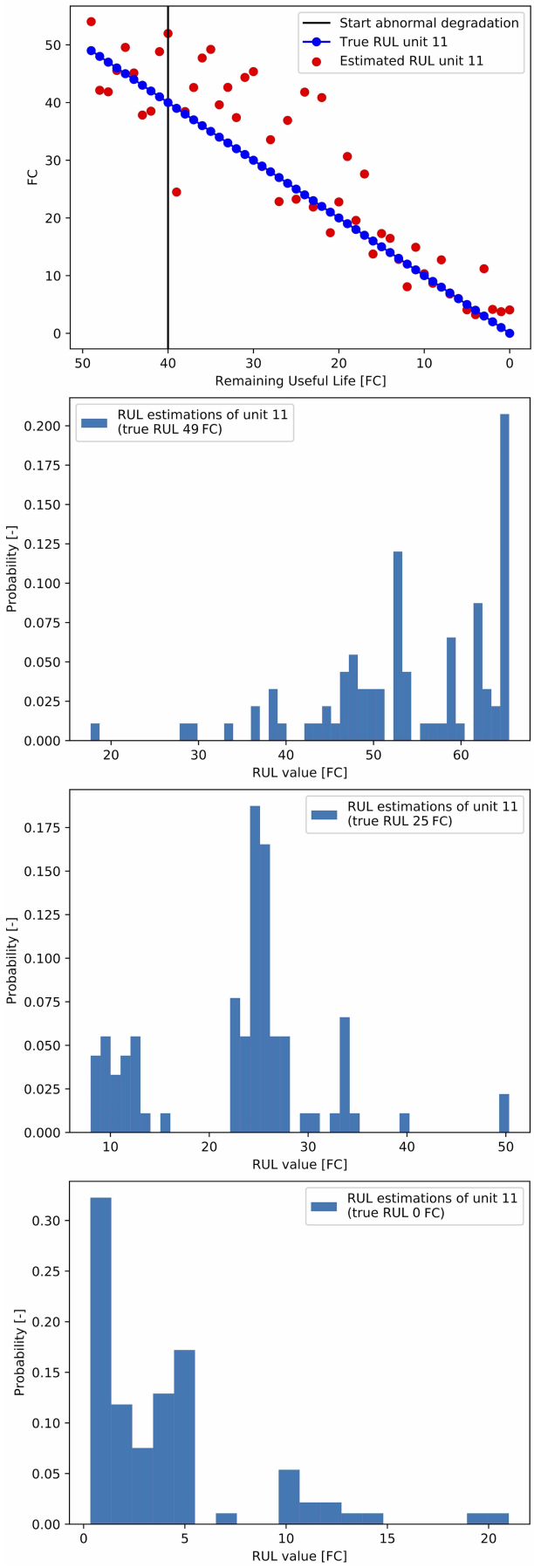


Figure E.5: RUL point estimations and probability distributions of first, middle and last RUL point estimations of unit 11, DS02.

U