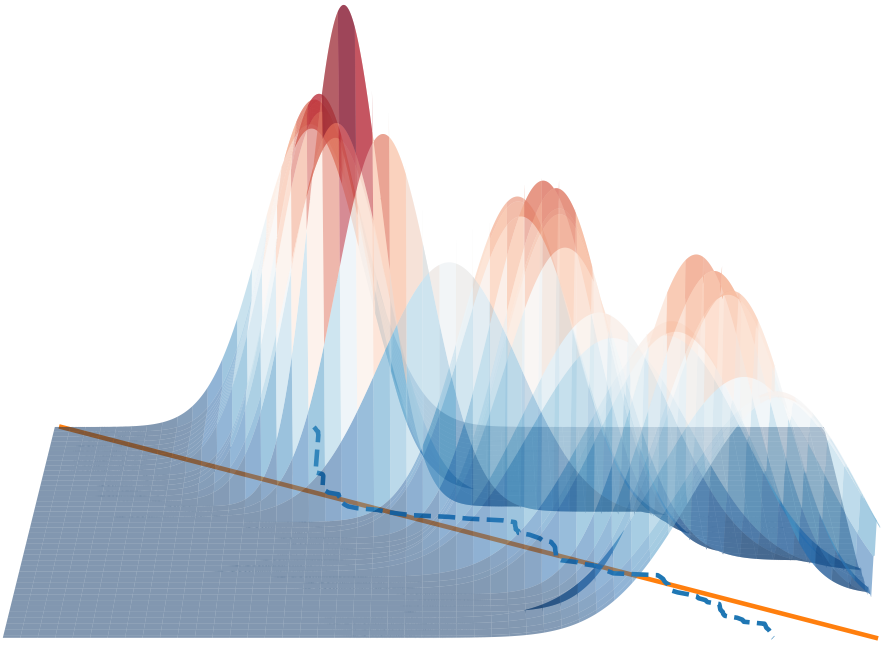


Prognostics for notched carbon/epoxy composites under variable loading using probabilistic methods

Master of Science Thesis

N. van den Bos

TU Delft Delft University of Technology

EWEM

DTU

Prognostics for notched carbon/epoxy composites under variable loading using probabilistic methods

by

N. van den Bos

to obtain the degree of:
Master of Science in Aerospace Engineering at the Delft University of Technology and
Master of Science in Engineering (European Wind Energy) at the Technical University of Denmark
to be defended publicly on Monday September 21, 2020 at 13:00.

Student number:	4274083 (TU Delft)	183013 (DTU)
Project duration:	November 1, 2019 – August 31, 2020	
Thesis committee:	Dr. D. Zarouchas,	supervisor (TU Delft)
	L.P. Mikkelsen,	supervisor (DTU)
	Prof. dr. S.J. Watson	chair (TU Delft) & censor (DTU)

An electronic version of this thesis is available at repository.tudelft.nl and findit.dtu.dk.

TU Delft DTU

Abstract

The field of prognostics on composites is relatively young, and research is focused on constant amplitude fatigue (CAF) loading, whereas variable amplitude fatigue (VAF) loading is more common in actual use-cases. Therefore in this research, the feasibility of different in-situ, data-driven probabilistic models is studied for prognostics on carbon fibre reinforced polymer (CFRP) specimens under VAF. Fatigue data with recorded acoustic emissions (AEs) is available from an earlier performed experimental campaign. Three models were selected: a statistical model to be used as a baseline comparison, a Gaussian process (GP) regression using cumulative AE energy, and a recurrent neural network (RNN) using all available AE features and load data. The models were compared for performance of remaining useful life (RUL) predictions on specimen under VAF loading for three different cases; when trained on CAF, VAF, and the combination of these two. Seven performance metrics were used to quantify their performance, as well as a qualitative comparison. The statistical method's performance varied per test specimen and can, therefore, only be used in practical applications when used very conservatively. It did not perform better in any of the three training cases as compared to the others. The GP regression was deemed not feasible due to high variability in its RUL predictions, high uncertainty due to the setting of a failure threshold as probability distribution based on other specimens, and high computational costs. The performance differed per test specimen as well. Finally, the performance of the RNN increased when trained on VAF data as compared to training on solely CAF data. It increased further when including both CAF and VAF in the training data. It is not yet feasible to be used in practice, due to variability in its predictions, and the inability to handle outliers. The latter is an issue for the other two models as well. The RNN outperformed the other two models when trained on VAF, and CAF and VAF data. Due to the definition of the failure threshold in the GP, the GP did not perform better than the statistical model. In the case of CAF training data, there was not a clear distinction between the performance of the statistical model and the RNN, except for one performance metric related to the precision of predictions. During this research, AE data from glass fibre reinforced polymer (GFRP) specimens tested on tension-tension (T-T) fatigue under different load levels became available. In a case study, the feasibility of a RNN, trained on AE data, was analysed for prognostics on this data-set. Due to large differences in the life-times between the specimens in this data-set, this was not feasible. Extending this RNN with a feedforward neural network (FFNN) which uses load data as well as input, provided worse predictions. The main conclusion drawn in this thesis is that with the current implementation of the used models, in-situ, data-driven prognostics on composite specimens under VAF is not yet feasible.

Acknowledgements

Completing this thesis has not been possible without the help from the following people (in no particular order), whom I am sincerely grateful to.

- Dimitrios Zarouchas, my TU Delft supervisor. After sparking my interest in prognostics in one of his lectures, he was immediately open to discuss the possibility of supervising a master's thesis in this direction and guide me during this thesis.
- Lars Pilgaard Mikkelsen, my Danmarks Tekniske Universitet (DTU) supervisor, who was quickly on board as well. His expertise in composites provided valuable input, and he made it possible for me to jump aboard an ongoing research project at DTU.
- Tue Herlau, my DTU co-supervisor. The discussions we had at the start of the thesis helped me stay critical of the obtained results throughout the last months.
- Nick Eleftheroglou, who conducted the experiments at TU Delft on which the majority of this thesis is based. Not only that but especially helping me out concerning the matching of the acoustic emission (AE) data and applied loads is much appreciated.
- Malcolm McGugan, who made it possible for me to use data from an ongoing research project at DTU. He did this by installing an AE sensor and sharing all the data with me, together with answering all my questions on the testing.

I would also like to thank all my fellow students from EWEM, who made the last two years unforgettable. Especially the activities outside classes such as climbing a wind-turbine, seeing the Northern lights in a remote cabin in Norway, visiting German wind energy companies, or just going to a bar made this an unforgettable period.

Finally, I could not have done this without my family and parents in particular. They have unconditionally supported me and encouraged me to take this opportunity, as well as many other opportunities during my studies.

Niek van den Bos
Rotterdam, August 2020

Preface

Before you lies the thesis which completes two unforgettable years in the European Wind Energy Master program, consisting of an MSc in Wind Energy at Danmarks Tekniske Universitet (DTU), and an MSc in Aerospace Engineering at TU Delft. During one of my courses on composites, I was intrigued by the concept of prognostics based on data-driven approaches. Thanks to Dimitrios Zarouchas, my TU Delft supervisor, I got the possibility to investigate this topic. Together with Lars Pilgaard Mikkelsen, my DTU supervisor, we had interesting discussions, and we shared plenty of ideas to tackle this challenge in the past nine months. The fact that one of us was always 700 km away from the others, partly with a pandemic going on, did not in any way stop them from helping me out, listening to my ideas (while trying to understand my explanatory sketches in MS Paint), and providing valuable feedback.

Contents

List of tables

List of figures

List of abbreviations

Notation	Description
AE	acoustic emission
AR	autoregressive
ARIMA	autoregressive integrated moving average
ARMA	autoregressive moving average
BNN	Bayesian neural network
C-C	compression-compression
CAF	constant amplitude fatigue
CBPM	cumulative bounded probability mass
CDF	cumulative distribution function
CFRP	carbon fibre reinforced polymer
CM	condition monitoring
CNN	convolutional neural network
CRA	cumulative relative accuracy
DI	damage index
DIC	digital image correlation
DTU	Danmarks Tekniske Universitet
EOL	end of life
EOUP	end of useful predictions
eu	energy unit
EV	extreme value
FFNN	feedforward neural network
FI	failure index
FRP	fibre reinforced polymer
GFRP	glass fibre reinforced polymer
GP	Gaussian process
GRU	gated recurrent unit
HI	health index
HPC	high performance computing
i.i.d.	independent and identically distributed
IR	infrared
kNN	k-nearest neighbours
KS	Kolmogorov–Smirnov
L-BFGS-B	limited memory Broyden–Fletcher–Goldfarb–Shanno bound constrained
LEV	largest extreme value
LRFD	load and resistance factor design
LSTM	long short-term memory
MAPE	mean absolute percentage error
MARSE	measured area of the rectified signal envelope
ML	machine learning
MLE	maximum likelihood estimation
MSE	mean squared error
NHCTHSMM	non-homogeneous continuous time hidden semi Markov model
NN	neural network
PCA	principal components analysis
PDF	probability density function
PH	prognostic horizon
PI	prediction interval
RA	relative accuracy

Notation	Description
ReLU	rectified linear unit
RMS	root mean square
RNN	recurrent neural network
RUL	remaining useful life
SE	squared exponential
SEV	smallest extreme value
SOM	self-organizing map
T-C	tension-compression
T-T	tension-tension
TTF	time to failure
UD	uni-directional
UTS	ultimate tensile strength
VAF	variable amplitude fatigue

List of symbols

Mathematical notation

$\odot$	element-wise multiplication
$\|\boldsymbol{x}\|$	magnitude of $\boldsymbol{x}$
$[X]$	Iverson bracket, returns 1 if condition X is satisfied, else 0
$\Gamma(z)$	gamma function
$\gamma(s,x)$	lower incomplete gamma function
δ	Kronecker delta
$\kappa(\boldsymbol{x},\boldsymbol{x}')$	covariance function
$\phi(x)$	activation function
A, or a	variable
$\hat{A}$, or $\hat{a}$	estimated/predicted variable
$\boldsymbol{A}$	matrix
$\boldsymbol{a}$	vector
$\boldsymbol{a}_i$	i^{th} row of $\boldsymbol{A}$
A_{ij}	j^{th} entry of the i^{th} row of $\boldsymbol{A}$
a_j	j^{th} entry of $\boldsymbol{a}$
$\boldsymbol{A}^{\mathrm{T}}$, or $\boldsymbol{a}^{\mathrm{T}}$	transpose of a matrix, or vector
$\boldsymbol{A}^{-1}$	inverse of a matrix
$\operatorname{cor}[X,Y]$	correlation between X and Y
$\operatorname{cov}[X,Y]$	covariance between X and Y
det	determinant
$E[X]$	expected value of a random variable
$\exp(x)$	exponential function
$f(x)$	function
$F_X(x)$	cumulative density function
$f_X(x)$	probability density function
$f_{XY}(x,y)$	joint probability density function
$\mathcal{GP}\left(m(\boldsymbol{x}),\kappa(\boldsymbol{x},\boldsymbol{x}')\right)$	Gaussian process with mean $m(\boldsymbol{x})$ and covariance $\kappa(\boldsymbol{x},\boldsymbol{x}')$
$K_v(x)$	modified Bessel function
log	natural logarithm
$m(\boldsymbol{x})$	mean function
$\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, or $\mathcal{N}\left(\mu,\sigma^2\right)$	(multivariate) normal distribution
$\mathcal{O}(n)$	computational complexity of an n-sized problem
$\mathrm{p}(X)$	probability of an event X
$\mathrm{p}(X \mid Y)$	conditional probability of X given Y
$\mathbb{R}$	real number set
$\mathbb{S}$	hypersphere set

Greek symbols

α	likelihood of a value occurring outside this confidence/prediction interval	-
α^+, or α^-	required confidence in remaining useful life (RUL) predictions	-
β	criterion for minimum probability mass within required confidence bounds	-
Δ	difference operator, or difference between prediction and actual RUL	-
ε	strain	-
ϵ	independent and identically distributed (i.i.d.) error, or extremely small value for numerical purposes	-
θ	scale parameter of the gamma distribution	-

λ	time window modifier for $\alpha - \lambda$ performance	-
μ	mean	-
ν	smoothness parameter of the Matérn kernel	-
ξ_i	components of spherical coordinate matrix	-
$\boldsymbol{\Sigma}$	covariance matrix	-
σ	stress, or standard deviation	N/m^2, -
σ_f	function variance in covariance functions	-
σ_y	white noise	-
$\boldsymbol{\tau}$	scaling vector	-
ϕ	angular spherical coordinate	rad

Latin symbols

B	mean bias	-
C	Convergence of a metric	-
$\boldsymbol{c}$	x-intercept of linear covariance kernel	-
$\boldsymbol{c}_t$	cell state in long short-term memory (LSTM) cell	-
$\tilde{\boldsymbol{c}}_t$	candidate values in LSTM cell	-
$\mathcal{D}$	data-set	
D	Kolmogorov–Smirnov (KS) test statistic, or dimensionality of data	-
d	distance	-
d_M	Mahalanobis distance	-
d_E	Euclidean distance	-
E	cumulative energy (random variable), or error, or Young's modulus	eu, -, Pa
e	cumulative energy	eu
F	force, or load	N
$\boldsymbol{f}$	vector of function evaluations	-
$\boldsymbol{f}_t$	forget gate in LSTM cell	-
h	hidden state	-
$\boldsymbol{I}$	identity matrix	-
$\boldsymbol{i}$	input gate in LSTM cell	-
$\boldsymbol{K}$	covariance matrix	-
k	shape parameter of the gamma distribution	-
$\boldsymbol{L}$	Cholesky factorisation	-
L	number of labels in data-set, or loss	-
l	length scale in covariance function, or label of time-series	-
$\mathcal{M}$	model	
M	performance metric	-
N	matrix dimension	-
n_E	number training epochs	-
n_h	number of hidden nodes in neural network (NN)	-
n_M	number of model architectures	-
n_p	number of parameters	-
n_R	number of model repetitions	-
$\boldsymbol{o}$	output gate in LSTM cell	-
P_f	probability of failure	-
R	stress ratio	-
r	RUL	-
$\boldsymbol{S}$	matrix of spherical coordinates)	-
S	survival of a specimen, or precision (variability between predictions)	-
s	number of survived cycles	-
T	number of cycles at failure	-
T_E	cumulative energy threshold (random variable)	eu
t	number of cycles	-
t_P	start of predictions	-
$\boldsymbol{U}$	weight matrix for input data in recurrent neural network (RNN)	-
$\boldsymbol{V}$	weight matrix for previous output in RNN	-
$\boldsymbol{W}$	weight matrix in NN	-

w	weight	-
x	input	-
x_c	centroid in x-direction	-
y	observation	-
y_c	centroid in y-direction	-

1 Introduction

The use of fibre reinforced polymers has been growing in multiple industries, including the aerospace and wind-energy industry. However, structures in these industries have to cope with fatigue loads. Some sources of fatigue in aircraft are due to take-off and landing, pressurization and gusts. The wind-energy industry is constantly moving towards larger blades for offshore turbines, resulting in more extreme fatigue loads. With larger blade lengths come higher aerodynamic forces, masses and moment arms, while in a blade's lifetime of 20 years it typically has to survive over 100 million cycles. An example for the wind-energy industry is that of GE's Haliade X, with 107 m long rotor blades (GE Renewable Energy). With the resulting loads in these blades, designing for fatigue is critical.

Considering the cost of maintenance, there are three traditional maintenance strategies (figure 1.1) (Tchakoua et al., 2014). The first is preventive maintenance, which is time-based. This type of maintenance is often too frequent but results in low repair costs. On the other side of the spectrum is reactive maintenance. Running up to failure gives a low prevention cost, but high repair cost. In the case of a wind turbine blade, the latter is unacceptable since failures of the blade often lead to catastrophic failures of the complete turbine. This is not only a financial loss, but it can also impact safety as well as the environment around it. The final type of maintenance strategy is that of intelligent maintenance. This is a condition-based approach, where the condition of the structure is determined from a distance, and maintenance is carried out if a fault is predicted to occur in the near future, i.e. the remaining useful life (RUL) is too low. This is referred to as prognostics. While prognostics has an impact on maintenance planning, operations, and profitability, it is also of value for re-use and recycling of components after their service (Si et al., 2011). Three main model categories exist in the field of prognostics. One of these is a data-driven approach. Requiring no prior knowledge of the case or physics behind it, these models can make RUL predictions solely based on available data.

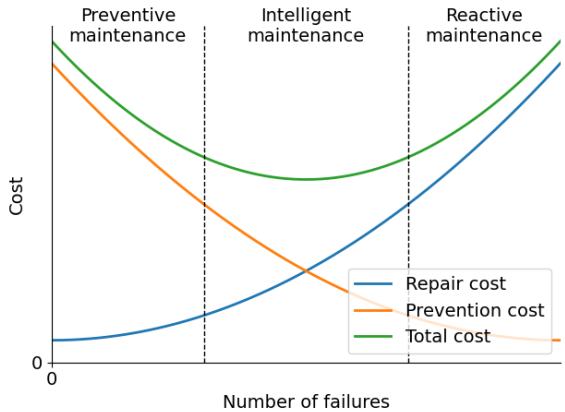


Figure 1.1: Different maintenance strategies and their associated cost, adapted from Tchakoua et al. (2014)

The degradation process of composites under fatigue loading is a complicated mechanism, however. Multiple failure mechanisms take place while also influencing each other. Next to this, there is high variability

in the properties of composites, and the (stiffness) degradation of composites varies per specimen (Yang et al., 1990). The risk of this stochastic nature is that deterministic designs overestimate the reliability of composite structures, as Lekou and Philippidis (2008) have found for a case of a wind turbine blade. Therefore, prognostics could be used to perform in-situ predictions of the RUL of an asset, and determine on time if maintenance or replacements should be carried out, to prevent catastrophic failure.

Current research on prognostics for composites is relatively young, dating back to a maximum of roughly 15 years. Furthermore, research is focused on constant amplitude fatigue (CAF) loading, while in practice, the amplitude varies. Therefore, this project aims to find out whether data-driven prognostics can be used on variable amplitude fatigue (VAF) loading. It is not only useful to know how different models perform on subjects under VAF when trained on VAF. Can a model also predict RUL of subjects under VAF, while only trained on CAF? Is there a significant difference between the two, which causes incompatibility? Multiple data-driven models will therefore be compared in their performance based on training on a data-set containing only CAF, only VAF, and a combination of the two.

1.1. Objective and research questions

The objective of this research is to investigate the feasibility of in-situ, data-driven prognostics on composites under VAF, by training multiple probabilistic models on CAF and/or VAF data and assessing their performance in the prediction of RUL. In order to fulfil this objective, the following research questions must be answered:

1. What is the performance of different data-driven prognostics models in in-situ predictions of the RUL of a composite under VAF using acoustic emission (AE) data, when trained on CAF data?
2. What is the performance of different data-driven prognostics models in in-situ predictions of the RUL of a composite under VAF using AE data, when trained on VAF data and using the known load path as a second input?
3. How is the performance of a data-driven prognostics model improved when training the model on data from both CAF as well as VAF, using the known load path and AE data as input?
4. To what extent are the used models able to adapt to possible differences in the lifetimes of specimens under VAF?

These four research questions go hand in hand. Because of this, the following sub-questions which allow the main research questions to be answered apply to all four.

The first sub-question is related to the models: "*which probabilistic, data-driven models for in-situ prognostics can be implemented within the timeframe of this thesis?*" Choosing models is, of course, one of the fundamentals of studies like these. In choosing these models, it is essential to keep in mind which possible requirements models have for their input data.

The second sub-question will be investigated partially parallel to the first; "*which features from AE data can be used in prognostics?*" The features have to follow the degradation process as much as possible, for the models to make accurate predictions. Next, "*what is an appropriate threshold to set for the selected features?*" While generally, the feature series first has to be extrapolated, it also has to be determined at which point the end of life (EOL) occurs.

Finally, when the models have made their predictions, they have to be compared to each other, for all three cases mentioned in the research questions. To be able to do this, the following has to be determined: "*what is an objective way of comparing the outcome of the models both quantitatively as well as qualitatively?*" Within this comparison, attention should not only be paid to a general trend but especially to outliers.

The data which is used for this research is AE data from a testing campaign at TU Delft, conducted before this research. Both CAF as well as VAF tests were performed on notched carbon fibre reinforced polymer (CFRP) specimens, with a quasi-isotropic layup.

Case study: varying load levels
During this research, a testing campaign at Danmarks Tekniske Universitet (DTU), on uni-directional (UD) glass fibre reinforced polymer (GFRP) specimens took place. In this campaign, GFRP specimens were loaded under CAF, in tension-tension (T-T). In this research, specimens were loaded to different maximum strain levels, which caused differences in the lifetimes. The recurrent neural network (RNN) was thought to be able to make the most accurate predictions in this case, which therefore raised the following sub-question: *To what extent can a RNN predict the RUL of GFRP specimens which are loaded under different levels, purely on AE data?* Naturally, this also raised the question of whether these predictions could be improved: *Can these RUL predictions be improved when feeding the model information about the load levels and stress ratios?*

1.2. Structuring

These research questions are answered throughout this thesis, according to the following structure:

- Chapter 2 will discuss the current state of the art by first giving an introduction to composites and the relevant failure mechanisms for this research. This is followed by a section on AEs, and the research which was done to quantify their relation with damage. Next, the field of prognostics is covered. Finally, different models used in prognostics are discussed, together with the choices made on which models to use for this thesis.
- Chapter 3 covers the methodology behind the research. The theory behind the models (a statistical model, Gaussian process (GP) regression, and RNN), is thoroughly covered, as well as the implementation, followed by a section which covers the performance metrics to compare these models objectively.
- Chapter 4 treats everything concerning the data which is used in the models. This starts with the setup of the experiments and the processing of the acquired data for both the experiments on CFRP as well as the GFRP specimens. Then, the computational setup is briefly discussed. Finally, the selected features for this thesis are discussed.
- Chapter 5 discusses the results of the RUL predictions, model by model. The research questions are answered per model. The models are also compared to each other, determining which model shows the best performance, both quantitatively, but also qualitatively. Finally, the results from the case study are discussed.
- Chapter 6 marks the end of the thesis; the conclusions are drawn, as well as recommendations for further research.

2

State of the art

This chapter covers the relevant research for this thesis. First, composites and their failure mechanisms are introduced. This is followed by the research on acoustic emission (AE) data, focusing on the classification of signals. Next, the field of prognostics is covered, together with relevant research in this field. Finally, different kinds of data-driven models are introduced, which is concluded with the choices made for the models which are to be used in this research.

2.1. Fibre reinforced polymer composites

While a composite can, in general, be any type of material made of multiple constituents, the focus of this section will be on fibre reinforced polymer (FRP) composites, the family of composites which is central to this research. This section first gives a general introduction on FRP composites concerning its constituents and imperfections. It is concluded by a description of the complex series of failure mechanisms within composites, for three different fatigue load types.

2.1.1. Properties

As the name suggests, a composite consists of multiple materials, as can be seen in representation in figure 2.1. The choice of materials for both the load-bearing fibres as well as the matrix which holds them together is extremely important for the properties of a composite and discussed first. This is followed by the orientation of the fibres and the stacking of different layers, which also impacts the properties of a composite. Finally, the scatter of material properties and imperfections are being discussed. These are present in practically every composite sample ever fabricated and lead to unpredictable behaviour.

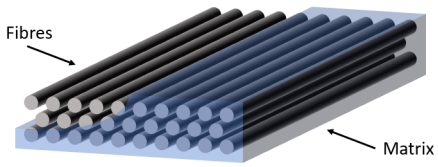


Figure 2.1: A depiction of a FRP, with aligned fibres, embedded in a matrix

Fibres

The fibres are the load-bearing components of a polymer matrix composite. There are many different types of fibres, starting with a division between natural and artificial fibres. Examples of natural fibres are hemp and flax. Within man-made fibres, there are again multiple different types; carbon, glass, polyethylene and Kevlar, for example. Furthermore, there are fibre types with unique properties within these groups. All these fibre types have different properties (e.g. stiffness, strength, fibre diameter), and therefore cause the composite to have different properties (e.g. S-N curve, density). This makes a fibre type suitable or not for a specific use case. Whereas fibre types perform strongly in one area, they perform poorly in others; there is not a single perfect fibre type (yet).

Typical carbon fibres are T300 (baseline), IM6 (high modulus) or AS4 (high strength). Examples of glass fibres are E-glass (named after initial electrical applications), S-glass (high strength), and H-glass (high modulus). (Wicaksono and Chai, 2013)

Due to the complex manufacturing process of carbon fibres, they are more expensive to produce than glass fibres. On the other hand, carbon fibres show a greater specific modulus and -strength than glass fibres, making them more suitable for high-performance applications, where weight is a driving design factor.

Matrix types

The matrix is the material which keeps the fibres together. Because this research focuses on polymer matrix types, only these will be discussed, and, e.g. metal matrix composites or ceramic matrix composites will not be covered. Just as in fibres, there are different types of matrix materials, of which there are two subgroups in the polymer matrix class; thermoset and thermoplastic resin. A thermoplastic resin is said to have an advantage over thermoset material in ductility and fracture toughness, shown by Wicaksono and Chai (2013). This is important since damage often starts as cracks between matrix and fibre material in fibre layers which are not oriented to the load direction (Mikkelsen, 2020). A tougher resin will result in higher interfacial toughness, i.e. a higher resistance to delamination if the fibres allow this. Finally, the interfacial strength of the matrix and the fibres is another important factor in the toughness of a composite. A lower interfacial strength effectively resists the crack in propagating from the matrix to fibres, therefore increasing toughness of the composite. (Wicaksono and Chai, 2013)

Thermoset matrices are often used in the aerospace- and wind industry. Nevertheless, thermoplastic matrix composites are appearing more and more within the aerospace industry (Airbus, 2015). Thermoset composites can better deal with higher temperatures then thermoplastics. However, the manufacturing time of composites using thermoset matrices is often longer than that of thermoplastics because they need time to harden. Thermoplastics are also easier to recycle because their structural properties do not degrade significantly when they are heated up and reshaped. This also allows them to be welded together. Due to an initial high supply of thermoset matrices, their cost was perceived as lower than that of thermoplastic matrix types. As more thermoplastic matrix composites started to appear in the market, this difference in cost is decreasing. (Airbus, 2015)

Fibre orientation and stacking sequence

Not only the materials, but especially the orientation of the fibres makes an enormous difference in the properties of a composite. Because fibres provide uni-directional (UD) strength and stiffness to the composite, their orientation is crucial.

A composite can be tailored to its specific needs by the customisation of its layup. In the case of a wind turbine blade, for example, high UD strength and stiffness are required because the load is primarily in one direction. A typical layout is then (quasi-)UD, where all/most fibres are aligned in the direction of the applied loads, and some are aligned in other directions in order to account for remaining transverse loads. A cross-section of such a layup is shown below in figure 2.5.

In other cases, isotropic strength can be required. Now, a layup of fibres in for example the 0°, $\pm 45^\circ$ and 90° directions can be used. This is an example of a quasi-isotropic layup. While it can handle loads from all in-plane directions, more material is needed to support a specific load as compared to a UD layup. On the other hand, the downside of a UD layup is that there can be no exceptions to the load case for which it is designed.

Finally, the order of the laminae, the stacking sequence, is also important. A symmetrical stacking sequence, for example, does not bend/twist while in pure tension, while some asymmetrical ones do. Another example is that of the bending strength and stiffness, which can be increased by having plies on the top and bottom of the stack in the desired direction. This effect can be compared with the increased bending resistance of an I-beam.

Scatter of properties and imperfections

While from the properties of fibres and matrices composites sound very promising, there are also some disadvantages. Due to the complexity of the material, there is more scatter in fatigue resistance and material properties than in, for example, metals. There are different kinds of imperfections which can cause scattered, lower material properties. The ones discussed below are purely imperfections within the material; factors such as load misalignment are seen as external factors and are not discussed. The amount of scatter can be seen in figure 2.3, where the dots indicate specific measurements. The scatter also leads to differences in the number and timings of AE events, which can be seen in the plots in section 4.4.2.

First of all, the properties of the fibres and matrix themselves can scatter. In figure 2.2 below, different fibre diameters can already be spotted, for example. Due to the large numbers of fibres however, these variations do not have a significant impact on the properties of a composite. Secondly, during the manufacturing process, more imperfections relating to the fibres can be created. Fibres can break, or they can be misaligned. The former is not always an issue, since loads can be transferred into the matrix, and back into the fibre again; bridging the failed region. The latter can cause regions in the composite to have with higher loads within the fibres than they are designed for. Furthermore, when in compression, this may lead to micro buckling. Finally, a design variable is the volume- or weight fraction of fibres and matrix. This can vary locally within the composite. Furthermore, in an ideal world, these would make up all of a composite. However, during fabrication imperfections occur, causing voids to be left in the composite. These are also referred to as porosities. Three kinds of porosities can be seen in figure 2.2. An interface porosity (B) is a location where there is no bonding between a fibre and the matrix; thus allowing no load transfer. Next, an impregnation porosity at spot C is a place where the matrix material could not reach, since the fibres encapsulated this region. The last type of porosity is that of voids in the matrix, shown at point D. These three types of porosities cause a decrease in the possibility to transfer loads, as well as possible stress concentrations.

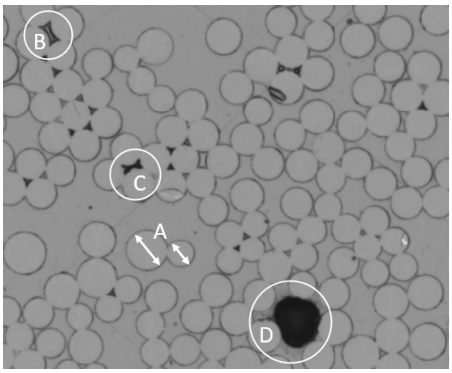


Figure 2.2: A group of glass fibres within a glass/carbon FRP, taken from Malte Markussen (2015). Four key areas are shown: different sized glass fibres (A), an interface porosity (B), impregnation porosity (C), and matrix porosity (D).

2.1.2. Failure mechanisms

Failure mechanisms in FRP composites are vastly different as compared to those in other materials such as metals. First of all, failure does not occur locally at an imperfection, where a crack opens and extends into the material. Instead, it is a much more global phenomenon in composites; faults occur in multiple locations in the material. Secondly, the failure mechanisms are complicated in composites. Depending on the composite and load case, multiple damage accumulation mechanisms are present; matrix cracking, fibre-matrix debonding, fibre failure, and delamination. The mechanisms may, of course, also influence each other. (Wicaksono and Chai, 2013)

Failure mechanisms differ based on the type of fatigue loading applied on a composite, which can be distinguished between tension-tension (T-T), tension-compression (T-C), and compression-compression (C-C). Figure 2.3 shows the S-N curves for the three different load cases for multi-directional composites. The stress ratio, or R-ratio, is the ratio between the minimum and maximum stress, $\sigma_{min}/\sigma_{max}$. It can be seen that the T-T ($R = 0.1$) and T-C ($R = -1$) case have relatively similar slopes, also called Basquin exponents, suggesting a similar failure mechanism. This is not the case for C-C ($R = 10$), which has a different failure mechanism. (Mikkelsen, 2020)

Environmental effects such as temperature, moisture, and acidic corrosion also play a role in the fatigue behaviour and resistance of composites (Wicaksono and Chai, 2013), but are outside the scope of this thesis. In the following sections, T-T, T-C and C-C loading are covered respectively. T-T is extensively researched in the past decades, and a thorough understanding of failure mechanisms has been developed. There is little research available on T-C and C-C failure mechanisms, especially due to the hardships in testing under these conditions. Because C-C is not the load case for the samples used in this theses, it is briefly covered at the end.

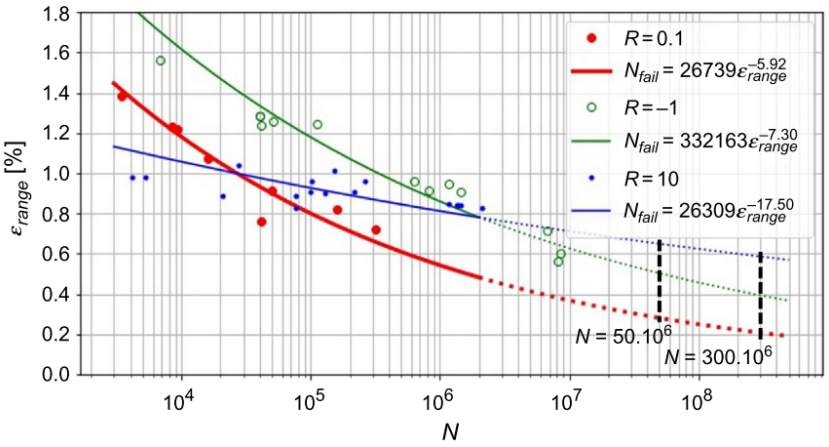


Figure 2.3: S-N curves for different load cases on multi-directional composites, taken from Mikkelsen (2020). The data from this figure is from the OptiDat database (Nijssen et al., 2006).

Tension-tension loading

T-T fatigue is by far the most researched type, since it is the most experienced load type due to the longevity of composites under this load type. It is also relatively easy to test, as compared to fatigue tests with compressive loading, in which control of the stress state and buckling effects are hard to achieve (Gamstedt and Sjögren, 1999).

In a typical composite specimen under T-T loading, there are three distinguishable phases in the development of damage in quasi-isotropic, cross-ply and quasi-UD composites (Reifsnider and Jamison, 1982). The three phases are shown in the context of stiffness degradation in figure 2.4. The three phases of degradation are respectively:

Stage I: initial stiffness drop;

Stage II: gradual stiffness degradation;

Stage III: fast final stiffness degradation and failure.

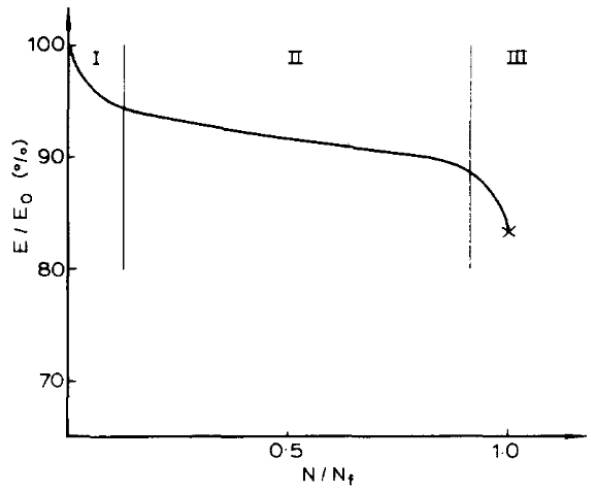


Figure 2.4: Typical stiffness degradation curve (normalised), taken from Ye (1989)

It has to be noted that the amount of stiffness degradation varies considerably based on the type and layup of a composite; Reifsnider and Jamison (1982) reports 18% stiffness degradation on a quasi-isotropic carbon/epoxy specimen when at failure, whereas a cross-ply fails below 10% stiffness degradation already.

Stage I Transverse matrix cracks started occurring first in Reifsnider and Jamison (1982) their experiments on quasi-isotropic ($[0, 90, \pm 45]_s$) and cross-ply ($[0, 90_2]_s$) carbon/epoxy laminates. The cracks were distributed throughout the specimen but did not reach the characteristic damage state (CDS) yet (see figure 2.6a). This state is reached when no new cracks are created. When a matrix crack occurs, internal forces in the matrix are transferred to the fibres through shear. This load transfer requires a certain length along a fibre. Now, the matrix material close to cracks does therefore carry less load. At a certain stage of crack saturation, the distance between cracks is small enough such that there is no effective stress transfer to the matrix material anymore, and therefore no new cracks will form. The eventual saturation of crack formation is achieved in every ply, where the crack spacing depends on the specific type of material and geometry. This mechanism is likely the main driver of the sharp stiffness degradation in stage I. (Reifsnider and Jamison, 1982)
In figure 2.6

Quasi-UD glass fibre reinforced polymer (GFRP) specimen are tested in T-T loading by Zangenberg et al. (2014) and Jespersen et al. (2016), who analysed damages using 2D and 3D methods respectively. The composites consist of mainly UD, non-crimp laminae, together with a few backing bundles. Both authors report matrix cracking in stage I in the backing bundles, at cross-over points (as shown in figure 2.5).

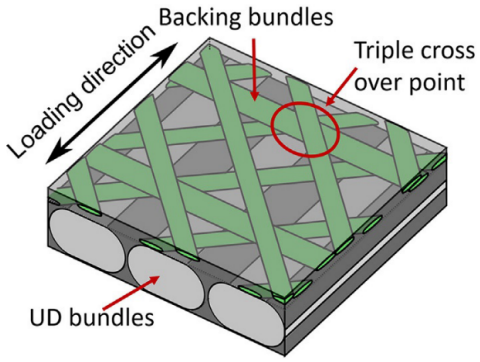


Figure 2.5: Sketch of UD- and backing bundles, taken from Mikkelsen (2020)

Stage II This stage lasts for the majority of a subjects life; damage accumulation is constant in this phase. The rate of stiffness degradation decreases relative to stage I due to the fact that fibres act as obstacles to matrix crack growth (Ye, 1989).

At the end of this stage, delamination growth along the outer plies was complete for the quasi-isotropic and cross-ply specimen from Reifsnider and Jamison (1982). Longitudinal cracks started appearing and growing in the 0° plies in their cross-ply laminate, as can be seen in figure 2.6b-d. Reifsnider and Jamison (1982) notes that this phenomenon occurred due to too high transverse stress in these plies, caused by the surrounding 90° plies' constraint.

For quasi-UD laminates, damage grows into the UD layers adjacent to the backing bundles due to sliding friction from rubbing and fretting, especially near intertwining regions in the backing bundles. The friction naturally causes increases in temperature, which can be picked up by infrared (IR) thermography. This eventually causes fibre failures and fibre-matrix debonding in the UD laminae close to the backing bundles. This process naturally leads to less stiffness, but also higher damping of the composite due to the energy which is taken up by the friction.

Stage III In the quasi-isotropic specimen, Reifsnider and Jamison (1982) note the appearance of micro-cracks at this phase, which occur closely to transverse cracks. The crack density increased towards the specimen's end of life (EOL). Eventually, coalescence and interaction of micro-cracks, and the quick growth of 'favourable cracks' lead to catastrophic failure (Ye, 1989). The cross-ply laminates of Reifsnider and Jamison (1982) showed stepwise degradation in this phase. First, the 90° plies fail, followed by the ultimate failure of the specimen when the 0° plies fail. Transverse matrix cracks eventually cause fibres to fail as well, as illustrated in the right of figure 2.7. Damage localisation also takes place in the quasi-UD laminate, especially for longitudinal cracks. Zangenberg et al. (2014) mentions that this often occurs close to clamps, due to the local stress concentrations at these locations. This last stage generally lasts less than 20% of a composite's lifetime, whereas sometimes localised cracks grow fast on a macroscopic scale, causing it to fail extremely quickly in this phase (Ye, 1989).

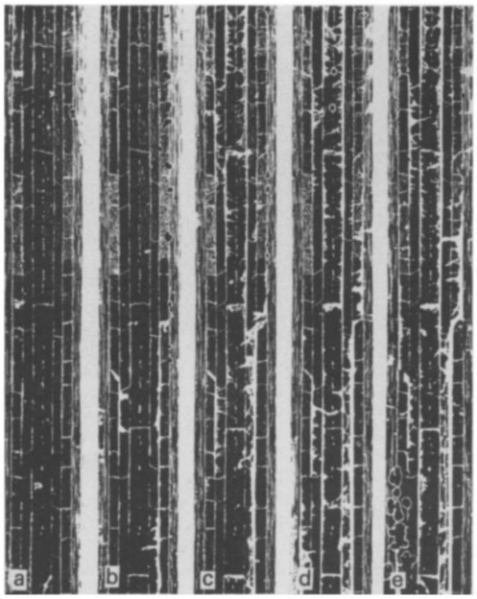


Figure 2.6: Edge replicas of the quasi-isotropic specimen at stiffness degradations of 2% (a), 4% (b), 8% (c), 12% (d) and 15% (e), taken from Reifsnider and Jamison (1982)

Tension-compression loading

Mikkelsen (2020) observed a similar failure mechanism for T-C (R=-1) as for T-T loading, which is also initiated by transverse cracks in the matrix material in the backing bundles of a quasi-UD GFRP composite. In the compression part of the loading, two additional mechanisms were observed; fibre crushing and the limited formation of kink-bands. The failure mechanism fibre breakage occurs in the tensile part of the loading.

Transverse cracks in plies not in the loading direction are also the initial damages according to Gamstedt and Sjögren (1999), who note that these type of damages are again not critical. However, these cracks might interact with surrounding laminae, resulting in eventual delamination in compression and fibre failure in tension, as illustrated in figure 2.7. These transverse cracks start from cavities caused by fibre-matrix debonding. This process, as compared to T-T fatigue, occurs at a much higher rate, due to the compression component in the loading. The micromechanisms of this phenomenon are thoroughly studied by Gamstedt and Sjögren (1999). The increase in both density and rate of occurrence of transverse cracks is detrimental for the fatigue performance in this load case. Mall et al. (2009) finds that on both notched and unnotched carbon/epoxy specimens, the initial process of matrix cracking and delamination causes the matrix to be unable to hold the compressive loads. This causes both micro buckling and the kinking of fibres. Fibre breaking eventually occurs at the EOL of a specimen and is the cause of final failure.

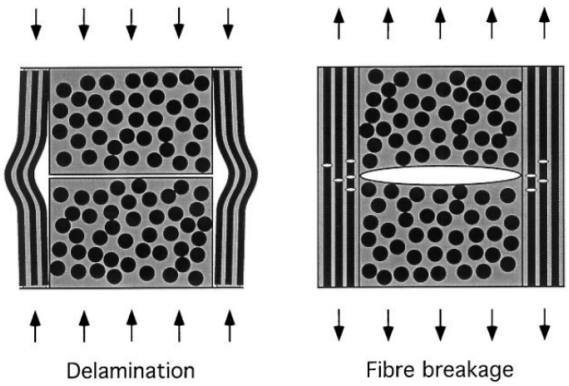


Figure 2.7: Transverse cracks leading to delamination in compression, while leading to by fibre failures in tension, taken from Gamstedt and Sjögren (1999)

Compression-compression loading
As discussed above, the failure mechanism for C-C loading is different from that of T-T and T-C. This causes the S-N curve in figure 2.3 to have a different slope than the other two. Instead of matrix cracking, delamination, fibre kinking (in T-C) and eventually fibre failures, shear cracks are driving for final failure (Fraisse and Brøndsted, 2017).

2.2. Acoustic emissions

A common feature which is be tracked during the life of a composite and one used in this thesis is that of AEs. Upon the occurrence of damage, stored elastic energy is transformed into a transient mechanical wave, propagating through the structure, which can be captured by a transducer (Eitzen and Wadley, 1984). This method is a very common condition monitoring method, not only in composites but also in other materials.

An AE signal contains a wide array of data, all of which can be used to possibly identify events within the examined structure. A single event is captured in a waveform, shown in figure 2.8. It is common practice to set a threshold in order to filter out unwanted background noise. The duration of the AE covers the entire AE event, from where it first crosses the threshold up until the last crossing. Such a crossing is called a count, the number of which is another feature of an AE event. The rise time is the time the signal takes from the start of the signal until the peak of the signal. These times are in the order of μs. Another feature which is extracted from this signal is the root mean square (RMS) of the voltage. From this signal, the amplitude, frequency and energy are determined as well. While this signal contains information on both the location and characteristics of the AE event, characterising it is difficult. This is due to the fact that the wave is modified in its transmission, due to factors such as material non-homogeneity, geometry and loading (Eitzen and Wadley, 1984).

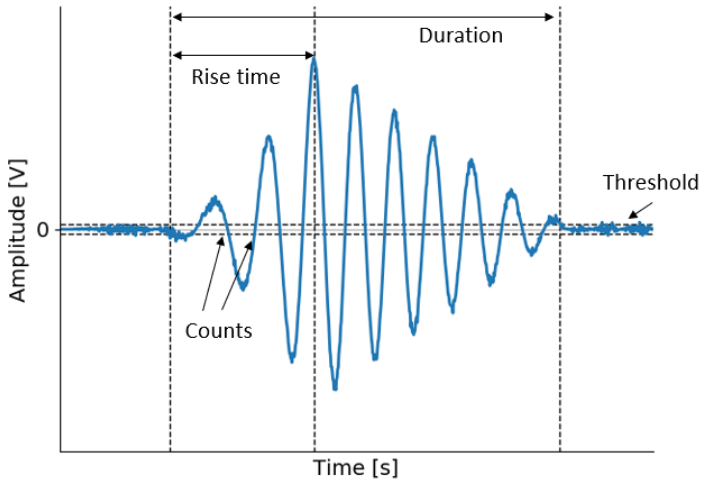


Figure 2.8: Depiction of an AE signal and its parameters

Current research has focused on three main feature sets from AE: single parameter (mostly amplitude), multi-parameter, and wavelets. As can be seen below, much of the research is related to diagnostics; it concerns the identification of different failure mechanisms. The type of failure mechanism is also a feature, which can be used for prognostics, as some of the papers (for example, Arumugam et al. (2010); Godin et al. (2019)) suggest. However, there is not yet a consensus about the identification of specific failure mechanisms; as the research below will show.

2.2.1. Single parameter

Barré and Benzeggagh (1994) performed a study on damage mechanisms in short glass fibre reinforced thermoplastics in which they reported that the AE amplitude varied with the damage mechanism. The four mechanisms and corresponding AE amplitudes which they identified were: matrix cracking (40-55 dB), interface fracture (60-65 dB), fibre pull-out (65-85 dB) and finally fibre fracture (85-95 dB).

Similarly, Arumugam et al. (2010) try to predict the failure load in uniaxial tension of (impacted) carbon fibre reinforced polymer (CFRP) based on cumulative counts of AE events, in bins of 10 dB, from 50–100 dB. The authors perform predictions at 50% and 75% of the failure load and manage to obtain a maximum error of

15%.

In fatigue testing on open-hole carbon/epoxy specimen by Eleftheroglou et al. (2016), the cumulative energy of AE events is taken, and shows a correlation with the stiffness degradation, some specimens better than others. The authors conclude that cumulative AE energy is promising for describing the damage process in composites in fatigue.

Liu et al. (2009) take cumulative AE counts as input for their Gaussian process model to estimate remaining useful life (RUL), due to their monotonic increase. They work with a damage index (DI), the normalised value of a parameter describing the damage process, e.g. the stiffness degradation. This parameter is 0 at the start of fatigue testing, and 1 at the EOL. This requires knowledge of the entire series of counts. Therefore, this research is not about in-situ/online prognostics, but rather offline prognostics.

Finally, Surgeon and Wevers (1999) uses a modal technique on different CFRP laminates in tensile and bending tests, with two AE sensors. The technique distinguishes between high-frequency extensional modes and lower frequency flexural modes. The former are assumed to relate to matrix cracking and travel faster to the medium than the latter, which are assumed to be related to fibre fracture. Discrimination between these two modes was done in a qualitative manner by manually comparing the signals at both sensors. The author therefore correctly notes that this technique was not yet ready to be used in practical cases, where this expensive technique would diminish the resulting cost savings from AE techniques.

2.2.2. Multi-parameter

Gutkin et al. (2011) uses five parameters (peak amplitude, peak frequency, energy, rise time and duration). His clustering analyses show that most clusters are defined by peak frequency. Testing was done on different CFRP laminates (0°, 90°, 0/90° and ±45°) in various load conditions (tensile, compact tension and -compression, double cantilever beam and and four-point bend end notched flexure). A comparison was made between three clustering methods: k-means, a combination of a Kohonen's self-organizing map (SOM) and k-means and a competitive neural network (NN). The SOM and k-means combination turned out to have the best combination of quality and computational efficiency. Using this model; matrix cracking, fibre/matrix splitting and delamination could be identified. Fibre pull-out and fibre fracture were identified, but not consistently present (or absent) in certain tests, and therefore needed further study.

The group of Huguet et al. (2002) tried to eliminate as much damage mechanisms as possible by testing on pure matrix material, 90° off-axis to UD GFRP samples and 45° off-axis to the samples, in order to capture specific AE events belonging to their damage types. They gathered six components from each AE signal: rise time, counts, energy, duration, amplitude and counts to peak, of which amplitude and duration proved to be the main differentiating factors between damage types. Using an unsupervised clustering method (Kohonen's map), two different signals for matrix cavitation were identified. An "A-type" (55-70 dB) occurs in pure resin samples and continuously throughout the loading in the 90° off-axis specimen. These events were thought to have been caused by mode I matrix fracture. The second, "B-type", occurred after a certain damage level was reached. The researchers point that this had another mechanical origin, likely decohesion, since this occurred more frequently than A-type for tests on 45° off-axis loaded samples. The same group (Godin et al., 2004) showed that using a k-nearest neighbours (kNN) classifier proved to be successful as well, while it was easier to implement than the Kohonen's map. Furthermore, they attempted to identify a "C-type" event. Fibre breakage could be captured by loading a single fibre composite in conditions where the failure strain of the fibre was lower than that of the matrix. When however combining the data from their multiple experiments and training the kNN and Kohonen's map, there was a significant overlap between especially the B- and C-type clusters. This caused the kNN and Kohonen's map to classify respectively 10% and 5% as C-type signals in the 90° off-axis loaded UD specimen. This does not make physical sense, since fibre failure should not occur in these samples.

A damage mode which is not discussed in the previous papers, but which is present in composites, is delamination. McCrory et al. (2015) identify this mechanism, as well as matrix cracking in a CFRP plate under a buckling load. The authors use three classification methods. The first is an NN based on a SOM with k-means distance measure. The input data from AE events are counts, rise-time, duration, absolute energy, amplitude, and average- central and peak frequency. Unsupervised waveform clustering is the second, based on a principal components analysis (PCA) analysis of the shape of the waveforms. Measures were taken to overcome signal attenuation over distance. Finally, a method called the measured amplitude ratio calculates the in- and out-of-plane ratio of the damage, based on the zero-order longitudinal and transverse mode amplitudes of the caused Lamb waves. Using Delta T mapping (Baxter et al., 2007), the authors were also able to locate a large part of the damage correctly. On the other hand, the method is not fool-proof, since it also attributed a group of AE events to a location without noticeable damage. Delta T mapping seems particularly suited for more

complex structures, with 3 or more AE sensors to measure and locate AE events. In addition to this, it requires prior calibration. All three classification methods show coherence in the identification of AE events, as well as correctly locating most of them. It must be noted that the authors investigated just one plate. Therefore the determination of the number of clusters may be dependent on this specific plate, and all models are tested and trained on the same data.

In estimating RUL, Eleftheroglou and Loutas (2016) require a monotonically increasing parameter. They use a rolling window of 1500 load cycles, in which they sum rise-time over amplitude ratio's for all AE events in that window. The result shows a -generally- increasing trend. This allows a failure threshold to be set at a certain μs/dB, which occurs no earlier than 70% of each specimen's total cycles.

Finally, Zarouchas (2017) divided the loading- and unloading phase in tension-tension fatigue tests on CFRP and GFRP composites into four sections each. He finds that during one of the loading phases, the AE hits correlate strongly with the stiffness degradation of the specimen, therefore likely being matrix cracking. During unloading there are hits especially towards the end of tests, likely originating from internal friction in delaminated regions.

2.2.3. Wavelets

An AE signal can be decomposed into a sum of wavelets, oscillatory functions which are zero on average. This process is much like a Fourier transform. Every signal can be decomposed into a number of wavelets until their sum sufficiently represents it. This method allows for a joint time-frequency analysis (Loutas et al., 2006).

By applying both a continuous- and discrete wavelet transform, Marec et al. (2008) extract three new features specific to A- and B-type signals from the resulting wavelets. A PCA shows that clusters overlap much less with this extra information and that they, therefore, are able to discriminate better between A- and B-type damages.

Loutas et al. (2006) present a method to analyze the AE events using a discrete wavelet transform, based on a 'db20' wavelet. A decomposition to six levels is sufficient according to the authors, and they notice that the majority of the energy content is present in four levels. From the fact that one of these has an average energy content of roughly 50%, they derive that this must be related to the failure mode, which releases the most energy; fibre failure. Although the writers do not identify the other levels, they do point out the varying energy levels per waveform decomposition level. They tested on multiple GFRP samples, each with one AE sensor at a varying distance to a small hole, where the damage is initiated.

2.3. Prognostics

The field of prognostics is best described by Kim et al. (2016): *"Prognostics is to predict future damage/degradation and the RUL of in-service systems based on the measured damage data."* Prognostics is a relatively new field of research, with most research done in the last 30 years. This also causes the fact that there is no uniform way in which to perform prognostic predictions, as Uckun et al. (2008) mentions. This section covers the basics of prognostics, and the many differences there are within this field.

2.3.1. Model types and algorithms

Liao and Köttig (2014) describe three different model approaches for prognostics: experience-based, physics-based and data-driven. As already mentioned, the data-driven method will be used in this research.

The first is based on experience and knowledge of experts. These models mostly consist of IF-THEN rules and is much like how a human would solve a problem. The results are therefore easy to interpret. The downside to these kinds of models is that the model complexity explodes when more parameters are introduced. Furthermore, the performance of the model depends heavily on the ability of the expert to define a specific ruleset for the problem.

A physics-based model also relies on an understanding of the physical processes within the subject. It adjusts its mathematical model parameters based on obtained data. When the system is correctly described, physics-based models can outperform other model types. A downside to these models is the fact that a thorough physical understanding of the problem is needed. It is possible that a feature is not included, which leads to non-sensible results. Unless using a high-fidelity model, physical models generally require less computational power than data-driven models (Kim et al., 2016). Due to uncertainty, algorithms mostly define parameters as probability distributions instead of fixed values (Kim and Soni, 1984).

A data-driven model is purely based on historical data. This allows a model to be trained and give predictions based on extrapolation of this data, based on series and trends from other subjects. As long as there is a

sufficiently large set of training data available such that trends and behaviour can be identified and correctly extrapolated, these models are quick and easy to implement. Because the results are obtained from extrapolation, the inner workings of these models may be difficult to interpret (Kim et al., 2016).

In order to harvest the advantages of all approaches, a fourth, hybrid approach can be used (Liao and Köttig, 2014; Kim et al., 2016). Richardson et al. (2017) even advise not to use a purely data-driven model, should there be information available on the underlying process. In this way, no valuable knowledge is lost.

Next, Coble and Hines (2008) categorised prognostics algorithms in three types, according to which data is used:

Type I: Reliability data-based algorithms. These algorithms are purely based on historical reliability data and therefore give an estimate of an average subject under average environmental conditions. An example would be the fitting of a Weibull curve to a data-set.

Type II: Stress-based algorithms. Environmental conditions are now taken into account, but the algorithm still takes an average subject into account. An example of such an algorithm would, for example, be a proportional hazards model.

Type III: Effects-based algorithms. These algorithms use the response of the subject to the environmental conditions by tracking or identifying damage parameters. A general path model is an example of an effects-based algorithm.

While type III algorithms provide most accurate results in a test case with a large variation of failure times with little influence of environmental factors, Coble and Hines (2008) warn that enough data should be available in order to minimise the impact of noise.

Going from type I to III, the amount of information which is fed into models is increased. The complexity of the models is however also increased, as these features are increasingly harder to interpret. In the case of this research, AE data is used in two models, therefore classifying these as type III algorithms, since AEs are 'responses' of the subjects. Information can be contained in these responses, possibly telling something about the progression of damage. This makes the type III models, in principle, more able to give adaptive predictions, based on the current state of the subject. However, this also introduces uncertainty, since the data which is captured must reflect the degradation process well and consistently.

2.3.2. Failure thresholds

One could say that the end of life is simply the point at which a subject fails catastrophically. However, it could be argued whether it is desired to operate until such a point. First of all, testing until failure of coupons is not an issue. Doing this on, for example, wind turbine blades or aircraft engines is often unacceptable due to safety and cost reasons. This raises the prognostics paradox (Saxena et al., 2010); running to failure is avoided by maintenance, therefore the exact time to failure (TTF) is unknown, resulting in a guess for actual TTF which has to be used in training the prognostics algorithms. In the case of wanting to avoid failure, one should define the so-called minimum allowable prognostics horizon (Saxena et al., 2010), which is the amount of time needed to repair/replace a component. Secondly, is it desirable to test until catastrophic failure, or is a component already unable to fulfil its function before this point? Finally, as discussed above in section 2.1.1 and shown in section 4.4.2, there is a large degree of variability in both EOL and AE features between specimen. Therefore, each specimen likely fails after a different amount of cycles, while also its AE features at failure may be different.

This makes the definition of failure for a prognostic algorithm a choice which has to be made. In a prognostics algorithm, a threshold is often used to determine failure (Coble and Hines, 2008). These can be categorised as follows:

- Manual threshold;
- Based on engineering knowledge;
- Catastrophic failure.

When there is a predictable condition monitoring (CM) curve, a threshold can be set in order to determine the practical EOL, and therefore determine RUL for a specimen. Setting the threshold is a somewhat arbitrary process, based on data of other specimen or engineering experience/standards (Si et al., 2011). An example of the manual threshold is that in Eleftheroglou and Loutas (2016), who base the threshold on the rise time/amplitude

of AEs. Examples of thresholds based on engineering knowledge are from, for example, Wei et al. (2010), who define failure at a 40% stiffness drop. In another field, Richardson et al. (2017) set a limit at a certain percentage of battery capacity loss.

For these methods to be accurate, the prognostic feature will have to either be a feature on which a threshold based on engineering knowledge can be set such as stiffness loss, or it has to follow the degradation of such a value closely. These methods are, however, susceptible to outliers in terms of life-time or degradation behaviour. If a specimen performs differently than the 'status quo', a threshold can be set too conservatively or too optimistically, depending on whether it fails later or earlier than other specimens, respectively.

The final method is that of going to catastrophic failure; the type of failure which is undesired in applications, but which is possible to achieve in a lab setting. This is an easy threshold to set for existing data, but hard to predict on a specimen which is about to fail, since there are no direct features which lead up to this in the longer term.

Finally, as mentioned briefly before, (Liu et al., 2009) use a DI. They defined the DI at a measurement i as a normalizing function of the total time-series y (equation (2.1) (Liu et al., 2009)). While this does return a value which is in the range [0,1], the flaw of this method is that the minimum and maximum values of the time-series y have to be known at every point in time. Therefore, one must know the final value of the time-series. This value is often unknown in composite specimens, due to the large scatter in material properties. Whereas one specimen might, for example, fail after 10,000 cumulative acoustic emission counts, another may fail at 25,000. This final value, to which the DI is normalised, is not known during operation. Therefore, this method does not allow in-situ prognostics.

$$DI_i(y) = \frac{y_i - \min(y)}{\max(y) - \min(y)} \tag{2.1}$$

2.3.3. Prognostic performance metrics

There is not yet a standard method for prognostics (Uckun et al., 2008), which also results in the fact that there is not a standard metric to assess the quality of a prognostics algorithm (Saxena et al., 2009, 2010). Therefore, the two papers by Saxena et al. (2009, 2010) investigate which metrics could be used to judge and compare the performance of different prognostic algorithms. Common methods used were: "accuracy, precision, mean squared error (MSE), and mean absolute percentage error (MAPE)" (Saxena et al., 2009). These metrics, however, did not fully capture the essence of prognostics; the quality of predictions should increase towards the EOL, whereas these metrics were mainly measured at certain points only. They can be aggregated over the full life of a subject, but then an arbitrary weight must be included to take into account the fact that higher performance is required towards the EOL. Therefore, Saxena et al. (2010) introduces the following four metrics:

1. Prognostic horizon (from which point in time on is the EOL predicted with a desired accuracy, between bounds $\pm\alpha$, and is this time-span large enough to perform repairs or replacements?);
2. $\alpha - \lambda$ performance (what is the slope of the cone in which the accuracy increases towards EOL?);
3. Relative accuracy (what accuracy does the algorithm have at a certain point in time? For multiple points in time cumulative relative accuracy may be used, including a weight factor);
4. Convergence (does the performance converge, and if so, how fast?).

These metrics will have to be taken into account when trying to quantify the performance of the different models objectively.

2.4. Model selection

In order to select the models which shall be used in this research, the requirements for the models are first discussed. Then, from the initial literature study, three main model categories were distinguished. These are discussed here, together with both advantages and disadvantages of these models. Finally, the definitive choice of models for this research will be discussed.

2.4.1. Requirements

From the available literature on prognostics, and a preliminary analysis of especially the AE features in the available experimental data, four main requirements were drawn for the model selection. In order for a model to be used in this research it must:

1. be able to provide an in-situ regression, based on training data from other samples;
2. give probabilistic outputs;
3. be able to handle nonlinearities in training and prediction data;
4. be possible to implement within the timeframe of this research (9 months).

The first requirement relates to the training data. A model must be able to make predictions on RUL purely based on data which is available until the prediction point. It could be argued that the field of offline prognostics is not prognostics in its purest form, since this requires data from the future. This is therefore not possible to apply in practice.

Next, Uckun et al. (2008); Saxena et al. (2010) argue that uncertainty is a key element in prognostics. Not only should prognostic methods give accurate and precise estimates of RUL, but they should also give their confidence. Examples of sources of uncertainty, collected by Saxena et al. (2010), are listed below. Therefore, it is key to not only provide an estimation of the RUL but especially of the associated confidence.

- modeling uncertainties;
- measurement uncertainties;
- operating environment uncertainties;
- input data uncertainties.

Especially the AE data will be of nonlinear nature. Therefore, models must be able to handle this, trying to capture this behaviour. As Liao and Köttig (2014) point out, at the initiation of faults, health indicators are often suddenly very noisy, and these will therefore not show linear or stationary behaviour anymore.

Finally, it must be possible to implement a model within the timeframe of 9 months. Especially due to the goal to compare multiple models, they cannot become too complicated or computationally expensive. This therefore also makes this research a search for relatively simple models which can still fulfil this goal of making prognostic predictions on variable amplitude fatigue (VAF) data.

2.4.2. Linear models

Two types of linear models are covered; Gaussian processs (GPs) and the autoregressive moving average (ARMA) class. While there are many other linear models available, these two were found to be used in prognostics.

Gaussian processes

A GP is a stochastic process which can represent any possible function. This method is used by, for example, Liu et al. (2009), for offline prognostics on a carbon/epoxy beam. Two damage indices are compared, from both AE counts and energy content of low-frequency wavelet decompositions of Lamb waves. A major shortcoming is how the authors define the DI, which requires prior knowledge of the final state of a specimen, which is discussed above in section 2.3.2. Despite this, the authors note that the predictive capability of the in-situ GP regression improves during the life of a specimen. In another field, a GP regression is applied by Richardson et al. (2017) on in-situ battery capacity degradation data, based on data from other batteries. It must be noted that the cells were loaded under the same variable loading and conditions, resulting in similar degradation curves.

ARMA class

Liao and Köttig (2014) mentions two cases of an ARMA model being used in prognostics. In the prediction of RUL of elevator doors, an ARMA model was used to predict the trend of the failure probability. In another case, an ARMA model was used to regress between a measured health indicator (pump rotation speed) and a system degradation severity indicator. This allowed for prognostics of an engine fuel pumping unit in aircraft.

When comparing the group of autoregressive (AR), moving average and ARMA models to nonlinear models, there are both disadvantages as well as advantages according to Dorffner (1996). These models generally require less computational power than their nonlinear counterparts, are less prone to overfitting and do not have a learning phase which can lead to sub-optimal minima. However, the time-series has to -of course- be of a linear nature. Next to this, these models require that the time-series is stationary. In other words, both the mean and standard deviation cannot change with time. The fact that health indicators tend to become noisy when faults occur generally makes ARMA models less suited for long-term predictions (Liao and Köttig, 2014).

Some time-series can be made stationary by applying a difference operator Δ with order d. By checking the variation of the mean and standard deviation of the series $\Delta^d x(t)$, it can be determined if it is now stationary. The new model is called an integrated ARMA model, or ARIMA$[p,d,q]$ model.

An example of an autoregressive integrated moving average (ARIMA) model being used in prognostics is by Wu et al. (2007). This group aims to predict the RUL of a rotor test rig, from vibration data. A vibration severity measure is established from the data, and this measure is extrapolated using an ARIMA model to estimate when it reaches a certain threshold. The conclusion from this research is that the extended ARIMA model performs better than a regular ARIMA model. However, the authors did not include prediction intervals (PIs).

2.4.3. Neural networks

NNs are models which can theoretically model anything. The idea behind this set of models is taken from nature, based on neurons within a brain which communicate with each other. There are numerous types of NNs. Examples are a feedforward neural network (FFNN), Bayesian neural network (BNN), convolutional neural network (CNN), or recurrent neural network (RNN).

A FFNN is the most common type of NN. An input layer is connected to a number of hidden layers, where all nodes from the previous layer are connected to all nodes in the current layer. The final hidden layer is connected to the output layer. FFNNs can be used in time-series prediction by feeding it a number of most recent data-points. Based on this, the model can predict future values, making it an AR model (Dorffner, 1996). The nonlinear nature of NNs make them able to approximate any function (Dorffner, 1996). Having more than one hidden layer in a NN is often referred to as deep learning. Generally, NNs output single, deterministic values. They are trained by maximum likelihood estimation (MLE) of their parameters. To make the output probabilistic, a BNN can be used. This model category sets a probability density function (PDF) over its weights and biases, making the output a posterior. Yet, evaluating the output of the model is complicated, and with an increasing scale of the model it can become computationally too demanding (MacKay, 1992). An example of a FFNN in prognostics is by Arumugam et al. (2010), who use a FFNN with 10 neurons in each of the 5 hidden layers, in their prediction of failure load of a composite in tension (see section 2.2.1). However, the authors did not motivate their specific choice of FFNN architecture. Finally, FFNNs have proven to be able to model S-N curves by Al-Assaf and El Kadi (2001).

A CNN is a variant to the FFNN. Instead of each node being connected to every node at the adjacent layers, inputs are 'grouped' together and sparsely connected to the next layer. This type of network architecture is often used in image or sound processing, where there is a large array of input values in the order of thousands of nodes. Connecting all nodes of them to all nodes in a hidden layer would result in a very complex network, which is not only hard to train but is also prone to overfitting.

Finally, RNNs are specifically suited for time-series. What makes a RNN differ from other NNs is that nodal outputs are fed into the NN again. This allows the handling of temporal data, e.g. music or language processing. The difference from a AR model is that in this category, information from far in the past can still be used if the model is implemented correctly. An example of the usage of a RNN in prognostics is that by Heimes (2008), why use one to tackle the IEEE 2008 Prognostics and Health Management conference challenge problem. This problem consists of sensor data from 218 complex units, from an unknown initial state to failure. The goal is the provide an as accurate RUL prediction as possible. Their RNN was able to predict when a system was starting to fail, and had a consistent response towards failure, as opposed to a FFNN which they also tested.

The random initialization of NNs, just as the random choice of training and testing data, makes them suitable for probabilistic modelling (Kim et al., 2016). An approach where multiple models are trained could be used. With these models giving varying predictions on the data due to their varying initial conditions, the different

outcomes can form a probability distribution based on their relative frequencies. Another possibility is through bootstrapping, where different models are trained on random subsets of the data. This is complicated with temporal data however; data-points cannot just be left out. Complete series can be skipped in training, but it is decided to not use this method due to the low number of time-series in the training data.

2.4.4. Other models

In their overview, Si et al. (2011) discusses other statistical data-driven models, based on directly observable states as well as indirectly observable states. Three noteworthy models in the first category are Wiener processes (Brownian motions with drift), Gamma processes and Markov processes. A disadvantage of the first two is that they solely use degradation data of the specimen subjected to prognostics. There are more disadvantages, such as degradation state evolving based on the current state of a subject only (Weiner, Gamma, Markov), and requiring monotonic degradation data (Gamma). A general downside of these models is that they require a directly observable state, whereas in structural health monitoring of composites, this is often not available. Taking a hidden Markov model allows for modelling of indirectly observable states (Si et al., 2011; Eleftheroglou and Loutas, 2016). Still, there is the fact that the sojourn state time (the time spent in a (current) state) is exponentially distributed in a (hidden) Markov model, which is not always the case in engineering appliances (Si et al., 2011; Eleftheroglou and Loutas, 2016). In order to overcome this, Eleftheroglou and Loutas (2016) mention a hidden semi Markov model. Finally, they take an extended version of this; the non-homogeneous continuous time hidden semi Markov model (NHCTHSMM). This makes the state transitions now also dependent on the total age of the subject and the sojourn state time.

2.4.5. Choice of models

Due to the limited timeframe of this research, choices had to be made concerning which models would seem most feasible. Furthermore, they must also meet the set requirements above.

In order to see if the more complex models presented below are of any use, they are compared to a so-called baseline model as well. This is a relatively simple statistical model, based on a statistical analysis of the experimental data. For this analysis, the RUL for each specimen under VAF, is estimated using the failure distribution from all other specimens (both constant amplitude fatigue (CAF) and VAF). It is, therefore, a model based purely on failure data.

Models in the ARMA class did not meet all requirements above. No noteworthy research was found on multivariate input data, and the requirement of stationary input data in the ARMA class cannot be fulfilled by AE data without any severe modifications. Within the class of linear models, GP regressions seemed better suitable. This model would have to be implemented by extrapolating a set of features or a single feature based on itself and other specimens until it passes a threshold.

Within the class of NNs, the RNN seems most promising due to its ability to model temporal dependencies, in comparison to other types of NNs which would have to work in an AR manner. In this way, possibly more complex patterns can be picked up. An advantage to NNs is that they do not need to compared to a threshold of some kind, but can be trained based on for example the EOL time or degradation state of other specimens. In this research, it was decided to train the models on the failure index (FI), which linearly increases from 0 at the start of the fatigue test, to 1 when the specimen fails. The degradation state is then captured as a hidden state within the model.

In the category of the other models, the NHCTHSMM class seems most promising. However, due to its complexity, it is not possible to implement this within the timeframe of this thesis.

Finally, it must be stressed that the models must provide in-situ/online prognostic predictions. Having information from the future makes in-situ prognostics impossible. Therefore, it is key that when testing a model on a specimen, it must not have information after the testing point.

3
Methodology

In this chapter, the theory behind the three selected models, and their implementation will be discussed. First, the statistical model will be described, followed by the Gaussian process (GP) regression. The final model discussed is the recurrent neural network (RNN). Finally, a section dedicated on performance metrics will describe different metrics used, for quantitatively comparing the results of each of the three models.

3.1. Statistical model

The first model is the statistical model, which will primarily aid as a baseline in order to compare the other two models with. This section is divided into three parts. First, the appropriate statistical distributions to describe the failure times in the data-set are covered. Next, the methodology behind basic, static statistical distributions is discussed. Finally, a second set of statistical distributions is covered, one which is adapting during the testing phase, based on how long a specimen has survived already.

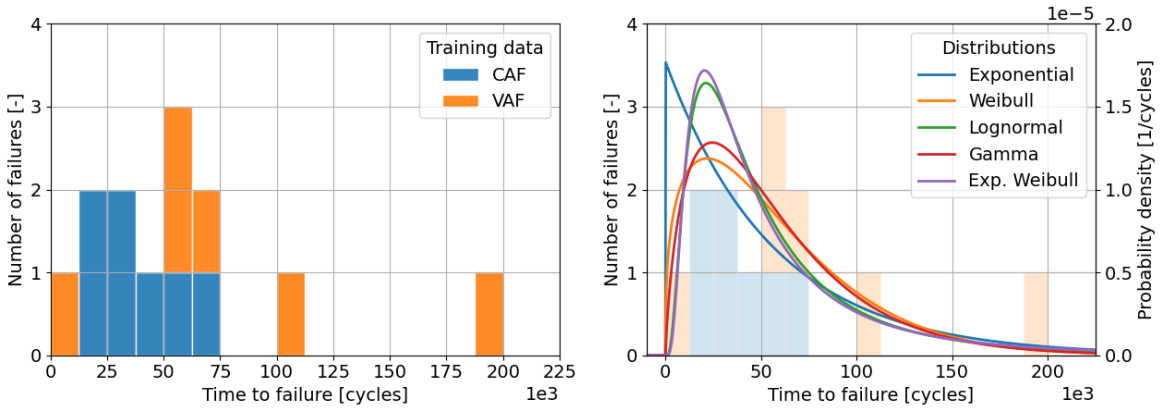
3.1.1. Determining the statistical distribution

Figure 3.1a shows the time to failure (TTF) of all specimens in a histogram. The first feature which stands out is that this is not a symmetrical distribution. The majority of the specimens lies around 50,000 cycles, with a tail towards higher numbers of cycles. Furthermore, from a physical perspective, the probability density function (PDF) cannot be non-zero for cycles lower than zero; a negative number of cycles does not have a physical meaning. These two statements likely eliminate the possibility for symmetrical distributions (e.g. normal, logistic). The last statement also rules out the class of extreme value (EV) distributions. In these, the smallest extreme value (SEV) distribution suggests a high concentration of failures after a certain point in life. The largest extreme value (LEV) distribution is more suitable since it is left-skewed. Both the SEV and LEV distributions, however, give non-zero PDF values at cycles ≤ 0. (Meeker and Escobar, 1998)

When comparing non-symmetrical distributions, there are many options. The most common ones are now briefly discussed. It is expected that the failures follow a Weibull distribution, a PDF which is 0 at 0 cycles, is left-skewed and often used in failure time analysis. Another common model for this is, for example, the lognormal distribution, which can take roughly the same shape as the Weibull. The exponential distribution is another type of non-symmetrical distribution. However, this distribution is usually unfit for modelling the life of components under fatigue. Another possibility is that of the gamma distribution, of which the failure rate converges to a constant value at the end of life. This makes it suitable for some specific failure time analyses. (Meeker and Escobar, 1998)

The distributions mentioned here are fitted to the data and shown below in figure 3.1b. Another Weibull distribution is modelled as well; the exponentiated Weibull. This is a Weibull distribution whose cumulative distribution function (CDF) is raised to a power, allowing for a more flexible type of distribution. Especially the (exponentiated) Weibull, lognormal and gamma distributions seem to model the data correctly.

In order to confirm these beliefs, the Kolmogorov–Smirnov (KS) test is used. It is implemented in SciPy. This test is a goodness-of-fit test, which compares two distributions to each other and returns a test-statistic D and p-value. D is the value which describes the difference between the two distributions. The null hypothesis is that the two distributions are equal, and it is rejected if D is larger than a critical value. This is described



(a) Distribution of failure times

(b) Fits of different PDFs

Figure 3.1: Distribution of failure times and their fitting distributions, of all carbon fibre reinforced polymer (CFRP) specimens

by the p-value; which should therefore be above a certain threshold in order not to have the null-hypothesis rejected.

The results of the KS test are shown below in table 3.1. The test shows that in fact, the gamma distribution is the most suitable for describing the failure time distribution. Due to the small sample size, the test is only performed on all specimens. Therefore, no different distribution types will be taken when looking at subsets of the data. Within all subsets discussed, the p-values for the gamma distribution were larger than 0.39. Therefore there was no need to reject the hypothesis; something which is commonly done when $p < 0.05$.

Table 3.1: KS test results for lifetime distributions on the CFRP data-set, sorted by goodness-of-fit

Distribution	p-value [-]	D [-]
Gamma	0.879	0.152
Weibull	0.807	0.167
Exponential	0.783	0.171
Lognormal	0.693	0.186
Exponentiated Weibull	0.672	0.189
Normal	0.351	0.246

The PDF at t cycles for the gamma distribution is shown in equation (3.1). This is with the location parameter set to 0, as this is under the assumption that there is an immediate probability of failure in the first cycle. Its shape and scale parameters, k and θ, are found by maximum likelihood estimation (MLE). $\Gamma(k)$ denotes the gamma function of k. Its CDF is presented in equation (3.2), where $\gamma(k, t/\theta)$ is the lower incomplete gamma function. Both the gamma- and incomplete gamma function can be found in appendix A.1. (Meeker and Escobar, 1998)

$$f_T(t) = \frac{t^{k-1} \exp[-t/\theta]}{\theta^k \Gamma(k)} \tag{3.1}$$

$$F_T(t) = \frac{\gamma(k, t/\theta)}{\Gamma(k)} \tag{3.2}$$

3.1.2. Static predictions

The first set of predictions is based on failure times of all specimens except for the test specimen. In this way, the model does not have any prior knowledge of the shape and scale parameters of the distribution of the test specimen. It does, however, from the analysis above, have knowledge of the type of distribution.

The CDF of the distribution can be used to obtain a confidence of the predictions, prediction intervals (PIs). To obtain a $(1-\alpha)$ confidence level, the lower- and upper PI bounds are calculated by solving equation (3.3). This is done numerically. The same is done to obtain the median, by now solving for 0.5.

$$\big(F_T(t) = \alpha/2, \quad F_T(t) = 1-\alpha/2\big) \tag{3.3}$$

With a prediction for the failure time, the remaining useful life (RUL) at t cycles can be calculated by subtracting the passed time. The same goes for the lower- and upper PI bounds.

3.1.3. Adapting predictions

The statistical model is taken one step further, to where it adapts during the testing phase. If after s cycles the specimen has survived (meaning $\mathrm{p}\big(S=s\big)$), then the failure distribution can be updated with this knowledge to $f_{T|S=s}(t)$. This is done using Bayes' theorem in equation (3.4).

Two terms in this equation can be simplified. First of all, the probability that the specimen has survived, given the time of failure is simple; the specimen survives if $t > s$. Therefore, this term can be written as the Iverson bracket $f_{T|S=s}(t) = [t > s]$, returning 1 if the condition is satisfied, and 0 if this is not the case. Secondly, the denominator, is simply the survival function of the PDF, evaluated at s; $\mathrm{p}\big(S=s\big) = 1 - F_T(s)$. These two operations simplify equation (3.4) to equation (3.5):

$$f_{T|S=s}(t) = \frac{\mathrm{p}\big(S=s \,\big|\, T=t\big) f_T(t)}{\mathrm{p}\big(S=s\big)} \tag{3.4}$$

$$= \frac{[t>s] f_T(t)}{1-F_T(s)} \tag{3.5}$$

All terms except $f_T(t)$ are constants, making the integration of equation (3.5) a quick process of taking the CDF of $f_T(t)$. This results in the CDF $F_{T|S=s}(t)$:

$$F_{T|S=s}(t) = \frac{[t>s]\big(F_T(t) - F_T(s)\big)}{1-F_T(s)} \tag{3.6}$$

With the obtained PDF and CDF, the same methods as above can be used to obtain the median failure times and PIs. The RUL is then calculated in the same manner as well.

3.2. Gaussian process regression

This section discusses the GP regression model. It starts off with a description of GPs, followed by kernel functions used in these models. Next, the parameterisation is covered, together with possible issues. This is followed by a section on making the predictions, where a threshold will need to be set to determine RUL from the extrapolated GP predictions. Next, a new method of possibly improving the setting of a threshold is discussed; the adjustment of the threshold PDF according to the correlation between time-series in the data. Finally, the implementation of this model category is discussed.

3.2.1. Description

GPs are supervised methods, which can be used for both regression and classification. For the purpose of this thesis, regression will be used. Although sometimes regarded as common knowledge, the theory below is based on the works of Rasmussen and Williams (2006); Murphy (2012). The nature of GPs is that they are probabilistic; each prediction will come with a certain confidence.

Because a GP regression is a supervised method, there is a training set (available data) $\mathcal{D}$, consisting of N data-points. At each point, there are inputs and an output. The inputs $\boldsymbol{x}_i$ are D-dimensional, and a value y_i is observed. In order to perform a regression, two approaches can be taken. The first one, a regular regression, takes a function $f\left(\boldsymbol{y},\boldsymbol{\theta}\right)$, and infers a distribution of its function parameters $\mathrm{p}\left(\boldsymbol{\theta}\,\middle|\,\mathcal{D}\right)$. This method can, however, underperform when the wrong function is chosen to represent the data. A GP regression on the other hand infers a distribution of functions $\mathrm{p}\left(f\,\middle|\,\mathcal{D}\right)$. It is essentially a distribution of possible functions which agree with the training set. The prior of the regression function, a GP, is written as in equation (3.7).

$$f\left(\boldsymbol{x}\right) \sim \mathcal{GP}\left(m\left(\boldsymbol{x}\right), \kappa\left(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}}\right)\right) \tag{3.7}$$

The prior follows a mean $m\left(\boldsymbol{x}\right)$ and has a covariance- or kernel function $\kappa\left(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}}\right)$, which are defined as follows:

$$m\left(\boldsymbol{x}\right) = E\left[f\left(\boldsymbol{x}\right)\right] \tag{3.8}$$

$$\kappa(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{T}}) = E\left[\left(f(\boldsymbol{x}) - m\left(\boldsymbol{x}\right)\right)\left(f\left(\boldsymbol{x}^{\mathrm{T}}\right) - m\left(\boldsymbol{x}^{\mathrm{T}}\right)\right)\right] \tag{3.9}$$

The function evaluations at different points $\boldsymbol{f} = \left[f\left(\boldsymbol{x}_1\right), ..., f\left(\boldsymbol{x}_N\right)\right]^{\mathrm{T}}$ are assumed to be jointly Gaussian, with the prior in equation (3.10). In other words, function values are all dependent on each other through a multivariate Gaussian distribution, with mean vector $\boldsymbol{\mu} = \left[m\left(\boldsymbol{x_1}\right), ..., m\left(\boldsymbol{x_N}\right)\right]^{\mathrm{T}}$ and covariance matrix $\boldsymbol{K}$, constructed using positive definite kernel functions: $K_{ij} = \kappa\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$. This is therefore a discrete process. Several kernel functions will be discussed below in section 3.2.2.

$$\mathrm{p}\left(\boldsymbol{f}\,\middle|\,\boldsymbol{X}\right) = \mathcal{N}\left(\boldsymbol{f}\,\middle|\,\boldsymbol{\mu}, \boldsymbol{K}\right) \tag{3.10}$$

When a prediction has to be made on a set of points $\boldsymbol{X}_*$, sized $N_* \times D$, the function values at these points $f\left(\boldsymbol{X}_*\right)$, written as $\boldsymbol{f}_*$, behave according to the joint distribution in equation (3.11). Due to the flexibility of GPs, it is common to ignore the mean function and set it to 0, since the mean function can still be modelled well (Murphy, 2012; Rasmussen and Williams, 2006). This will be done from this point on. This simplifies equation (3.11) to equation (3.12).

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^{\mathrm{T}} & \boldsymbol{K}_{**} \end{bmatrix}\right) \tag{3.11}$$

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K}_* \\ \boldsymbol{K}_*^{\mathrm{T}} & \boldsymbol{K}_{**} \end{bmatrix}\right) \tag{3.12}$$

The different covariance matrices are constructed from the kernel functions by: $\boldsymbol{K} = \kappa\left(\boldsymbol{X}, \boldsymbol{X}\right)$, $\boldsymbol{K}_* = \kappa\left(\boldsymbol{X}, \boldsymbol{X}_*\right)$ and $\boldsymbol{K}_{**} = \kappa\left(\boldsymbol{X}_*, \boldsymbol{X}_*\right)$. The matrices are dimensioned respectively as $N \times N$, $N \times N_*$, and $N_* \times N_*$. Now when

wanting to predict the distribution of function values at $\boldsymbol{X}_*$, the prior from equation (3.10) is conditioned on $\boldsymbol{X}_*$:

$$\mathrm{p}\left(\boldsymbol{f} \mid \boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}\right) = \mathcal{N}\left(\boldsymbol{f}_* \mid \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right) \tag{3.13}$$

From this Gaussian distribution, the mean and variance of the new set of points can be determined by the rules for conditioning Gaussians. The full proof behind this can be found in, for example, Murphy (2012).

$$\boldsymbol{\mu}_* = \boldsymbol{K}_*^{\mathrm{T}} \boldsymbol{K}^{-1} \boldsymbol{f} \tag{3.14}$$

$$\boldsymbol{\Sigma}_* = \boldsymbol{K}_{**} - \boldsymbol{K}_*^{\mathrm{T}} \boldsymbol{K}^{-1} \boldsymbol{K}_* \tag{3.15}$$

First of all, it should be noted that the assumption that $\boldsymbol{\mu} = 0$ is already incorporated in equations (3.14) and (3.15). Secondly, the above method assumes that there is no noise surrounding the available data $\boldsymbol{f}$. This model interpolates the training data exactly, fitting through the available data-points; $y_i = f\left(\boldsymbol{x}_i\right)$. This assumption, however, is not always correct. Training data can be noisy, and having to fit the model exactly through each point might not be realistic. Furthermore, the inversion of $\boldsymbol{K}$ may also give numerical issues in some cases. An example is where there is noise, and two data-points $\boldsymbol{x}$ lie in the same position. The resulting covariance matrix will be singular in this case, and cannot be inverted. Therefore, it is common practice to allow for noise in the training data by adding a white noise kernel to the model.

3.2.2. Kernel functions

There are multiple kernel functions which can be used. The choice of kernel function is the single factor which influences the predictive performance of a GP (Murphy, 2012). Different kernel functions and their parameters will be discussed in this section. Two different distance measures can be used; the Euclidean distance and Mahalanobis distance. These are elaborated upon in appendix A.2.

Besides the four kernel types listed below, there is a large variety of other kernel functions. They are either not commonly found in literature concerning this field of research, or not applicable to the problem in this thesis. An example of this is the periodic kernel since no periodic behaviour is present in the time-series used for the GP regression model. For more kernel functions, Rasmussen and Williams (2006) provide an excellent overview.

White noise kernel

The first kernel function is the most basic and is already mentioned briefly before. By using this kernel, it is assumed that the available data-points are noisy. In this way, a measured point y_i is related to not only the true function, but also a independent and identically distributed (i.i.d.) normally distributed error term $\epsilon \sim \mathcal{N}\left(0, \sigma_y^2\right)$, thus $y_i = f\left(\boldsymbol{x}_i\right) + \epsilon$. This gives the GP a bit more flexibility, in the sense that it does not have to pass through the available data-points exactly.

The white noise kernel is shown in equation (3.16). In this equation, δ_{pq} depicts the Kronecker delta function, such that the noise σ_y^2 is added to the diagonal of the covariance matrix only. The covariance matrix is therefore simply constructed by the multiplication of the identity matrix $\boldsymbol{I}$ and the white noise σ_y^2, as in equation (3.17).

$$\kappa_{WN}\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right) = \sigma_y^2 \delta_{pq} \tag{3.16}$$

$$\boldsymbol{K}_{WN} = \sigma_y^2 \boldsymbol{I} \tag{3.17}$$

In literature, the noise kernel is often not explicitly used. The covariance matrix of the noisy observations is then constructed by adding the white noise, defined as $\boldsymbol{K}_y \triangleq \boldsymbol{K} + \sigma_y^2 \boldsymbol{I}$. From this notation, it is easy to overlook the fact that the noise also has to be added to $\boldsymbol{K}_{**}$, since this matrix's diagonal terms also satisfy the Kronecker delta function.

With a covariance matrix of sole zeros except for the diagonal; the process is simply white noise with standard deviation σ_f. A sample drawn from this process is shown in figure 3.2. There is indeed no covariance between the different points.

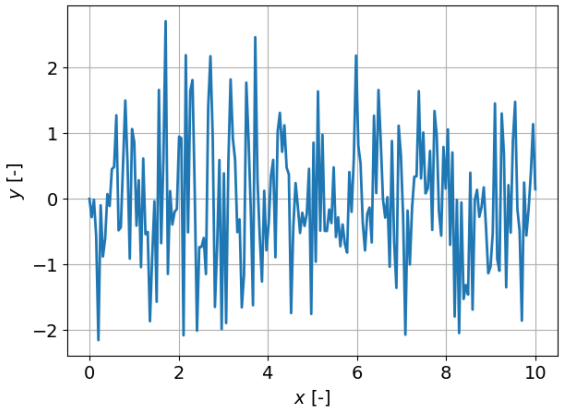


Figure 3.2: A sample drawn from a GP constructed with a white noise kernel ($\sigma_f^2 = 1$)

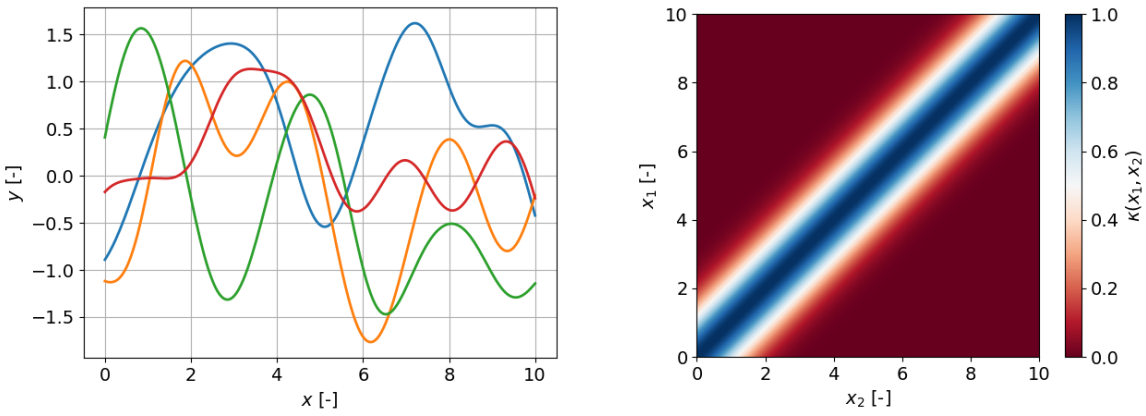
Squared exponential kernel

The squared exponential (SE) kernel is commonly shown in the literature. The hyperparameters of this kernel (equation (3.18)) are the length scale l, and function variance σ_f^2. The former dictates the smoothness of the function, while the latter determines its scale. The function uses the distance d between two points $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ as input. In case the Mahalanobis distance is used instead of the Euclidean distance l^{-2} can be captured in the covariance matrix $\boldsymbol{\Sigma}_M$. As a consequence, the entries in this matrix are an additional set of hyperparameters which must be optimised for. A way of rewriting the matrix as a set of parameters is by using the spherical representation, which is explained in detail in appendix A.3.

$$\kappa\left(d\right) = \sigma_f^2 \exp\left(-\frac{d^2}{2l^2}\right) \tag{3.18}$$

An issue of the SE function is that it is infinitely smooth. This is generally not applicable to many physical processes, and therefore not recommended to use. Instead, the Matérn class (below) is recommended. (Rasmussen and Williams, 2006)

An example of a kernel is used to draw a few function realisations, which are shown in figure 3.3a. The covariance matrix is shown in figure 3.3b. It can be seen that the function is stationary due to the linear ridge in the covariance matrix; points which are close to each other have higher covariance than those further away, and covariance is solely dependent on this distance between points.



(a) Four samples drawn from GPs constructed with a SE kernel

(b) Covariance matrix

Figure 3.3: Characteristics of a SE kernel. The kernel is set with $\sigma_f^2 = 1$, $l = 1$.

Matérn class

There is no single Matérn kernel; it is a full class of kernels. The Matérn class is often used in GP regression models, and shown below in equation (3.19) (Murphy, 2012). The length scale ($l > 0$) and function variance can be spotted again. K_ν is a modified Bessel function, with $\nu > 0$.

$$\kappa_{Ma}(d) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} d}{l} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} d}{l} \right) \tag{3.19}$$

When $\nu \to \infty$, the SE kernel function is obtained (Rasmussen and Williams, 2006; Murphy, 2012). For half-integer values larger than 0, the function simplifies significantly. The most interesting cases for machine learning are possibly $\nu = 3/2$ and $\nu = 5/2$, stated by Rasmussen and Williams (2006). These Matérn kernels are often depicted as Ma3- and Ma5 kernels (equations (3.20) and (3.21)). With these values for ν, the Bessel function disappears, and the remaining function is that of the product of a polynomial and an exponential function. Just as in the SE kernel, the length scale can be captured in $\mathbf{\Sigma}_M$ when using the Mahalanobis distance. The hyperparameters are then σ_f^2 and $\mathbf{\Sigma}_M$.

$$\kappa_{Ma3}(d) = \sigma_f^2 \left(1 + \frac{\sqrt{3} d}{l} \right) \exp\left(- \frac{\sqrt{3} d}{l} \right) \tag{3.20}$$

$$\kappa_{Ma5}(d) = \sigma_f^2 \left(1 + \frac{\sqrt{5} d}{l} + \frac{5 r^2}{3 l^2} \right) \exp\left(- \frac{\sqrt{5} d}{l} \right) \tag{3.21}$$

The difference in the smoothness of these processes, as compared to those from the SE kernel, can be immediately noticed in figures 3.4 and 3.5. Both covariance matrices again show the stationarity of the functions; being only dependent on the distance between coordinates. Because they seem similar to that of the SE kernel, they are not shown here.

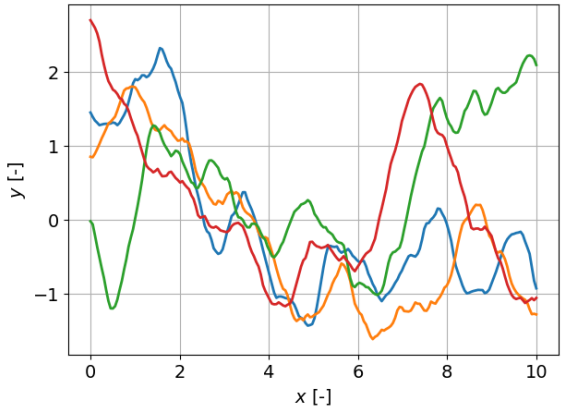


Figure 3.4: Four samples drawn from GPs constructed with a Ma3 kernel. The kernel is set with $\sigma_f^2 = 1$, $l = 1$.

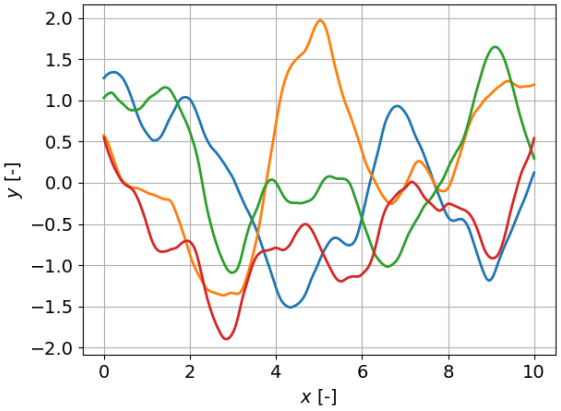


Figure 3.5: Four samples drawn from GPs constructed with a Ma5 kernel. The kernel is set with $\sigma_f^2 = 1$, $l = 1$.

The covariance matrices of the functions above seem similar to that of the SE kernel. In order to spot differences, the covariance function and its dependence on d is shown in figure 3.6. What can be seen is that the Matérn class kernels are steeper near the centre of the function. This explains the higher roughness for the Matérn kernels. Furthermore, the Ma5 kernel is more similar to the SE kernel than the Ma3 kernel. This is consistent with the fact that when $\nu \to \infty$, the SE is obtained for the Matérn kernels; ν is higher in the Ma5 kernel than in the Ma3 kernel.

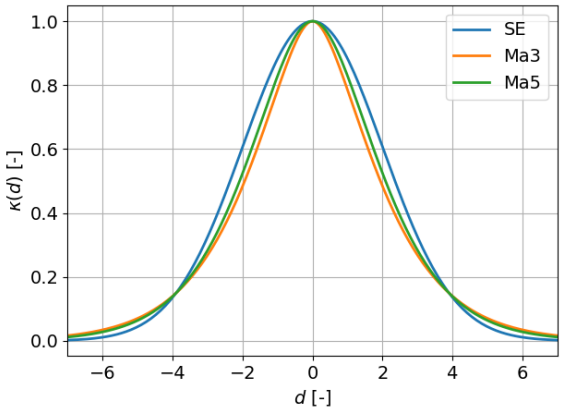


Figure 3.6: Covariance functions as a function of distance from 0. For all functions, $\sigma_f^2 = 1$ and $l = 1$

.

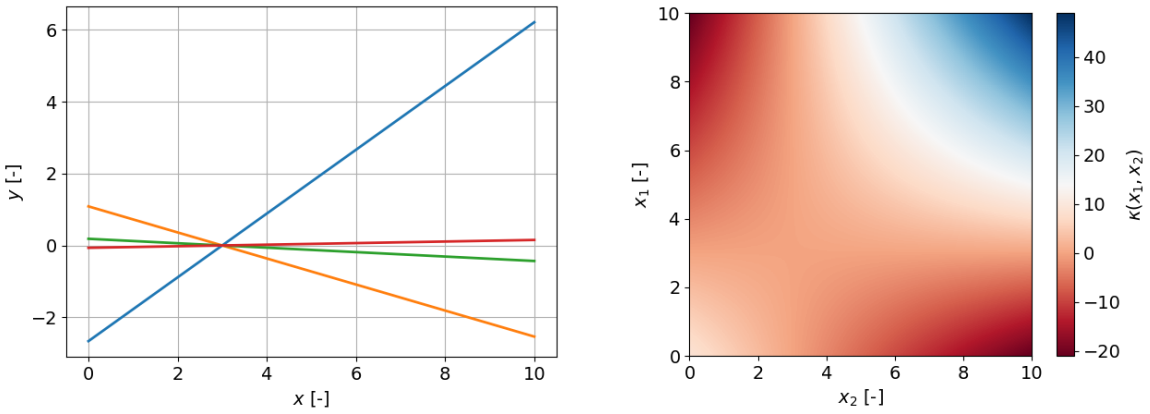
Linear kernel

The final kernel type is a linear kernel (equation (3.22)). While the SE and Matérn type kernels are stationary, this kernel is not. It is dependent on the locations of the two points x_p and x_q. Were the data-points to move, the model would produce different results, in contrast to the kernels mentioned above. One hyperparameter is present again in this kernel; the function variance again scales this kernel's output. The vector $\boldsymbol{c}$ determines the x-intercept of the prediction.

$$\kappa_{lin}\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right) = \sigma_f^2(\boldsymbol{x}_p - \boldsymbol{c})^{\mathrm{T}}(\boldsymbol{x}_q - \boldsymbol{c}) \tag{3.22}$$

Figure 3.7a shows a few samples from a linear kernel. The influence of c can be spotted in the drawn samples. The covariance matrix in figure 3.7b shows that this kernel is non-stationary; the covariance differs based on the magnitude of the points instead of on their difference.

When a GP regression would be based solely around this kernel, it would be wiser to use a Bayesian linear regression, since this is a more efficient method (Duvenaud, 2014).



(a) Four samples drawn from GPs constructed with a linear kernel

(b) Covariance matrix

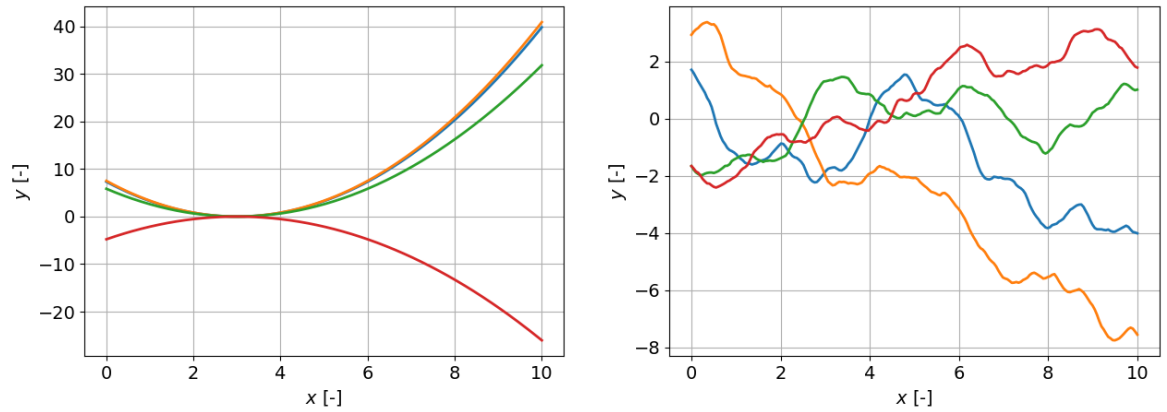
Figure 3.7: Characteristics of a linear kernel. The kernel is set with $\sigma_f^2 = 1$, $c = 3$.

Combining kernels

It is possible to combine kernels, in order to build a model which can predict more complex structures which are not possible to model using a single kernel function. Kernels can be added, as well as multiplied with other kernels. (Duvenaud, 2014)

This fact can be used in order to model more complex GPs, with for example (exponentially) increasing means. When multiplying a linear kernel with another linear kernel, an exponential process is obtained. Taking the kernel from figure 3.7, and multiplying it with itself yields the curves in figure 3.8a. A linear kernel can also be used to model a linear trend behind another process, such as in figure 3.8b.

The shape of the eventual function depends on the hyperparameters of the individual kernels. Not only their shape can be altered, but also their relative share in the final function. When adding two kernels and giving the first a 100 times lower σ_f^2 than the other, for example, the shape from the second kernel will be much more present in the process.



(a) Multiplication of two linear kernels

(b) Addition of a linear and Ma3 kernel

Figure 3.8: Samples drawn from different kernel combinations. The kernels use $c = 3$, $\sigma_f^2 = 1$, $l = 1$.

There is an endless list of possible kernel combinations, and there are different methods which can test different combinations. An example is that from the work of Duvenaud (2014), who uses a tree-like method. First, single functions are tested. The best performing kernel will then be combined (by addition and multiplication) with all kernels. Again, the best combination will be picked, and this will be repeated until accepted model performance is reached, or when a certain depth is reached. Another possibility is constructing a multi-layered neural network (NN) from kernel functions, used by Sun et al. (2018). The weights which connect the neurons, which are in this case kernel functions, are optimised for optimal representation of the process in the training data.

While these methods may give an optimal combination of kernels, they are also complex and harder to implement. Another drawback is the fact that when increasing the number of kernels, the number of hyperparameters also increases in the model. All these parameters have to be optimised, where an increase in the number of parameters yields an increase in required computation time. Therefore it was decided not to perform a complex, deep search for kernels. Instead, an approach was taken with the knowledge of the data. The decision was made to model the cumulative energy, as discussed below in section 3.2.6. From the data in figure 3.13, the following observations can be made:

1. The cumulative energy time-series is not smooth;
2. The cumulative energy is a monotonically increasing time-series.

A linear kernel on its own would be able to perform a regression along the data, but would not be able to follow the time-series exactly. Its predictions would therefore have high uncertainty. Furthermore, a Bayesian linear regression would be a more efficient method, as stated above. Based on observation 1, the SE kernel is also deemed unfit to model the time-series. Therefore, the Ma3- and Ma5 kernels remain options for modelling the time-series.

A GP based on one of these two kernels would however always end up at its mean, which was set at zero. Far away from the training data, the model would therefore predict zero cumulative energy. This would clash

with observation 2. In order to overcome this, a combination by addition of a linear- and Ma3- or Ma5 kernel is therefore proposed. The linear kernel can be used to model the long-term, increasing trend, while the Ma3- or Ma5 kernel can pick up on smaller changes in the time-series. Addition is preferred over multiplication, since the variance in the data is not expected to increase with the number of cycles.

In order to allow for noise in the data and make the model more flexible regarding the existing data, a white noise kernel is finally added. Therefore, the covariance matrix will be built up from a Ma3 or Ma5 kernel, a linear kernel, and a white noise kernel. The difference in performance between the Ma3 and Ma5 kernels will be discussed in section 5.2.1.

Multiple time-series

The final set of operations using kernels is the product correlation rule. This rule enables the full potential of GPs in this thesis; the ability to model multiple time-series. Using a GP regression on L different time-series, requires multiple outputs for the model, which are presumably related to each other. A simple way to deal with this is to label the L separate time-series and treat them as separate inputs. The labels $\boldsymbol{l} = [1, 2, ..., L]$ are now used as additional dimension for the input; $\boldsymbol{x}_i$ in the l_i^{th} time-series would now be treated as a $D+1$-dimensional point $\boldsymbol{x}_i^{(l)} = [\boldsymbol{x}_i, l_i]$. This approach does not require that data-points need to be in the exact same points in time for the different series. (Osborne, 2010)

Instead of having to use $\boldsymbol{x}_i^{(l)}$ as an input for a (set of) kernels, the product correlation rule decomposes the covariance function into the multiplication of the covariance of the inputs $\kappa^{(x)}$ and that of the labels $\kappa^{(l)}$, as in equation (3.23).

$$\kappa\left(\boldsymbol{x}_{\boldsymbol{p}}^{(l)}, \boldsymbol{x}_{\boldsymbol{q}}^{(l)}\right) = \kappa^{(x)}\left(\boldsymbol{x}_{\boldsymbol{p}}, \boldsymbol{x}_{\boldsymbol{q}}\right)\kappa^{(l)}\left(l_p, l_q\right) \tag{3.23}$$

The covariance between labels cannot be modelled using a kernel function; since the order of labelling would possibly influence the result. Because there are just 13 specimens used in the CFRP data-set, $\kappa^{(l)}\left(l_p, l_q\right)$ can also be represented by an $L \times L$ covariance matrix $\boldsymbol{\Sigma}_L$. This covariance matrix can be represented by a vector of parameters using the spherical representation (appendix A.3). This results in $L/2(L+1)$ parameters which have to be quantified in order to determine the covariance between all time-series.

3.2.3. Parameterisation

With hyperparameters in kernel functions and in the label covariance matrix, there are multiple parameters to be set in order accurately regress on the data. This is done by maximising the marginal likelihood of the observations $\boldsymbol{y}$, given the inputs; $\mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$. The log of the marginal likelihood for GPs is given in equation (3.24) (Rasmussen and Williams, 2006).

$$\log \mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{X}\right) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}_y^{-1}\boldsymbol{y} - \frac{1}{2}\log\left(\det\left(\boldsymbol{K}_{\boldsymbol{y}}\right)\right) - \frac{N}{2}\log\left(2\pi\right) \tag{3.24}$$

In the equation, the first term accounts for the data fit, while the second accounts for the complexity of the model. These two terms therefore perform the bias-variance trade-off, a classic problem in machine learning (ML). This avoids over-fitting of the data by making the model too complex. The final term is a constant, depending on the number of samples. This is not influenced by setting the hyperparameters, of course.

A faster way of computing $\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$ is by taking the Cholesky factorisation $\boldsymbol{L}$ of $\boldsymbol{K}_y$, where $\boldsymbol{K}_y = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ (Rasmussen and Williams, 2006). The term $\boldsymbol{K}_y^{-1}\boldsymbol{y}$ can be replaced by a vector $\boldsymbol{\alpha}$ (equation (3.25)). Another advantage is that the determinant of $\boldsymbol{K}_y$ can be computed faster if $\boldsymbol{K}_y$ is decomposed. The determinant of $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ is equal to the squared product of the diagonal entries; $\prod_{i=1}^{N} L_{ii}^2$. The log of this is then twice the sum of the diagonal entries. These operations simplify equation (3.24) to equation (3.26).

$$\boldsymbol{\alpha} = \left(\boldsymbol{L}^{-1}\right)^{\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{y} \tag{3.25}$$

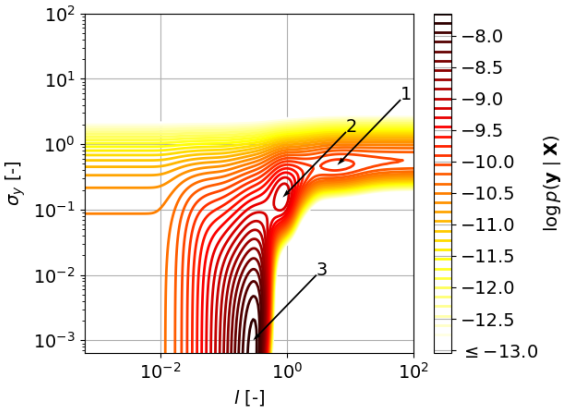
$$\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha} - \sum_{i=1}^{N}\log\left(L_{ii}\right) - \frac{N}{2}\log\left(2\pi\right) \tag{3.26}$$

Despite this optimisation, the computational complexity is still $\mathcal{O}\left(N^3\right)$ for the Cholesky factorisation, and $\mathcal{O}\left(N^2\right)$ for computing its inverse and solving equation (3.25) (Rasmussen and Williams, 2006). The computa-

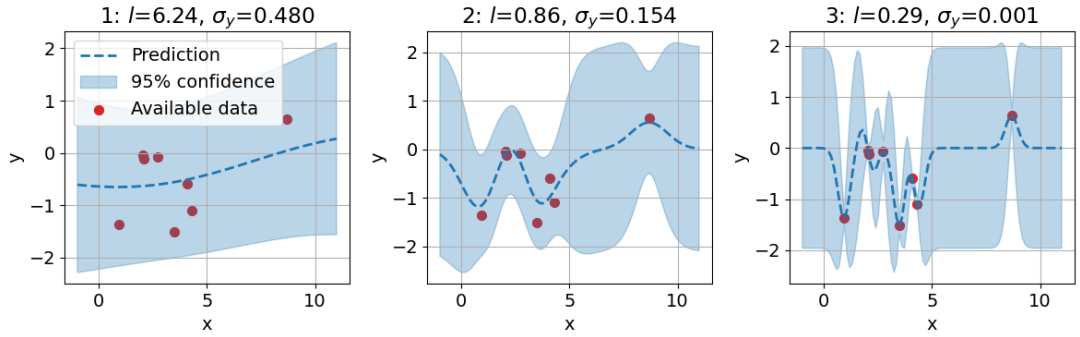
tional cost therefore increases exponentially and can become a problem for large data-sets. In fact, an optimisation for hyperparameters can take up to one core-hour for a model with 1-dimensional input data and 13 different time-series, which is an actual use-case for this thesis.

Another issue is that depending on the process modelled, there can be local optima in $\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$. This is shown in figure 3.9, where data was generated using a GP. Values for $\boldsymbol{X}$ were randomly taken between 0 and 10. Three local optima were found. The global optimum lies at point 3, however, this prediction does not seem to be the optimal prediction; prediction 2 lies much closer to the hyperparameters of the original GP.

When relating this to the bias-variance trade-off, it can be seen how these two interact; with increasing length scales the model complexity decreases, but therefore the noise has to grow with it. If more data-points were taken, the shape of the process would become more apparent, and eventually, there will be a single optimum in $\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$.



(a) Grid search for $-\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$



(b) Different optimal models

Figure 3.9: Three local optima arise in a search for optimal parameters for a GP with SE kernel with σ_f set at 1. $\boldsymbol{y}$ was drawn from a GP with SE kernel, white noise and parameters $\sigma_f = 1, l = 1, \sigma_y = 0.1$. Inspired by figure 5.5 of Rasmussen and Williams (2006).

A grid search could be performed over all possible combinations of hyperparameters in order to overcome the issue of landing at local optima. This method would, of course, become very expensive for higher numbers of hyperparameters. Therefore, due to limited computational power and time, another approach is taken. A model is initialised with random hyperparameters. From a random initial position, the model will converge towards a nearby optimum. This may be the global optimum or a local one. The hyperparameters will be saved, as well as $\log p\left(\boldsymbol{y} \mid \boldsymbol{X}\right)$ at this point. This process of initialisation and optimisation is repeated 20 times, trying to find the global optimum in at least one of the tries. This is an arbitrary number of tries, primarily based on the time the Danmarks Tekniske Universitet (DTU) high performance computing (HPC) cluster would take to perform a full run on one of the data-sets. This time is in the order of days, depending on which research question is answered. From the results, the predictions from the optimal sets of hyperparameters generally showed good accordance with both the training- and testing data, although there were still cases where local optima led to improper predictions. This led to the belief that 20 times would be a suitable number to find the best possible sets of hyperparameters in this context.

3.2.4. Probability of failure and remaining useful life

With a prediction of the future behaviour of the energy parameter, the probability of failure or RUL is not yet known. As stated in section 2.3.2, a commonly used method is to set an arbitrary threshold for the -in this case- energy parameter. If the predicted energy crosses this threshold, it is assumed that the specimen fails. This is quite an arbitrary and subjective process, as discussed in section 2.3.2.

Because of this issue, a new method is proposed. This method is based on the principles of load and resistance factor design (LRFD) (Galambos, 1981). In structural design, it is the goal to design a structure which can withstand a certain load. The amount of load a structure can withstand is called resistance in this method. Due to uncertainties in, e.g. design, materials, and conditions, the load and resistance are often not deterministic values. Instead, they are modelled as probabilities. This is illustrated in figure 3.10. When the load exceeds the resistance, the structure fails. Due to the uncertain nature, there is no single point of failure. Instead, there is a probability of failure P_f in the region where the load PDF is greater than the resistance PDF.

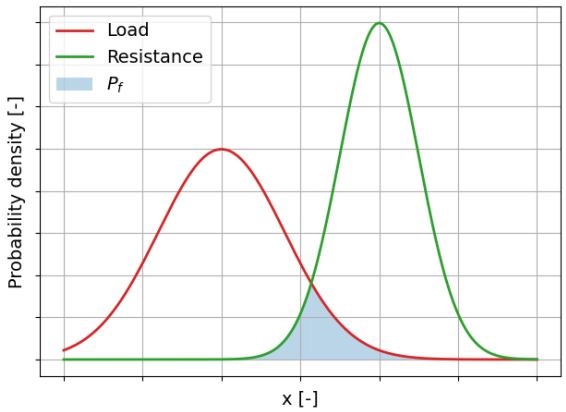


Figure 3.10: Load and resistance PDFs. The intersection depicts the probability of failure P_f.

This concept is translated to the context of this thesis by taking the (predicted) cumulative energy as load, and taking the PDF of the failure energies from other specimens as the resistance. Therefore, the threshold now becomes a continuous random variable T_E, for energy threshold values t_E. Letting E denote the predicted amount of cumulative energy e, then P_f can be defined as $\mathrm{p}(E > T_E)$. This can be written as the integral of the joint PDF $f_{T_E E}(t_E, e)$:

$$\mathrm{p}(E > T_E) = \int_{e=-\infty}^{\infty} \int_{t_E=-\infty}^{e} f_{T_E E}(t_E, e) \mathrm{d}t_E \mathrm{d}e \tag{3.27}$$

Now, it is assumed that the T_E and E are independent. Therefore, the joint PDF $f_{T_E E}(t_E, e)$ can be formulated as the product of the individual PDFs. The integrals become:

$$\mathrm{p}(E > T_E) = \int_{e=-\infty}^{\infty} \int_{t_E=-\infty}^{e} f_{T_E}(t_E) f_E(e) \mathrm{d}t_E \mathrm{d}e \tag{3.28}$$

$$= \int_{e=-\infty}^{\infty} f_E(e) \left[\int_{t_E=-\infty}^{e} f_{T_E}(t_E) \mathrm{d}t_E \right] \mathrm{d}e \tag{3.29}$$

The term between square brackets in equation (3.29) is the definition of the CDF of the threshold $F_{T_E}(e)$. Now the two terms within the integrals are both dependent on energy and can be integrated numerically. Furthermore, the lower bound for the integral can be set to 0, since this both the CDF of the threshold should be 0 for inputs below 0 since negative cumulative energies are not physically possible in this case. This results in equation (3.30).

Finally, one last addition is made. Since the PDF of the GP is a normal distribution, there is sometimes a significant probability of cumulative energy below 0 eu. This depends on both the mean and standard deviation of each prediction, but examples where this happens are shown in section 5.2.1. This means that infinitely far in the future, there will be a probability that the cumulative energy is below 0 eu, which is not physically possible. This results in the issue that, at this infinitely far point in the future, the integral in equation (3.30) and hence the probability of failure stays significantly less than 1. Therefore, a scaling factor is applied. The

$f_E(e)$ distribution is scaled by its CDF at 0 eu. This results in the fact that the area under the PDF of the cumulative energy prediction will always be 1, which is the cumulative probability.

$$\mathrm{p}\left(E > T_E\right) = \int_{e=0}^{\infty} f_E(e) F_{T_E}(e) \mathrm{d}e \tag{3.30}$$

$$= \int_{e=0}^{\infty} \frac{f_E(e)}{1 - F_E(0)} F_{T_E}(e) \mathrm{d}e = P_f \tag{3.31}$$

Now, the probability of failure P_f is obtained. This is still a scalar variable at this point. Keep in mind that this is at one point in the future, in a single prediction by the GP regression. When evaluating the GP prediction further in the future, another P_f can be determined, which is larger than the previous one, since $f_E(e)$ shifts to the right, with more probability mass exceeding the threshold. When this is done for all points in time, a CDF of the probability of failure is obtained; $P_f(t)$. Some energy predictions are significantly more optimistic than their predecessors, resulting in a decrease in the probability of failure. Because this is physically impossible, a monotonicity constraint is enforced; a probability of failure should always be larger than or equal to its predecessor, i.e. $P_f\left(t_{i+1}\right) \geq P_f\left(t_i\right)$.

The median end of life (EOL) can now be calculated by solving the CDF for 0.5. The PIs can be determined by solving the CDF for $\alpha/2$ and $(1-\alpha/2)$.

This CDF is still for just one prediction. Therefore for this prediction, the expected EOL and the 95% PI can now be determined. This process is performed for all predictions in order to get a live RUL prediction.

Just as in the statistical model, the probability distribution type of the cumulative energies at failure had to be determined. Again, the KS test was used. The results of the test are shown in figure 3.11 and table 3.2. Just as for the cycles at failure in section 3.1.1, the gamma function (equations (3.1) and (3.2)) turned out to fit the data best.

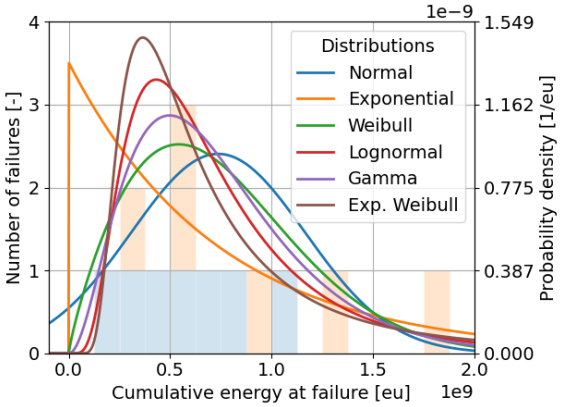


Figure 3.11: Distributions of cumulative energies at failure, for all CFRP specimens

Table 3.2: KS test results for different distributions of the cumulative energy at failure, on the CFRP data-set, sorted by goodness-of-fit

Distribution	p-value [-]	D [-]
Gamma	0.998	0.100
Lognormal	0.972	0.125
Weibull	0.969	0.126
Exp. Weibull	0.761	0.175
Normal	0.760	0.175
Exponential	0.291	0.260

3.2.5. Correlation adjustment

Just as the distribution of failure times, this distribution of cumulative energy is very wide. For setting the failure threshold, this will likely result in a general, wide PDF, not tailored towards the specimen under testing. Therefore, an extension is made to the failure PDF; an adjustment for correlation.

It is hypothesised that the correlations between the specimen under testing and other specimens can be used to centre the threshold PDF more towards the failure energy of the test specimen. This would then be under the assumption that cumulative energy series which correlate with each other will also fail at similar cumulative energies. This does, however, not yet suggest that positively correlated series share the same damage mechanisms at the same times.

Specimens whose time-series shown high correlation with the test specimen should therefore have more weight when establishing the failure threshold PDF than those with zero or negative correlation.

Extracting the correlation between time-series from a GP regression is a simple task; the label covariance matrix $\mathbf{\Sigma}_L$, constructed by $\kappa^{(l)}\left(l_p, l_q\right)$ can be extracted from the model. This function's hyperparameters are fitted by MLE, and should therefore correctly represent the covariance between labels. From the relation between covariance and correlation in equation (3.32), the correlation matrix can be extracted.

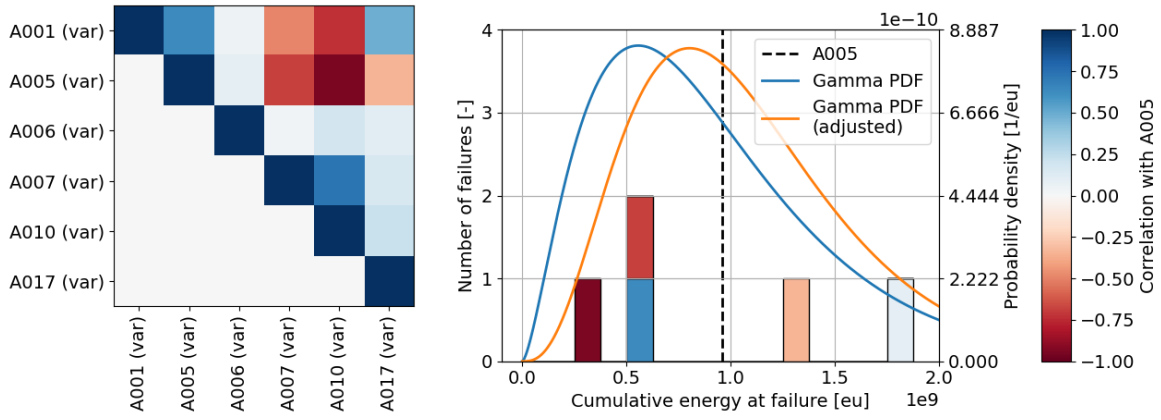
$$\text{cor}\left[X, Y\right] = \frac{\text{cov}\left[X, Y\right]}{\sigma_X \sigma_y} \tag{3.32}$$

The decision to go with the correlation matrix instead of the covariance matrix is due to how it is scaled; all values are between -1 (negative correlation) and 1 (positive correlation). Now, the weights can be determined. Because a new distribution is drawn based on weighed data, there are two conditions for the weights. First of all, the weights should all be larger than or equal to zero. A distribution cannot be drawn on negative numbers of samples. Secondly, the sum of the weights should be larger than zero; if all weights are zero, a probabilitiy distribution cannot be drawn. From this and the bounds from the covariance matrix, it was chosen to set the weights as in equation (3.33), for a test specimen with label l_p under test, and l_q labelled specimen from the training data.

$$w_q = \text{cor}\left[l_p, l_q\right] + 1 + \epsilon \tag{3.33}$$

Using this methodology, w_q is an extremely small value of ϵ if there would be a purely negative correlation. The maximum weight would be $2 + \epsilon$. This ensures both requirements for the weights. The maximum weight is set at just above 2, in order not to cause any too extreme new distributions. This is also needed for the assumption that PDF drawn on the weighted distribution is still a gamma function.

An example of desired behaviour is shown in figure 3.12. Because the correlation matrix is symmetric, only the upper triangular half is shown in figure 3.12a. The specimen for which the RUL is predicted is A005. The correlation matrix shows especially negative correlation with specimens A007 and A010. This causes the adjusted PDF in figure 3.12b to shift towards the right, where the actual cumulative energy at failure lies. In this case, the model therefore finds a correlation between specimens which have failure energies close to that of A005, which is used in the RUL predictions.



(a) Correlation matrix

(b) Original and adjusted PDF

Figure 3.12: Original and adjusted PDFs of the failure energy threshold. The model, with Ma3+lin kernels, is trained on variable amplitude fatigue (VAF) data, and tested on A005 after approximately 17,500 cycles.

3.2.6. Implementation

Due to the time-consuming training phase, a single feature was chosen to be extrapolated using this model, although using more features could be possible. Because a GP can model any function, every feature could, in theory, be used for making predictions. However, the value must also cross a threshold. Furthermore, if data is very noisy, the GP will likely pick this up as noise, therefore resulting only in a general trend with low confidence. It would therefore be wisest to take a cumulative acoustic emission (AE) parameter.

From the research of Eleftheroglou et al. (2016), the cumulative energy of AE events shows good correlation with the stiffness degradation, some specimen better than others. The authors conclude that cumulative energy is promising for describing the damage process in composites in fatigue. Therefore, it was decided to use this feature on the GP regression. All series are shown below in figure 3.13. It can be seen that for most specimens, there is a relatively sharp increase in the cumulative energy early in their life, followed by a lower gradient later on. The values at the EOL differ significantly, and will therefore likely cause a low precision in the predictions.

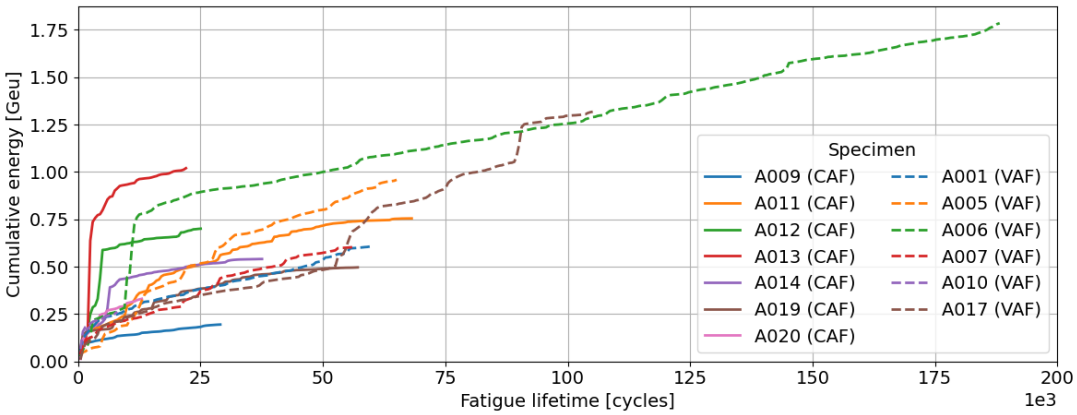


Figure 3.13: Cumulative AE energy for the CFRP specimens

This model is not used to answer the first research question. Training on solely constant amplitude fatigue (CAF) data is impossible since the model uses the available data of the specimen which it is being tested on as training data as well.

Due to the required computational resources, a prediction is not made at every 500 cycles except for specimen A010. It was decided to make predictions at a maximum of around 40 points in the life of each specimen. This meant that the intervals between predictions were higher than 500 cycles for some specimens. The most extreme case was specimen A006; due to its long life there were intervals of 5,000 cycles between predictions. It was ensured that the last prediction was madeat the actual EOL. The training data was not altered, so this consisted of blocks of approximately 500 cycles.

All of the above theory had to be implemented in a program. This is showed schematically in algorithm 1. Because current available Python dictionaries did not seem to be flexible enough to incorporate all of the theory and extensions discussed above, it was decided to develop a new Python GP regression model. The model is built using the NumPy library for its efficient storage of- and operations on arrays, and the SciPy library for the optimisation algorithm and matrix operations. It uses the limited memory Broyden–Fletcher–Goldfarb–Shanno bound constrained (L-BFGS-B) optimisation algorithm, which allows for the use of bound constraints. In this way, the constraints regarding the spherical parameterisation (appendix A.3) can be fulfilled, thus decreasing the search space for hyperparameters.

Algorithm 1: Steps taken for a single RUL prediction in the GP regression

input : Data $\mathcal{D}_l$ for each specimen labelled l in $\boldsymbol{l} = [1, ...L]$ total specimens, containing the input (cycles) $\boldsymbol{x}_l$ and output (cumulative energy) $\boldsymbol{y}_l$
Specimen l^{test} to perform prediction on
Time-step t_0 at which to make the RUL prediction
Number of model repetitions n_R

output: Median RUL and the 95% PI

Let the training specimens be $\boldsymbol{l}^{train} = \{l | l \in \boldsymbol{l}, l \neq l^{test}\}$
Split the data-set into training data $\mathcal{D}^{train}$, containing $\boldsymbol{x}$, $\boldsymbol{y}$ of $\boldsymbol{l}^{train}$ and $\boldsymbol{x}_{l^{test}}$, $\boldsymbol{y}_{l^{test}}$ until t_0
Standardise $\mathcal{D}^{train}$
for *r=1* **to** n_R **do**
    Initialise $\mathcal{M}_r$ on $\mathcal{D}^{train}$ with random hyperparameters
    Maximise $\log p(\boldsymbol{y} | \boldsymbol{X})$ by varying the hyperparameters using L-BFGS-B optimisation
end
Take $\mathcal{M}_r$ with highest $\log p(\boldsymbol{y} | \boldsymbol{X})$
if *using the correlation adjustment* **then**
    Extract $\boldsymbol{\Sigma}_L$ from the hyperparameters, construct correlation matrix and determine $\boldsymbol{w}$
    Resample the cumulative energy at failure distribution of $\boldsymbol{l}^{train}$ according to $\boldsymbol{w}$
    Construct $F_{T_E}(e)$ based on this distribution
else
    Construct $F_{T_E}(e)$ based on the cumulative energy at failure distribution of $\boldsymbol{l}^{train}$
end
while $P_{f,n} < 0.975$ **do**
    Predict cumulative energy on $\boldsymbol{x}^{test}$, running from t_0 to t_n
    Construct normal distribution $f_E(e)$ from predicted mean and covariance
    for t_0 **to** t_n **do**
        Solve the integral in equation (3.31), append $P_{f,t}$ to $\boldsymbol{P}_f$
    end
    Extend t_n further into future if $P_{f,n} < 0.975$
end
Enforce $P_{f,t} \geq P_{f,t-1}$
Calculate median RUL, lower and upper PI by interpolating $\boldsymbol{P}_f$ at 0.5, 0.025, 0.975 respectively

3.3. Recurrent neural network

In this section, the RNN is covered. First, the concept of NNs will be introduced, together with that of a RNN. Next, a more advanced type of RNN cell is covered; the long short-term memory (LSTM) cell. Next is the implementation of the model, followed by a validation scheme. Then, because NNs tend to be somewhat of a black-box, a sensitivity analysis is included. This is followed by an explanation of the conversion from failure index (FI) to RUL since it was decided to have the FI as model output. Finally, the case on varying load levels, using data from the glass fibre reinforced polymer (GFRP) experiments is covered.

3.3.1. Neural networks

A neural network is a group of so-called 'neurons'; nodes which interact with each other. Although on a much smaller scale, there are similarities with the structure of a brain, hence the term neurons. Depending on the architecture of the network, the neurons are connected in different ways.

A relatively simple network is that of a feedforward neural network (FFNN) with one hidden layer. Data is processed from an input layer to the output layer, via a layer which contains a number of neurons; the hidden layer. In such a network, all nodes in adjacent layers are connected to each other. Each connection is altered by a weight. An example of a FFNN is shown in figure 3.14.

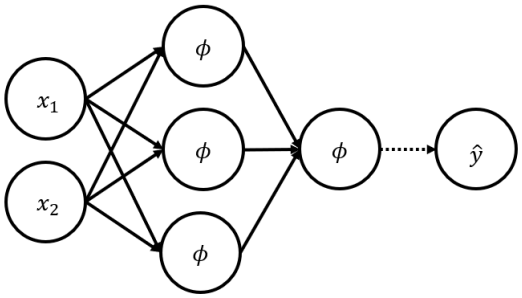


Figure 3.14: A typical FFNN architecture with two inputs, a hidden layer with three nodes, an output layer and one output. The connections going into the activation functions ϕ are weighed.

At each neuron, an operation takes place which sums the weighed input from connected neurons and feeds this through a so-called activation function ϕ. The input to this function $\boldsymbol{x}$ is a D-dimensional vector. Sometimes a bias term is included as well, resulting in $\boldsymbol{x} = \left[1, x_1, x_2, ..., x_D\right]^{\mathrm{T}}$. When this is multiplied with a set of weights $\boldsymbol{w}$, the cell's activity is obtained. This is then passed through the activation function, resulting in the output h of the i^{th} neuron (equation (3.34)). Instead of doing this per neuron, this can also be done for the entire layer at once, as in equation (3.35). In this equation, $\boldsymbol{W}$ consists of the individual weight vectors for each neuron in that layer, and is, therefore, $D+1 \times n_h$-dimensional.

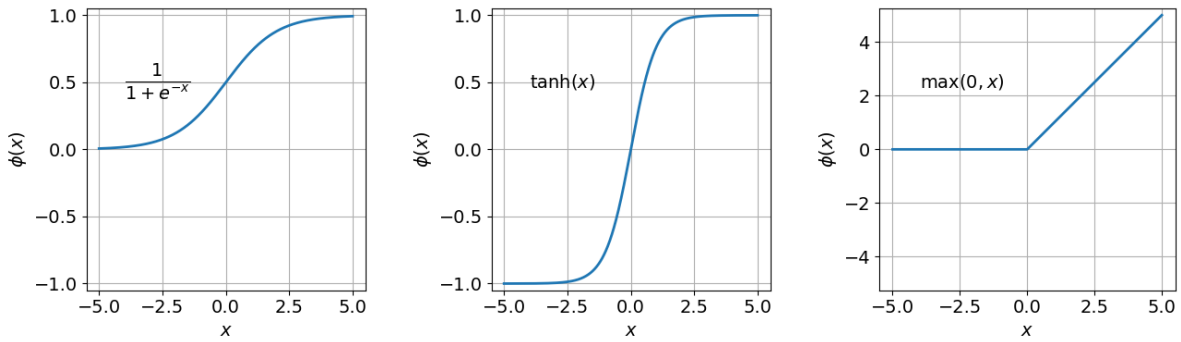
$$h_i = \phi\left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{x}\right) \tag{3.34}$$

$$\boldsymbol{h} = \phi\left(\boldsymbol{W}\boldsymbol{x}\right) \tag{3.35}$$

An activation function can, in principle, be any function. If the network is required to model nonlinear behaviour, these functions must do so as well. With nonlinear activation functions, NNs are able to approximate any function (Dorffner, 1996). Some commonly used activation functions are shown below in figure 3.15. The sigmoid function (figure 3.15a) outputs values between 0 and 1, and is often used in classification tasks because of this behaviour. Next, the hyperbolic tangent (figure 3.15b) shows the same S-shaped curve, however outputting values between -1 and 1. Finally, the rectified linear unit (ReLU) is a half rectified function, with outputs ranging from 0 to infinity. Many other activation functions exist, but these three are common ones and used in this thesis.

A network as described above is essentially a (nonlinear) function of input variables and its weights: $\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{w}\right)$. Before training, the unknowns in this function are the weights. A method which is generally applied for finding the weights is the MLE approach (Herlau et al., 2019). This method results in the loss function L, which is a function of the model's output, and the actual output $\boldsymbol{y}$:

$$L(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \left\| \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{w}\right) - \boldsymbol{y}_i \right\|^2 \tag{3.36}$$



(a) Sigmoid (b) Hyperbolic tangent (c) ReLU

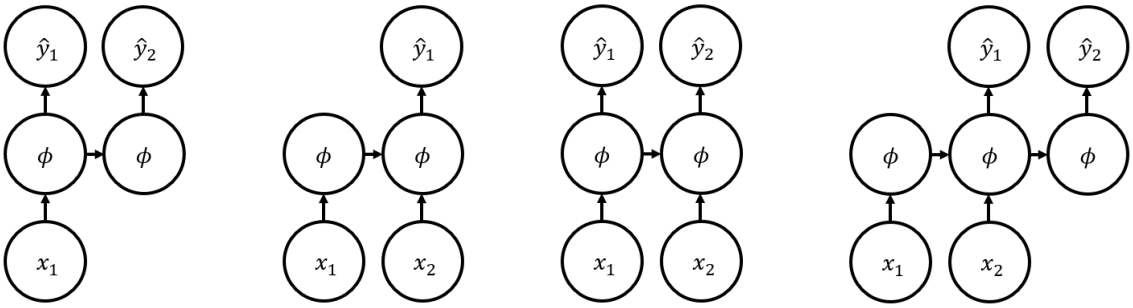
Figure 3.15: Three commonly used activation functions

The goal is now to minimise the loss of the model, and find accompanying optimal weights $\boldsymbol{w}^*$. Because the above equation is not analytically solvable, this is done using gradient descent. The loss is calculated for a specific $\boldsymbol{w}$, as well as the gradient at this point. The sign of the gradient then tells us in which direction the weight should move in order to result in a lower loss. Just as in the GP regression, NNs optimisers have to deal with the possibility of hitting local minima. When weights are initialised randomly, different 'optimal' models will be found.

The loss function can also be customised. Although this is not done for this thesis, it could be, for example, customised to add more weight to the loss close to the EOL of a specimen. In this way, the priority of predictions near the EOL is increased.

Recurrent neural networks

Whereas a FFNN takes a vector as an input, resulting in an output vector, a RNN is capable of handling entire multidimensional time-series. A RNN consists of cells, which do not only take inputs from a previous layer, but also from the previous time-step. With this addition, multiple architectures can be used, see figure 3.16. All of these are with one hidden layer, but numerous hidden layers can, of course, be used, as well as combinations of these architectures.



(a) One-to-many (b) Many-to-one (c) Many-to-many (d) Another version of many-to-many

Figure 3.16: Four possible architectures for RNNs. A cell thus has two inputs, one from the input x, and its previous output. The data is therefore passed through the same cell every time.

The first, one-to-many in figure 3.16a, can be used for generating word sequences or music, based on a single input. An output would then be dependent on a previous output, in order to form coherent sentences or melodies. The other way around is done in a many-to-one architecture (figure 3.16b). An array of input data can be compressed into one output. Unlike in a regular FFNN, this output takes the order of inputs into account. Finally, there are two versions of many-to-many architectures (figures 3.16c and 3.16d). The first can be used for processing a series where at a time-step an output is required, based also on the previous time-steps. The second is used in for example translations, where a translated word depends on not only one word, but possibly also on the following word.

For this thesis, the first many-to-many architecture (figure 3.16c) shall be used. In this way, an input can

immediately produce a corresponding output. In the context of this thesis, the inputs would then consist of measured AE data and possibly load data, and the output of the RNN will be a parameter which relates to the state of the specimen under testing.

Vanilla RNN cell
A so-called vanilla RNN cell is the simplest form of a RNN. Just as in the FFNN, a weighted input is passed through an activation function. The cell is shown in figure 3.17

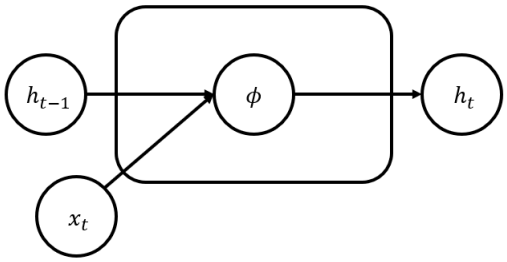


Figure 3.17: A vanilla RNN cell

For every time-step t, a D-dimensional input vector $\boldsymbol{x}_t$ is fed into the cell. The output, or hidden state, of a cell $\boldsymbol{h}$ at time t is a function of both the input vector $\boldsymbol{x}_t$, as well as the preceding output $\boldsymbol{h}_{t-1}$. It is defined as in equation (3.37). Just as in a FFNN, there is a weight matrix $\boldsymbol{U}$ for the input. A bias can be again be included in $\boldsymbol{x}$. The size of the weight matrices depends on the number of hidden nodes n_h; $\boldsymbol{U}$ is $(n_h + 1) \times D$-dimensional, where the 1 is due to the bias. The previous output is multiplied by a weight matrix $\boldsymbol{V}$. This matrix is of course $n_h \times n_h$-dimensional, since the output is a n_h-dimensional vector. The sum of these is passed through an activation function, just as in any NN.

In the equations hereafter, $\boldsymbol{U}$ and $\boldsymbol{V}$ are written as one weight matrix $\boldsymbol{W}$. This $D + 1 + n_h \times n_h$-dimensional matrix is multiplied with two stacked vectors $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$, as in equation (3.38).

$$\boldsymbol{h}_t = \phi\left(\boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{V}\boldsymbol{h}_{t-1}\right) \tag{3.37}$$

$$= \phi\left(\boldsymbol{W}\left[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}\right]\right) \tag{3.38}$$

During training, the values in $\boldsymbol{U}$ and $\boldsymbol{V}$ have to be optimised in order to fit the training data. The number of parameters n_p which have to be trained in a simple RNN cell is, therefore, a function of the number of hidden nodes, as well as the number of dimensions of the input:

$$n_p^{(RNN)} = n_h^2 + n_h D + n_h \tag{3.39}$$

An issue of vanilla RNN cells is the problem of exploding/vanishing gradients (Hochreiter and Schmidhuber, 1997). The information from a previous cell is handled together with the cell input. In this way, the information from cells in the past is overwritten multiple times. This causes difficulties in training, as well as the simple fact that long-term information is not stored. Because of this, FFNNs with time windows would perform equally well as RNNs with vanilla cells. (Hochreiter and Schmidhuber, 1997)

3.3.2. Long short-term memory cell

In order to overcome this vanishing gradient problem, Hochreiter and Schmidhuber (1997) proposes a new type of cell: the LSTM cell. Recently, the gated recurrent unit (GRU) was introduced by Cho et al. (2014). This type of cell contains one less gate as compared to the LSTM. The LSTM cell was however preferred over the GRU, due to a wide array of documentation online, making the implementation easier. In terms of performance, no real difference between the two was found in, for example, the study by Chung et al. (2014). The two significantly outperformed a regular RNN in multiple case studies, however.

The entire LSTM cell can be seen in figure 3.18. What makes the LSTM differ from a vanilla RNN cell is the fact that it has one additional in- and output. This is the so-called cell state. The cell state is adjusted every cycle, such that information is forgotten or added. The cell state interacts with the input and previous hidden state, leading to a new hidden state.

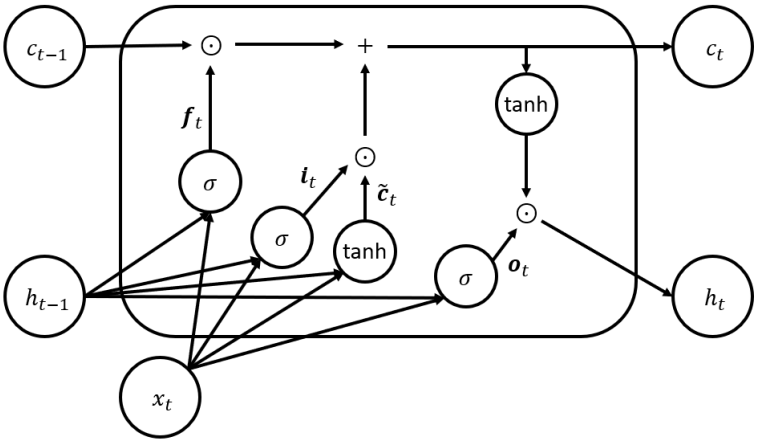


Figure 3.18: A LSTM cell

The LSTM cell contains three so-called gates. The gates are the combinations of sigmoids and element-wise multiplications in figure 3.18. The sigmoid will determine if information should be let through (output of 1), or be deleted (output of 0). Firstly, the cell state is altered within the cell through two operations, shown in equation (3.40). They will be explained below.

$$\boldsymbol{c}_t = \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t + \tilde{\boldsymbol{c}}_t \odot \boldsymbol{i}_t \tag{3.40}$$

The first operation in the cell state in equation (3.40) is through the forget gate, where $\boldsymbol{c}_{t-1}$ is multiplied element-wise with the output of the forget gate $\boldsymbol{f}_t$ (equation (3.41)). This gate determines which elements of the cell state should be forgotten or kept, based on the input and previous hidden state. The trainable parameters in this equation are the weight matrix $\boldsymbol{W_f}$ (also containing bias). Just as in a vanilla RNN, the weight matrix is $D + 1 + n_h \times n_h$-dimensional. This is passed through a sigmoid function σ, meaning that all values in $\boldsymbol{f}_t$ are between 0 and 1. Because of this, elements of $\boldsymbol{c}_{t-1}$ multiplied with values close to 0 will diffuse, while elements multiplied with values close to 1 will be kept.

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \left[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}\right]\right) \tag{3.41}$$

The input gate $\boldsymbol{i}_t$ proposes candidate values $\tilde{\boldsymbol{c}}_t$. Their corresponding equations are shown in equations (3.42) and (3.43). The gate uses again the output of a sigmoid function, whereas the candidate values come from a hyperbolic tangent function. This causes the candidate values to range between -1 and 1, allowing the cell state $\boldsymbol{c}_t$ to decrease as well.

$$\tilde{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_c \left[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}\right]\right) \tag{3.42}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \left[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}\right]\right) \tag{3.43}$$

Now, the cell state is adjusted before it is fed into the cell in the next time-step. The output gate, which adjusts the output of the cell based on the cell state, input and previous output, remains. Again, the sigmoid function in equation (3.44) determines which parameters to output, and what their weight is. Next, this is multiplied element-wise with the cell-state which has been passed through a hyperbolic tangent function in equation (3.45). Each element of $\boldsymbol{h}_t$ therefore contains values between -1 and 1.

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \left[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}\right]\right) \tag{3.44}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{c}_t\right) \tag{3.45}$$

From the fact that four weight matrices, including bias vectors, are present in an LSTM-cell, the number of trainable parameters becomes much higher than in a vanilla RNN cell:

$$4(n_h^2 + n_h D + n_h) \tag{3.46}$$

3.3.3. Implementation

In order to predict RUL using a LSTM-cell, several changes had to be implemented to the basic many-to-many architecture with LSTM cell. These changes are discussed below. The optimisation of the number of hidden nodes is discussed after this section in section 3.3.4. The model itself is built in Python using Keras, an open-source library for NNs (Chollet and others, 2015).

In- and output

Due to the ability of NNs to distinguish between different features if they are trained properly, it was decided to feed the RNN all features which are presented in section 4.4.3. The sensitivity analysis will point out which features are deemed most important.

For the output of the model, it was decided not to use the RUL as a direct output variable. This variable is scaled differently between specimens, also after standardisation. This meant that not all activation functions are suited for reaching RUL values. A variable which lends itself extremely well is the FI or health index (HI), with values always between 0 and 1. These variables linearly increase from 0 to 1, or decrease from 1 to 0 respectively. All three activation functions discussed above would be able to output these values, and would therefore not require additional weights or scaling afterwards.

It should be noted that these functions describe the percentage of life passed or still left to pass. Therefore, they are not representing the actual damage within a structure, which is generally not linearly increasing.

Eventually, the choice was made for the FI. This was done in conjunction with choosing the final activation function, discussed below. The FI could, in principle have values above 1, which say that the specimen should have failed by then. A value below 0 would, however, be physically impossible; this implies that a specimen is in better health than before the test. The ReLU function, with outputs larger than 0 can model this behaviour. Therefore, the combination of a ReLU function and FI as output was deemed the best combination.

Architecture

The architecture was already briefly discussed above, with a many-to-many with a direct output being the best architecture for this problem. It was also decided to keep the number of hidden layers at one. While increasing the number of hidden layers generally leads to better-fitting of the training data, there are also downsides. First of all, the number of trainable parameters increases, and therefore the required computation time. In addition to this, the training set is already extremely small for a NN with just 13 specimens in total. With more added complexity, the model will be bound to overfit on such a small sample set.

Summation layer

An extra layer is added after the output of the cell, in order to make the output of a cell a single number, instead of a vector with n_h entries, which is the original output of a LSTM cell. This layer (equation (3.47)) is a simple summation layer with weight vector $\boldsymbol{w}_s$, hence containing an additional $n_h + 1$ trainable parameters.

$$y_t = \phi\left(\boldsymbol{w}_l^{\mathrm{T}} \boldsymbol{h}_t\right) \tag{3.47}$$

Now, the network with n_h hidden nodes is able to be fed a batch of time-series with D dimensions, and output a 1-dimensional series. The output can now be trained to match and predict a single output variable.

A model parameter which can significantly affect the results is the choice of activation function in this final layer. Because the model is trained on the FI, values should fall between 0 and 1. Initially, a sigmoid function was thought to adhere to this requirement, since this function will always result in values between 0 and 1 (see figure 3.15a). However, in order to get output values close to 0 or 1, large negative or positive activities are required. This proved to be a problem. In test runs, the outputs generally floated between 0.2 and 0.8; the models were not able to train towards giving more extreme activities to the final layer. In order to overcome this, ReLU activation functions were used in the network. This resulted in predicted FIs which were closer to 0 and 1.

Dropout

A relatively new regularisation method is dropout, introduced by Srivastava et al. (2014). The idea behind this method is to randomly eliminate nodes and their connections during training.

Srivastava et al. (2014) motivate their idea for dropout from a theory of the role of sex in evolution. Organisms which have evolved through sexual reproduction are much more advanced than those who reproduce asexually. Through sexual reproduction, the offspring receives roughly 50% of the genes of the parents, together with some random mutations. In asexual reproduction however, the offspring receives all genes, together with some random mutations, from one parent. In the latter case, genes which work well together are passed on to the offspring. It seems plausible that this results in more advanced lifeforms through evolution than reproducing sexually because here, the genes of the parents are split and must work together with a new set of genes. The fact that organisms which evolved through sexual reproduction are however more advanced than their asexual counterparts is likely that over the long term, natural selection favours the mix-ability of genes. This ability for sets of genes to work together with unseen other sets makes them more robust. Furthermore, when genes are then working together in small, compatible subsets, mutations can also more quickly lead to more successful offspring. When a favourable mutation would be introduced in an asexually reproducing organism, the mutated genes are less likely to be able to work together with the almost monolithic set of genes which was passed on for generations already.

Srivastava et al. (2014) argue that the same could be applied to a NN. When forcing units to randomly work together with other units, preventing co-adaptations between nodes, this makes the model more robust. Furthermore, nodes will be forced to create useful features themselves, instead of relying on the entire infrastructure around the units. Typical values for dropout range from retaining 50% to 80%. Smaller retainment values can lead to underfitting, whereas larger values may not enforce enough dropout to effectively regularise the model to prevent overfitting. (Srivastava et al., 2014)

The motivation behind implementing this method in this thesis is because the RNN has many inputs (discussed in section 4.4.3). With many relatively similar inputs and some possibly not of interest to the FI predictions, it was decided to add dropout to the input layer as a form of input regularisation. A value of 50% was chosen, based on the typical values by Srivastava et al. (2014). Also, eliminating 50% of all inputs would not lead to useless predictions, due to the fact that there are up to 31 inputs. With, for example, all 9 cumulative features form AEs, missing half of them will not result in the inability to function.

Furthermore, the learning rate and momentum of the optimiser are also advised to be changed by Srivastava et al. (2014), because of noise which is introduced in the stochastic gradient descent. Through trial and error, a learning rate of 0.005 and momentum of 0.9 was found to give the highest learning speed and reduce noise as much as possible. Note that this was done on one model, on one specimen. Including this in another cross-validation loop would too computationally expensive.

3.3.4. Hidden nodes and validation

In order to objectively determine the best model architecture and its performance, a two-level, K-fold cross-validation scheme is implemented. The key to this scheme is that this is as objective as possible, by validating and testing on unseen data. A model cannot ever be objectively evaluated when it is fed the same training data as it is tested on. Two-level cross-validation is a way of mitigating this and determining the optimal model architecture and performance based on 'unseen' data. The optimal architecture is a combination of the number of hidden nodes, as well as the number of training epochs. In order to capture increasingly more complex models, the number of hidden nodes is exponentially increased from 1 to 128.

Having a low amount of epochs usually results in underfitting, whereas a high amount of epochs results in overfitting. An example of this phenomenon is shown in figure 3.19. Whereas the training loss keeps decreasing over time, the validation loss increases again due to overfitting. Note the oscillating losses; this is due to noise from the dropout. Setting the maximum amount of training epochs at 1000 proved to capture most optima in validation losses, as well as keeping the computation time for training a set of models relatively low.

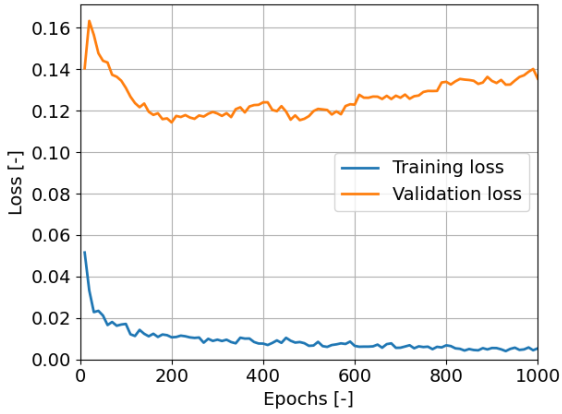


Figure 3.19: Training- versus validation loss for a model with $n_h = 16$, with specimen A001 left out, and validated on A010

The algorithm for the cross-validation scheme is shown in algorithm 2. The loops for the two levels can be seen in lines 1 and 6. For each test specimen l^{test}, the data is split into a test case $\mathcal{D}^{test}$, and the data from the other specimens $\mathcal{D}^{inner}$. The optimal model architecture is now determined in the inner loop, where for each combination of a validation specimen l^{val} and training set $\boldsymbol{l}^{train}$, the validation error is determined. In every one of these combinations, all model architectures are evaluated. The validation errors are also stored per training epoch in order to decide on the best number of training sessions eventually.

This is then repeated $R = 10$ times since due to the random initialisation of weights in a NN, results vary per model. In order to mitigate the impact of outliers, the median over these repetitions is taken. The average of these validation errors is taken per model architecture, resulting in an approximation of the generalisation error. The architecture and number of epochs with the lowest generalisation error is taken to train a new model, based on all data used in this inner loop. In this way, the optimal model architecture is determined on the other specimens, without any knowledge of the test specimen.

When training on solely CAF data, the outer loop is omitted. The validation can be done on all CAF specimens. From this, the optimal model architecture can be determined and tested on a VAF specimen. When trained on both CAF and VAF data, it was decided to loop over just the VAF specimens as validation and test specimens. This is because, for this research, there was no interest in the performance of the models on predictions of CAF specimens, as well as to save time on computations. Although not used for calculating errors, the CAF specimens were used as training data in all steps.

Algorithm 2: Two-level K-fold cross-validation for the RNN

input : Data $\mathcal{D}_l$ for each specimen labelled l in $\boldsymbol{l} = [1, ...L]$ total specimens, containing the input $\boldsymbol{X}_l$ and output $\boldsymbol{y}_l$
n_M number of model architectures $\mathcal{M}$, with n_h hidden nodes from $[1, 2, 4, 8, 16, 32, 64, 128]$
Number of training repetitions n_R
Number of training epochs n_E

output: Optimal network architecture $\mathcal{M}_i^*$ and predictions $\boldsymbol{y}_i$ for each specimen

for $i = 1$ **to** L **do**
  Let the test specimen be $l_i^{test} = l_i$
  Let the inner fold specimens be $\boldsymbol{l}_i^{inner} = \{l | l \in \boldsymbol{l}, l \neq l_i^{test}\}$
  Split the full data-set into $\mathcal{D}_i^{inner} = \mathcal{D}_{l \in \boldsymbol{l}_i^{inner}}$, $\mathcal{D}_i^{test} = \mathcal{D}_{l_i^{test}}$
  Standardise $\mathcal{D}_i^{inner}$, save $\boldsymbol{\mu}_i^{inner}$ and $\boldsymbol{\sigma}_i^{inner}$ and use these to standardise $\mathcal{D}_i^{test}$
  for $j = 1$ **to** $L - 1$ **do**
    Let the validation specimen be $l_{i,j}^{val} = l_{i,j}^{inner}$
    Let the set of training specimens be $\boldsymbol{l}_{i,j}^{train} = \{l | l \in \boldsymbol{l}_i^{inner}, l \neq l_i^{val}\}$
    Take training and validation data-sets $\mathcal{D}_{i,j}^{train} = \mathcal{D}_{l \in \boldsymbol{l}_{i,j}^{train}}, \mathcal{D}_{i,j}^{val} = \mathcal{D}_{l_{i,j}^{val}}$
    for *m=1* **to** n_M **do**
      for *r=1* **to** n_R **do**
        Train $\mathcal{M}_{i,j,m,r}$ on $\mathcal{D}_{i,j}^{train}$ for n_E epochs, minimising the mean squared error (MSE)
        Validate $\mathcal{M}_{i,j,m,r}$ on $\mathcal{D}_{i,j}^{val}$, giving validation error $E_{i,j,m,r}^{val}$ at every epoch
      end
      Take the median validation error for each model $\tilde{E}_{i,j,m}^{val}$
    end
  end
  Compute the estimated generalisation error $\hat{E}_{i,m}^{gen} = \frac{1}{L-1} \sum_{j=1}^{L-1} \tilde{E}_{i,j,m}^{val}$ for each model
  Select the optimal model $\mathcal{M}_i^*$ and number of epochs E^* to train it, based on the minimal $\hat{E}_{i,m}^{gen}$
  for *r=1* **to** n_R **do**
    Train $\mathcal{M}_i^*$ on $\mathcal{D}_i^{inner}$ for E^* epochs, and feed it $\boldsymbol{X}_i^{test}$ to obtain the estimate output $\hat{\boldsymbol{y}}_{i,r}$
  end
  Compare the group of estimated outputs $\hat{\boldsymbol{y}}_{i,r}$ to the actual outcome $\boldsymbol{y}_i$
end

3.3.5. Sensitivity analysis

Due to the number of weights and operations in a NN with multiple hidden nodes, it tends to become a 'black-box'. With information going in and being fed through all nodes and layers, the net spits out an output. How this output is constructed based on all inputs is impossible to analyse by just looking at all weights matrices of the network. Yet, it is important to know which inputs affect the output of the model. Therefore, a sensitivity analysis is performed. Due to the temporal dependence of the RNN, not all methods can, however, be used. The perturbation method was found to be possible to implement in this context. This method is able to classify variables, according to their importance of inputs (Gevrey et al., 2003).

The perturbation method adds changes δ to the i^{th} input variable. These changes are usually steps of 10% of the input, up to 50% (Gevrey et al., 2003). This was followed in this analysis, with the steps calculated as 10-50% of the maximum value of an input. This was done before standardising the data. After adding the δ, the input data was standardised, using the mean and standard deviation from the other specimens' data. Now for each input variable, the model's MSE loss is calculated when adding δ to this input. By then ranking the change in MSE, a ranking of the input variables can be made.

3.3.6. Failure index to remaining useful life

The RNN predicts the FI, while the RUL is the required prognostic feature. Fortunately however, there is a simple relationship between the FI and RUL. Since the FI is a function of RUL and passed time or cycles t, equation (3.48) can be derived from this. The unfortunate part of this relationship is the fact that FI is in the denominator. If the model therefore predicts a FI of 0, this leads to an infinitely high RUL. Therefore, predictions with an FI of 0 had to be omitted from the final RUL prediction.

From the 10 repetitions with predictions for the FI, a RUL, as well as the PIs have to be determined. Assuming that the repetitions result in a normally distributed set of predictions, the median, as well as the PIs of the RUL prediction, can be easily obtained.

$$RUL = \frac{t}{FI} - t \tag{3.48}$$

One final adjustment which is made to the FI is enforcing monotonicity. This is because the predictions from the RNN have a high variability from one time-step to another, with sometimes also decreasing FIs. An example is shown below in figure 3.20. From a physical perspective, this behaviour is not possible. A damage parameter cannot decrease in a structure when it is not repaired, let alone when it is under active loading. From the perspective of data analysis, this behaviour is possible. A model has more information at time t compared to time $t-1$. When having this additional information, it might turn out that the FI is found to be lower than in the previous prediction.

It was decided to go for more realistic predictions, and therefore enforcing a monotonicity constraint on the FI. This was implemented on the results by setting FI at t equal to the FI at $t-1$ if it were smaller than its predecessor. This also smoothens the RUL predictions. Compared to other smoothing methods which may rely on the data on both sides of t, this method does not require prior information on future FIs.

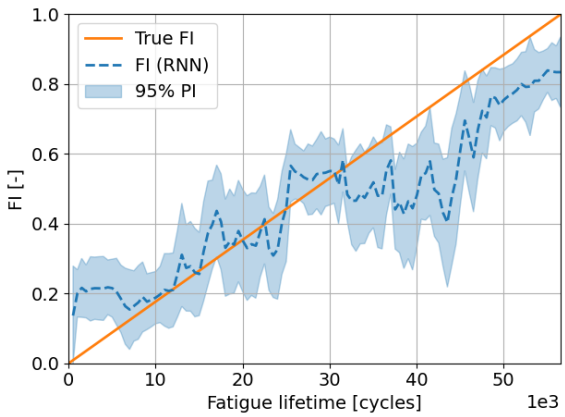


Figure 3.20: The original FI output from the RNN for specimen A007, trained on CAF data

3.3.7. Case study: varying load levels

For the case study, the same approach from above was used, to predict RUL based on solely AE data and the number of cycles as time-indicator. These predictions were, however, not sufficiently good enough; models seemed to be unable to handle the large scale differences between the specimens.

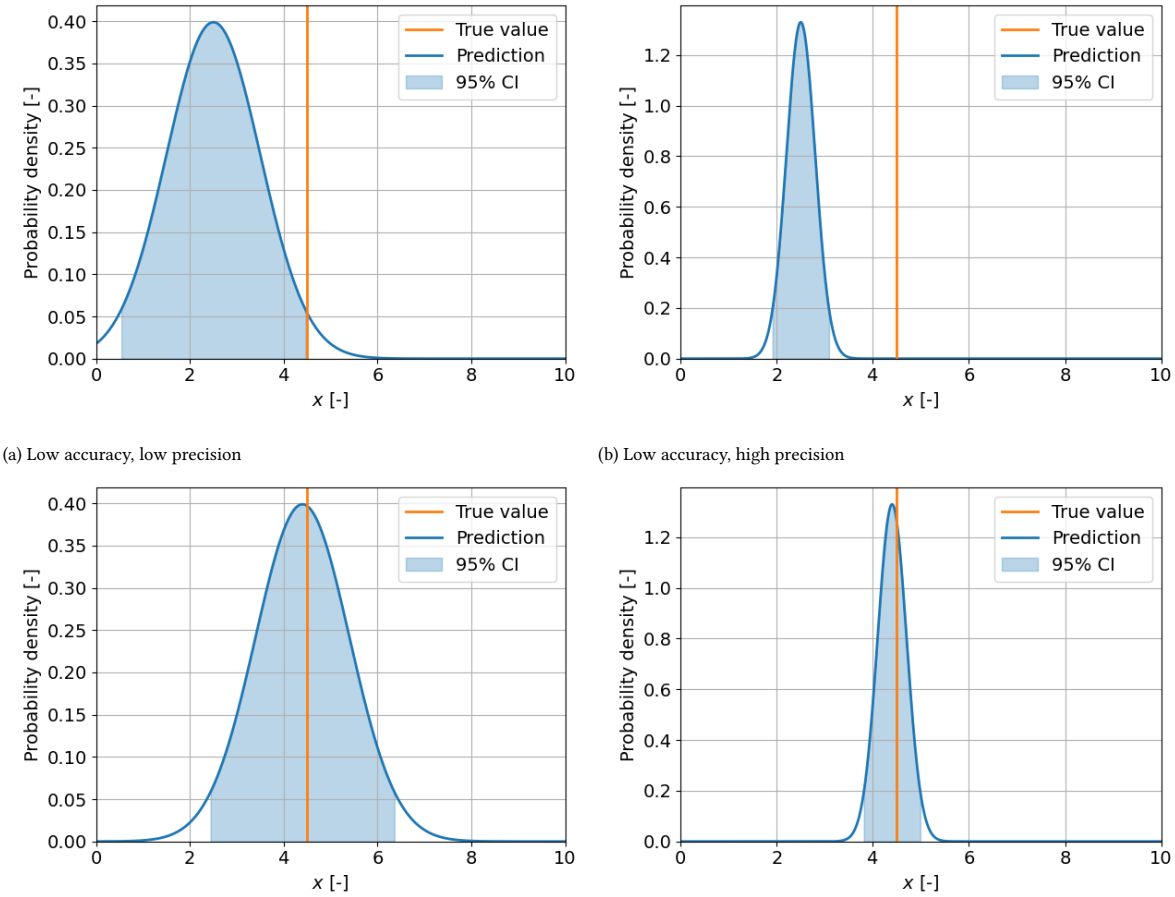
Therefore, it was decided to include the load levels in the input features. From a physical standpoint, the magnitude of the applied load impacts the EOL of the specimen in a nonlinear nature. An example can be seen in the S-N curve in figure 2.3. There is research done on modelling S-N curves using FFNNs, as done by for example Al-Assaf and El Kadi (2001), who discuss a FFNN for successfully modelling the S-N curve of uni-directional (UD) GFRP specimens under tension-tension (T-T) and tension-compression (T-C) loading. They use the maximum stress, R-ratio and fibre orientation angle as inputs. The latter does not apply to this research.

It is hypothesised that the predictions from a RNN can be enriched by adding a FFNN at the end of a RNN. Feeding the load levels into this second layer next to the output of the RNN could result in a nonlinear weight factor on the output of the RNN, leading to better predictions for this data-set. The FFNN will use ReLU activation functions, just like the final layer of the RNN, which was used to generate a single output. This final layer was bypassed; the output from the LSTM cell is directly fed into a number of hidden nodes in the FFNN. After the hidden layer, a single node is used to generate one output value per time-step; the FI.

In order to validate this approach, the same cross-validation scheme is used above first to determine an optimal RNN architecture for this problem. FI predictions will be made using this setup. Next, the cross-validation scheme will be used again, but now for determining the number of nodes in the hidden layer of an added FFNN at the end of the RNN. The hidden nodes from the RNN are kept the same as before, therefore varying only the nodes of the FFNN.

3.4. Performance metrics

When comparing predictions to actual values, there are two notions which seem similar but are actually different. These are accuracy and precision. The difference between the two is illustrated in figure 3.21. As can be seen, accuracy is a measure of bias; how far the mean or median is situated from the actual value. Precision, on the other hand, is the width of the PI. More precision leads to more certainty about the mean or median, but not necessarily to better predictions.



(a) Low accuracy, low precision

(b) Low accuracy, high precision

(c) High accuracy, low precision

(d) High accuracy, high precision

Figure 3.21: The difference between accuracy and precision

In this thesis, both the prediction and accuracy of predictions will be taken into account. Scoring low on one of them means that a model does not yet provide good predictions. The 95% PIs are illustrated to show the precision of the models. The predicted values for the RUL are based on the median of the predictions, or 50% probability of failure. This choice was made because of the implementation of the GP regression. In this model, the probability of failure was calculated numerically. In order to calculate a mean, the CDF would have to be integrated. This was an array of points, however, and not a function. Numerical integration over this array proved to give unstable results, and it was decided to not fit a distribution on this, since there can be no solid arguments behind specific distributions. Therefore, taking the median from interpolating the CDF at 0.5 gave more reliable results. In order to stay consistent, this was also done for the other models.

3.4.1. Classical metrics

According to Saxena et al. (2009), the four most commonly used classical metrics in forecasting are accuracy, precision, MSE, and mean absolute percentage error (MAPE). All these functions utilise the prediction error Δ; the difference between the actual RUL of the n^{th} prediction r_n, and the predicted RUL at this point $\hat{r}_n$:

$$\Delta_n = r_n - \hat{r}_n \tag{3.49}$$

The mean bias B is then simply the sum of the errors of all predictions (equation (3.50)). The precision measure S is used for quantifying the variability between predictions (equation (3.51)). Each error is compared to the mean bias, resulting in the standard deviation of the errors. This measure does not relate to the precision defined above.

$$B = \frac{1}{N}\sum_{n=1}^{N} \Delta_n \tag{3.50}$$

$$S = \sqrt{\frac{\sum_{n=1}^{N} \left(\Delta_n - B\right)^2}{N-1}} \tag{3.51}$$

Both these functions have the issue that negative and positive errors cancel each other out. Luckily, the MSE and MAPE exist; these do not have this issue. They are shown in equations (3.52) and (3.53). The MAPE is a scale-independent number. This makes it possible to compare different samples with different EOL to each other. However, using this metric causes the loss of information about the absolute error. This may be a valuable metric to determine if, for example, a specimen with a relatively short lifespan still has an acceptable prediction error in order to perform maintenance in time.

$$MSE = \frac{1}{N}\sum_{n=1}^{N} \Delta_n^2 \tag{3.52}$$

$$MAPE = \frac{1}{N}\sum_{n=1}^{N} \left|\frac{\Delta_n}{r_n}\right| \cdot 100\% \tag{3.53}$$

The MAPE is the most widely used accuracy metric. A downside in RUL predictions is that forecasts which exceed r_n are more heavily penalised than those which are less; since in the latter, the error cannot be larger than 100%. Furthermore, the MAPE of the last prediction, where the real value is 0, cannot be calculated. For these reasons, MSE and MAPE are used as basic metrics.

3.4.2. Prognostic metrics

As mentioned in section 2.3.3, Saxena et al. (2010) propose four prognostic metrics. Three of these are actually used here; the $\alpha - \lambda$ convergence is left out. This is because one of the requirements for this metric is that there is a prognostic horizon (PH). In just two predictions in this thesis, there was a PH. Therefore this method was deemed not useful. Therefore, a fourth metric is proposed, based on the PH.

Prognostic horizon

The prognostic horizon is defined as the time to the EOL, where the probability mass of a prediction $\pi\left[\hat{r}_n\right]\big|_{\alpha^-}^{\alpha^+}$ between required confidence bounds $\alpha^{\pm}$ around the actual RUL, is larger than a threshold β. This is valid only where this is true for an uninterrupted time period until the EOL. Therefore, right predictions at the start of a specimen's life do not count towards the PH if the requirement is not met later in the specimen's life. When given a CDF of the RUL prediction $F_T(r)$, the PH is calculated as follows:

$$\pi\left[\hat{r}_n\right]\big|_{\alpha^-}^{\alpha^+} = F_T\left(r_n + \alpha^+\right) - F_T\left(r_n + \alpha^-\right) \tag{3.54}$$

An advantage of this metric compared to the metrics mentioned above is that this metric keeps track of the confidence of the prediction. Wider PIs are disadvantageous for this metric.

The minimum required probability mass β, is for now set at a value of 50%. This is relatively low and should not be acceptable in the industry. The required confidence $\alpha^{\pm}$ is set at ±5,000 cycles. This is mainly based on the lifetimes of specimens in the test set. In, for example, the application of a wind turbine blade, this would be an 8-hour window based on the turbine operating at 10 rpm, and looking at a section of the blade which suffers edgewise fatigue loads. These loads are caused by gravity, and therefore a fatigue cycle occurs at every rotation.

Cumulative bounded probability mass

As the results will show, many predictions' $\pi\left[\hat{r}_n\right]\big|_{\alpha^-}^{\alpha^+}$ do not exceed a reasonable value for β. Therefore, a new metric is introduced. This metric, the cumulative bounded probability mass (CBPM) (equation (3.55)), tracks the cumulative probability mass within the required confidence bounds, with weights which are relatively higher near the EOL of the specimen. The advantage of this metric, as compared to the other mentioned metrics, is that it captures precision as well as accuracy. Wide PIs are automatically penalised because their mass within the required confidence bounds is small. Predictions with low accuracy will logically have lower CBPMs.

$$CBPM = \sum_{n=1}^{N} w\left(r_n\right)\pi\left[\hat{r}_n\right]\big|_{\alpha^-}^{\alpha^+} \cdot 100\% \tag{3.55}$$

A linear weight function, following the fraction of total life T is taken. The weights are normalised to ensure that the total sum of the weights is 1, and therefore all weighted CBPMs can be added up. The weight function thus becomes:

$$w\left(r_n\right) = \frac{1}{\sum_{n=1}^{N} r_n / T}\frac{r_n}{T} \tag{3.56}$$

Cumulative relative accuracy

Just as the CBPM, the cumulative relative accuracy (CRA) is a weighted metric over the entire life of the specimen. Proposed by Saxena et al. (2009), it keeps track of the relative accuracy (RA) of the predictions. Therefore for this metric, the expected RUL is used. Wide PIs are not penalised. CRA is calculated using equation (3.57). The weights in the equation are the same as those in equation (3.56). A CRA as close to 100% as possible is desired. CRA values become smaller than 0 if $\left|\Delta\right| / r_n$ is larger than 1. In this case, there would be at least a 200% bias. The CRA should be as close to 100% as possible.

$$CRA = \sum_{n=1}^{N} w\left(r_n\right)\left(1 - \frac{\left|\Delta_n\right|}{r_n}\right) \cdot 100\% \tag{3.57}$$

Convergence

The convergence is the final metric introduced by Saxena et al. (2009, 2010). This is a so-called meta-metric, which quantifies how fast a metric M improves. By using the convergence, it will be quantified how the accuracy and precision improve over time.

The convergence C_M is defined from the start of the predictions t_P up until the time where it is too late to perform any necessary repairs, the end of useful predictions (EOUP), t_{EoUP}. In the case of this research, this means 5,000 cycles before the EOL. Over this time period, the Euclidean distance from $(t_P, 0)$ to the centre of mass of the area under the metrics curve, (x_c, y_c), defines the convergence. The lower this distance, the faster the metric converges. The convergence assumes that the performance of the algorithm improves. Therefore this must be checked first.

$$C_M = \sqrt{\left(x_c - t_P\right)^2 + y_c^2} \tag{3.58}$$

$$x_c = \frac{\frac{1}{2}\sum_{i=P}^{EOUP}\left(t_{i+1}^2 - t_i^2\right)M\left(i\right)}{\sum_{i=P}^{EOUP}\left(t_{i+1} - t_i\right)M\left(i\right)} \tag{3.59}$$

$$y_c = \frac{\frac{1}{2}\sum_{i=P}^{EOUP}\left(t_{i+1} - t_i\right)M\left(i\right)^2}{\sum_{i=P}^{EOUP}\left(t_{i+1} - t_i\right)M\left(i\right)} \tag{3.60}$$

It is chosen to calculate the convergence on accuracy (Δ convergence) and precision (PI convergence). The latter is also used in the research by Eleftheroglou et al. (2018b), for example. In this way, the simplest metrics can be used to compare the prognostic performance of each model.

4

Data acquisition

In this chapter, everything related to the acquisition of data which will be used in the remaining useful life (RUL) predictions will be discussed. First, the two experimental campaigns regarding the carbon fibre reinforced polymer (CFRP) specimens and the glass fibre reinforced polymer (GFRP) specimens will be covered, containing the test setups and pre-processing of the data such that data from multiple sources can be used in one go. Next, another aspect concerning the experiments will be discussed; the computational setup. The chapter is concluded with a section on features, their handling, and aggregation.

4.1. CFRP data

The experimental data was already available for constant amplitude fatigue (CAF) and variable amplitude fatigue (VAF) loading. The data from the CAF specimens was used for the research by Eleftheroglou and Loutas (2016); Eleftheroglou et al. (2016); Loutas et al. (2017); Eleftheroglou et al. (2018a,b). The VAF loaded specimens were tested in during the same campaign, but were not included in any published research.

The eight and eleven coupons for CAF and VAF respectively, are from carbon/epoxy prepreg material, manufactured using an autoclave process. The layup of the coupons is quasi-isotropic; $[0, \pm 45, 90]_{2s}$. The coupons measure 300x30 mm, and are open-hole, or notched; all coupons have a central hole with a diameter of 6 mm.

Due to this shape, there will be stress concentrations at the edges of the hole in each specimen. When analytically evaluating this, this stress concentration would be three times the nominal stress in the specimen. Therefore, damage accumulates faster in this area. In the research of Eleftheroglou et al. (2016) on CAF, macroscopic cracks are indeed seen to originate in this area before propagating towards the sides of a specimen. This allows for damage to localise at the centre of the specimen, unlike at the clamps as mentioned above. The failed specimens, as well as the test setup, can be seen in figure 4.1.

4.1.1. Loading

The fatigue tests were performed at 10 Hz, in an Instron hydraulic universal testing machine. The machine can be seen above in figure 4.1. All specimens are loaded to the same specific load in N, under the assumption that they share the same cross-sectional area and hence endure the same stress.

The eight specimens under CAF are loaded at about 82% of their ultimate tensile strength (UTS) with $R = 0.06$, varying a little per specimen. This UTS was determined by static tests on three specimens, which averaged 42.7 kN. The frequency of the loading was at 10 Hz. Every 500 cycles, the testing was interrupted for a few seconds to take digital image correlation (DIC) measurements at a static load of 43% of the UTS.

While advanced tools such as X-ray computed tomography were not used to monitor the damage evolution in detail, Eleftheroglou and Loutas (2016) discuss two major damage mechanisms which were present; fibre splitting and delaminations. These were present in all plies and interfaces respectively, occurring in parallel, but at different rates. Eventually, near the end of life (EOL), the delamination fronts propagated out-of-plane. This caused the failure of large amounts of fibre bundles, resulting in catastrophic failure.

Another eleven specimens are tested under VAF. The different load levels for two spectra are shown in tables 4.1 and 4.2. The other spectra are not shown, since they were eventually not used in the analyses. Loads from these blocks are randomly placed in a time-series as half-cycles, meaning that a low load F_{min} will be

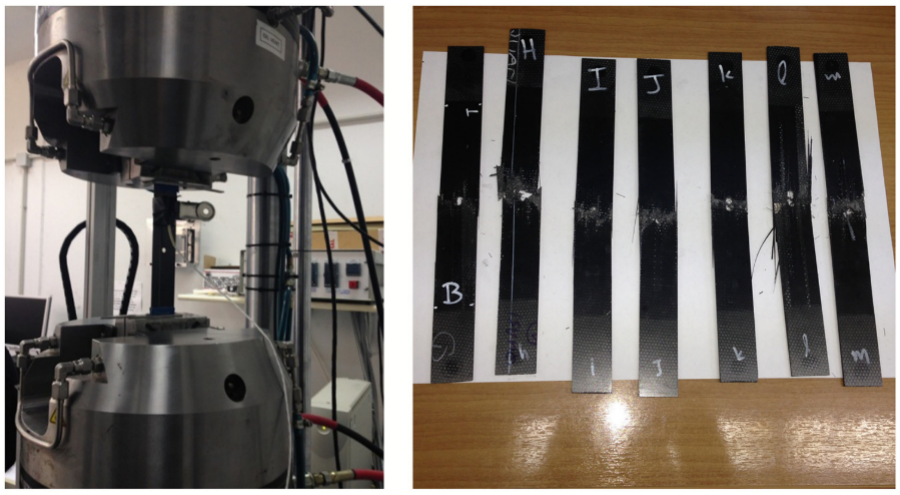


Figure 4.1: Test setup (a) for the CFRP specimens, and failed specimens (b). Taken from Eleftheroglou and Loutas (2016).

followed by a high load F_{max}, based on the method described by de Jong et al. (1973). This process is repeated in order to create different load paths, to be used on different specimens. Just as the CAF, the testing was interrupted every 250 and 500 seconds for DIC measurements, for NE6 and NE9 respectively, and held at a static load.

Table 4.1: Distribution of loads in the VAF load spectrum NE6

Cycles	F_{min} % of UTS	F_{max} % of UTS
20	-5.4	88
40	-4.5	85
100	-2.7	79
360	-1.35	74.5
1,040	0.3	69.7
3,040	4.8	65.2
16,000	9.6	60.4
83,400	14.1	55.9

Table 4.2: Distribution of loads in the VAF load spectrum NE9

Cycles	F_{min} % of UTS	F_{max} % of UTS
5,000	-3.5	88
10,000	-3.5	85
16,000	-3.5	80
18,000	-3.5	79
6,000	0.3	88
11,000	4.8	85
18,000	9.6	80
20,000	14.1	75

The expected failure pattern is likely similar to that in tension-compression (T-C) observed by Mall et al. (2009): matrix cracking, delamination, micro buckling and fibre kinking, followed by eventual fibre failure and finally, catastrophic failure. This is due to the fact that this is a combination of tension-tension (T-T) and T-C, where the compression component is relatively small. Microbuckling and fibre kinking are most likely not observed in the CAF cases, which raises the question whether these mechanisms, which are the reason for worse fatigue performance of T-C as compared to T-T, can be picked up by a data-driven algorithm which is solely trained on CAF cases, or if this is maybe not relevant for data-driven predictions.

It is expected that the total damage will be correlated with the load levels of previous cycles. As Ye (1989) noted, both the magnitude and rate of the development of damage are directly proportional to the load level at any number of load cycles in the first two stages of damage development (see figure 2.4). Therefore, the applied load could possibly be used as a feature in a prognostics algorithm.

4.1.2. Acoustic emission data

The acoustic emission (AE) data of both the CAF and VAF data-set was captured using an AMSY-6 Vallen system, with one AE sensor attached to the specimen. A threshold of 50 dB was found to be enough to filter out background noise. The signal itself was pre-amplified by 34 dB, and a band-pass filter of 20-1200 kHz was applied. Each event is captured in six features, listed in the table 4.3 below. The unit for energy is an energy unit (eu), where 1 eu = 10^{-14} V^2s.

Table 4.3: Recorded AE parameters for the CFRP specimens

Parameter	Unit
Rise time	μs
Duration	μs
Energy	eu
Counts	-
Amplitude	dB
Root mean square (RMS)	mV

4.1.3. Data matching

The AE data and load paths were initially from different files and therefore had to be matched. First, so-called load blocks were identified in the (spectrum) load paths, based on the breaks between them. For all specimens except A010 (250 cycles), the blocks contained 500 cycles. The blocks of specimen A010 were treated per two, such that they contained approximately 500 cycles as well. Because specimens were held at a static load during the breaks, the change of load to the static load was documented as a half cycle. Therefore in the remainder of this thesis, there are references to numbers of cycles which are not exact multiples of 500.

From the log files of the fatigue machine, the start- and end times of the test were extracted. Then, the respective load path was shortened to this duration. Because failure often occurs with a load-block, this block was included.

Next, the AE data could be merged on this load data. This was done under the assumption that the AE system started recording at the time loading started. From inspecting the correlation between the load paths and AEs, this was indeed the case. An example of this is shown for specimen A001, in figure 4.2. The times without AE activity align exactly with the load breaks.

Unfortunately, several tests during the testing campaign were found to be unsuccessful while examining the data. The decisions to dismiss specific specimens was based on the log files from the fatigue machine, as well as those from the AE records. For several specimens, the life of the specimens was extremely short. Therefore it is assumed that something went wrong during these tests. For specimens A003 and A015, the life was longer, but there seemed to be a mismatch in the AE data and the applied loading. The case of A015 is shown below in figure 4.3. As opposed to A001 in figure 4.2, it can be seen that the pauses between the load phases are not only not in phase with pauses in the AE, but the load phases and pauses also seem to have a different length than those of the AE hits. Therefore, this specimen was discarded as well. An overview of all tests is shown below in table 4.4.

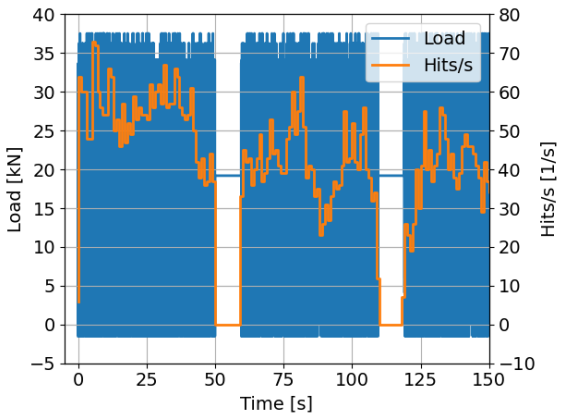


Figure 4.2: Load path and AE hits for specimen A001 in the first 200 s

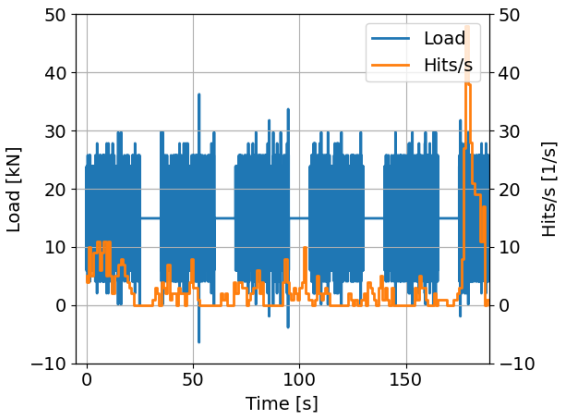


Figure 4.3: Load path and AE hits for specimen A015

For the specimens which were analysed, the failure times and cumulative energies at failure (used for the Gaussian process (GP) regression) are shown below in table 4.5. The failure times are rounded per 500 cycles in order to improve readability.

Table 4.4: Overview of all specimens and reason for dismissal in the CFRP testing campaign

Specimen	Type	Load sequence	Reason to dismiss
A001	variable	NE9	
(A002)	variable	NE9	Life of 2 cycles
(A003)	constant		Two load log files, both not showing coherence with AE data
(A004)	variable	NE9	Life of 2 cycles
A005	variable	NE9	
A006	variable	NE9	
A007	variable	NE9	
(A008)	variable	NE9	Life of 7 cycles
A009	constant		
A010	variable	NE6	
A011	constant		
A012	constant		
A013	constant		
A014	constant		
(A015)	variable	NE3	Short life (140 cycles), no coherence with AE data
A017	variable	NE9	
A019	constant		
A020	constant		
(A021, A021b, A021c)	variable	NE9	Multiple interrupted load log files, as well as interrupted AE files

Table 4.5: Overview of CFRP specimens used in this research, their EOL, and cumulative energy at failure

Specimen	Type	EOL [cycles]	Cumulative energy [Meu]
A001	variable	59,500	606
A005	variable	65,000	959
A006	variable	188,000	1,785
A007	variable	56,500	605
A009	constant	29,000	195
A010	variable	8,000	262
A011	constant	68,000	755
A012	constant	25,000	701
A013	constant	22,000	1,020
A014	constant	37,500	541
A017	variable	105,000	1,318
A019	constant	57,000	496
A020	constant	13,000	331

4.2. GFRP data

In the period from February to March 2020, T-T fatigue tests on GFRP specimens have been performed in by Danmarks Tekniske Universitet (DTU) Wind Energy, in their composite labs in Risø.

The specimens were made of non-crimp glass fibres, reinforced with epoxy. They have a $[\text{biax}/0°/0°]_s$-layup; the primary load-carrying part of the laminates are 4 uni-directional (UD) layers, surrounded by bi-axial layers. This results in specimens which have a thickness of approximately 4 mm. The shape of the specimens is a so-called butterfly shape. It is the same as in a previous research campaign by Jespersen and Mikkelsen (2017). The dimensions can be seen in figure 4.4. In such a shape, the peak stresses are concentrated in the 60x15 mm gauge section in the centre, forcing failure in this part. The specimen tapers both in width and thickness towards the ends, where it is held in the hydraulic clamps of the fatigue machine. An AE sensor is attached on the gauge section, as well as two 50 mm strain gauges. The entire setup is shown in figure 4.5

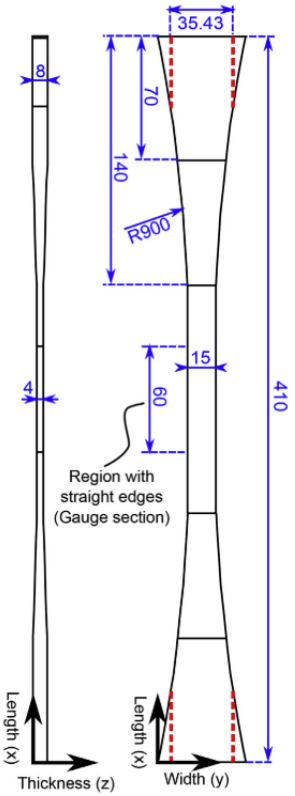


Figure 4.4: Dimensions of the test specimen (taken from Jespersen and Mikkelsen (2017))

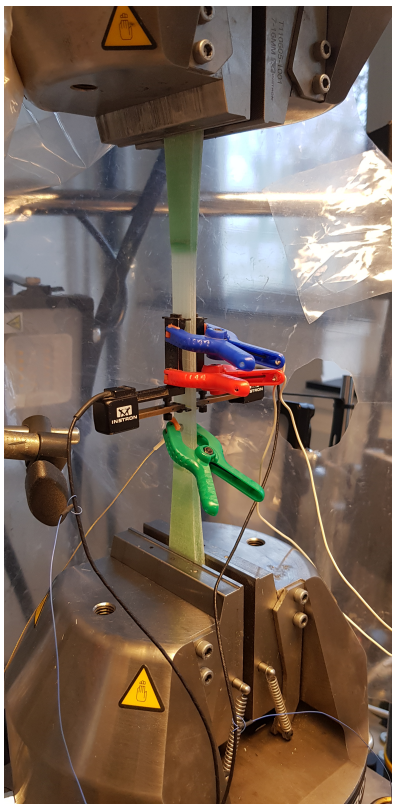


Figure 4.5: Testing setup

4.2.1. Loading

The tests were performed using an Instron 88R8501 machine, with a 100 kN load cell. The aim was to load the specimens for different maximum strain ε values, shown in table 4.6. The load sequence consisted of two parts. First, the Young's modulus of the specimen was determined using a static test. During ramping up the load, the initial Young's modulus E_0 was determined (table 4.6). With the predetermined ε, the maximum stress σ_{max} could be determined using Hooke's law:

$$\sigma_{max} = E_0 \varepsilon \tag{4.1}$$

During this ramping up, the load was increased up to 90% of σ_{max}, followed by loading to 0.30% strain. This load sequence is plotted for specimen 6, below in figure 4.6. Next, the specimens were subjected to CAF loading at 5 Hz, until failure. Following a stress ratio R_σ of 0.1 for all specimens, the minimum load F_{min} was set at $R_\sigma F_{max}$. The S-N plot of the coupons in this experimental campaign is plotted in figure 4.7. Because the fatigue machine was not able to reach the exact same stress values for every cycle, especially in the first 100 cycles, the median values have been plotted in this figure. The different stress levels already show large differences in the lifespans of the different specimens.

Table 4.6: Load settings and properties of the GFRP specimens

Specimen	Maximum strain [%]	Young's modulus [GPa]	Area [mm^2]
6	0.90	35.07	70.35
7	0.95	34.91	70.53
8	0.87	35.16	70.67
9	0.95	35.14	70.14
10	1.00	35.70	70.78
11	1.20	35.47	70.33
12	1.10	34.88	70.52

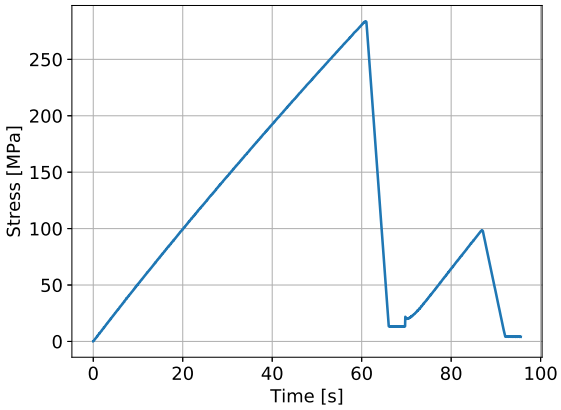


Figure 4.6: The static test for specimen 6

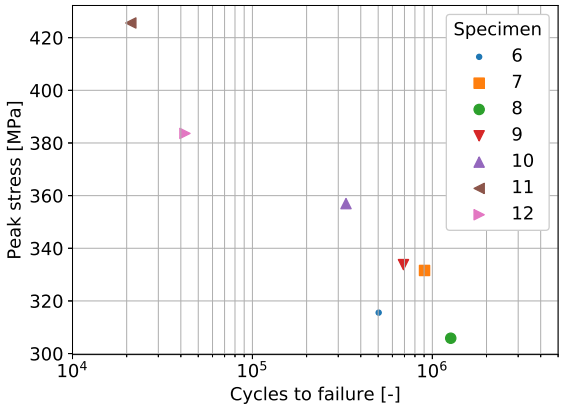


Figure 4.7: S-N plot for the GFRP data-set

4.2.2. Acoustic emission data

The AE events were captured by a MicroExpress Digital AE System, with a single AE sensor in the gauge section of the tested specimens. A threshold is set at 55 dB in order to filter out background noise. The AE events are amplified using a 2/4/6 pre-amplifier with 40 dB gain. The system captures eight parameters, shown below in table 4.7. Most of these parameters are discussed in section 2.2. Measured area of the rectified signal envelope (MARSE) is a dimensionless waveform characteristic. To calculate the MARSE, the AE signal is first rectified. Next, the envelope over this rectified signal is determined. The area under this envelope is then the MARSE.

Table 4.7: Recorded AE parameters for the GFRP specimens

Parameter	Unit
Rise time	μs
Duration	μs
Frequency	kHz
MARSE	-
Counts	-
Amplitude	dB
Absolute energy	aJ
RMS	mV

4.2.3. Data matching

Due to the fact that the AE system and the fatigue machine were not connected or synchronised in time, the fatigue load sequence had to be matched to the AE events. This was done based on the assumption that the first high-energy AE event was associated with the final failure of a specimen. At the EOL, multiple high energy AE events were measured. Because most were within a second of each other, the first occurrence was taken, assuming that this depicted failure.

Figure 4.8 shows the difference in timestamps for the two machines, as well as the measured stiffness and absolute energy of AE events. The large jumps in stiffness indicate failure; the machine stops right after this behaviour. Two AE events occur which have energies above 10 pJ (10^7 aJ), which are considered high-energy events, considering the scale of other events which occur a few minutes before these events. Hence, the difference in time between the first high-energy event and the last fatigue machine measurement is used to match these two files. While this method is not accurate to several milliseconds, it is accurate enough for this research; since all AE events will be grouped into bins of 500 load cycles, which at 5 Hz take 100 s.

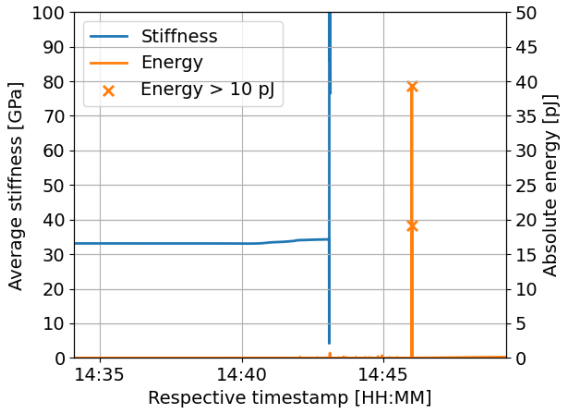


Figure 4.8: Measured stiffness and AE events for specimen 6, near its EOL

4.3. Computational setup

While experiments above were conducted in a lab, the computational efforts behind this thesis also have to be regarded as an experiment.

All calculations were done in Python 3.7. Python, together with Spyder (an integrated development environment) and a group of packages (such as Numpy, Scipy, matplotlib), is obtained through Anaconda, which is free package management software. Python is selected due to the fact that it is open-source and free to use. Furthermore, a wealth of information and packages for Python can be found online, especially regarding data analysis and machine learning.

Two hardware systems are used. The first is a laptop with an i5 processor, 20 Gb RAM and a 1 TB hard disk. This laptop is primarily used for feature processing, results analysis and training the statistical model. The second system is the high performance computing (HPC) cluster from DTU. The cluster, available for students and employees of DTU, lets users queue jobs of up to 24 hours, on up to roughly 100 cores at a time (based on best practices). Through the use of job scripts, multiple cores could be run in parallel. All training sessions for the GP regression, and the cross-validation runs for the recurrent neural network (RNN) could be performed in the timeframe of roughly a week.

4.4. Features

This section covers the features which were extracted from the experimental data. First, it is briefly discussed how all features are standardised before they are fed into the models. Next, the acoustic emission features are discussed, and especially their aggregation to come to new features. The section is concluded with features which are specific to the CFRP or GFRP data-set.

4.4.1. Standardisation

Before features are fed into either the GP regression or the RNN, they are first standardised (equation (4.2)). This means that for each feature vector $\boldsymbol{x}$ the mean μ is subtracted, followed by a division over the standard deviation σ of the feature.

$$\boldsymbol{x}_{i,s} = \frac{\boldsymbol{x} - \mu_i}{\sigma_i} \tag{4.2}$$

In this way, all values are centred around 0 and are all in the same, low order of magnitude, generally

between -5 and 5. This is done because, in this way, there are no massive differences in the scales of input features, which may offset the models; the weights of the models do not have to be extraordinary high or low. The means and standard deviations are saved in order to scale the model predictions, which are also standardised, back to actual values.

The standardisation is, of course, performed based on data which the model is 'allowed to see'. If for example a GP regression would be trained on series A and B, and tested on C where 50% of C is already known, the μ and σ would be determined on A, B, and the first 50% of C. In this way, the methods can still be used for in-situ prognostics.

4.4.2. Acoustic emission data

As discussed above, all events are grouped in bins of 500 cycles. It was decided not to perform wavelet analyses on the AE events. Therefore, the observed events and their describing parameters make up the AE data which will be used. At first, these AE events seem to be of a random nature. Take for example the number of events per bin, shown in figure 4.9. Although there is a high concentration of events at lower numbers of cycles, this time-series is hard to analyse. It can, however, be fed into a RNN, which can identify complex relations in the data. These time-series do not seem to be very suitable for regression purposes, since they are noisy and are not monotonic.

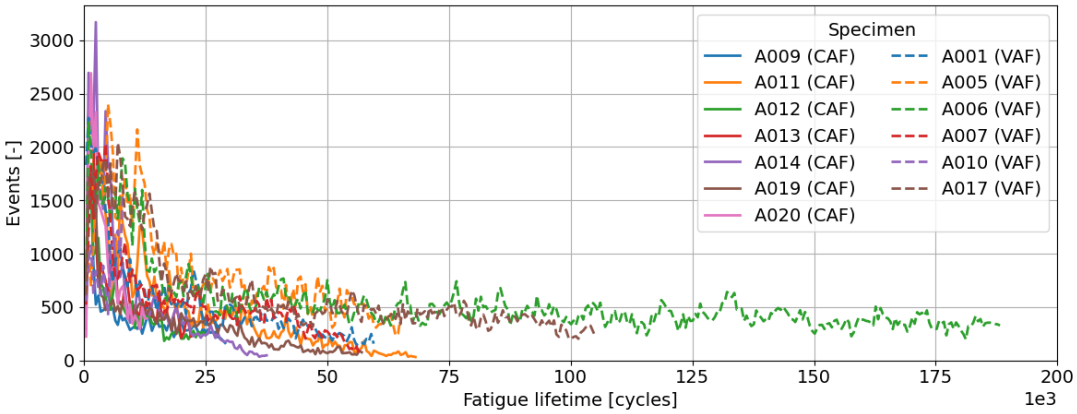


Figure 4.9: AE events for the CFRP specimens, per bin of 500 cycles

To get a better understanding of the events inside the specimens, a set of features which show monotonic behaviour would be more appreciable. Monotonicity is when the function is not decreasing or increasing. Hence, a monotonically non-increasing function is where each value $y_i \leq y_{i-1}$. Vice versa, a monotonically non-decreasing function is where $y_i \geq y_{i-1}$. The available data can be easily altered to make it monotonic; by taking the cumulative sum of all events over time. This operation gives a better view of events over time, not only for regression models but also for humans. Taking the same series as above, this results in figure 4.10. These series are better to perform a regression on, due to their increasing values. This is the reason why in the GP regression, a cumulative parameter is used.

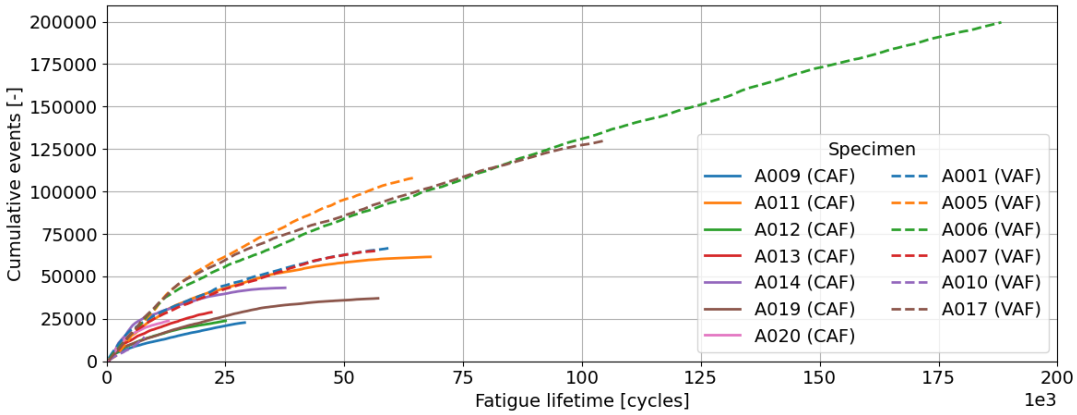


Figure 4.10: Cumulative AE events for the CFRP specimens

Next, the different studies discussed in section 2.2 classify damage mechanisms according to AE parameters. Because each study uses different bin sizes to classify these, as well as different parameters, it is decided to take the mean of an AE parameter per event, within the bins of 500 cycles. If there is a trend in these series, it could be picked up by the RNN. This would, however, not immediately imply that a similar failure mechanism is the cause of this trend.

Three aggregated parameters are introduced. The research of Eleftheroglou and Loutas (2016) uses a windowed cumulative rise time/amplitude on the CAF part of this data-set. This is, however, also a noisy and non-monotonic feature. Therefore, its cumulative sum was used in this research.

From further feature exploration, it was found that when dividing this cumulative feature over the number of passed cycles, a feature was created which converges later in life. This feature is thus calculated by taking the rise time/amplitude ratio per bin of 500 cycles. Of this series, the cumulative sum is taken. This results in the cumulative rise time/amplitude discussed above. An extra step is now to divide this over the passed cycles at the respective point in time. The feature is shown in figure 4.11. The same operation is performed over the energy/counts ratio, in figure 4.12. There is no motivation from a physical standpoint to introduce these features. These features also do not aid in the prediction of the final failure. Although they converge to roughly the same y-value, the time which they spend at this value varies. However, it can be said that if they have not converged, this could mean that the specimen is not near its EOL yet. Therefore, these features contain information about the fact that a specimen is in the early stage of its life.

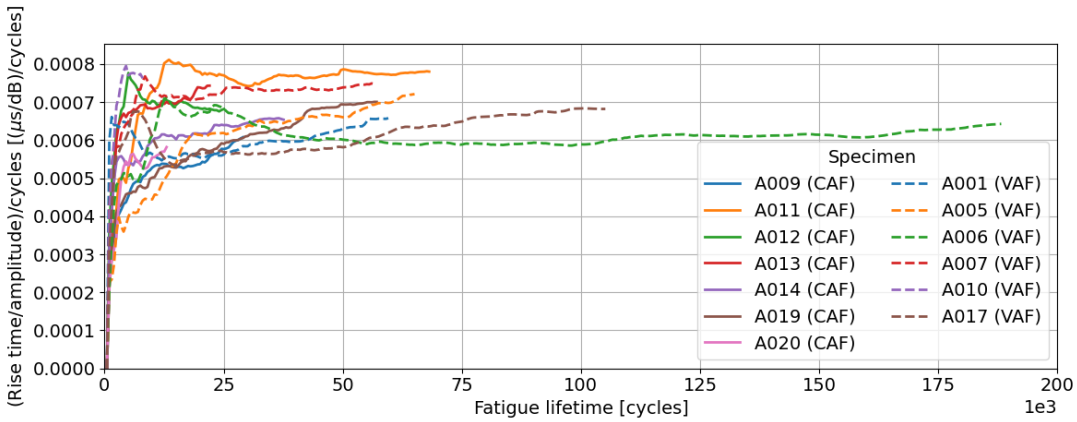


Figure 4.11: Cumulative rise time/amplitude, divided by the passed cycles for the CFRP specimens

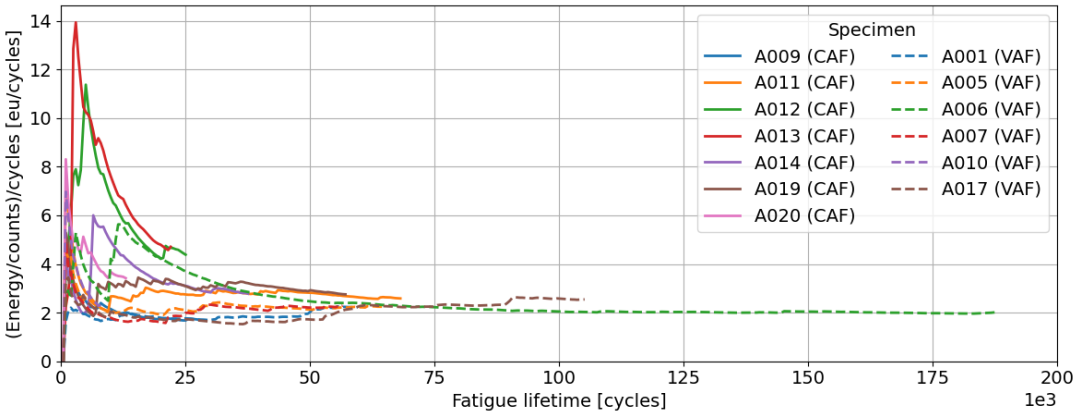


Figure 4.12: Cumulative energy per count, divided by the passed cycles for the CFRP specimens

From these plots, it can be seen that not only the lifetimes of the specimens are varying, but the trends and scales in the AE data as well. As expected, this will become challenging in the RUL predictions, likely increasing the uncertainty that goes with predictions.

4.4.3. CFRP specimens

For the CFRP specimens, a combination of AE features and load data can be used as input. All AE parameters, as well as their cumulative sum, mean per event, and the three aggregated features result in 23 AE features.

For both CAF and VAF loading profiles, each 500-cycle time-series of the load had to be converted to a set of features. While the 500 highs and lows per block could in theory all be linked to a set of input nodes in a RNN, this would result in a large number of inputs and therefore trainable parameters. Another option could be to feed these time-series into a preceding neural network (NN), which would then output a single value to the main RNN, representing a sort of impact related to the loading, if the model is able to distinguish this.

This would, however, overcomplicate the matter for now. Therefore a more simple, physics-based approach is taken. From Miner's rule, a cumulative damage model, a piece of material can endure a number of cycles in different stress ranges n_i. Based on how much cycles the material can sustain in this range N_i, the damage fraction is calculated as n_i/N_i. The damage fractions can then be summed up for each stress bin, resulting in the total damage. Because it is unknown how much cycles the coupons can endure in each stress bin, the damage fractions cannot be used as an input in the models. What is possible, however, is giving feeding the model numbers of cycles in specific load ranges.

The load ranges were grouped in blocks of 5 kN, resulting in 7 bins, linearly ranging from 5 kN to 40 kN. Blocks of 5 kN were chosen as not to give too much extra inputs to the RNN. The numbers of half-cycles per load range were determined using rainflow counting. This is made possible by fatpack, an open-source fatigue analysis package for Python (Frøseth and Capponi, 2019). A histogram of these blocks is shown in figure 4.13. The majority of the loads is situated in the three highest bins. Specimen A010 is responsible for the largest portion of the 15-20 kN bin, due to the different load path (NE6). A handful of cycles can be found in the bins from 5-10, 10-15, and 20-25 kN, containing respectively 14, 171, and 1506 cycles.

Now, these seven additional features can be used in the predictions, each containing a number of cycles in this load range. In the perfect scenario, a RNN will attach higher weights to the inputs of the higher load bins. In the case where training data consists of solely specimens under CAF, the loads are all in the same bin, with other bins containing zeros. Because this would not lead to any added value, they are not included as input parameters for this training case.

The AE features combined with the load bins and the elapsed cycles result in a total of 30 input features for models trained on VAF and the combination of CAF and VAF data. When trained on solely CAF, there are 24 input features.

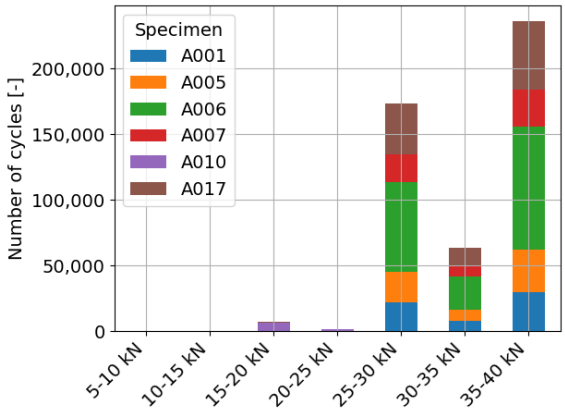


Figure 4.13: Histogram of the number of cycles in different load bins, for the CFRP specimens under VAF loading. The bins of specimens are stacked upon each other.

4.4.4. GFRP specimens

In the GFRP testing campaign, there are two additional AE parameters; the frequency and MARSE. This results in a total of 29 features, using the same aggregation methods as above.

When again investigating the cumulative events (figure 4.14), the scale of the difference in failure times between the specimens can be seen. When compared to the CFRP data, there are periods with low AE activity in these specimens, as well as a difference in the scales of the number of events, and fatigue lifetime.

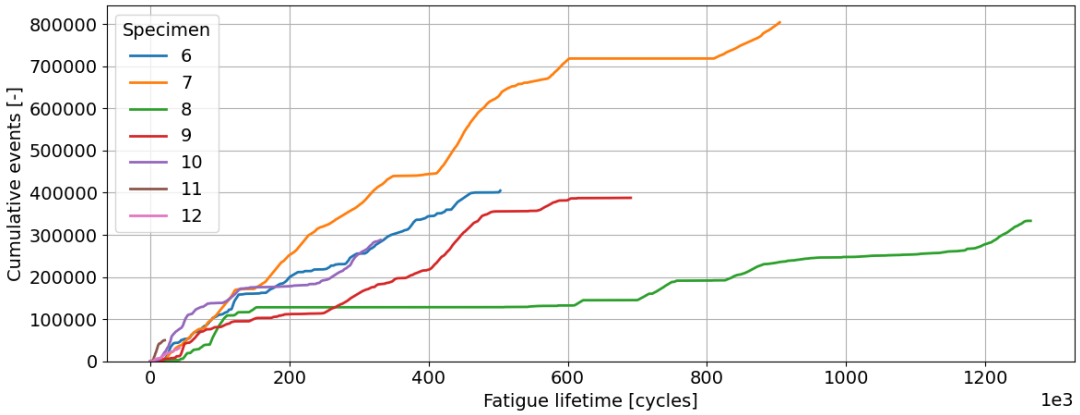


Figure 4.14: Cumulative AE events for the GFRP specimens

In order to possibly enhance these predictions, load features can be used as additional inputs. Because the fatigue loading is constant for this data-set, the minimum- and maximum load are used as inputs, as well as the stress ratio. This leads to a total of 33 features for the GFRP data-set when the fatigue lifetime is also included. When solely training on AE data and fatigue lifetime as time-scale, 30 features will be used.

5

Results and discussion

This chapter covers the results of each model category, as well as the results of the comparison between the different models. First, the results from each model are discussed, not only the remaining useful life (RUL) predictions, but also results of intermediate steps in order to get a grasp of the inner workings of these models. Next, a comparison is made between the models, determining which is most fit for prognostics on specimens under variable amplitude fatigue (VAF). Finally, the results from the case study on varying load levels are discussed.

5.1. Statistical model

This section covers all results from the statistical model. First, the RUL predictions are analysed for this model. Some predictions are shown in this section, but due to the number of specimens, the ones not mentioned in this section can be found in appendix B.1.1. This is followed by a comparison of the impact of training data on the quality of the predictions in order to answer the research questions. The section is concluded with a discussion on this model category.

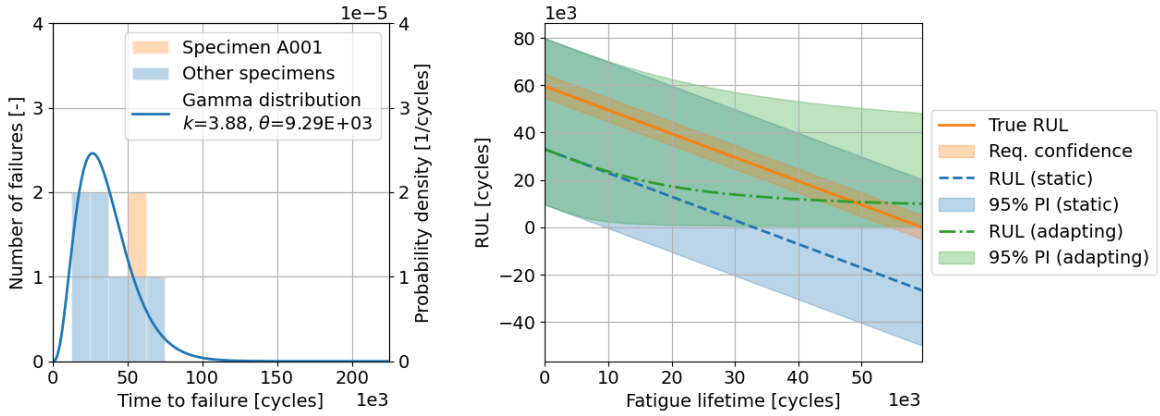
5.1.1. Remaining useful life predictions

From figure 3.1a in section 3.1.1, it can already be concluded that the failure times of the specimens are scattered. Furthermore, no clear distinction can be spotted between specimens under VAF and constant amplitude fatigue (CAF) loading. Therefore, it can be expected that the precision of the predictions will be relatively low using this method. Furthermore, since the failure times of CAF loaded specimens are relatively concentrated, lower precision can be expected for cases when the model is based on VAF (together with CAF) data. The accuracy may be positively impacted, however.

The statement concerning precision can immediately be confirmed by taking one of the predictions where the distributions are based on CAF data only. Take for example specimen A001 in figure 5.1. Figure 5.1a shows the fitted probability density function (PDF) on the CAF data, as well as where specimen A001 lies. Again, note that specimen A001 is not used for fitting this PDF, as that would insinuate having prior knowledge of its failure. It can be seen that A001 is not a complete outlier. Combined with the fact that A001 lies to the right of the peak of the PDF, results in a conservative static prediction.

The 95% prediction interval (PI) of both the static and adapting predictions, however, is extremely wide. With an end of life (EOL) of just below 60,000 cycles, having PIs of 70,000 cycles is far too great, especially when put into contrast with the earlier defined required confidence of $\pm$5,000 cycles. This results in the fact that the probability mass contained in the required confidence interval is low.

The behaviour of the adapting predictions can also be seen in figure 5.1b. As time progresses, the prediction converges, but not to 0 cycles. This can be explained by equation (3.6) in section 3.1.3. This cumulative distribution function (CDF) is always zero at the point of observation s. Therefore the expectation, as well as the low bound of any PI, will always lie in the future; the specimen is predicted to never fail. The convergence of the PI can be explained by the hazard function of the gamma distribution, which converges later in life. Taking a Weibull distribution with a shape parameter >1, for example, would lead to convergence in the PIs, although the function will also not cross zero.

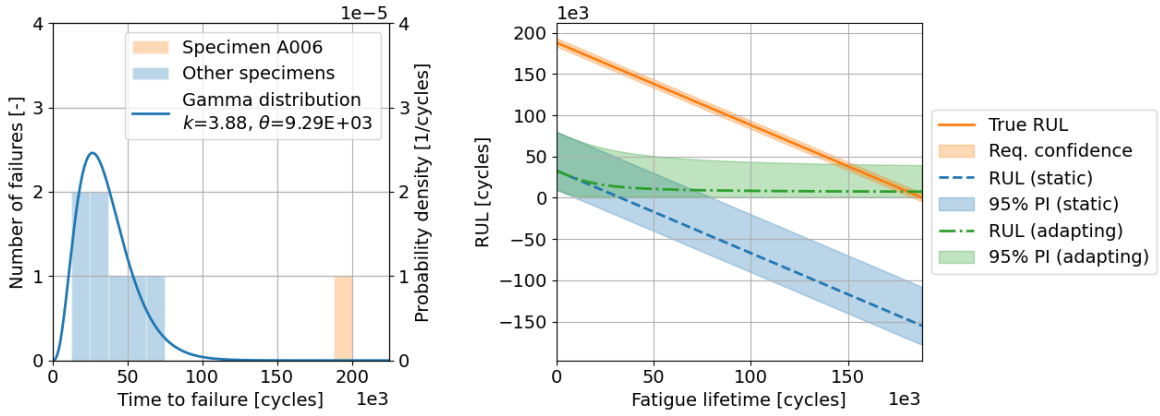


(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure 5.1: Results of the statistical model, trained on CAF data, tested on specimen A001

Whereas specimen A001 is an average specimen, there are also outliers in the data-set. Being an outlier has a great influence on the accuracy of the model. The predictions for specimen A006, trained on CAF data, are shown in figure 5.2. Since this model is trained on CAF only, the PDF shares the same parameters with the PDF for the prediction above. The accuracy of these predictions is low, which can already be determined from the fact that this specimen lies far away from the majority of the probability mass in figure 5.2a. The fact that this prediction is far too conservative may lead to excessive maintenance, but not to unexpected failures of components, should this model be used in practice.

Furthermore, the adapting prediction in figure 5.2b makes one remind of the saying: 'a broken clock is right twice a day'. Because it converges, the expected RUL will always cross the actual RUL if the EOL of the specimen is located to the right of the training data's median.

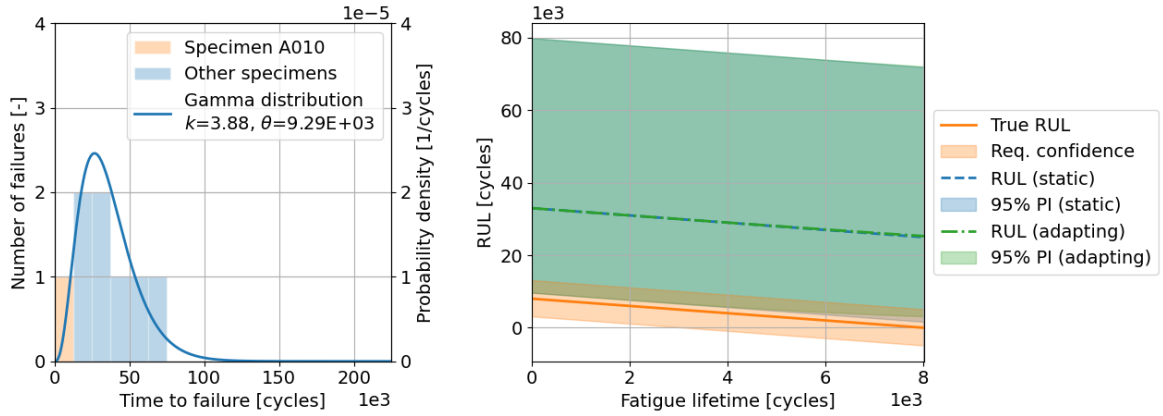


(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure 5.2: Results of the statistical model, trained on CAF data, tested on specimen A006

The statement above is of course not the case when a specimen's EOL is shorter than average, as in for example specimen A010 in figure 5.3. In this case, the adapting prediction's convergence will never hit the actual RUL. This is also one of the specimens for which the prognostics using this method is dangerous. Whereas the predictions for the specimens above are conservative, both the static and adapting predictions are too optimistic in this case. In real-world applications, this could lead to catastrophic failures of components before maintenance or replacement of the component is expected to be required.

From this initial investigation on specimens which are trained on CAF data, it can be concluded that there are multiple flaws to both the static and adapting predictions. First, the PIs are very wide, spanning 70,000 cycles for models trained on CAF data. Next, the accuracy is purely dependent on the distribution of the training set.



(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure 5.3: Results of the statistical model, trained on CAF data, tested on specimen A010

This is -of course- also the essence of a purely statistical model. Finally, the adapting predictions are not useful for prognostics. Since they will essentially never expect failure, a decision for maintenance cannot be made based on these predictions. Therefore for the comparison of training data below, the adapting predictions are not included in the analysis. For the reader's interest, tables containing the performance metrics of the adapting predictions can be consulted in appendix B.1.2.

5.1.2. Effect of training data

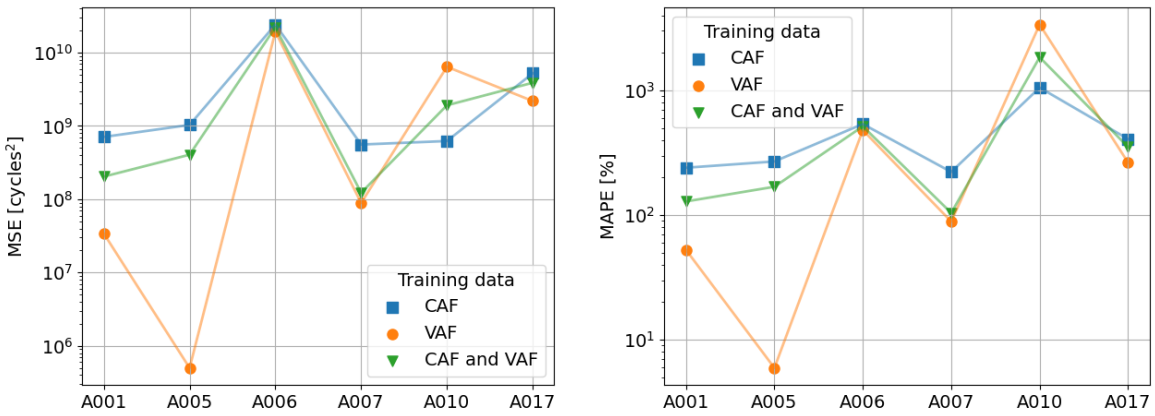
In order to analyse the impact of different training data sets on the static predictions, three prognostic measures are used here; mean squared error (MSE), mean absolute percentage error (MAPE) and cumulative bounded probability mass (CBPM). All three metrics are shown in figure 5.4. Convergence is not analysed here, since static predictions do not converge. Furthermore, due to a lack of prognostic horizon (PH) in all predictions, this is also not taken into account in the comparison. The fact that there is no PH is, however, something which should not be overlooked. Finally, because the predictions are not converging/diverging, the cumulative relative accuracy (CRA) is essentially the same as the MAPE and is therefore also omitted in the analysis. Tables containing all metrics are shown in appendix B.1.2.

A few specimens stand out. First, specimen A006 has the highest MSE and generally lowest CBPM. Not only is this an outlier compared to the CAF specimens, as shown in figure 5.2a, but also to the VAF data, although less. Its CBPM shows that having VAF data as training data is best for this metric. This is because with this data-set, the PDF is wider, and therefore more probability mass is located at the tail, where the actual RUL of A006 is located.

The second specimen which stands out is A005, with low MSE and MAPE compared to the other specimens, when trained on VAF data. Specimen A005 is a specimen which lands exactly on the expectation of the PDF drawn on the VAF data, thus resulting in low MSE and MAPE. With a MAPE of 6%, this could be considered an extremely good prediction. Yet, the CBPM is not higher for this training set at all. This phenomenon is due to the low precision of this type of model; the accuracy does not make a significant difference on the CBPM since the PDF of the predictions are so spread out.

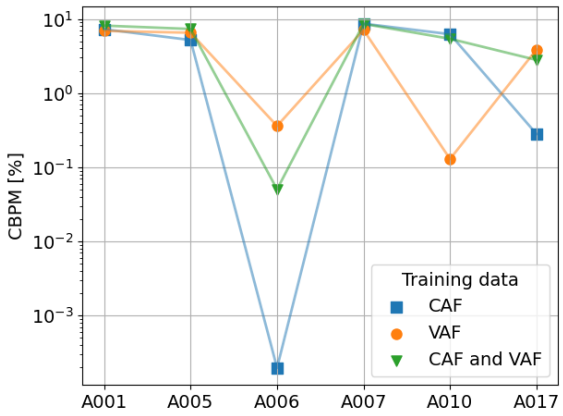
Finally, specimen A010 is the one with the shortest life. Because the distribution of CAF failure data is centred at a relatively low number of cycles, the predictions for this low outlier are relatively good. In the VAF data-set, however, specimens are more spread out and have higher numbers of cycles at failure. This causes worse predictions for this data-set on specimen A010.

At first glance, the VAF data seems to provide better predictions than the other two training sets. This result is, however, very dependent on the testing specimen, as specimen A005 and A010 tell, and could change when more specimens are tested. This can be seen in the difference in performance over all the specimens, as well as the non-consistency in the best models per specimen. Therefore, training on the most available data, thus on CAF and VAF, can result in 'safer' predictions, which are more likely to cover outliers.



(a) MSE

(b) MAPE predictions



(c) CBPM

Figure 5.4: Three performance metrics of the static statistical model, plotted on log-scales

5.1.3. Discussion

The results from this statistical analysis show that there is a significant spread in the data, causing low precision of predictions, as well as high sensitivity for outliers. When comparing the static and the adapting models, it can be concluded that the adapting versions are not fit for prognostics, even though the performance metrics for the adapting model may be better in specific cases. This is because the adapting predictions never predict failure, and can therefore not be used in practice. This said the static predictions could be used in practice. However, the accuracy and precision of this method are low due to the large spread between the samples. Therefore, it should be used very conservatively. In fact, this method is used in the industry, through the means of safety factors which are based on the distribution of failure data.

A significantly larger sample set would need to be tested in order to determine whether training on CAF, VAF or the combination may be better for the results of this method. As of now, the results are too sensitive to the behaviour of specific specimens in order to make the right judgement. This statistical model is, however, a good baseline model, in order to compare the other models in this thesis with.

An idea for a follow-up study would be to perform more of these CAF and VAF tests and determine the difference in statistical distributions between these two data-sets. If the difference is negligible, then it would not matter if a specimen would be trained on CAF and/or VAF data. If there is a difference, it would be more suitable to train on VAF only, since this would represent the distribution of the testing specimen better. The test results would, however, be restricted to this load level and setup of VAF load path.

5.2. Gaussian process regression

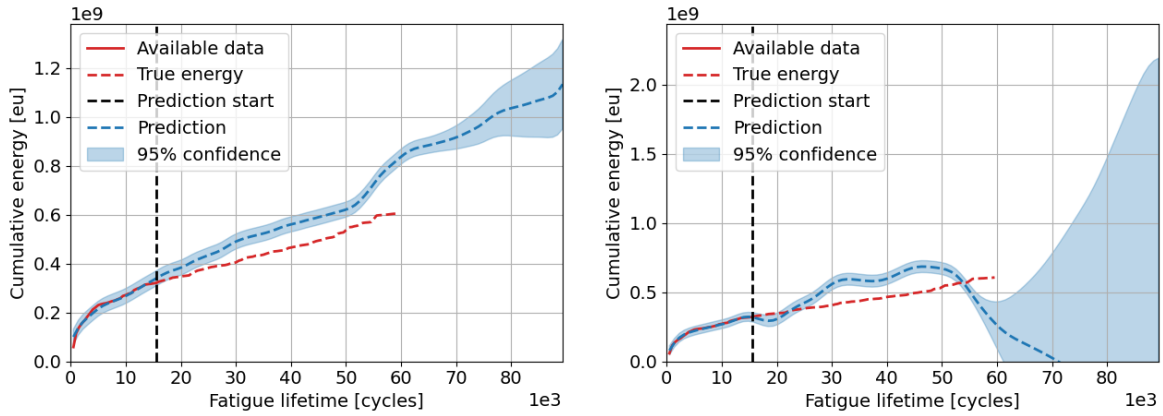
This section covers the results from the Gaussian process (GP) regression. First, the performance of the Ma3+lin and Ma5+lin kernel combinations is compared. This is followed by the analysis of RUL predictions and the effect of the correlation adjustment. The effect of training data on the overall performance is then covered. Finally, there is a discussion on the potential and shortcomings of the model. All figures which are relevant are covered here; others have been included in appendix B.2.

5.2.1. Kernel performance

Two kernel functions were used, as well as two different training sets. Therefore, before comparing the performance of the RUL predictions, it is good to know what happens in the preceding phase in the predictions. At each point in time when a prediction is made, a time-series of the cumulative energy is predicted. The quality of these predictions can be assessed by taking the MAPE over the actual predicted part of the series. The comparison is based on the MAPE since MAPE is a scale-independent measure, unlike the MSE.

Take, for example, specimen A001 after approximately 15,500 cycles. The predictions of cumulative energy are shown in figure 5.5. The models in the figure are trained on VAF data and use the Ma3+lin and Ma5+lin kernel combinations. The first observation which can be made is that the former model performs better than the latter. Note that these predictions are not only made based on this series but also on the cumulative energy until failure from the other VAF specimens.

The better performance of figure 5.5a is not only in the sense that its error is lower than that in figure 5.5b (MAPE of 18.5% versus 25.0%), but especially because it predicts physical behaviour, i.e. the cumulative energy prediction is monotonically increasing. While this non-physical behaviour is undesired, this can unfortunately not be enforced through a GP.



(a) Ma3+lin kernel combination

(b) Ma5+lin kernel combination

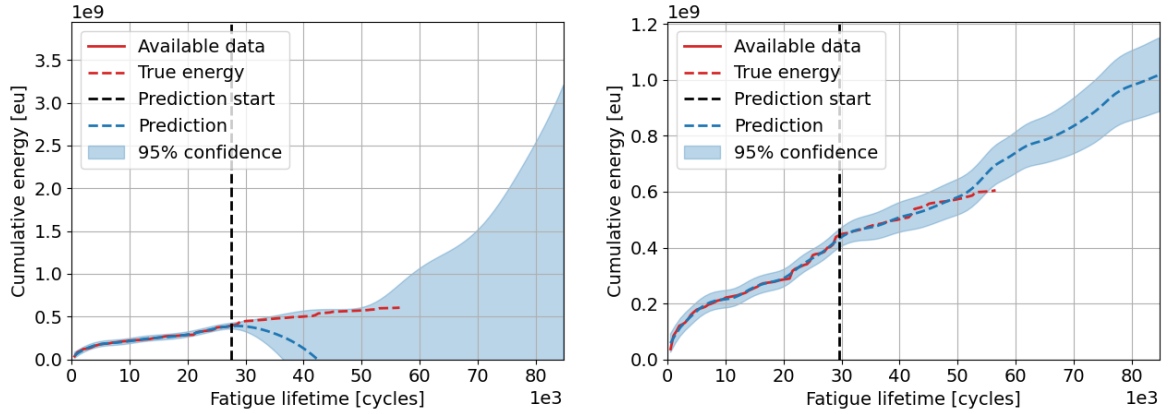
Figure 5.5: Cumulative energy predictions by the GP for specimen A001 at 15,500 cycles, trained on other VAF data

Another issue is that of the local optima, as discussed in section 3.2.3. Two consecutive predictions are shown in figure 5.6. The log-likelihood of the second prediction (figure 5.6b) is roughly 20% higher, and the prediction is much better (MAPE of 2.6% versus 114.5%). It is therefore likely that all 20 replicates of the prediction at around 27,500 cycles have stumbled upon local optima, with this replicate resulting in the best possible log-likelihood. Due to the high dimensionality of the hyperparameter search space, it is impossible to know where and if a better (global) optimum can be found.

In order to get a grasp of the performance of the different kernel combinations and the effect of training data, a comparison was made between all combinations. The results of this analysis are shown in figure 5.7. Note that the MAPE is plotted on a logarithmic scale, due to the vastly different magnitude of the errors between different predictions. This shows that although median MAPEs are in the order of 10%, there are numerous outliers for each combination. A possible explanation for these outliers is that their models' hyperparameters were found at local optima.

When inspecting the values for the median, and 1st and 3rd quartiles, a tentative judgement can be made. These three metrics are all lowest for the Ma3+lin kernel combination for models trained on only the VAF data-set. Because these numbers are relatively close, a clear distinction cannot be made yet. Hence, the RUL

predictions of all four will be analysed. The exact values can be found in appendix B.2.1.



(a) Prediction after approximately 27,500 cycles, $\log p\left(\boldsymbol{y} \middle| \boldsymbol{X}\right) = 1479$

(b) Prediction after approximately 29,500 cycles, $\log p\left(\boldsymbol{y} \middle| \boldsymbol{X}\right) = 1817$

Figure 5.6: Cumulative energy predictions by the GP with Ma5+lin kernels for specimen A007, trained on other VAF data

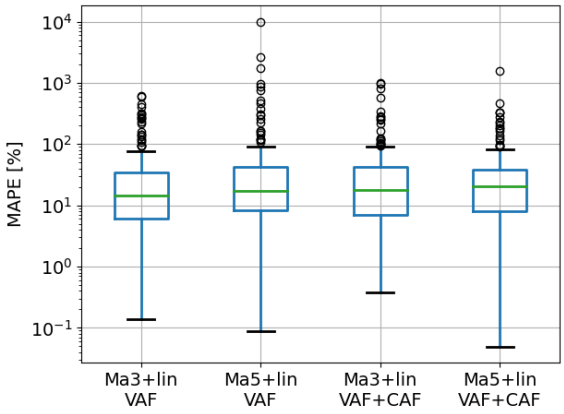


Figure 5.7: Box plot of the MAPE of the cumulative energy predictions by the GP model, grouped by kernel functions and training data. The green line indicates the median, with the box encapsulating the 1st and 3rd quartiles Q_1 and Q_3. The whiskers extend up to $1.5(Q_3 - Q_1)$. Outliers are plotted as dots.

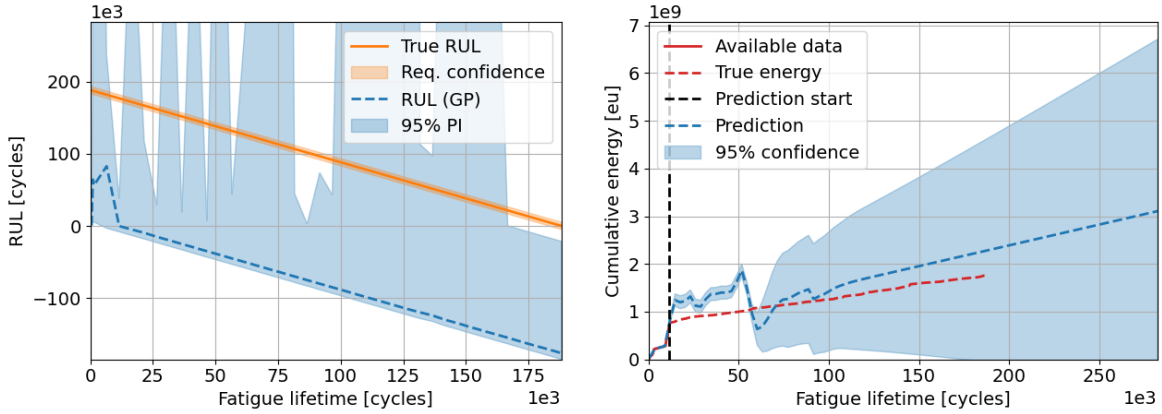
5.2.2. Remaining useful life predictions

With predictions for the cumulative energy made at each point in time, these series could be converted to RUL estimates. The main issue from the predictions in section 5.1 arises again; the large spread in the failure data. This could also already be seen in the histogram of the cumulative energies at failure in figure 3.11. This spread impacts the PDF of the cumulative energy threshold for failure, and therefore also the PI width of the RUL predictions. A few varying examples of RUL are shown here. The ones which are omitted can be consulted in appendix B.2.2.

An example of this phenomenon is shown in figure B.21. Whereas the expected RUL follows the general decreasing trend, the 95% PI is extremely wide. The probability density of the predictions is therefore low, resulting in a low $\pi\left[\hat{r}_n\right]\Big|_{\alpha^-}^{\alpha^+}$. This, in turn, leads to the fact that there is no PH for all models in this class.

The fact that the energy threshold is based on the distribution of failure energies also implies that outliers are often not modelled correctly. An outlier which appears at the far right of the distribution will always yield conservative predictions, whereas an outlier on the low end of the spectrum will yield too optimistic predictions. Specimen A006 is one of these outliers on the far right, with an EOL of 188,000 cycles and cumulative energy of 1.79 Geu at failure (see table 4.5). Because the majority of specimens lie at roughly one-fourth of these values, the specimen is expected to fail much earlier than others. Next to this, there is a sharp increase in cumulative

energy at around 10,000 cycles. This can be seen in figure 5.8b. The measured (and predicted) cumulative energies acquire values above the mean of the failure threshold distribution, and therefore it is already likely that the specimen fails after this point.



(a) RUL prediction from plain GP regression

(b) Cumulative energy prediction after approximately 11,500 cycles

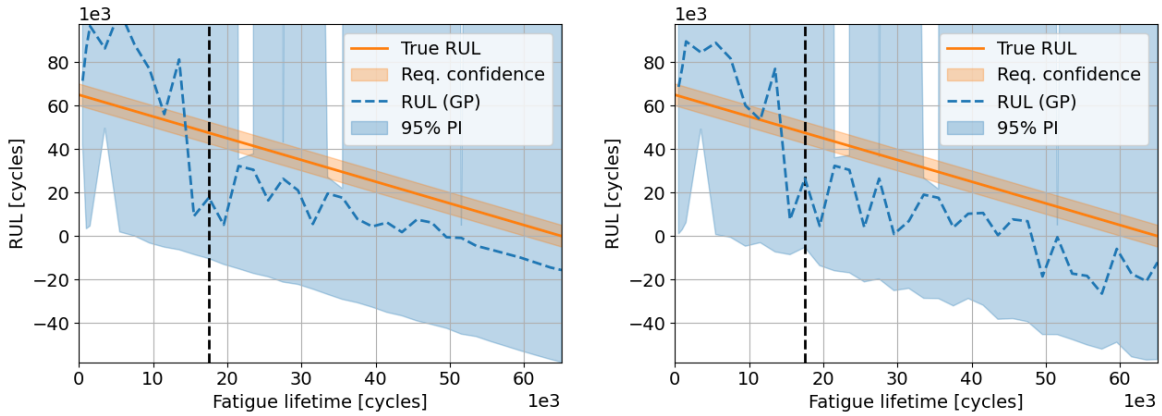
Figure 5.8: RUL and cumulative energy predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A006

5.2.3. Correlation adjustment

The effectiveness of the correlation adjustment on the failure threshold varies. Going back to the example in section 3.2.5, an improvement was shown in the expectation of the threshold PDF at 17,500 cycles; it was closer to the cumulative energy at failure than that of the original distribution.

The RUL predictions in figure 5.9 show that the expected RUL is indeed 'pulled towards' the actual RUL of the specimen at this point. The predictions are 17,700 and 26,600 cycles for the plain and adjusted model (figures 5.9a and 5.9b respectively), compared to the real RUL of 47,500 cycles.

Especially near the EOL there are fluctuations in the adjusted predictions. Because this method is applied after the prediction of the cumulative energy, these fluctuations can be completely attributed to the adjustment of threshold PDFs. From this region, it becomes clear that the adjustment does not only lead to more accurate predictions of RUL.



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure 5.9: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A005

In order to quantify the performance of the PDF adjustments, the change in the MAPE between the expectation of the energy threshold PDF T_E and actual failure energy E_f is shown below in figure 5.10. In a more readable fashion, this change Δ at point i is calculated as:

$$\Delta_i = \left(\left| \frac{E\left[T_{E,original,i}\right] - E_{f,i}}{E_{f,i}} \right| - \left| \frac{E\left[T_{E,adjusted,i}\right] - E_{f,i}}{E_{f,i}} \right| \right) \cdot 100\% \tag{5.1}$$

A positive Δ_i will therefore indicate better performance of the adjusted distribution versus the original, and vice-versa. The distribution of Δ is shown per model type and training set in figure 5.10. The figure is zoomed in; some outliers are therefore not showed. The full figure is shown in appendix B.2.3. Multiple conclusions can be drawn. First of all, the adjustment is not significantly improving the expectation of the threshold. Each median is close to zero, and the distribution is roughly symmetrical around the zero-line. This means that roughly half of the adjustments lead to better $E\left[T_E\right]$, while the other half leads to worse $E\left[T_E\right]$.

Next, the effect of the adjustment seems to be larger for the models which are trained on VAF data only, both in the positive as well as the negative direction. The 1st and 3rd quartiles are wider for these models. This can be explained by the fact that the number of specimens is higher when the distribution is based on both CAF and VAF. With a higher number of specimens in the sample set, the effect of removing a specimen or doubling its weight has a lower effect on the new distribution.

Two variants can be written off after this; the Ma3+lin model trained on VAF and the Ma5+lin model trained on CAF and VAF. These versions show a increase in respectively median (0.56%) and mean (0.06%) MAPE after applying the correlation adjustment. For the other two, the largest mean improvement can be found in the Ma5+lin model, trained on VAF data. The mean improvement is MAPE of 2.6% versus 1.8%. The median of the other (Ma3+lin, on CAF and VAF) is slightly higher; 0.54% versus 0.41%.

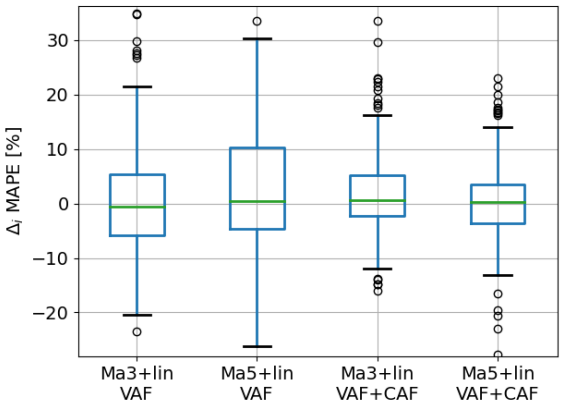


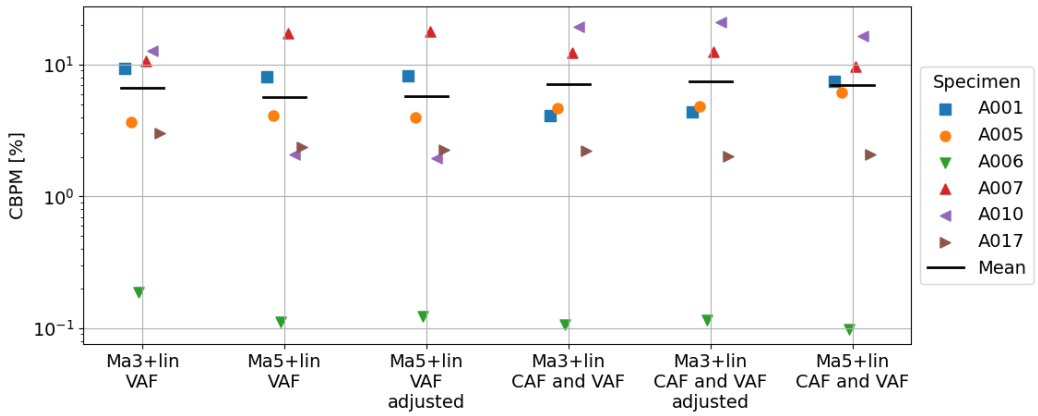
Figure 5.10: Zoomed in box plot of the change in MAPE (expected energy threshold versus actual failure energy) by applying the correlation adjustment to the threshold PDF. The difference is calculated by $MAPE_{original} - MAPE_{adjusted}$.

5.2.4. Effect of training data and overall performance

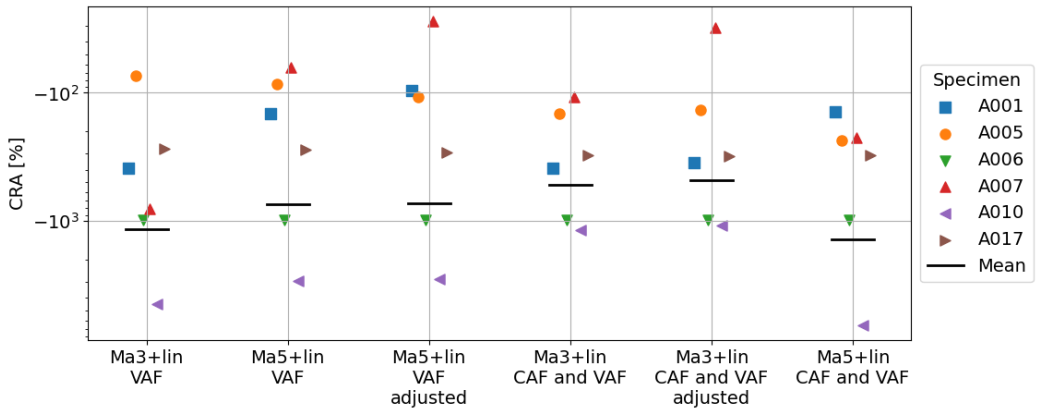
By now it is clear that for all variants of the GP regression, there is a great deal of variability in the predictions, as well as uncertainty. Still, it must be decided what the best possible configuration is. This is done according to two prognostic metrics; the CBPM and CRA. They are shown in figure 5.11. The convergence of accuracy and prediction precision is not analysed for this set of models. This is because of the fact that sometimes a cumulative energy prediction seems to land at a local optimum at the last prediction. The accuracy may be impacted by this and/or the PI width may skyrocket. This results in a final value which is then worse than the first, resulting in no convergence. Because this phenomenon seems to occur randomly, including these metrics in the comparison would not be reasonable.

Both metrics are plotted on a logarithmic scale, due to large differences in the magnitude between the metrics of specimens. The CRA (figure 5.11b) was smaller than zero for all cases. This means that the difference between the predicted and true values was generally larger than 100%, which is unacceptable.

A general trend can be observed for most specimens throughout all the model variations. The samples with average lifespans all show relatively the same performance. Specimen A006, the specimen with the longest life, has by far the lowest CBPM and also shows a low CRA; all models do not seem to be able to cope with this outlier. There does not seem to be a specimen whose performance in both metrics is clearly affected by whether its model is trained on just VAF, or also on CAF data.



(a) CBPM



(b) CRA

Figure 5.11: CBPM and CRA for all RUL predictions by multiple variants of the GP regression model

What happens in the case of specimen A010, the specimen with the shortest life, is peculiar. While the CRA is extremely low, the CBPM is in models trained on the combination of CAF and VAF data the highest at around 20%. The low CRA can be explained by the fact that this specimen is another outlier, and that the expectation of the threshold PDF is significantly higher than the cumulative energy at the failure of this specimen. This leads to RUL predictions which are too optimistic. This can be seen in figure 5.12. Apart from the last prediction, which is likely off due to a local optimum causing its energy prediction to hover around zero, there is already a significant bias. Compared to the predictions for A006 (e.g. figure 5.8a above) however, the predictions are now higher than the actual values. Together with the probability of failure distributions which are right-skewed, this leads to the fact that half of the probability mass is in a relatively small interval, below its expectation. Therefore, for this specimen, this leads to a part of this 'condensed' probability mass overlapping with the required confidence bounds, causing a relatively high CBPM.

It is impossible to deliver a final verdict based on these varying results per specimen. Therefore in order to get an estimation, the means are taken to select the best variation of the GP regression. While also shown in figure 5.11, the exact means can be seen in table 5.1. What could already be seen in the figure is that the Ma3+lin kernel combination, trained on CAF and VAF perform best. Table 5.1 shows that the adjusted threshold PDFs perform marginally better than the 'plain' ones, confirming the conclusion from figure 5.10. Again, these results are dependent on these specific specimens, and there is no general trend in all specimens. Therefore, a conclusion on whether training on a specific data-set is more favourable than on another set cannot be drawn.

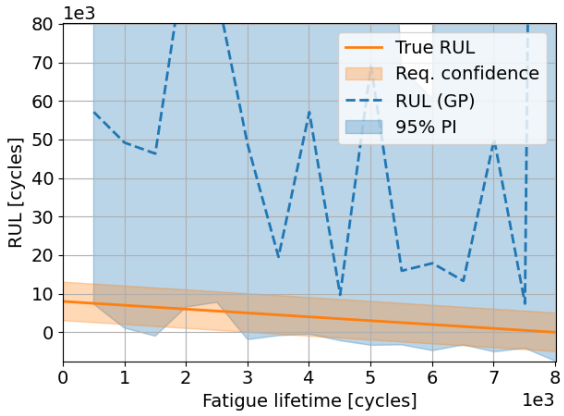


Figure 5.12: RUL prediction by the plain GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A010

The tables containing the performance metrics of each RUL prediction, for all variations of the GP regression can be found in appendix B.2.4.

Table 5.1: Mean CBPM and CRA for variations of the GP regression model

Model variation	Mean CBPM [%]	Mean CRA [%]
Ma3+lin, VAF	6.64	-1.17e+03
Ma5+lin, VAF	5.68	-748
Ma5+lin, VAF, adjusted	5.74	-725
Ma3+lin, CAF and VAF	7.12	-521
Ma3+lin, CAF and VAF, adjusted	**7.48**	**-484**
Ma5+lin, CAF and VAF	6.96	-1.41e+03

5.2.5. Discussion

From these results, it can be concluded that the current implementation of a GP regression using acoustic emission (AE) data is not suited for prognostics on the current sample set. CBPMs from 0.1% to 10% and negative CRAs do not lend themselves for precise and accurate enough predictions. Referring back to the research questions, the performance of this type of model is marginally improved when training it on both CAF and VAF data, instead of just on VAF data, although this is highly dependent on individual specimens.

For the correlation adjustment, there is an improvement both in terms of median and mean MAPE of expectations of the energy threshold. It does vary per kernel function and training data-set. Therefore, more research should be done to confirm the hypothesis that this could lead to better predictions or to reject it altogether. One could, for example, analyse the effects of different weight functions, or the effect of drawing new probability distribution types over the weighted data. If the hypothesis can be further confirmed, this method is an excellent add-on to the GP regression, due to the fact that the correlation can be so easily extracted from the model.

There seem to be four major issues in this model category. The first two are related to the cumulative energy predictions. Local optima in the hyperparameter search space cause predictions which are not in line with other predictions at surrounding points in time. This causes great variability in the RUL predictions. A more intensive study on hyperparameter optimisation, or possibly the effect of reducing the number of hyperparameters could possibly lead to an increase in performance concerning this issue.

The second issue is that some cumulative energy predictions contain non-physical behaviour, i.e. a non-monotonically increasing cumulative energy, or negative cumulative energies. Again, this may lead to variability in the RUL predictions. While the linear kernel was hypothesised to enforce this behaviour, it was not successful in all predictions. A possible solution would be enforcing a mean function in the GP, instead of assuming a mean of 0, as is done now. Another possibility would be by enforcing a monotonically decreasing RUL prediction, as done in section 3.3.6. However, predictions would become very conservative; a sudden drop

in the predicted RUL cannot be undone. A compromise could be made by applying a smoothing function, or by allowing for maximal increases in RUL of for example 10%. Furthermore, the custom implementation of this model makes it probably sub-optimal with regards to the finding of the global optimum. A study could be performed in which more focus could be laid on finding an optimal optimisation algorithm of the model's hyperparameters.

Then, the definition of the energy threshold as a probability distribution causes wide PIs. Although this is the purest form of setting a threshold, it is not one which results in the best predictions, since the predictions are now 'indirect'; an additional uncertainty is introduced. The adjustment of the distribution for correlation improves the results somewhat, but not significantly. Two options are available for possible improvements. Firstly, more could be experimented with the adjustment of the threshold PDF. Weights could be altered, or potential outliers could be excluded to make the PDF narrower. Secondly, a step back could be taken to a hard threshold, based on for example a conservative maximum allowable cumulative energy. Take, for example, the research by Richardson et al. (2017), where a threshold is set based on the ability of the specimens to function. This does, however, not predict failure of a sample.

These issues make this model not feasible for use in real applications. The high variability of RUL predictions would be hard to interpret for an operator who would have to make decisions on repairs or replacements. This should be done on the first time a RUL prediction crosses zero, but this often happens early in the life of a specimen due to this variability. The high uncertainty that goes with the predictions is another pitfall. Because maintenance decisions are tended to be done conservatively, wide PIs would mean very early replacement. On the other hand, when PIs would be narrow, there would be more certainty of a component failing in the short term. In the first case, and thus for this model, this would result in significantly higher maintenance costs compared to the second case, since equipment can be utilised longer with high enough safety margins.

Finally, due to the $\mathcal{O}\left(N^3\right)$ complexity of the model, the computational power required for large data-sets grows exponentially. This is a major issue for validation. In this thesis, validation for this model is done based on the final results. In order to make an unbiased comparison between different kernel functions and possibly correlation adjustments, a two-level cross-validation scheme should have been employed, just as used in algorithm 2, in section 3.3.4. This issue is the cause of long training times. Since the model is trained on the available time-series of the test specimen as well, the model has to be re-trained every time more data becomes available. Depending on whether a model was trained on only VAF, or also CAF data, this would take 5 or 30 core-hours on the Danmarks Tekniske Universitet (DTU) high performance computing (HPC) cluster. In practice, this would mean that if this method were to employed, low-dimensional training data has to be fed into the model, or intervals between predictions should be kept sufficiently large for the model to have enough time to make predictions before new data comes in. Some computational time could be shaved off due to the fact that this model was now written in Python, for the purpose of this thesis. By writing it in a lower-level computer language and making use of faster optimisers and algorithms for matrix inversion, time can be saved. However, this does not eradicate the $\mathcal{O}\left(N^3\right)$ complexity of the model. Even if this method were to provide accurate and precise predictions, this issue holds it back from actual use in applications which require predictions more than once a day.

5.3. Recurrent neural network

The last model to be discussed in this series is that of the recurrent neural network (RNN). The determination of model architectures is covered first. Next, the results of the sensitivity analysis are discussed. With the inner workings of the model covered, the RUL predictions are analysed. Again, the effect of training data on the model's performance is covered, and the section is concluded by a discussion on this model category.

5.3.1. Cross-validation

From the cross-validation, the optimal model architectures and number of training epochs were determined, such that the final model could be constructed. It was decided to work in steps of 10 epochs, such that very local drop-offs were avoided. A number of, for example, 357 epochs would be too precise to train a new model on; what if the optimum of the new model lies at for example 359 epochs? Each case from the research questions will be discussed below. The optimal numbers of hidden nodes and epochs, as well as their corresponding validation losses, can be consulted in appendix B.3.1.

The first validation set is that from the case where a model is trained on solely CAF data. As discussed in section 3.3.4, the generalisation error can be calculated over all CAF specimens in the inner loop, and a single model can then be employed to perform predictions on all VAF specimens. The estimated generalisation errors for this case are shown in figure 5.13.

It can be seen that the more complex a model is in terms of the number of hidden nodes, the earlier the model starts to overfit. Furthermore, the error generally increases with the complexity of the models. This implies that there are no complex relations in the data which are picked up by the model, and by increasing the complexity, there is just overfitting. From this analysis, the optimal model architecture for models trained on CAF data contains 1 hidden node at each activation function in the long short-term memory (LSTM) cell, and the model should be trained for 320 epochs.

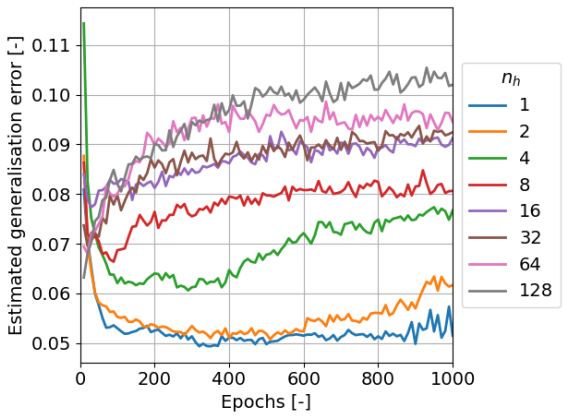
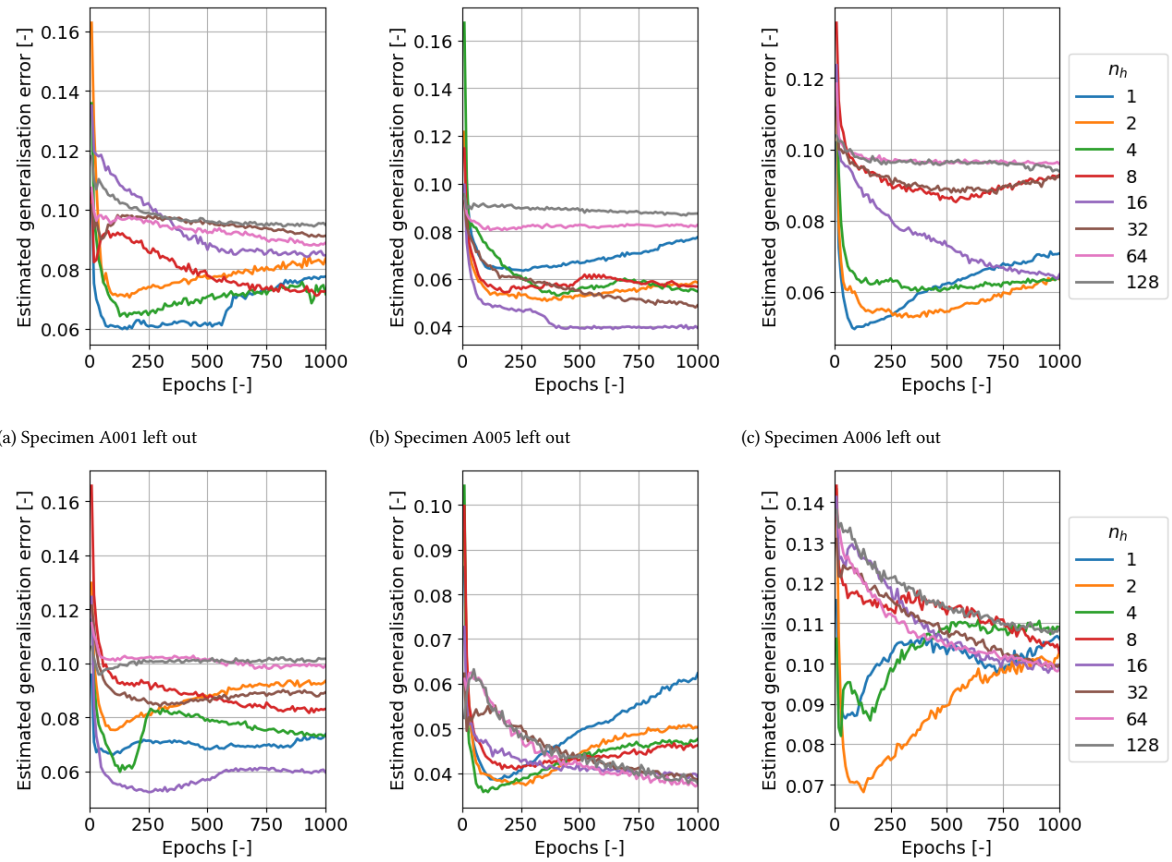


Figure 5.13: Estimated generalisation errors for the RNN, trained on CAF data

Next, the results for the cross-validation loop for VAF training data are shown in figure 5.14. Keep in mind that since this is the generalisation error, which is established in the outer cross-validation loop, the error is calculated over all VAF specimens except for the one which is left out. In the case of figure 5.14a for example, this generalisation error is established based on validation of A005, A006, A007, A010 and A017. Then from the lowest generalisation error for this set, the architecture for A001 is determined. Therefore, six different optimal architectures emerge for these six VAF test specimens.

The trend concerning model complexity from above can be spotted for this set of specimens as well. Generally, models with fewer nodes show lower generalisation errors. There are however exemptions for specimens A005 and A007 (figures 5.14b and 5.14d) where the subsets without them have the lowest generalisation errors for 16 nodes. Also, not in all specimens, a clear optimum of epochs can be spotted, as for example in the subset belonging to A005. In order to come to an unbiased decision on the number of epochs, the absolute minimum from each series was taken. Finally, in the subset where specimen A010 is left out (figure 5.14e), there is a possible optimum of more complex models at a higher number of epochs than 1000. This region was, however, not explored due to the additional computational cost. Within the defined 1000 epochs, a more simple architecture did prove to find an optimum, and therefore this was the go-to architecture in this case.



(a) Specimen A001 left out

(b) Specimen A005 left out

(c) Specimen A006 left out

(d) Specimen A007 left out

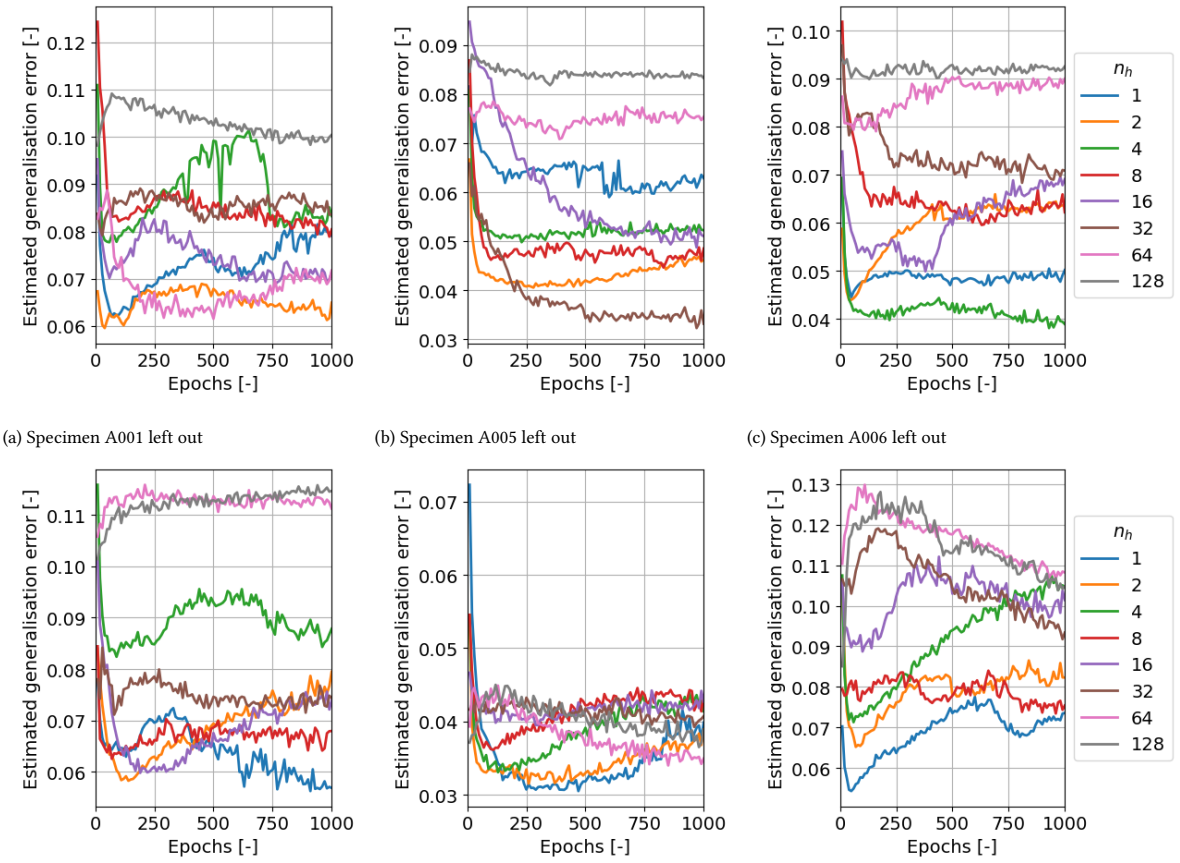
(e) Specimen A010 left out

(f) Specimen A017 left out

Figure 5.14: Estimated generalisation errors for the RNN, trained on VAF data

Finally, the cross-validation results for the last case, training on CAF and VAF data are shown in figure 5.15. Although there is much more variance between model complexities, the general trend of simplicity and lower generalisation errors is present again. When comparing this to the generalisation errors above, it can be seen that adding the CAF data does not necessarily result in significantly lower generalisation errors. It could be possible that having CAF and VAF training data does not lead to significantly better results. This is just a preliminary statement; the RULs should be analysed before any conclusions can be drawn.

Furthermore, three out of six cross-validation loops show lower generalisation errors near 1000 epochs. For these specimens, it could have been investigated if more epochs would lead to lower validation losses, or if a change in learning rate could lead to earlier optima.



(a) Specimen A001 left out

(b) Specimen A005 left out

(c) Specimen A006 left out

(d) Specimen A007 left out

(e) Specimen A010 left out

(f) Specimen A017 left out

Figure 5.15: Estimated generalisation errors for the RNN, trained on CAF and VAF data

5.3.2. Sensitivity analysis

Using the perturbation method, the effect of changes of input parameters on the MSE loss of the model was investigated. First, the actual changes in MSE loss were plotted. An example for the models trained on the CAF data-set is shown in figure 5.16 below. This figure is zoomed in to get a better view of the majority of the changes. Therefore, some outliers are excluded. The full figure can be seen in appendix B.3.2.

A counter-intuitive observation is made; perturbations lead to decreases in MSE loss as well, while it would be expected that perturbations would always lead to an increase in MSE. However, it is also possible that a change in the input results in an output value which is closer to the desired output than the unchanged output value. If the model would, for example, be too conservative, increasing an input value which is positively correlated with the output leads to an increase in the output, making the model less conservative and thus decreasing the loss.

For this set of RNNs, the events in the past 500 cycles before a prediction seem to be most influential on the MSE loss, whereas their cumulative features are rated as less important. While this was not expected, it makes sense. Within the LSTM, inputs are constantly combined with the previous cell state and output. If a cumulative

feature is therefore describing the failure process well, it can also be captured by adding the previous cell state to an input feature, essentially capturing the cumulative feature within the LSTM.

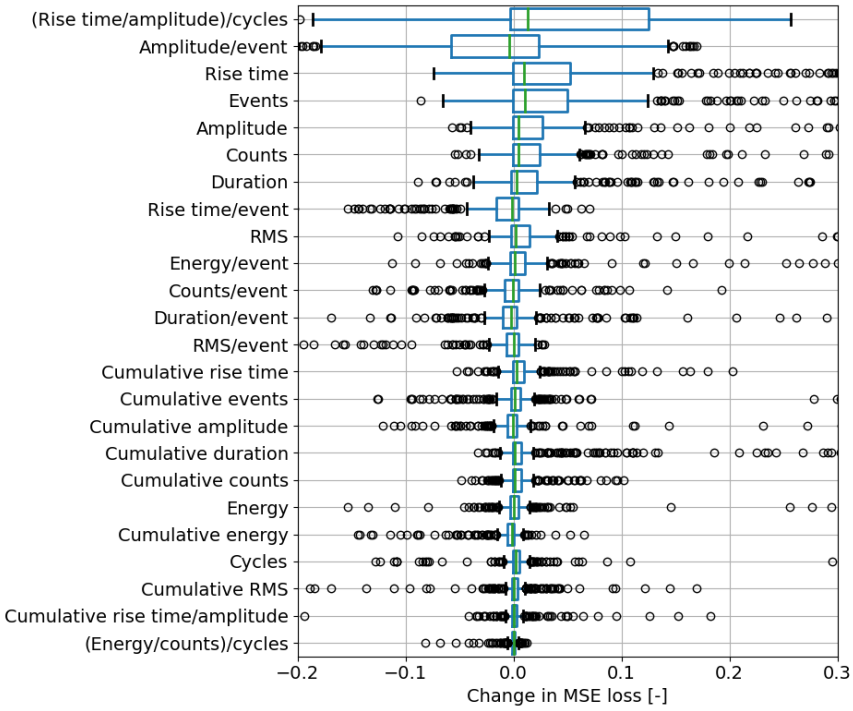


Figure 5.16: Zoomed in sensitivities of the MSE loss of the RNN, when trained on CAF data. The sensitivities are sorted by the width of the box encapsulating the 1st and 3rd quartiles. The green line indicates the median, with the box encapsulating the 1st and 3rd quartiles Q_1 and Q_3. The whiskers extend up to $1.5(Q_3 - Q_1)$. Outliers are plotted as dots.

In this set of models, trained on CAF data, the (rise time/amplitude)/cycles feature causes the most spread in the loss. It can be, however, that this is the case in a few of the models only. Therefore, another comparison was made. In this comparison, the order of feature importance was saved for every single model. This comes down to 60 models, for 6 different specimens times 10 repetitions. For each of these models, the ranking of a feature is shown in figure 5.17. Here, it can be seen that the (rise time/amplitude)/cycles feature has caused quite some spread in a few models, but certainly not in all. This is also the case for models trained on VAF, and CAF and VAF data, as can be seen in figures 5.19 and 5.20 below.

A feature which ranked highest in this set of models, as well as in the two sets below, is the amplitude/event ratio. There is a constant trend throughout all models, in which they are sensitive to this ratio. From a physical perspective, it does align with the results from the research of Huguet et al. (2002); Godin et al. (2004). However, the average amplitudes all fall in their A-type category, relating to matrix fracture. It can be, however, that the slight changes in average amplitude indicate different failure mechanisms, but this cannot be confirmed. On the other hand, this might also have to do with the spread of this feature's values (figure 5.18). Since all values are relatively close together, adding 50% of the maximum value causes relative outliers, which could have a larger effect on the output of the model than when a feature's values are more spread out.

It can be seen that almost every feature is at least once ranked lowest, and once highest. This large spread is likely caused by the fact that these features are not very different from each other. In the general trend, it can be seen that the features which describe events 500 cycle bins are again ranked higher than their cumulative counterparts.

Now for the other two sets of models, the load bins are introduced. While the loads are -from a physical standpoint- a leading factor in the degradation process, this is not fully captured by the RNN, as can be seen in both figures 5.19 and 5.20. From figure 4.13 in section 4.4.3, the load bins which are most common are those between 25-40 kN. This aligns with figure 5.19; the fact that there is relatively much input data for these specific loads likely resulted in the fact that the model learnt how to deal with these, giving them high enough weights in order to cause significant impact on the output. On the other hand, the loads below 20 kN are not common. Because of this, low weights are attached to this input, and therefore a change in this input does not lead to a

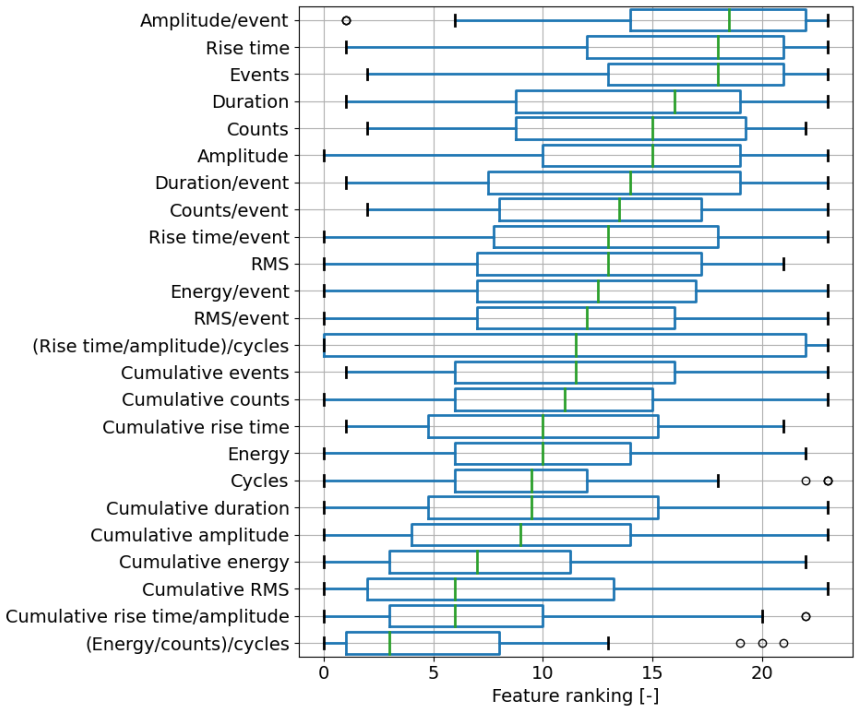


Figure 5.17: Feature ranking of input features based on their effect on the MSE loss of the RNN, when trained on CAF data. The features are sorted by their medians, with more important features having a higher score.

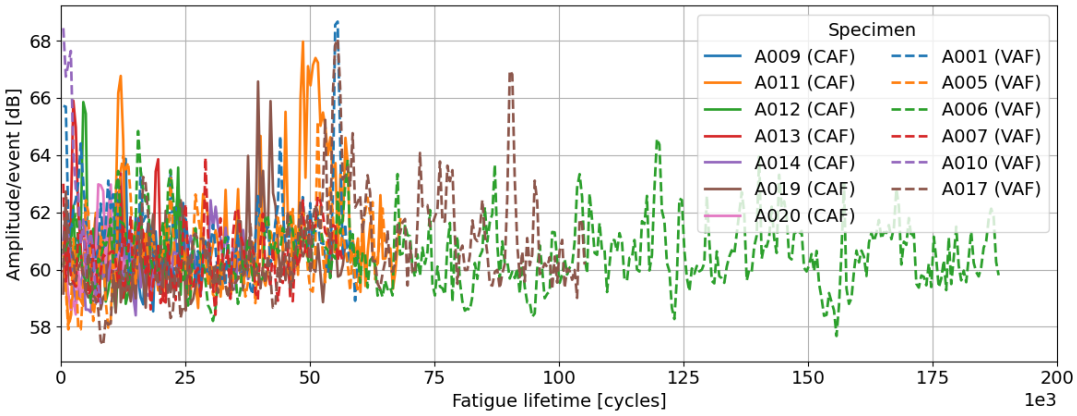


Figure 5.18: Amplitude/event per 500 cycles for the carbon fibre reinforced polymer (CFRP) specimens

significant change of the output of the model.

A very important factor in the loads is that all VAF specimen except for A010 share the same load sequence (table 4.4). Therefore, there is a significant possibility that the weights attached to the load bins are mainly suited for sequence NE9. Also, it would be worth investigating the influence of absolute changes in especially the load features, instead of relative changes. This is because these relative changes are dependent on the already available data. In the case of loads, a relative change in the number of cycles in the 5-10 kN bin is much smaller on an absolute scale than in the 30-35 kN bin, due to the vastly higher numbers of cycles in the latter.

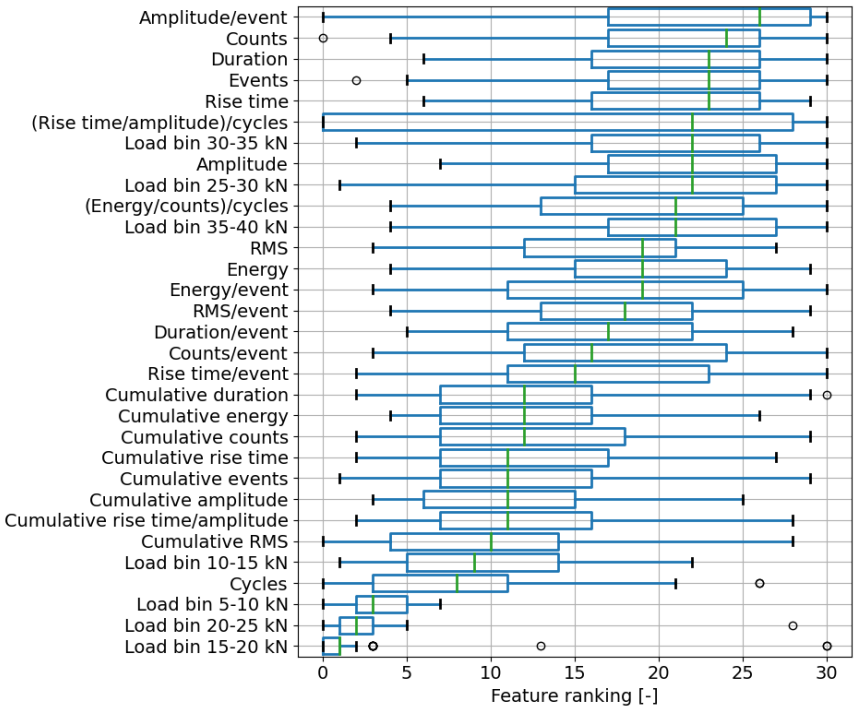


Figure 5.19: Feature ranking of input features based on their effect on the MSE loss of the RNN, when trained on VAF data

When the CAF data is added, the relative importance of load bins seems to drop again. This is probably since in CAF data all loads are situated in the 30-35 kN bin. The values at other bins are therefore 0. When a model is now trained on this combination, it would likely attach less weight to these bins, since half of them are relatively irrelevant.

5.3.3. Remaining useful life predictions

Next, the RUL predictions are covered. A few specimens will be discussed in this section, all other failure index (FI) and RUL predictions can be found appendix B.3.3.

A prediction for a specimen which is not an outlier and which shows desired behaviour is that of specimen A001. Both the FI and RUL predictions are shown in figure 5.21. The sawtooth pattern is due to constant FIs during certain time intervals. Because the RUL is a function of both passed time and the FI (equation (3.48)), if the FI stays constant and time increases, the RUL increases as well. This sawtooth pattern is especially present at the start, which is also the case for other specimens. The predicted FI here is higher than the actual FI. Combined with the low number of passed cycles, this results in very low predicted RULs. From an operator's perspective, it could therefore be wise to not only inspect the RUL prediction, but the FI prediction as well. If the FI prediction is still low, he or she could choose to neglect the RUL predictions, because they are still very likely to rise.

The trend from the previous models concerning outliers is seen again in most models in this category. Due to the relatively different behaviour, as compared to other specimens, the models cannot handle these specimen. Take, for example, specimen A006, when a model trained on CAF data is used to predict its behaviour in figure 5.22. Its FI moves up in the first 70,000 cycles but then stays constant. The fact that in the first 70,000 cycles, it is on the conservative side is likely due to the fact that other specimens' FIs have risen earlier than that

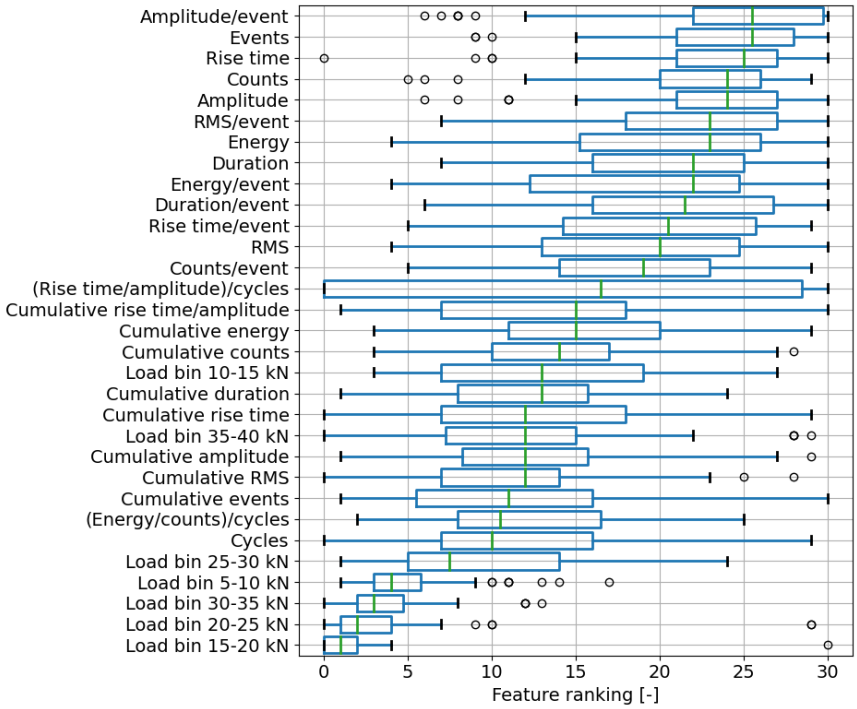
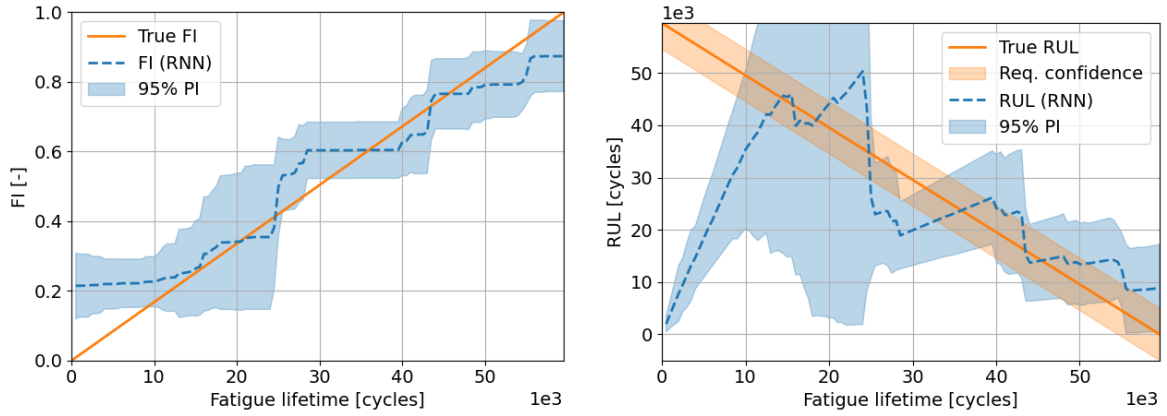


Figure 5.20: Feature ranking of input features based on their effect on the MSE loss of the RNN, when trained on CAF and VAF data

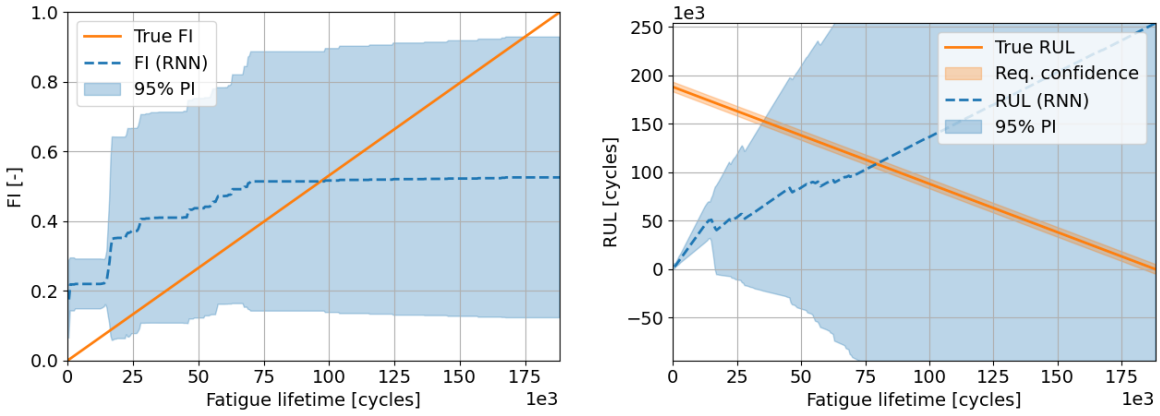


(a) FI prediction

(b) RUL prediction

Figure 5.21: RNN predictions for specimen A001, trained on CAF data

of A006. Then, the failure index stays constant after roughly 70,000 cycles, resulting in a continually increasing RUL. The exact reason behind this behaviour could not be figured out but is likely because there are no other specimens with data in this region.

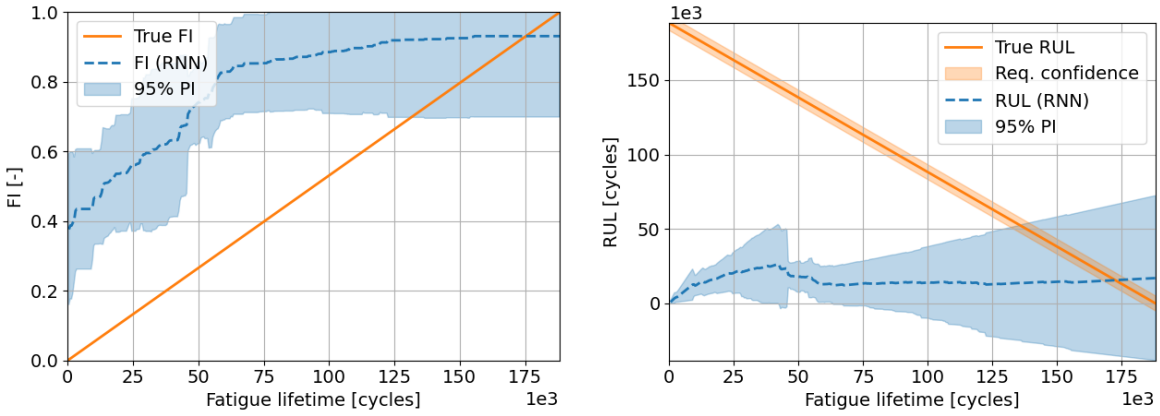


(a) FI prediction

(b) RUL prediction

Figure 5.22: RNN predictions for specimen A006, trained on CAF data

When trained on VAF data, or in the combination of VAF and CAF, the predictions for A006 seem to be slightly better; the convergence to a FI at around 0.5 is not seen in these predictions. Instead, they provide very conservative predictions. The models trained on VAF data show this behaviour in figure 5.23b. The fact that these predictions of these models are better is possibly because specimen A017 is included in these training sets as well. With a lifetime 0f roughly 105,000 cycles, this specimen is the closest related to A006.



(a) FI prediction

(b) RUL prediction

Figure 5.23: RNN predictions for specimen A006, trained on VAF data

The predictions for specimen A010, the one with the shortest life of all specimens, seem to be worst. Within the other models, the FI tends to stay constant in the first 10,000 cycles. This behaviour is observed in all three model variations. When trained on CAF data, the FI is still focused around low values, while in the other variations, it is scattered between 0 and 1, with a mean around 0.5. Because of the short life of A010, the specimen fails before the FI has changed significantly. In the sensitivity analysis, it was discovered that 3/10 and 6/10 variants of the models trained on VAF, and CAF and VAF data had zero sensitivity to parameters.

This behaviour is very likely caused by the fact that all training specimens use load sequence NE9, while A010 is loaded under sequence NE6 (table 4.4). It could very well be that the trained models overfit on that specific load sequence, making them unable to deal with sequence NE6.

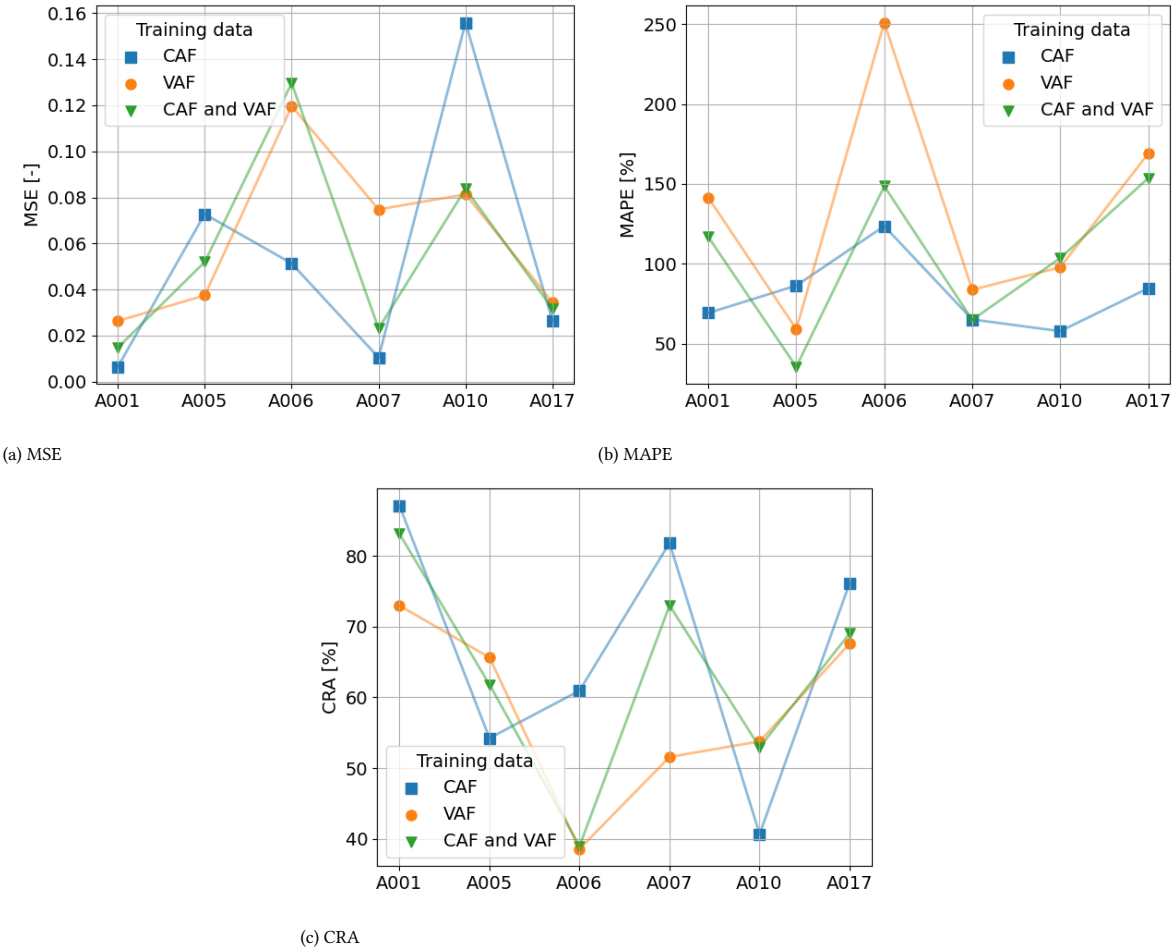
Another possible explanation is that for low numbers of cycles, the combination of input values at time t and output values from $t-1$ are not significant enough to 'open' the gates in the LSTM cell. As values slowly

accumulate over time, this could start to become significant, and thus actually influence the output.

5.3.4. Effect of training data

In order to assess the effect of the training data, the effect on FI predictions is investigated first, followed by the analysis of prognostic metrics on the RUL predictions. Figure 5.24 presents the MSE, MAPE, and CRA for the FIs predicted by the differently trained models. These performance metrics were chosen, because no required confidence is set on the FI. Therefore, only the bias/accuracy will be analysed.

A clear winner in training data cannot be chosen based on these metrics. They vary significantly per specimen. Furthermore, it can be seen that not all behaviour is captured in the metrics. Because specimen A010's predicted FIs have almost constant values of 0.2 (trained on CAF data) and 0.5 (trained on VAF, and CAF and VAF data) for its entire life, the performance metrics are not significantly worse. On top of this, due to the fact that the least important values (those early in life) have low target values, these end up in the denominator of the CRA equation (equation (3.57)), penalising differences more at the beginning. Therefore, it is important that not only the performance metrics are to be analysed, but also the actual predictions.



(a) MSE

(b) MAPE

(c) CRA

Figure 5.24: Performance metrics for the FI predictions of RNN variants, sorted by training data

With no model type which objectively performs best on FI predictions, the RUL predictions are analysed (figure 5.25). In these figures, there is still variability between specimens, although less than in the FIs.

Starting in the top left for the MSE (figure 5.25a), it seems as if the model performance improves when trained on VAF, and even more when trained on both CAF and VAF. It should be mentioned that there are differences between specimens. Specimen A006 has the highest MSE, due to its long life and therefore high maximum actual RUL values. Although specimen A010 was determined to have bad predictions due to the constant FIs, this cannot immediately be noticed in the MSE plot.

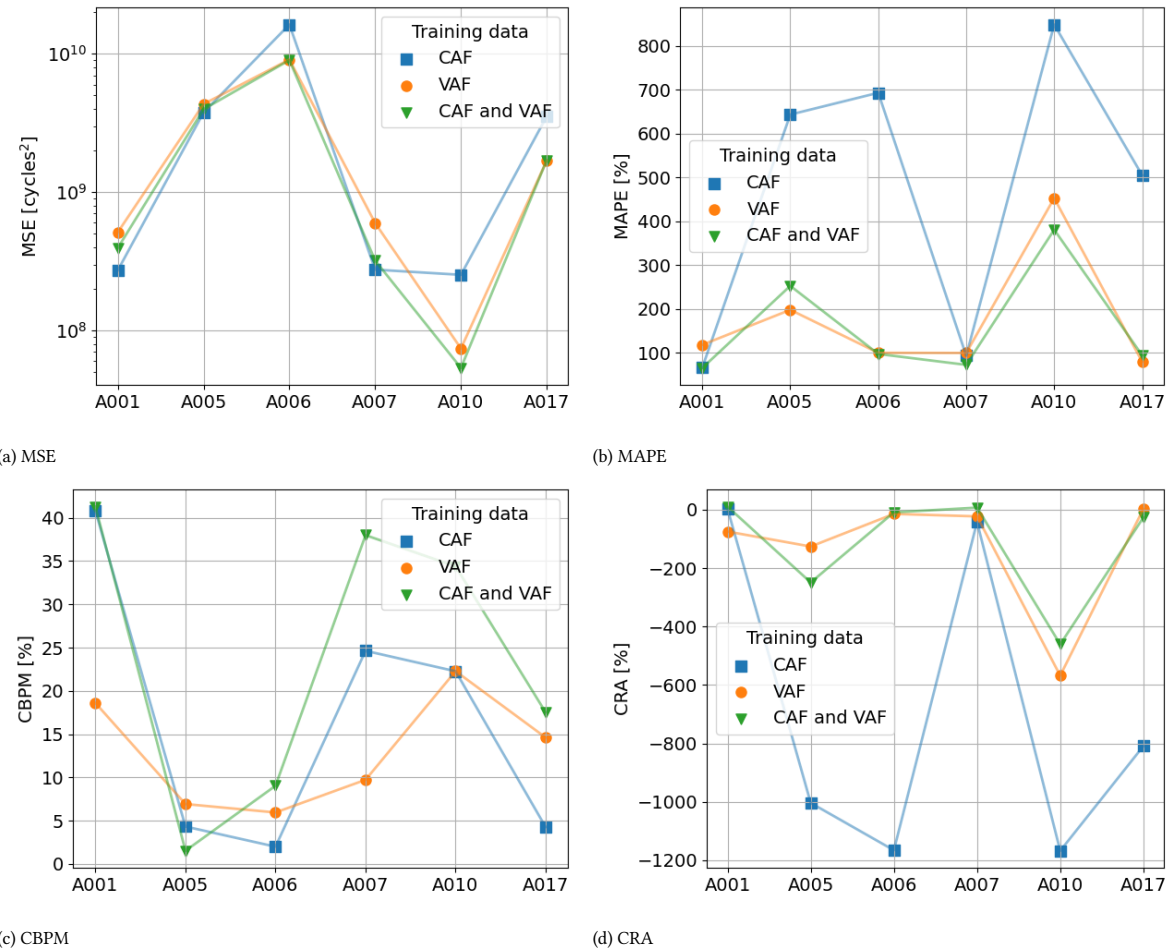
In the MAPE plot in figure 5.25b, the models trained on CAF data are generally performing badly. There

is no clear difference between the other two training sets. This metric does reflect the behaviour of A010, as this specimen has high MAPEs for all training sets. While the scale of MSE is hard to understand, a worrying observation can be made in the MAPEs plot; not a single specimen has a MAPE below 60%.

Next, figure 5.25c shows that some models have fairly high precision and accuracy near the EOL, especially when compared to the models in the previous sections. There is -yet again- a significant difference between specimens. The models trained with CAF and VAF data perform best in almost all specimens here. The difference between the other two is barely visible, but when the means are taken, training on CAF data (16.4%) gives a 3% higher CBPM as compared to training on VAF data (13%).

Finally, from the CRA in figure 5.25d, it is evident that predictions using solely the CAF data to train, give by far the worst predictions when compared with this metric. The mean difference between the other two is 13%, with models trained on both CAF and VAF data being again the best performing models, with an average CRA of -121%. Note that this is a negative CRA, so there is a weighted 221% difference. This metric should be put in perspective. Just as explained above for the FI, the actual RUL moves to increasingly lower values near the EOL, making relatively small absolute biases count as heavy bias in the CRA calculation. This is then further amplified by the weights.

The tables containing all metrics and their values are again displayed in appendix B.3.4. Here, the mean values can also be found for these four metrics.



(a) MSE

(b) MAPE

(c) CBPM

(d) CRA

Figure 5.25: Performance metrics for the RUL predictions of RNN variants, sorted by training data

When inspecting all RUL predictions based on solely CAF training data, the reason behind the low performance in these figures can be found. The median FI predictions for specimens A005, A006, A010, and A017 all stop increasing after approximately 0.6. Some stop earlier, and A017 slightly later. This has a tremendous impact on the RUL predictions; their median will keep growing when the FIs are constant. Therefore, when judging from the median prediction, one would think that these specimens will live forever. In these cases, it is crucial to keep the PIs in mind. These diverge rapidly, with at least some actually predicting failure.

The attentive reader can spot that the PH, as well as the convergence measures, are again not yet compared. For models trained on solely CAF data, there is only convergence of the accuracy, Δ convergence, for specimens A001 and A007. For the models trained on the other data-sets, there is Δ convergence for all specimens, except for specimen A010, which of course diverges due to its almost constant FI. When comparing the convergence of the last two data-sets, the mean of the Δ convergence is slightly lower for the models trained on CAF and VAF data. This implies that these converge slightly faster.

Almost all PIs show no convergence, with the exception of the predictions for A005 and A007 in some model variants. This is caused by the fact that most models are relatively confident of their first few predictions of the FI, which is close to zero. As time goes by, the spread between the predictions increases, reducing chances of convergence.

Finally, an honourable mention should go to the models trained on VAF data. This set of models has PHs of 1,000 and 500 cycles for specimens A005 and A007 respectively. Keep in mind, however, that this is with a 50% probability mass, as explained in section 3.4.

5.3.5. Discussion

The quality of the RUL predictions by the RNN varies, as discussed above. Through the performance metrics, it could be seen that predictions based on CAF and VAF data were generally the best, followed by predictions based on VAF data.

Although it was discovered in the sensitivity analysis that the loads were not the input with the most impact on the output, models trained on load data did perform better. Whether this is actually caused by the load in the inputs is unsure. Another reason for the increase in performance could be attributed to VAF feature time-series which are more identical to each other than to CAF time-series. When the model, trained on solely CAF data would have to make predictions for VAF specimens, the patterns in the test specimens might be 'new' to the model. The sensitivity analysis also showed that the models are not sensitive to cumulative AE features, which is counter-intuitive. They are more sensitive to events in the bin of 500 cycles which occurs before a prediction step. It is possible that the LSTM cell calculates its own cumulative features based on these. Hard conclusions regarding relations between in- and output can unfortunately not be drawn. The complexity of neural networks (NNs) makes them very hard to analyse and understand thoroughly, and therefore only suggestions can be given about its behaviour.

Having more data available in the training set, as in the case of training on CAF and VAF data, seems to result even in better predictions. This shows the decrease in the importance of load as input, which was found in the sensitivity analysis, is not significantly affecting the final predictions. It could very well be possible that increasing the number of samples in the training data would lead to better prediction results. This is likely the case for specimen A006 already. Models which have VAF data in their training sets perform better than the model trained on CAF data, by having a FI going to 1 instead of 0.5 for this specimen. This can be possibly be attributed to the fact that the VAF data contains specimen A017, which also has a relatively long life, just like A006. Having more training specimens in the lower EOL region could also increase performance on specimen A010. The sensitivity to loads can also be increased by having more VAF loaded specimens with different load sequences. As of now, the majority is loaded under one load sequence except for specimen A010. This might very well cause the predictions of A010 to be insufficient.

Continuing on the amount of training data, if it were not possible to generate more training data, it could be examined whether anomaly detection could result in better performance. An unsupervised learning method such as k-nearest neighbours (kNN) could, by comparing clusters and differences in data, spot outliers. It can be investigated whether this is possible during the life of a specimen. If the method were to be sure that the specimen under testing is an outlier, it could either warn the operator who can make a judgement call or even classify the type of outlier; will it live longer or shorter than the training specimens? This could be applicable to all models in this thesis.

The final RUL predictions by the RNN seem to converge towards the actual RUL in most cases. However, the behaviour of the RUL curves makes them bad indicators for practical usage. They generally start from almost zero and then grow to the actual RUL in the predictions which converge. If these models were to be used in practice, it would be hard to determine whether a specimen is actually likely to fail. It could therefore be better for an operator to keep an eye on the FI predictions as well. When the FI would cross a specific threshold, it could be decided to perform a check, maintenance or even replacement of the item which is monitored. The threshold would have to be set lower than 1, since failure occurs before the predicted FI is 1 in multiple specimens. This would, however, raise the discussion concerning the setting of thresholds, which was avoided by the RNN.

The models are currently trained on FIs. Although the use of a FI is practical from a modelling standpoint, the conversion to RUL proved to be very impractical. The conversion resulted in RULs which started from 0 and showed sawtooth-like behaviour in most specimens. It should be investigated whether directly predicting EOL could result in better predictions. The FI was chosen because it can be returned by many different activation functions, but the rectified linear unit (ReLU) seemed to provide the best predictions eventually. A ReLU activation function could also neatly model the RUL because of its output which is always larger than 0, but has no upper bound.

Based on hardware requirements, a trained RNN is definitely a feasible option for live prognostics. The emphasis is on the word 'trained' since the cross-validation is very time-consuming. The model sets used in this thesis needed about two days on the DTU HPC cluster to perform the cross-validation loop when spread out over 50x2 cores. When finally trained, predictions can be made in almost an instant on a standard desktop computer. Therefore, computational requirements would not hold this model type back from being used in the field.

5.4. Model comparison

This section covers the comparison between the different models. Not only a quantitative comparison based on performance metrics is made, but also a qualitative one, based on the usability of RUL predictions and computational cost of each model.

5.4.1. Quantitative

Using the performance metrics from section 3.4, all three model types are compared to each other, within their respective set of training data. From the statistical model, only the static predictions are compared to the other models. This is because the adapting predictions proved to be useless in practice. In case of the GP regression, the Ma3+lin model with adjustments to the failure threshold PDF was the best model by a small margin for predictions made based on CAF and VAF data. Therefore, this model type is compared to the others for this data set. For training on VAF data, there was no clear best model. Therefore, it is decided to take the model with the highest CRA; the Ma5+lin version with adjustments to the failure threshold. This model does not have the highest CBPM in this category, but it is just marginally lower than the best (Ma3+lin). Because the relative difference in CRA is larger compared to that in CBPM, this model was taken. Since the RNN was cross-validated for the best architecture, no choices had to be made for this set. Because a model gave a PH in just two predictions, this metric is not shown in the analyses below but discussed afterwards. The same is done for the PI convergence, with only 8 cases of convergence in total.

Constant amplitude fatigue data

Four performance metrics can be seen in figure 5.26, for the statistical model and RNN, trained on CAF data. Due to convergence in just two specimens for the RNN and of course none for the statistical model, this is not taken into account in the comparison. Apart from the CBPM in the figure 5.26c, there is no clear distinction between the performance of the two models. The results are highly dependent on specific specimens.

In the CBPM however, the RNN significantly outperforms its the statistical model, except for a slight difference in specimen A005. For specimen A001 and A007, this is because the RNN converges to the actual RUL. This behaviour is, of course, not possible for the static statistical model.

The RNN predictions for the other specimens are not very useful, while this is not clearly shown in the metrics below, but can be spotted in their RUL predictions in appendix B.3.3. The fact that the RNN performed poorly on this data-set was also concluded in section 5.3.4. Because their FI predictions stagnate, the RUL starts to grow near the end of life. The RUL in the predictions from the baseline model, however, is decreasing at a constant rate. Even though they are not conservative enough or too conservative, they do predict a point of failure, and could therefore be more useful in practical applications.

Variable amplitude fatigue data

Next, there is the case of VAF data. In this case, and the one hereafter, the Δ convergence is shown. The dots in these figures are however not connected because for some specimens there is no convergence. In this way, no false conclusions can be drawn on these figures.

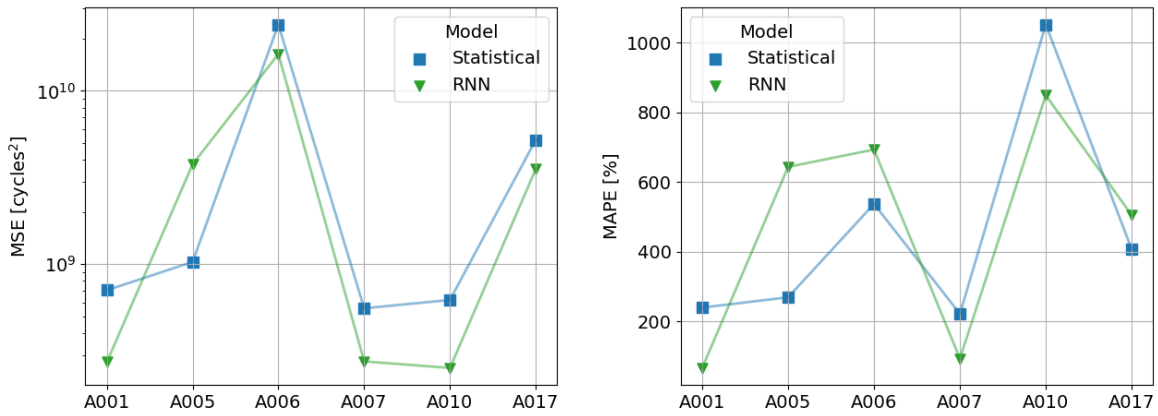
As was already concluded in section 5.3.4, the performance of the RNN increases. Now, the model shows convergence in accuracy (figure 5.27e) in most specimens. This is also lower than the convergence of the GP. It should be noted that, as was already discussed, the convergence of the GP seems to be quite situational.

The RNN is also seen to be more constant in its performance, as compared to the other two models, whose performance depends heavily on the specimen. Interestingly, the same trend can be spotted between these two. Despite the adjustment for correlation, the GP seems to effectively be just another statistical model, due to its dependence on the cumulative energy threshold.

When ignoring some situational highs and lows, the RNN is seen to generally provide the best predictions for this training set. A difference between the statistical model and GP regression cannot be clearly recognised.

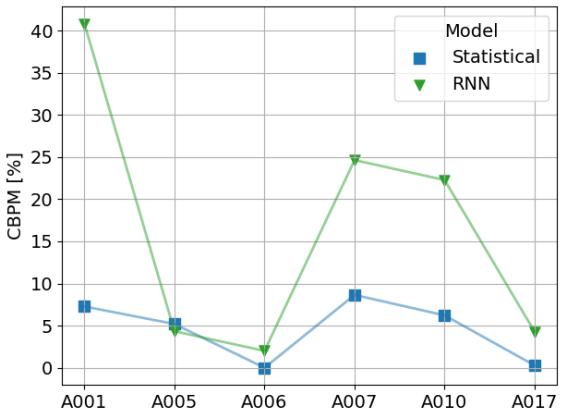
Constant- and variable amplitude fatigue data

Knowing that the RNN's performance increased further when training it on the CAF and VAF set together, and also knowing that there was no significant improvement in the other two models, the metrics below in figure 5.28 are no surprise. These show the same relations as above; the statistical model and GP regression seem to be dependent on specific specimens. The RNN shows a different trend over the specimens. Furthermore, the RNN significantly outperforms the other models again in CBPM (figure 5.28c). Again, there is not a clear difference between the statistical model's performance and that of the GP regression.

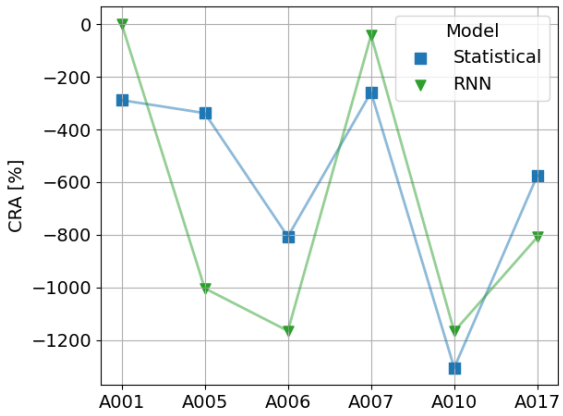


(a) MSE

(b) MAPE

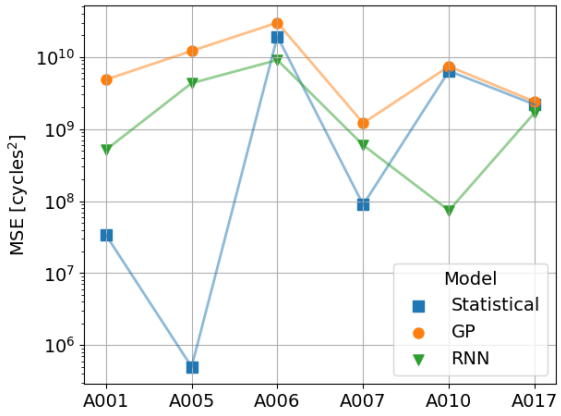


(c) CBPM

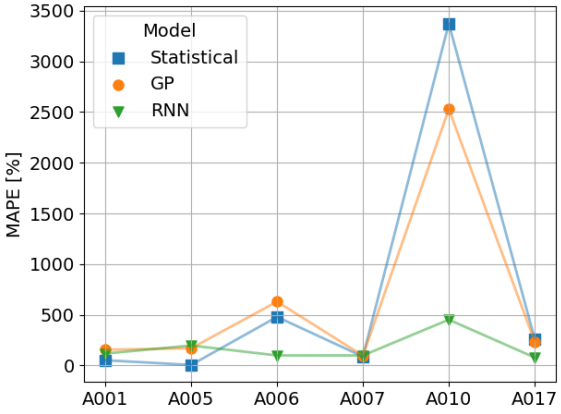


(d) CRA

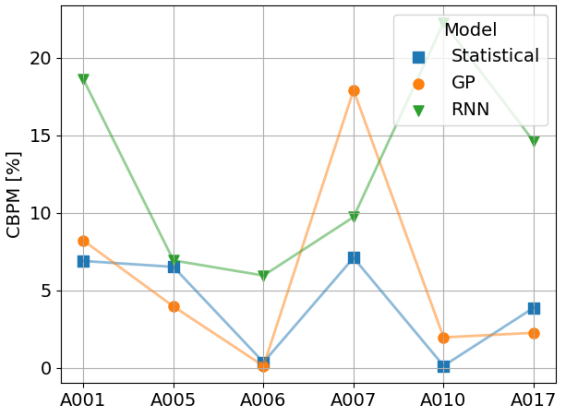
Figure 5.26: Performance metrics for the RUL predictions of the statistical model and RNN, trained on CAF data



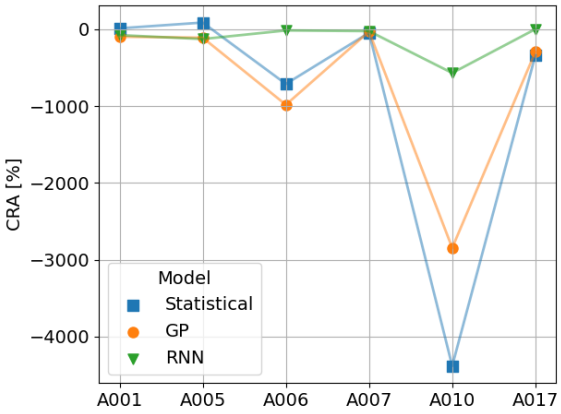
(a) MSE



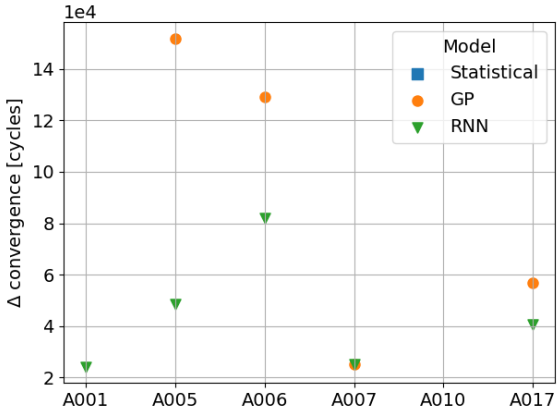
(b) MAPE



(c) CBPM

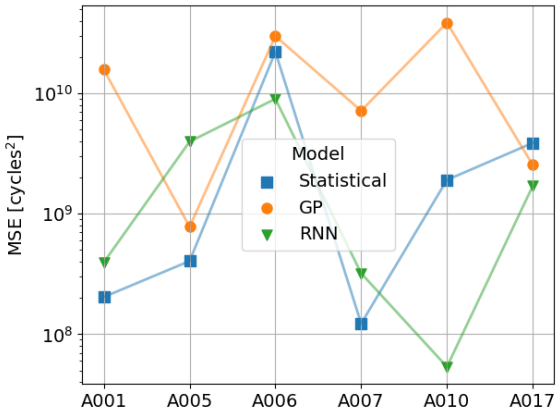


(d) CRA

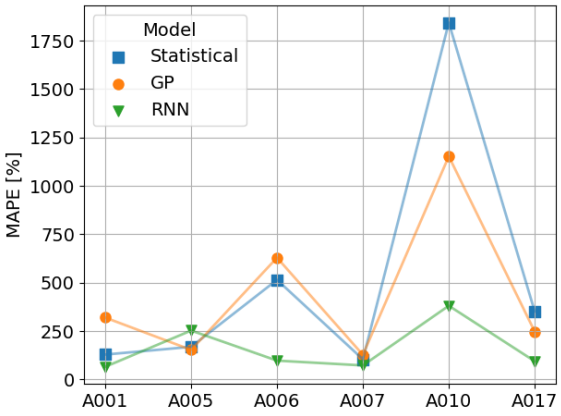


(e) Δ convergence

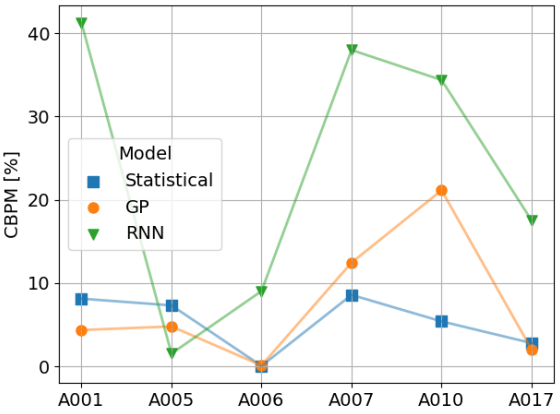
Figure 5.27: Performance metrics for the RUL predictions of all three models, trained on VAF data



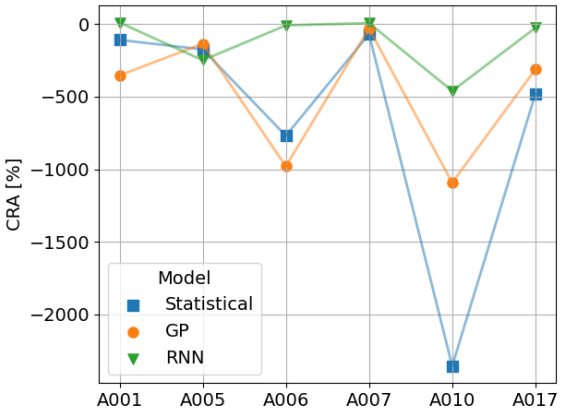
(a) MSE



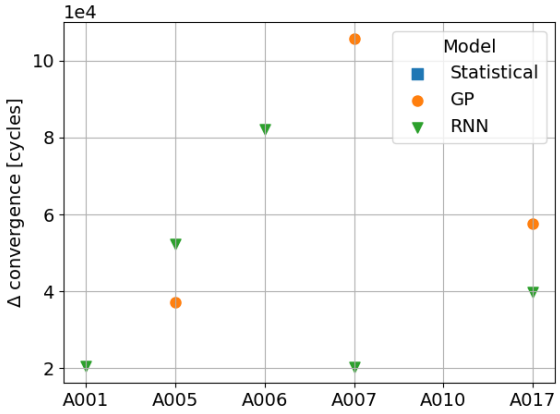
(b) MAPE



(c) CBPM



(d) CRA



(e) Δ convergence

Figure 5.28: Performance metrics for the RUL predictions of all three models, trained on CAF and VAF data

General observations
Although not visible in all performance metrics above, it should be noted that the predictions on specimen A010, the lower outlier in the VAF data, are all far from its actual RUL. Whereas the statistical model and GP regression are primarily based on the distribution of other specimen and are therefore too conservative in their predictions, the RNN is incapable of making sensible predictions at all.

As already mentioned in section 5.3.4, there is a prognostic horizon for two specimen only, for the RNN trained on VAF data. It has to be kept in mind that the definition of PH was set when 50% of the prediction's probability mass is between the required confidence bounds of ±5,000 cycles around the actual RUL. In practice, one would like to have higher certainty between these bounds. The fact that this occurred on two predictions only tells us that the models above are not at all ready yet to be employed in practice. If they would ever be, a PH must be present in a significant portion of the specimens, if not all. Furthermore, the required bounded probability mass should be higher than 50%, in order to be more confident about the predictions.

Finally, PI convergence is seldomly seen in the predictions. While this is of course not possible for the statistical model, it was expected for the other two. Due to a lot of variability in the GP predictions, there was no decrease between the first PI and the last in 7 out of 12 predictions. In the RNN the lack of PI convergence was due to a different reason. In almost all predictions, the models were confident about a low FI near the start. This confidence led to a small PI in this region. Even models which eventually converged towards the actual RUL did not see their PIs shrink enough to meet the definition of convergence.

To conclude, the RNN shows to be the best model in the cases where it is trained on VAF, and CAF and VAF data. From the fact that the performance of the RNN is best when trained on CAF and VAF data, it can be concluded that the RNN is, therefore, the best performing model in this context. A clear distinction cannot be made between the statistical model and the GP regression. In fact, the GP regression does not seem to differ significantly from the statistical model. The setting of the threshold as a probability distribution is the cause of this. The adjustment of this threshold for correlation does not lead to significant changes between these models.

5.4.2. Qualitative

Firstly, when comparing the RUL predictions, the statistical model stands out in its simplicity and consistency. Its RUL always linearly decreases, therefore always predicting failure at some point in time. The downside of this method is that the PIs are extremely wide. Also, this method is not a pure prognostic method, since predictions are not dependent on events during the life of a specimen.

A significant downside about the GP regression predictions is the high variability, as well as the wide PIs. The combination of these factors leads to the fact that with this model implementation, prognostics is not practically feasible. As discussed above, the predictions seem to be in trend with the statistical model due to the large influence of the failure threshold PDF on the result. Therefore it would be better to perform a statistical analysis before putting a subject into use than to rely on prognostics from this method.

Finally, the predictions from the RNN seem to be more constant than those from the GP regression. A downside of the current RNN however is that not in all cases the model tends to predict failure, especially when only trained on CAF data. Furthermore, the sawtooth-like behaviour of RUL predictions, as well as the fact that predictions start from a close to zero RUL, are a handicap due to the current implementation, which depends on the FI. When this model would put into use, FI predictions should also need to be taken into account. This is because, from the FI, a clearer trend can be spotted.

For all these three models, outliers are not handled well enough yet. Measures should be put in place in order to mitigate their impact. Providing more training data could likely lead to better results of the RNN, especially when more varying load sequences would be used in the VAF specimens. Due to the nature of the statistical model, and the dependence of the GP regression on the failure energy distribution, having more data would not necessarily lead to better results for these models. The only upside is that choosing their statistical distributions and parameterisation can be done with more certainty. For the GP regression, more training specimen would quickly become an issue. Due to the $\mathcal{O}\left(N^3\right)$ complexity of the model, computational requirements will skyrocket.

In terms of computation costs, the statistical model is the clear winner. This analysis can be performed in an instant on a laptop. Next, a single RNN takes roughly 20 core minutes to train on the DTU HPC cluster. Keep in mind that for a FI prediction 10 repetitions were used, therefore needing almost 3.5 core hours. This time did still allow for a thorough cross-validation scheme to determine the optimal model parameters objectively. Furthermore, just one set of trained models is needed to make predictions for one specimen. Therefore, online prognostics are possible using this method. Finally, the GP regression is the most computationally expensive

model. Also trained on the DTU HPC cluster, a single repetition takes 15 core minutes for a specimen trained on VAF data. When also trained on CAF data, the size of the data-set almost doubles, and the computation time indeed increases almost eight-fold; to roughly 1.5 core hours. Mind that for the GP regression, to be fairly confident that the models are not trained towards local optima, this process was repeated 20 times. Therefore, the computation times per prediction are roughly 5 and 32 core hours respectively. While this is still manageable, the model has to be trained again every time more data becomes available. Therefore if one wanted to make predictions after every set of 500 cycles, 5 or 32 core hours would be needed every time to train the models. With the current implementation, the repetitions can be split over different cores, therefore still requiring 15 minutes or 1.5 hours. This makes online prognostics very impractical using this model.

5.5. Case study: varying load levels

In this section, results from the case study for the glass fibre reinforced polymer (GFRP) data-set will be discussed. First, the model validation is covered, both for the single RNN, as well as the combination with a feedforward neural network (FFNN). Next, the FI predictions of both model types are discussed. The RUL predictions are not covered for this case, since the FIs tell enough about the performance of the two models. The section is concluded with a discussion on this matter.

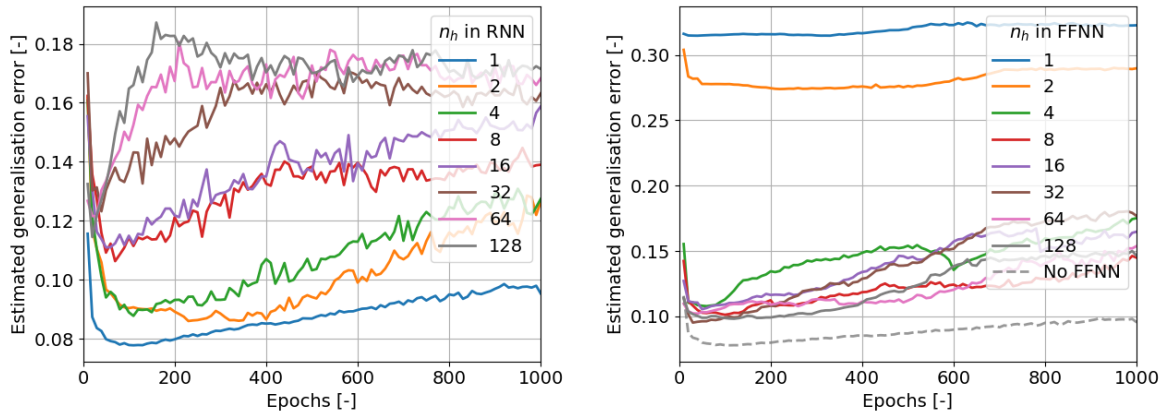
5.5.1. Cross-validation

First, the architecture and number of training epochs of the optimal RNN was determined for this data-set. The same method was used as in section 3.3.4. The results for specimen 6 are shown below in figure 5.29a. All specimens show the same trend in these results. The figures for the other models can therefore be found in appendix B.4.1. The optimal model architectures for the RNN+FFNN combinations are shown below in table 5.2. The table containing the optimal RNN architectures and training epochs can be found in appendix B.4.1.

From figure 5.29a, it is clear that a simple RNN with a low number of hidden nodes n_h is preferred over more complex models, just as in the case of CFRP specimens. Also, the model starts to overfit quite quickly, after around 100 cycles. The low complexity in the RNN is likely because of the varying scales between specimens. This likely causes a low complexity model with relatively bad predictions to be better than a variant with higher complexity, which will tend to overfit.

When adding a FFNN to this existing RNN architecture, the opposite is seen for this second layer in figure 5.29b. High numbers of hidden nodes in the FFNN are preferred over low numbers. The reason for this behaviour is not quite known. It is possible that for low numbers of nodes, the models converge to a solution where a constant output is given, which is independent of the inputs.

Using high numbers of nodes in the FFNN does not improve the estimated generalisation error when comparing this new model to the optimal RNN model, however. Based on this, it seems that adding a FFNN does not lead to added value.



(a) RNN

(b) RNN+FFNN

Figure 5.29: Estimated generalisation errors for specimen 6 in the GFRP data-set

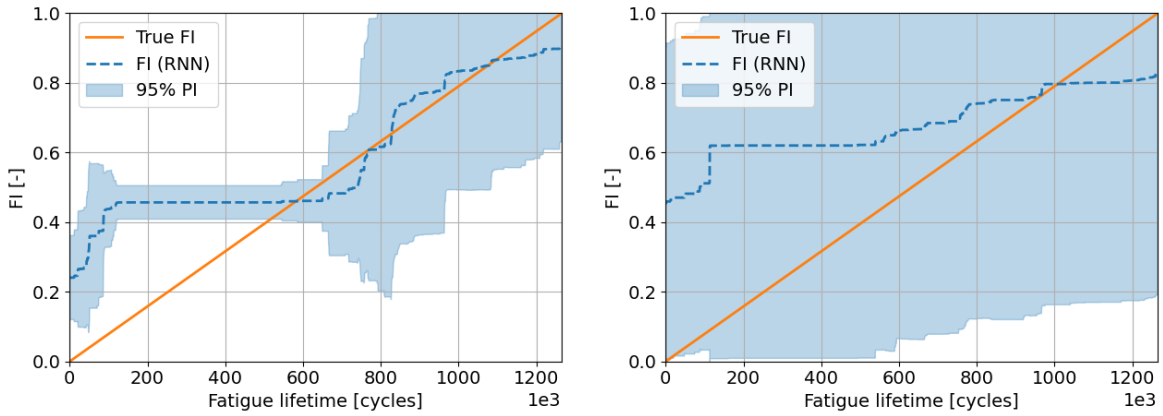
Table 5.2: Optimal RNN+FFNN architectures and their corresponding estimated generalisation loss for the GFRP data-set

Excluded specimen	Optimal model			
	n_h (RNN)	n_h (FFNN)	Epochs	Est. gen. loss
6	1	32	30	0.095
7	1	128	150	0.071
8	2	64	80	0.088
9	1	64	120	0.090
10	2	64	540	0.064
11	1	128	280	0.083
12	1	128	920	0.087

5.5.2. Failure index predictions

The best predictions for this case come from a single RNN, for specimen 8 (figure 5.30a). This specimen is an outlier in terms of EOL, but its AE data indicate failure at values close those of specimens 6, 9 and 10. Having similar data can be the reason for the relatively good predictions. The constant FI in the region between 100,000-500,000 cycles coincides with the region with very low AE activity, as could be seen in figure 4.14 in section 4.4.4.

As was expected through the analysis of the estimated generalisation loss, the addition of a FFNN does not lead to better predictions. In fact, they are significantly worse. This is likely caused by the high complexity of its hidden layer, causing it to overfit.



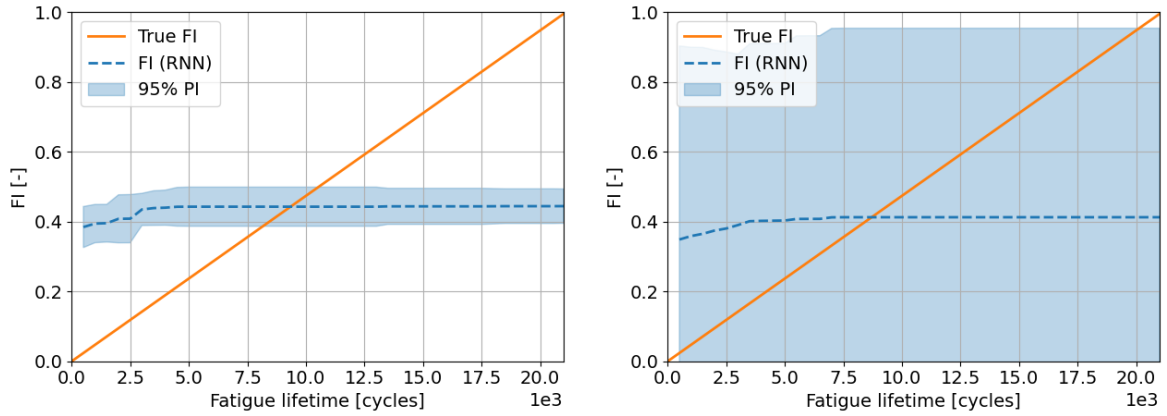
(a) FI prediction from the RNN

(b) FI prediction from the RNN+FFNN

Figure 5.30: FI predictions for specimen 8 in the GFRP data-set

Predictions for other specimens, such as specimen 11 below in figure 5.31 contain large regions with constant FIs. These regions are often around 0.5. It is therefore likely that the models are unable to make sensible predictions, and therefore converge to an architecture which is independent of its inputs. Bias terms within the models are then the driving factors behind the remaining output. This remaining output is at a FI of around 0.5 because for this value, the MSE loss is lowest. When adding the FFNN, the complexity of the model is increased, and it overfits, causing the PI to be extremely wide compared to the pure RNN prediction.

Because of the bad performance of these models on this data, the RULs nor any performance metrics are further analysed.



(a) FI prediction from the RNN

(b) FI prediction from the RNN+FFNN

Figure 5.31: FI predictions for specimen 11 in the GFRP data-set

5.5.3. Discussion

From these results, two conclusions can be drawn. First of all, the current implementation of the RNN is not suited for RUL prediction on specimens with different load levels. The resulting large differences in load levels cause significant differences in the lifetimes of specimens. This cannot be handled by the RNN currently. Secondly, adding a FFNN at the end of a RNN and feeding this the output of the RNN in combination with load levels of a specimen does not improve the predictions. Instead, they are deteriorated.

With the low complexity of the RNNs and the high complexity of the FFNNs, it could be possible that there is a middle ground, where the quality of the predictions does actually increase. Finding this middle ground is not possible with the current validation method, where the layout of the RNN is first determined, followed by adding a FFNN to this network. This was done to minimise the computational cost of this search for a model architecture; this method has $\mathcal{O}\left(2n_h\right)$ complexity, where n_h is the number of hidden nodes in one of the networks. A the cross-validation for a complete grid search has $\mathcal{O}\left(n_h^2\right)$ complexity.

6

Conclusion and recommendations

This chapter discusses the conclusions which can be drawn based on the research done, and the answers to the research questions. This is followed by a set of recommendations, based on the gathered insights during this thesis which can be used to enhance further research.

6.1. Conclusion

The objective of this research was to investigate the feasibility of in-situ, data-driven prognostics on composites under variable amplitude fatigue (VAF), by training multiple probabilistic models on constant amplitude fatigue (CAF) and/or VAF data and assessing their performance in the prediction of remaining useful life (RUL).

Experimental data for carbon fibre reinforced polymer (CFRP) specimens, tested under CAF and VAF, was available from earlier research. This was used to answer the main research questions, using three models. During this project, experimental data from a testing campaign on glass fibre reinforced polymer (GFRP) specimens, under different load levels in tension-tension (T-T) fatigue became available. This was used in a case study to assess the effect of different load levels on the feasibility of in-situ prognostics using a recurrent neural network (RNN).

The first model, a statistical model based on the distribution of failure times of specimens, performed as expected; the predictions were highly uncertain due to large variability in the failure times. The performance was heavily dependent on which specific specimen was tested. Therefore, it was concluded that it was not possible, using the current data-set, to identify improvements when going from CAF training data to VAF training data, or a combination of the two. When used conservatively, this model can be used (and is used) in practice.

The second model was a Gaussian process (GP) regression on cumulative acoustic emission (AE) energy data. There was not a clear difference in the prognostic performance between different versions of the model, with Ma3+lin or Ma5+lin kernels. A marginal improvement in performance was found when adjusting the cumulative energy threshold probability density function (PDF), according to the correlation between different time-series in the training data. These correlations were extracted from the trained model. The performance of this model type was marginally improved when trained on both CAF and VAF data, instead of just VAF data, although this was highly dependent on specific specimens. Due to a high degree of variability in the RUL predictions, low performance according to performance metrics, and high computational cost, it is not feasible to use this model in practice. From its definition, the model could not be trained on purely CAF data, and this case was therefore not analysed. It should be noted that an objective cross-validation scheme, for finding the right covariance kernels and objectively evaluating the effect of the correlation adjustment of the threshold, was not used in this model category due to its high computational costs.

The third and final model is a RNN based on a long short-term memory (LSTM) cell, where the failure index (FI) is predicted. Through cross-validation, it was found that an architecture with a low number of hidden nodes performed best on all three variations of the training data. From a sensitivity analysis, the models seemed most sensitive to AE events grouped in the preceding 500 cycles. The load was used as input in cases with VAF training data, in bins of 5 kN. The sensitivity of the RNNs for these features was not as high as for the AE events grouped per 500 cycles. The RUL predictions improved when moving from CAF, to VAF, to CAF and VAF training data. The current implementation of the RNN was, however, not deemed feasible enough yet for

practical applications, due to an inability to handle outliers and the variability in RUL predictions.

From all these models, the RNN performed best when trained on VAF data, and on the combination of CAF and VAF data. Due to the definition of the failure threshold as PDF, the GP did not perform better than the statistical model. In the case of CAF training data, there was not a clear distinction between the performance of the statistical model and the RNN, except for a higher cumulative bounded probability mass (CBPM) for the RNN, which converged to the actual RUL for some specimens. The varying lifetimes of different specimens posed an issue to all three models; they were not able to deal properly with outliers.

In the case study, it was found that using a RNN is not feasible yet for prognostics on specimens under different load levels using AE data, due to large differences between specimens in the training data. It was hypothesised that adding a feedforward neural network (FFNN), which uses the RNN output as well as information about the load level, could lead to better predictions. This was not the case.

The main conclusion drawn in this thesis is, therefore, that with the current implementation of the used models, in-situ, data-driven prognostics on composite specimens under VAF is not yet feasible.

6.2. Recommendations

While the four research questions in this thesis were answered, more questions and ideas arose. Furthermore, although these models are not yet capable of being implemented in the industry, there are possibilities to improve their performance. Unfortunately, due to the limited timeframe of this thesis, these had to be moved to this section; the recommendations. These are grouped into four categories. First general recommendations are discussed, followed by those for the three models used in this thesis.

General

A general recommendation is based on the amount of training data. With just 13 CFRP specimens which could be used in this research, it was not possible to confidently draw conclusions in multiple matters. Furthermore, all models will likely perform better when fed with more training data.

If having more data is not possible, it might be possible to look into unsupervised learning methods for outlier detection. A method such as k-nearest neighbours (kNN) could spot outliers, by comparing clusters in the data, and their differences in data. It can be investigated whether this is possible during the life of a specimen. If so, using this method could either warn an operator who can make a judgement call when this is used in practice or even classify the type of outlier; will it live longer or shorter than the training specimens?

Finally, the AE data used in this research is from the features of each signal. These features were not further processed. It could, however, be possible to obtain better results when a hybrid model is used. This hybrid model would use not only AE features, but also information concerning the type of damage with which they should be associated. This method can be based on the previously performed research on the relations between specific AE signals and the related damage mechanism. One could then also compare different damage type indicators and determine their ability to predict RUL. The most potent indicators can then be validated against the existing literature on the relationship between specific AE events and their parameters, and specific damage mechanisms.

Statistical model

Because the performance of the statistical model varied significantly per specimen, an idea for a follow-up study for this model class would be to perform more of these CAF and VAF tests and become more confident on the parameters and type of statistical distributions. The difference between these distributions can then be compared, and the research questions from this thesis can be answered for this model; if the difference is negligible, it would not matter if a specimen would be trained on CAF and/or VAF data.

Gaussian process regression

For the GP regression, there are multiple recommendations for possible further research. Firstly, the cumulative energy predictions of the GP regression were sub-optimal in multiple predictions due to local optima, and/or contained non-physical behaviour. More refined hyperparameter optimisation methods could be implemented to solve the first issue, and it could be researched whether enforcing a mean function can solve the second.

Secondly, a new method of defining a failure threshold was introduced in this thesis; by making it a PDF based on data from other specimens. Putting more weight on specimens which correlated with the time-series of the test specimen showed marginal improvements in the performance of the GP regression. This method has potential for the replacement of hard, arbitrary thresholds, by purely data-driven thresholds. However, more

research is needed to confirm the hypothesis that this could actually lead to better predictions, and if so, to look into different weight functions or PDFs for the failure threshold.

Finally, due to the high demand for computational power, a full cross-validation scheme could not be run. Therefore, the selection of best model architectures was made in hindsight, with knowledge of the performance of each architecture on all the data. It is a better practice to employ a cross-validation scheme for the different kernel functions and correlation adjustment, just as was done in determining the architecture of the RNN. Again, the computational effort required by the GP regression hindered this. This may therefore be another disadvantage of the GP regression as well; being unable to perform validation due to high computational cost.

Recurrent neural network

Although the RNN performed best in two out of three cases of training data, it is still far from optimal. Therefore, multiple recommendations for this model type are proposed. First of all, just as for the statistical model, having more available data is very likely to improve predictions. Additionally, there is currently only one specimen in the VAF data-set with a different load path than the others. This likely led to the poor performance of the RNN on this specimen. Having more data, with different load paths, will allow these complex models to find patterns and dependencies better, and therefore lead to better predictions.

The RNN currently outputs FIs. It should be investigated whether predicting the end of life (EOL) could result in better predictions. A rectified linear unit (ReLU) activation function could in theory neatly model the RUL because of its output which is always larger than 0, but has no upper bound. In this way, the sawtooth behaviour in the RUL predictions, caused by the conversion of FI to RUL can be prevented.

The loss function for the RNN (and added FFNN) is now a mean squared error (MSE) loss, in order to fit the FI. It could be changed in multiple ways. First of all, the models could be trained to maximise the prognostic horizon (PH) or any other performance metric directly. Another way to improve the feasibility of RNN predictions is to add a penalisation to the loss function, such that predictions become conservative. In order to force the models to be conservative, the errors in the MSE loss function could, for example, be weighted based on whether a prediction is below or above an actual value. The priority of predictions near the EOL can also be increased.

Finally, the sensitivity analysis of the RNN is now based on perturbations which are ratios of the input variables. In the case of AE, this seems sensible, due to the low interpret-ability of these (aggregated) parameters. In the case of loads, however, it could be worth investigating what happens when absolute values are used; if 100 cycles are added to the 20-25 kN load bin, will this have more impact than when 100 cycles would be added to the 5-10 kN bin? Under the current method, this is not analysed, but could be useful to help to quantify the actual value of these input variables for the model's performance.

Bibliography

Airbus. Airbus' focus on "thermoplastic" composite materials brings environmental and production improvements, 2015. URL https://www.airbus.com/newsroom/news/en/2015/01/airbus-focus-on-thermoplastic-composite-materials-brings-environmental-and-production-improvements.html.

Y. Al-Assaf and H. El Kadi. Fatigue life prediction of unidirectional glass fiber/epoxy composite laminae using neural networks. *Composite Structures*, 53(1):65–71, 7 2001. ISSN 02638223. doi: 10.1016/S0263-8223(00)00179-3.

V. Arumugam, R.N. Shankar, B.T.N. Sridhar, and A.J. Stanley. Ultimate Strength Prediction of Carbon/Epoxy Tensile Specimens from Acoustic Emission Data. *Journal of Materials Science and Technology*, 26(8):725–729, 2010. ISSN 10050302. doi: 10.1016/S1005-0302(10)60114-4.

S. Barré and M.L. Benzeggagh. On the use of acoustic emission to investigate damage mechanisms in glass-fibre-reinforced polypropylene. *Composites Science and Technology*, 52(3):369–376, 1994. ISSN 02663538. doi: 10.1016/0266-3538(94)90171-6.

M.G. Baxter, R. Pullin, K.M. Holford, and S.L. Evans. Delta T source location for acoustic emission. *Mechanical Systems and Signal Processing*, 21(3):1512–1520, 4 2007. ISSN 08883270. doi: 10.1016/j.ymssp.2006.05.003.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1724–1734. Association for Computational Linguistics (ACL), 6 2014. ISBN 9781937284961. doi: 10.3115/v1/d14-1179. URL https://arxiv.org/abs/1406.1078v3.

F. Chollet and others. Keras, 2015. URL https://github.com/fchollet/keras.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, 12 2014. URL http://arxiv.org/abs/1412.3555.

J.B. Coble and J.W. Hines. Prognostic algorithm categorization with PHM challenge application. In *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008. ISBN 9781424419357. doi: 10.1109/PHM.2008.4711456.

J.B. de Jong, D. Schütz, H. Lowak, and J. Schijve. A standardized load sequence for flight simulation tests on transport aircraft wing structures. *NLR-TR 73029 U, LBF Bericht FB-106*, 1973.

G. Dorffner. Neural Networks for Time Series Processing. *Neural Network World*, 6:447–468, 1996.

D.K. Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014. URL https://www.cs.toronto.edu/~duvenaud/thesis.pdf.

D.G. Eitzen and H.N.G. Wadley. Acoustic Emission: Establishing the Fundamentals. *Journal of Research of the National Bureau of Standards (United States)*, 89(1):75–100, 1984. doi: 10.6028/jres.089.008.

N. Eleftheroglou and T.H. Loutas. Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes. *Structural Health Monitoring*, 15(4):473–488, 2016. ISSN 17413168. doi: 10.1177/1475921716646579.

N. Eleftheroglou, D.S. Zarouchas, and T.H. Loutas. In-situ fatigue damage assessment of carbon-fibre reinforced polymer structures using advanced experimental techniques. In *ECCM 2016 - Proceeding of the 17th European Conference on Composite Materials*, 2016. ISBN 9783000533877.

N. Eleftheroglou, D.S. Zarouchas, and R. Benedictus. Extreme prognostics for remaining useful life analysis of composite structures. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 2018a. ISBN 9781936263295.

N. Eleftheroglou, D.S. Zarouchas, T.H. Loutas, R.C. Alderliesten, and R. Benedictus. Structural health monitoring data fusion for in-situ life prognosis of composite structures. *Reliability Engineering and System Safety*, 178: 40–54, 2018b. ISSN 09518320. doi: 10.1016/j.ress.2018.04.031.

A. Fraisse and P. Brøndsted. Compression fatigue of wind turbine blade composite materials and damage mechanism. In *ICCM International Conferences on Composite Materials*, volume 2017-Augus, 2017. URL http://www.vindenergi.dtu.dk/englishhttp://www.vindenergi.dtu.dk/english.

G.T. Frøseth and L. Capponi. fatpack, 2019. URL https://github.com/Gunnstein/fatpack.

T.V. Galambos. Load and Restance Factor Design. *Engineering Journal*, 18(3):74–82, 1981. ISSN 00138029.

E.K Gamstedt and B.A Sjögren. Micromechanisms in tension-compression fatigue of composite laminates containing transverse plies. *Composites Science and Technology*, 59(2):167–178, 2 1999. ISSN 0266-3538. doi: 10.1016/S0266-3538(98)00061-X. URL https://www.sciencedirect.com/science/article/pii/S026635389800061X.

GE Renewable Energy. World's Most Powerful Offshore Wind Turbine: Haliade-X 12 MW. URL https://www.ge.com/renewableenergy/wind-energy/offshore-wind/haliade-x-offshore-turbine.

M. Gevrey, I. Dimopoulos, and S. Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. In *Ecological Modelling*, volume 160, pages 249–264, 2 2003. doi: 10.1016/S0304-3800(02)00257-0.

N. Godin, S. Huguet, R. Gaertner, and L. Salmon. Clustering of acoustic emission signals collected during tensile tests on unidirectional glass/polyester composite using supervised and unsupervised classifiers. *NDT and E International*, 37(4):253–264, 6 2004. ISSN 09638695. doi: 10.1016/j.ndteint.2003.09.010.

N. Godin, P. Reynaud, and G. Fantozzi. Contribution of AE analysis in order to evaluate time to failure of ceramic matrix composites. *Engineering Fracture Mechanics*, 210:452–469, 4 2019. ISSN 00137944. doi: 10.1016/j.engfracmech.2018.03.006.

R. Gutkin, C.J. Green, S. Vangrattanachai, S.T. Pinho, P. Robinson, and P.T. Curtis. On acoustic emission for failure investigation in CFRP: Pattern recognition and peak frequency analyses. *Mechanical Systems and Signal Processing*, 25(4):1393–1407, 5 2011. ISSN 08883270. doi: 10.1016/j.ymssp.2010.11.014.

F.O. Heimes. Recurrent neural networks for remaining useful life estimation. In *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008. ISBN 9781424419357. doi: 10.1109/PHM.2008.4711422.

T. Herlau, M.N. Schmidt, and M. Mørup. Introduction to Machine Learning and Data Mining, 2019.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. URL http://www7.informatik.tu-muenchen.de/~hochreithttp://www.idsia.ch/~juergen.

S. Huguet, N. Godin, R. Gaertner, L. Salmon, and D. Villard. Use of acoustic emission to identify damage modes in glass fibre reinforced polyester. *Composites Science and Technology*, 62(10-11):1433–1444, 2002. ISSN 02663538. doi: 10.1016/S0266-3538(02)00087-8.

K.M. Jespersen and L.P. Mikkelsen. Three dimensional fatigue damage evolution in non-crimp glass fibre fabric based composites used for wind turbine blades. *Composites Science and Technology*, 153:261–272, 2017. ISSN 02663538. doi: 10.1016/j.compscitech.2017.10.004. URL https://doi.org/10.1016/j.compscitech.2017.10.004.

K.M. Jespersen, J. Zangenberg, T. Lowe, P.J. Withers, and L.P. Mikkelsen. Fatigue damage assessment of unidirectional non-crimp fabric reinforced polyester composite using X-ray computed tomography. *Composites Science and Technology*, 136:94–103, 11 2016. ISSN 02663538. doi: 10.1016/j.compscitech.2016.10.006.

N.H. Kim, J.H. Choi, and D. An. *Prognostics and health management of engineering systems: An introduction*. Springer International Publishing Switzerland, 2016. ISBN 9783319447421. doi: 10.1007/978-3-319-44742-1.

R. Y. Kim and S. R. Soni. Experimental and Analytical Studies On the Onset of Delamination in Laminated Composites. *Journal of Composite Materials*, 18(1):70–80, 1984. ISSN 1530793X. doi: 10.1177/002199838401800106.

D.J. Lekou and T.P. Philippidis. Mechanical property variability in FRP laminates and its effect on failure prediction. *Composites Part B: Engineering*, 39(7-8):1247–1256, 10 2008. ISSN 1359-8368. doi: 10.1016/J.COMPOSITESB.2008.01.004. URL https://www.sciencedirect.com/science/article/pii/S1359836808000097.

L. Liao and F. Köttig. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1):191–207, 2014. ISSN 00189529. doi: 10.1109/TR.2014.2299152.

Y. Liu, S. Mohanty, and A. Chattopadhyay. A Gaussian process based prognostics framework for composite structures. In *Modeling, Signal Processing, and Control for Smart Structures 2009*, volume 7286, page 72860J. SPIE, 3 2009. ISBN 9780819475466. doi: 10.1117/12.815889.

T.H. Loutas, V. Kostopoulos, C. Ramirez-Jimenez, and M. Pharaoh. Damage evolution in center-holed glass/polyester composites under quasi-static loading using time/frequency analysis of acoustic emission monitored waveforms. *Composites Science and Technology*, 66(10):1366–1375, 8 2006. ISSN 02663538. doi: 10.1016/j.compscitech.2005.09.011.

T.H. Loutas, N. Eleftheroglou, and D.S. Zarouchas. A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data. *Composite Structures*, 161:522–529, 2017. ISSN 02638223. doi: 10.1016/j.compstruct.2016.10.109.

D.J.C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3): 448–472, 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448.

S. Mall, D. W. Katwyk, R. L. Bolick, A. D. Kelkar, and D. C. Davis. Tension-compression fatigue behavior of a H-VARTM manufactured unnotched and notched carbon/epoxy composite. *Composite Structures*, 90(2): 201–207, 9 2009. ISSN 02638223. doi: 10.1016/j.compstruct.2009.03.015.

C. Malte Markussen. CXG25 Gigapan, 2015. URL http://www.gigapan.com/gigapans/3e820fd94450a193955640054baa9e0b.

A. Marec, J. H. Thomas, and R. El Guerjouma. Damage characterization of polymer-based composite materials: Multivariable analysis and wavelet transform for clustering acoustic emission data. *Mechanical Systems and Signal Processing*, 22(6):1441–1464, 2008. ISSN 10961216. doi: 10.1016/j.ymssp.2007.11.029.

J.P. McCrory, S.K. Al-Jumaili, D. Crivelli, M.R. Pearson, M.J. Eaton, C.A. Featherston, M. Guagliano, K.M. Holford, and R. Pullin. Damage classification in carbon fibre composites using acoustic emission: A comparison of three techniques. *Composites Part B: Engineering*, 68:424–430, 2015. ISSN 13598368. doi: 10.1016/j.compositesb.2014.08.046.

W.Q. Meeker and L.A. Escobar. *Statistical Methods for Reliability Data*. Wiley, New York, 1998. ISBN 978-1-118-62597-2.

L.P. Mikkelsen. The fatigue damage evolution in the load-carrying composite laminates of wind turbine blades. In *Fatigue Life Prediction of Composites and Composite Structures*, pages 569–603. Woodhead Publishing, 1 2020. ISBN 9780081025758. doi: 10.1016/b978-0-08-102575-8.00016-4. URL https://www.sciencedirect.com/science/article/pii/B9780081025758000164?via%3Dihub.

K.P. Murphy. *Machine learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN 978-0-262-01802-9.

R.P.L. Nijssen, A.M. van Wingerde, and D.R.V. van Delft. The OptiDAT materials fatigue database. In H. Lilholt, B. Madsen, T.L. Andersen, L.P. Mikkelsen, and A. Thygesen, editors, *Polymer composite materials for wind power turbines*, pages 257–263. Knowledge Centre WMC, 2006.

M. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010.

J.C. Pinheiro and D.M. Bates. Unconstrained Parameterizations for Variance-Covariance Matrices. *University of Wisconsin*, 1988.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006. ISBN 978-0262182539.

K.L. Reifsnider and R. Jamison. Fracture of fatigue-loaded composite laminates. *International Journal of Fatigue*, 4(4):187–197, 1982. ISSN 01421123. doi: 10.1016/0142-1123(82)90001-9.

R.R. Richardson, M.A. Osborne, and D.A. Howey. Gaussian process regression for forecasting battery state of health. *Journal of Power Sources*, 357:209–219, 2017. ISSN 03787753. doi: 10.1016/j.jpowsour.2017.05.004.

A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel. Evaluating algorithm performance metrics tailored for prognostics. In *IEEE Aerospace Conference Proceedings*, 2009. ISBN 9781424426225. doi: 10.1109/AERO.2009.4839666.

A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel. Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, 1(1), 2010. ISSN 21532648. URL https://ntrs.nasa.gov/search.jsp?R=20100039169.

X.S. Si, W. Wang, C.H. Hu, and D.H. Zhou. Remaining useful life estimation - A review on the statistical data driven approaches, 8 2011. ISSN 03772217.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. ISSN 15337928.

S. Sun, G. Zhang, C. Wang, W. Zeng, J. Li, and R. Grosse. Differentiable compositional kernel learning for Gaussian processes. In *35th International Conference on Machine Learning, ICML 2018*, volume 11, pages 7676–7696, 2018. ISBN 9781510867963.

M. Surgeon and M. Wevers. Modal analysis of acoustic emission signals from CFRP laminates. *NDT and E International*, 32(6):311–322, 1999. ISSN 09638695. doi: 10.1016/S0963-8695(98)00077-2.

P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T.A. Tameghe, and G. Ekemb. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges, 2014. ISSN 19961073.

S. Uckun, K. Goebel, and P.J.F. Lucas. Standardizing research methods for prognostics. In *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008. ISBN 9781424419357. doi: 10.1109/PHM.2008.4711437.

B.S. Wei, S. Johnson, and R. Haj-Ali. A stochastic fatigue damage method for composite materials based on Markov chains and infrared thermography. *International Journal of Fatigue*, 32(2):350–360, 2 2010. ISSN 01421123. doi: 10.1016/j.ijfatigue.2009.07.010.

S. Wicaksono and G.B. Chai. A review of advances in fatigue and life prediction of fiber-reinforced composites, 7 2013. ISSN 14644207.

W. Wu, J. Hu, and J. Zhang. Prognostics of machine health condition using an improved ARIMA-based prediction method. In *ICIEA 2007: 2007 Second IEEE Conference on Industrial Electronics and Applications*, pages 1062–1067, 2007. ISBN 1424407370. doi: 10.1109/ICIEA.2007.4318571.

J.N. Yang, D.L. Jones, S.H. Yang, and A. Meskini. A Stiffness Degradation Model for Graphite/Epoxy Laminates. *Journal of Composite Materials*, 24(7):753–769, 1990. ISSN 1530793X. doi: 10.1177/002199839002400705.

L. Ye. On fatigue damage accumulation and material degradation in composite materials. *Composites Science and Technology*, 36(4):339–350, 1989. ISSN 02663538. doi: 10.1016/0266-3538(89)90046-8.

J. Zangenberg, P. Brøndsted, and J.W. Gillespie. Fatigue damage propagation in unidirectional glass fibre reinforced composites made of a non-crimp fabric. *Journal of Composite Materials*, 48(22):2711–2727, 9 2014. ISSN 0021-9983. doi: 10.1177/0021998313502062. URL http://journals.sagepub.com/doi/10.1177/0021998313502062.

D.S. Zarouchas. A mechanics-informed method for real-time acoustic emission source classification during fatigue loading of composite structures. In *Structural Health Monitoring 2017: Real-Time Material State Awareness and Data-Driven Safety Assurance - Proceedings of the 11th International Workshop on Structural Health Monitoring, IWSHM 2017*, volume 2, pages 2147–2153. DEStech Publications, 2017. ISBN 9781605953304. doi: 10.12783/shm2017/14104.

A

Mathematical background

A.1. Gamma functions

The gamma function $\Gamma(z)$ is shown in equation (A.1). When taking other limits, the incomplete gamma function is obtained. The lower incomplete gamma function $\gamma(s,x)$, used in this thesis, is shown in equation (A.2).

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} \mathrm{d}t \tag{A.1}$$

$$\gamma(s,x) = \int_0^x t^{s-1} e^{-t} \mathrm{d}t \tag{A.2}$$

A.2. Distances

When the scale of each coordinate axis is the same, there is a need for an isotropic distance measure. Hence, the distance d between two points $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ is normally calculated as the Euclidean distance d_E:

$$d_E\left(\boldsymbol{x}_p, \boldsymbol{x}_q\right) = \sqrt{\left(\boldsymbol{x}_p - \boldsymbol{x}_q\right)^{\mathrm{T}} \left(\boldsymbol{x}_p - \boldsymbol{x}_q\right)} \tag{A.3}$$

In order to allow for correlation between different dimensions, the Mahalanobis distance (equation (A.4)) can be used (Osborne, 2010). This is an anisotropic distance measure, meaning that it provides different scales between dimensions, according to the covariance matrix $\boldsymbol{\Sigma}_M$.

$$d_M\left(\boldsymbol{x}_p, \boldsymbol{x}_q, \boldsymbol{\Sigma}_M\right) = \sqrt{\left(\boldsymbol{x}_p - \boldsymbol{x}_q\right)^{\mathrm{T}} \boldsymbol{\Sigma}_M^{-1} \left(\boldsymbol{x}_p - \boldsymbol{x}_q\right)} \tag{A.4}$$

The D-dimensional matrix $\boldsymbol{\Sigma}_M$ can take different forms. The simplest of these is $\boldsymbol{\Sigma}_M = l^2 \boldsymbol{I}$. The diagonal of $\boldsymbol{\Sigma}_M$ can also be adjusted in order to scale dimensions differently. Finally, adjusting the non-diagonal terms in $\boldsymbol{\Sigma}_M$ allows for correlations between input dimensions. (Osborne, 2010) When for the above function the covariance matrix is set to identity, i.e. $\boldsymbol{\Sigma}_M = \boldsymbol{I}$, the Mahalanobis distance simplifies to Euclidean.

A.3. Spherical parameterisation

Pinheiro and Bates (1988) describe how an $N \times N$-dimensional covariance matrix $\boldsymbol{\Sigma}$ can rewritten as the product of its Cholesky factorization:

$$\boldsymbol{\Sigma} = \boldsymbol{L}^{\mathrm{T}} \boldsymbol{L} \tag{A.5}$$

In this equation, $\boldsymbol{L} = \boldsymbol{S}\mathrm{diag}(\boldsymbol{\tau})$ is an $N \times N$-dimensional, upper triangular matrix. It is the product of the matrix $\boldsymbol{S}$ and scaling vector $\boldsymbol{\tau}$. In $\boldsymbol{S}$, the N^{th} column contains the spherical coordinates in $\mathbb{R}^N$, along the boundaries of a hypersphere in $\mathbb{S}^{N-1}$. The hypersphere set is defined as $\mathbb{S}^{N-1} = \{x \in \mathbb{R}^N : \|\boldsymbol{x}\| = 1\}$. To visualise this concept, one can think of a sphere. This structure lies in a 3-dimensional space $\mathbb{R}^3$, but all points on its boundary lie in the 2-dimensional plane $\mathbb{S}^2$. The components ξ_j of the $\mathbb{S}^{N-1}$ with radius 1, and $N-1$ spherical coordinates

$\{\phi_i : 0 < \phi_i < \pi\}$ are written as:

$$\xi_1 = \cos\phi_1 \tag{A.6}$$

$$\xi_j = \cos\phi_j \prod_{k=1}^{j-1} \sin\phi_k, \qquad (j = 2, ..., N-1) \tag{A.7}$$

$$\xi_n = \prod_{k=1}^{N-1} \sin\phi_k \tag{A.8}$$

Now, an example of $\boldsymbol{S}$ where $N = 3$, is as follows:

$$\boldsymbol{S} = \begin{bmatrix} 1 & \cos\phi_1 & \cos\phi_2 \\ 0 & \sin\phi_1 & \sin\phi_2\cos\phi_3 \\ 0 & 0 & \sin\phi_2\sin\phi_3 \end{bmatrix} \tag{A.9}$$

In this form, $\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}$ can represent the correlation coefficients between pairs of N discrete variables. All values in this form are between -1 and 1, with the diagonal containing ones. The vector $\boldsymbol{\tau}$ (where the elements $\{\tau_i : \tau_i > 0\}$) allows for scaling of the different dimensions. If the scale of each dimension is the same, diag(τ) can be set to $\mathbf{1}$ and therefore be omitted in the above equations.

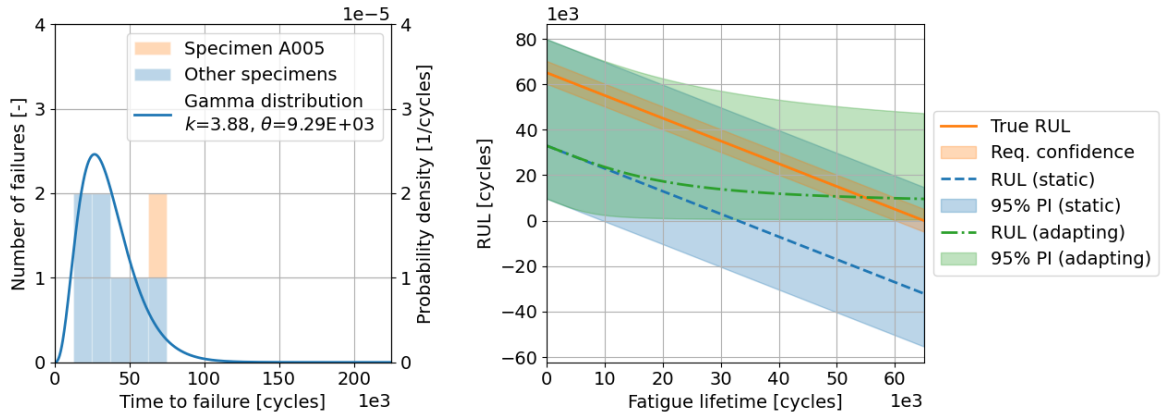
The total number of parameters required for $\boldsymbol{S}$ and $\boldsymbol{\tau}$ is $\frac{1}{2}N(N+1)$. While this is a large number of parameters, the spherical parameterisation is able to construct every possible covariance matrix (Osborne, 2010).

B

Remaining results

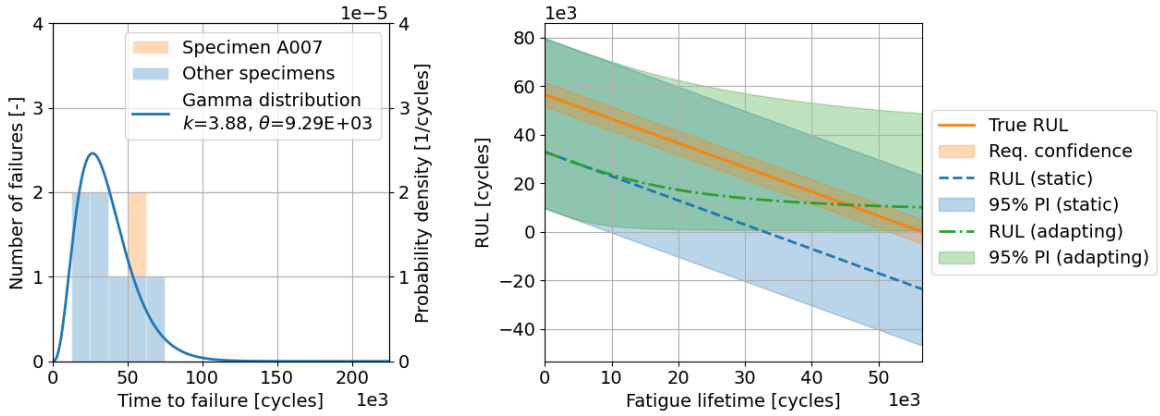
B.1. Statistical model

B.1.1. Remaining useful life predictions



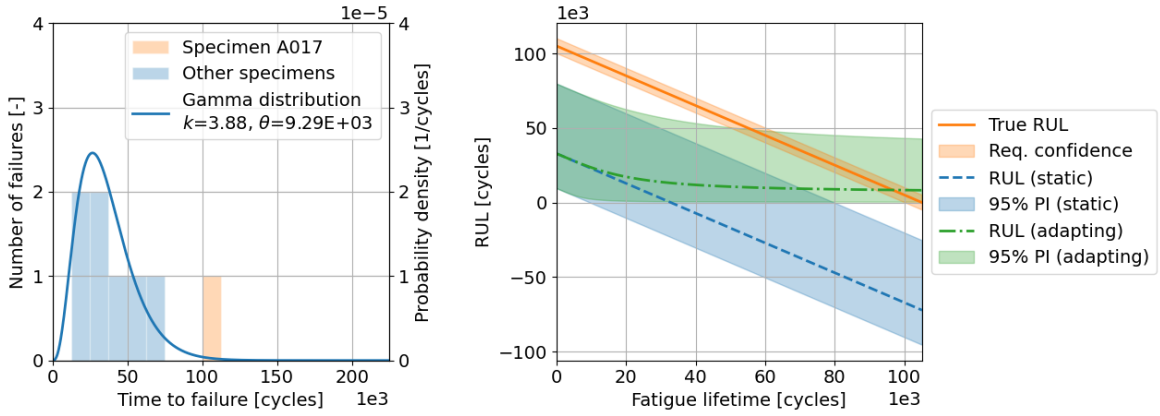
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.1: Results of the statistical model, trained on CAF data, tested on specimen A005



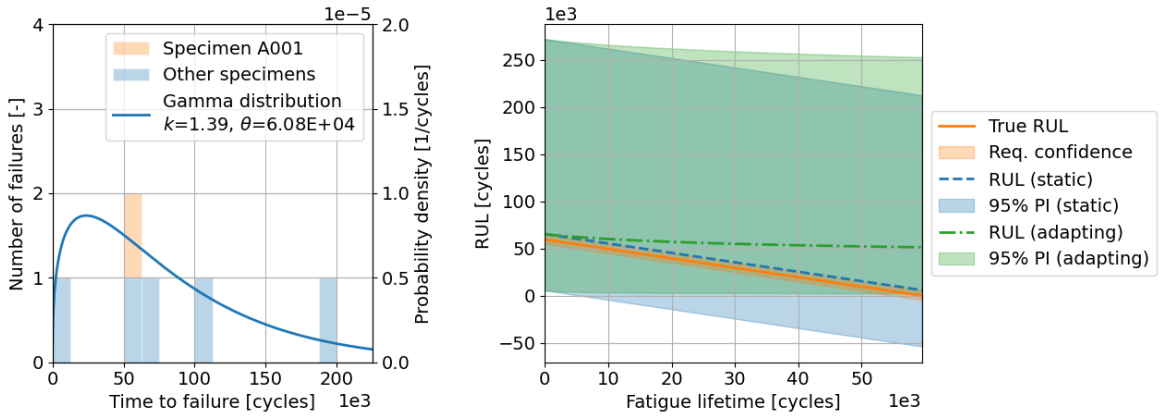
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.2: Results of the statistical model, trained on CAF data, tested on specimen A007



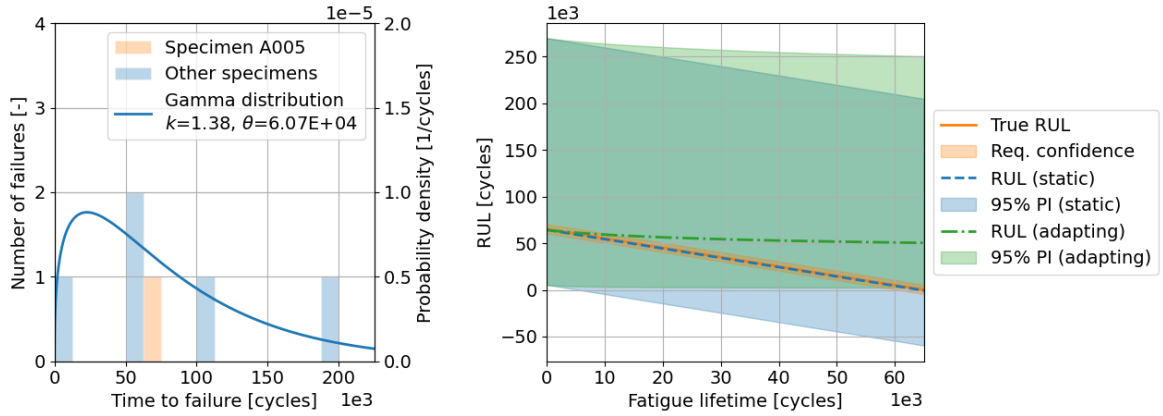
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.3: Results of the statistical model, trained on CAF data, tested on specimen A017



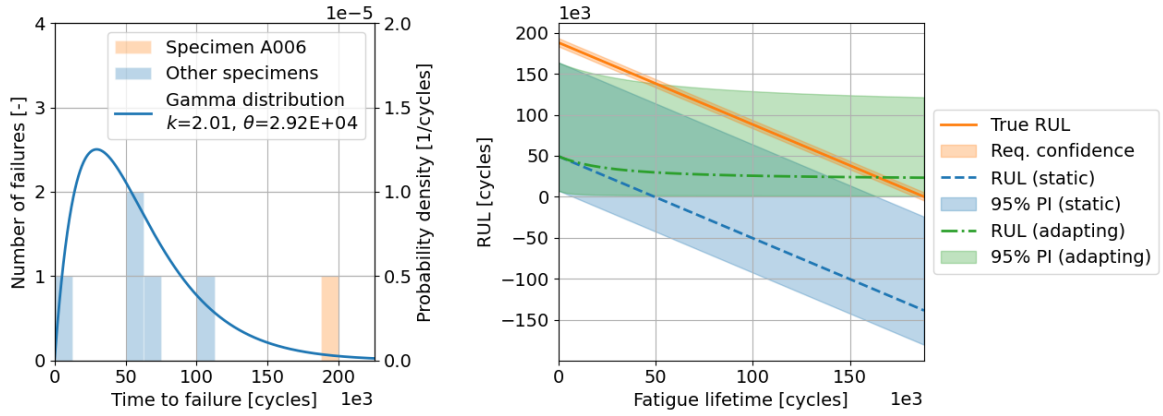
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.4: Results of the statistical model, trained on VAF data, tested on specimen A001



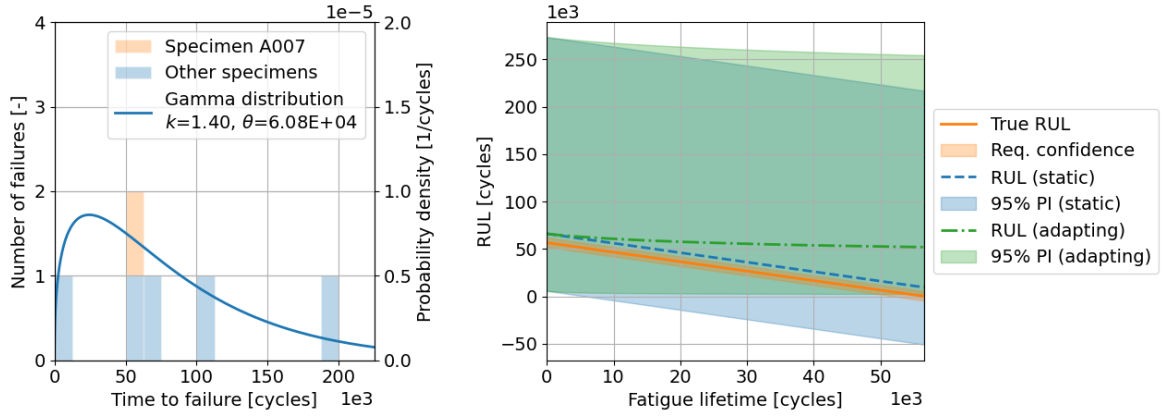
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.5: Results of the statistical model, trained on VAF data, tested on specimen A005



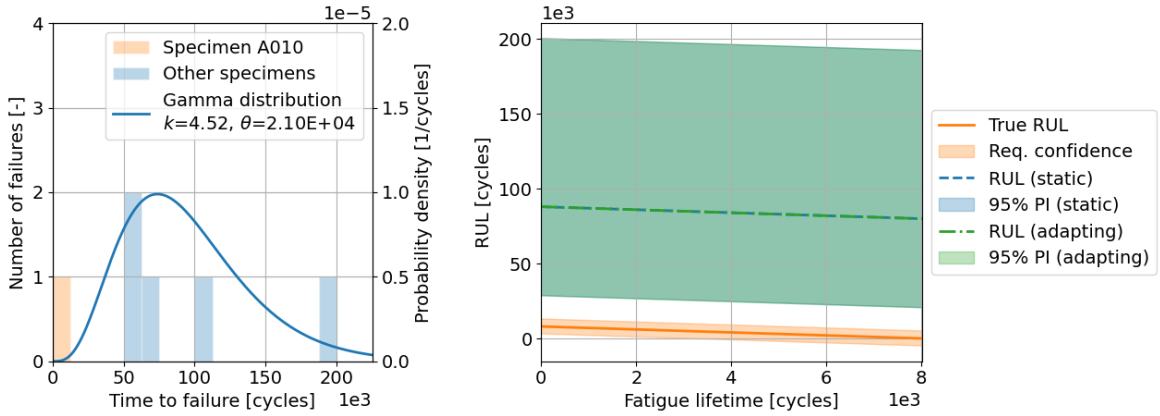
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.6: Results of the statistical model, trained on VAF data, tested on specimen A006



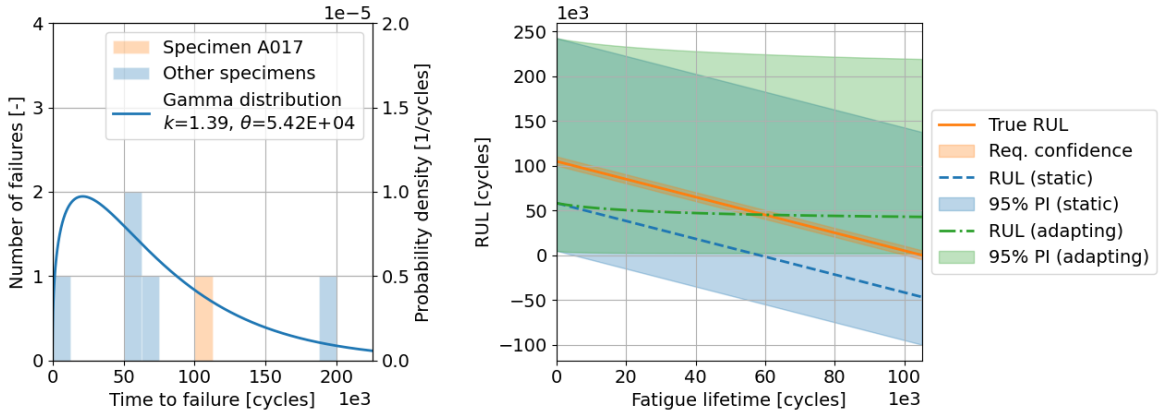
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.7: Results of the statistical model, trained on VAF data, tested on specimen A007



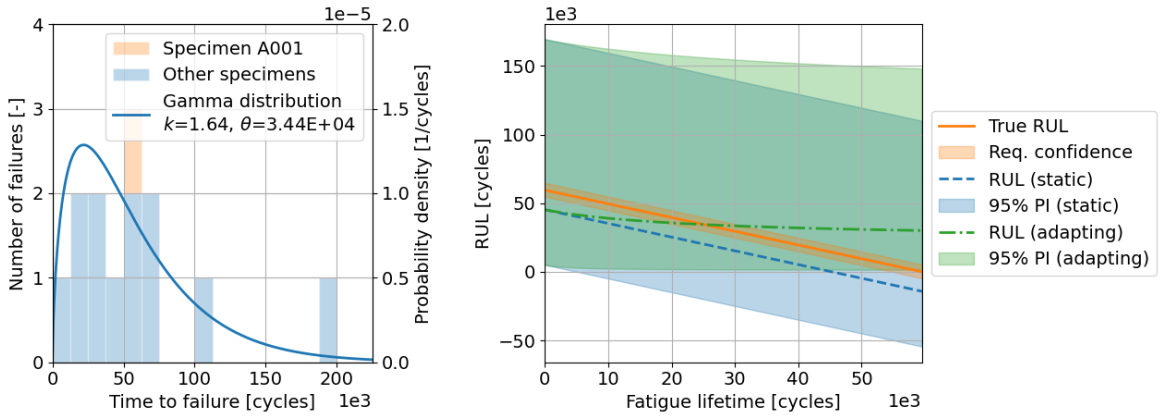
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.8: Results of the statistical model, trained on VAF data, tested on specimen A010



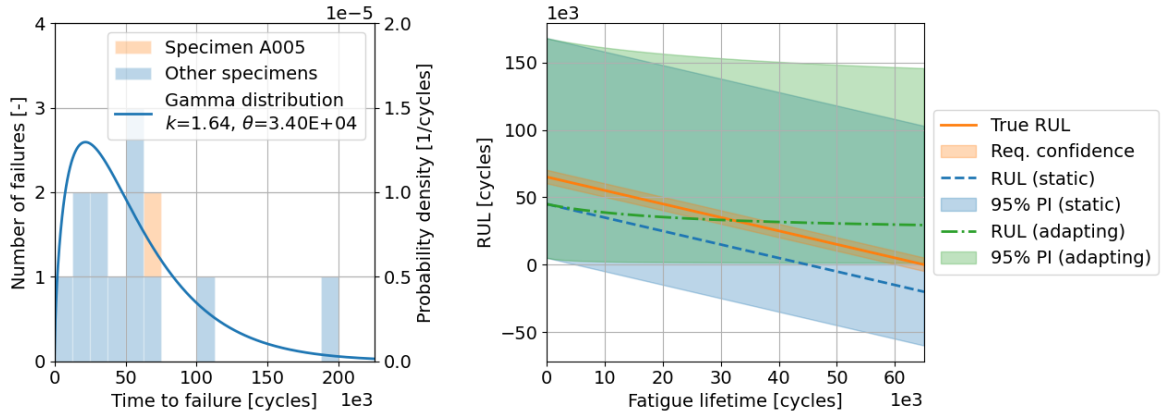
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.9: Results of the statistical model, trained on VAF data, tested on specimen A017



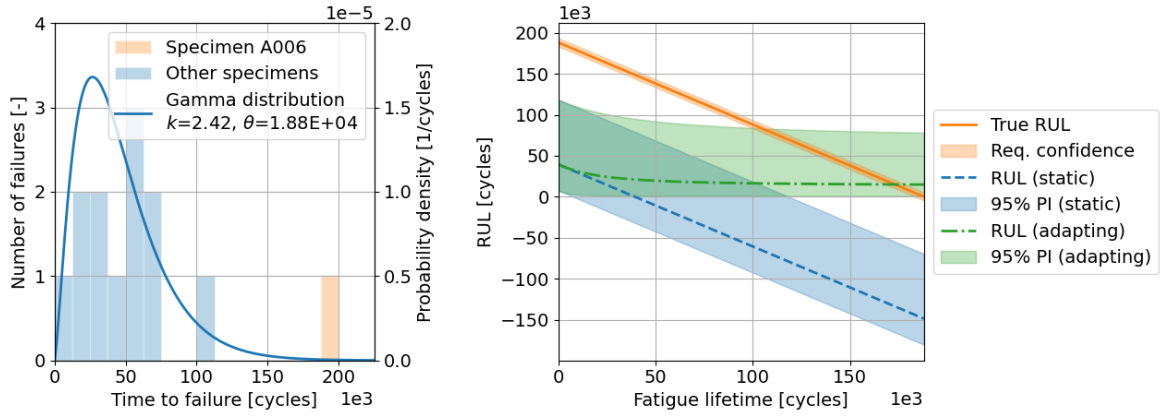
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.10: Results of the statistical model, trained on CAF and VAF data, tested on specimen A001



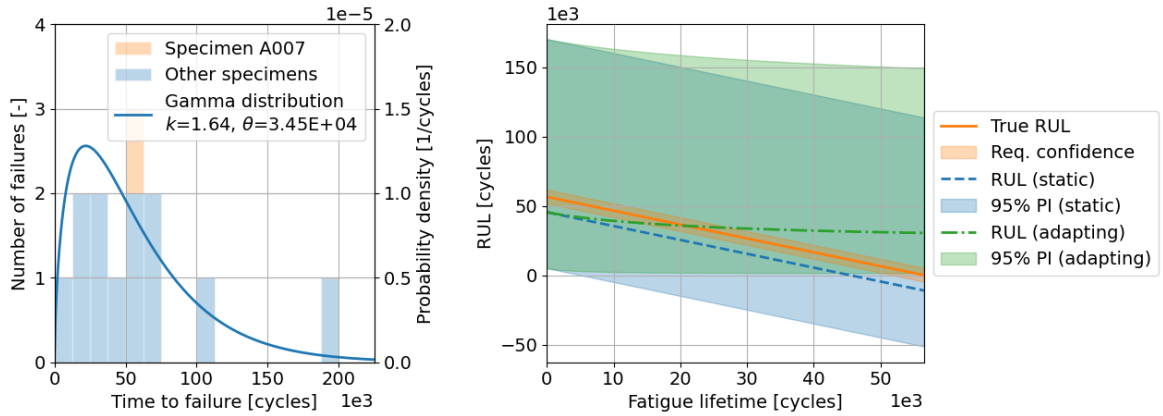
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.11: Results of the statistical model, trained on CAF and VAF data, tested on specimen A005



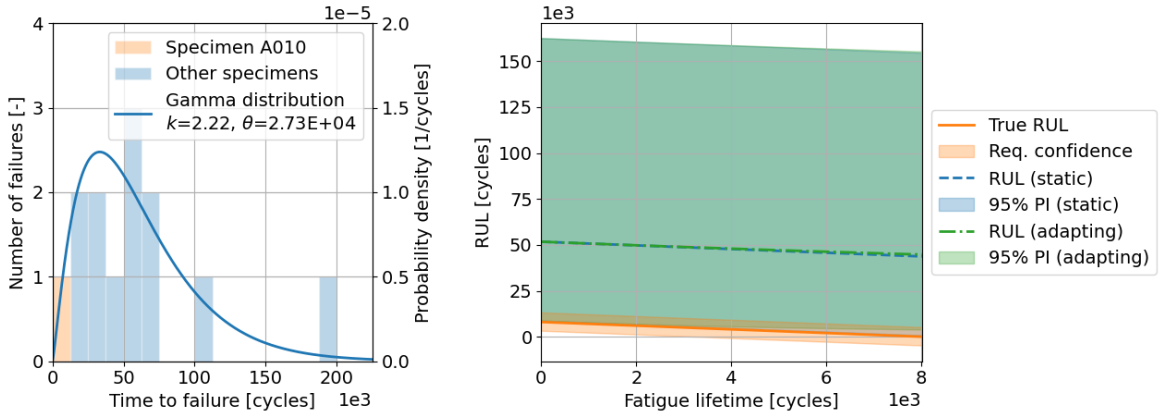
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.12: Results of the statistical model, trained on CAF and VAF data, tested on specimen A006



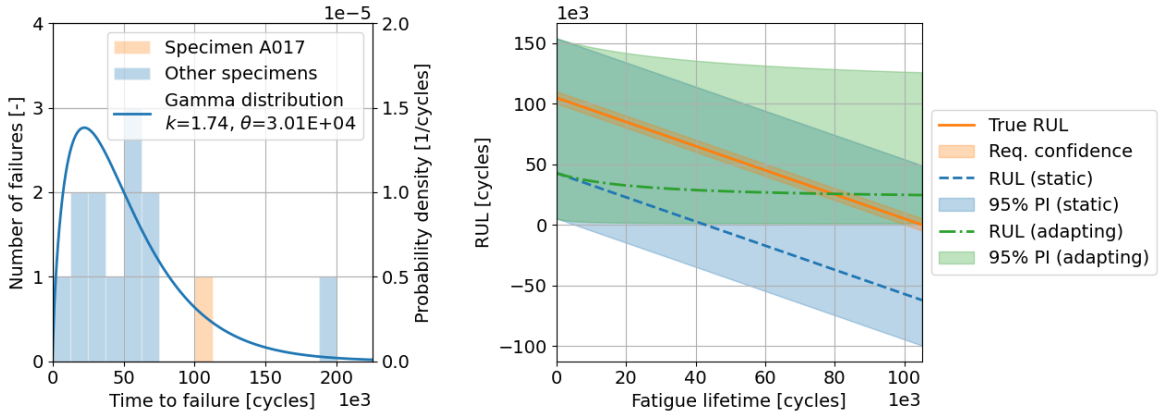
(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.13: Results of the statistical model, trained on CAF and VAF data, tested on specimen A007



(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.14: Results of the statistical model, trained on CAF and VAF data, tested on specimen A010



(a) Fitted PDF on the training data, and the test specimen (b) RUL predictions

Figure B.15: Results of the statistical model, trained on CAF and VAF data, tested on specimen A017

B.1.2. Performance metric tables

Table B.1: Performance metrics for the static statistical model, trained on CAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	7.08e+08	239	-	7.27	-288	-	-
A005	1.03e+09	269	-	5.2	-337	-	-
A006	2.41e+10	537	-	0.000197	-807	-	-
A007	5.57e+08	222	-	8.64	-258	-	-
A010	6.22e+08	1.05e+03	-	6.22	-1.31e+03	-	-
A017	5.21e+09	407	-	0.28	-574	-	-

Table B.2: Performance metrics for the adapting statistical model, trained on CAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	3.14e+08	82.7	-	24.9	-14.8	2.14e+04	3.93e+04
A005	4.54e+08	82.2	-	23.5	-8.88	2.41e+04	4.06e+04
A006	9.5e+09	90.2	-	9.1	8.38	8.39e+04	8.64e+04
A007	2.48e+08	83.2	-	25.7	-19.1	2e+04	3.87e+04
A010	6.26e+08	1.06e+03	-	5.7	-1.32e+03	-	3.51e+04
A017	2.2e+09	85.1	-	15.7	5.89	4.38e+04	5.31e+04

Table B.3: Performance metrics for the static statistical model, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	3.4e+07	52.5	-	6.9	13.6	-	-
A005	5.01e+05	5.93	-	6.51	88.9	-	-
A006	1.92e+10	480	-	0.36	-710	-	-
A007	8.9e+07	88.6	-	7.1	-44.3	-	-
A010	6.4e+09	3.37e+03	-	0.129	-4.38e+03	-	-
A017	2.18e+09	264	-	3.88	-337	-	-

Table B.4: Performance metrics for the adapting statistical model, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	8.78e+08	376	-	9.65	-560	-	1.32e+05
A005	7.38e+08	334	-	9.69	-496	-	1.31e+05
A006	6.78e+09	101	-	8.54	-28.4	7.63e+04	1.1e+05
A007	9.7e+08	402	-	9.62	-599	-	1.32e+05
A010	6.4e+09	3.37e+03	-	0.123	-4.38e+03	-	8.59e+04
A017	7.56e+08	184	-	9.53	-223	4.54e+04	1.23e+05

Table B.5: Performance metrics for the static statistical model, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	2.04e+08	128	-	8.13	-109	-	-
A005	4.05e+08	168	-	7.32	-175	-	-
A006	2.21e+10	514	-	0.0504	-769	-	-
A007	1.22e+08	104	-	8.58	-68.9	-	-
A010	1.91e+09	1.84e+03	-	5.4	-2.36e+03	-	-
A017	3.88e+09	351	-	2.8	-482	-	-

Table B.6: Performance metrics for the adapting statistical model, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	2.09e+08	198	-	14.3	-255	-	8.2e+04
A005	2.32e+08	180	-	14.2	-219	-	8.19e+04
A006	8.18e+09	92.9	-	9	-4.62	8.01e+04	9.72e+04
A007	2.09e+08	210	-	14.4	-278	-	8.21e+04
A010	1.94e+09	1.87e+03	-	4.56	-2.39e+03	-	7.68e+04
A017	1.26e+09	119	-	12.5	-82.5	3.87e+04	8.3e+04

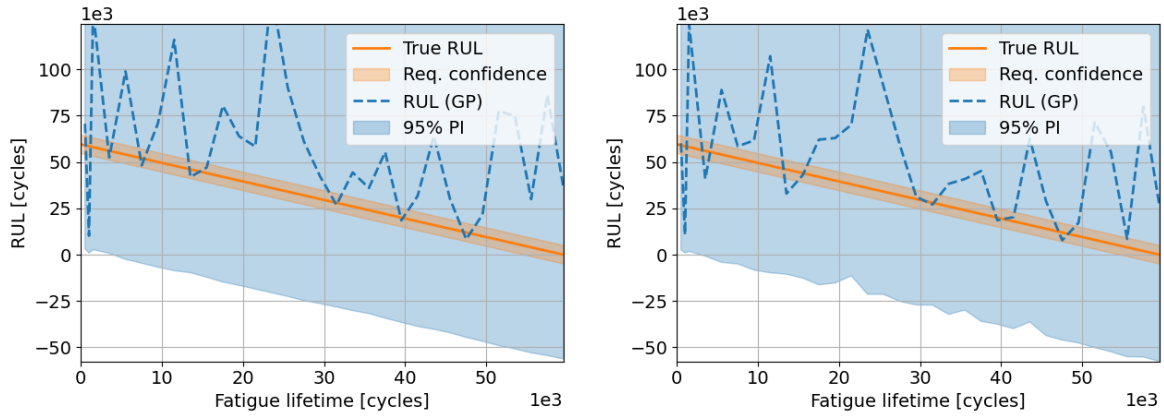
B.2. Gaussian process regression

B.2.1. Cumulative energy predictions mean absolute percentage error

Table B.7: Median, and 1st and 3rd quartile values for the MAPE of the cumulative energy predictions by the GP model, grouped by kernel functions and training data

Data-set	Kernel functions	MAPE [%]		
		1st quartile	median	3rd quartile
VAF	Ma3+lin	6.13	14.3	34.6
	Ma5+lin	8.35	17.5	42.7
CAF and VAF	Ma3+lin	7.03	18.0	41.9
	Ma5+lin	7.97	20.8	38.9

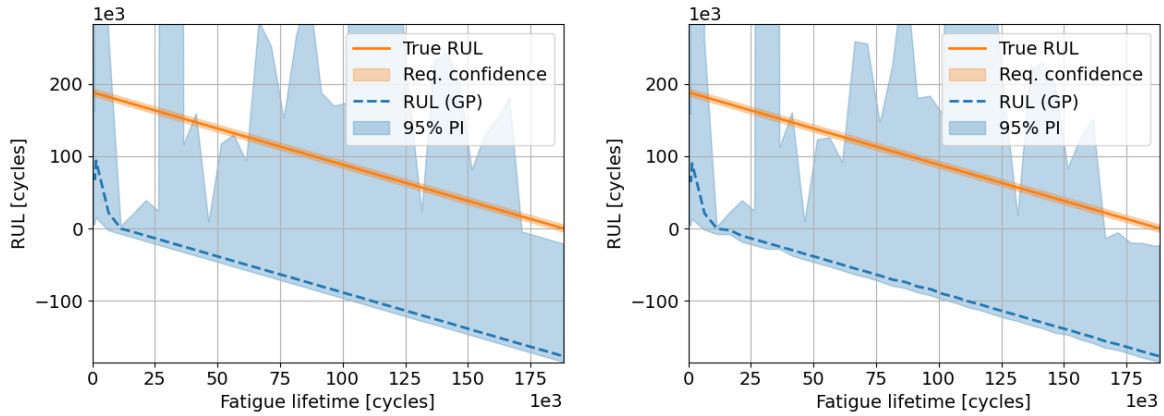
B.2.2. Remaining useful life predictions



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

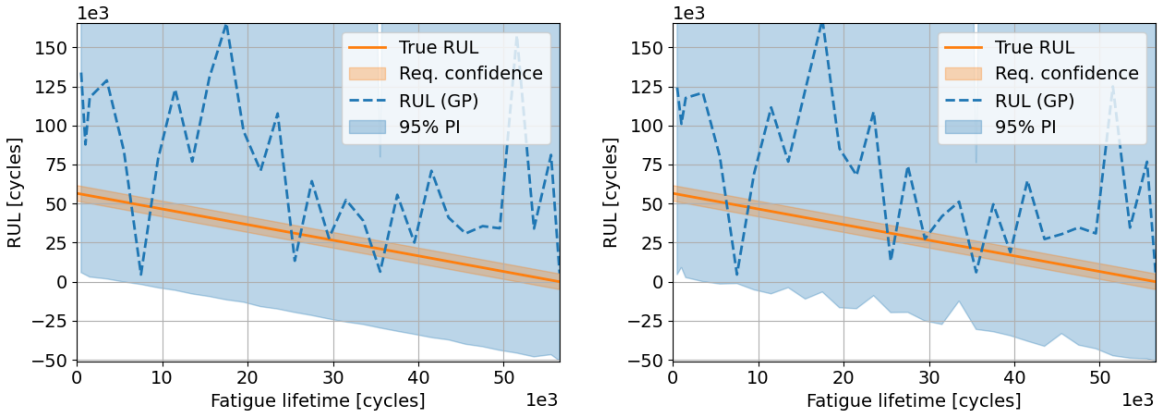
Figure B.16: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A001



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

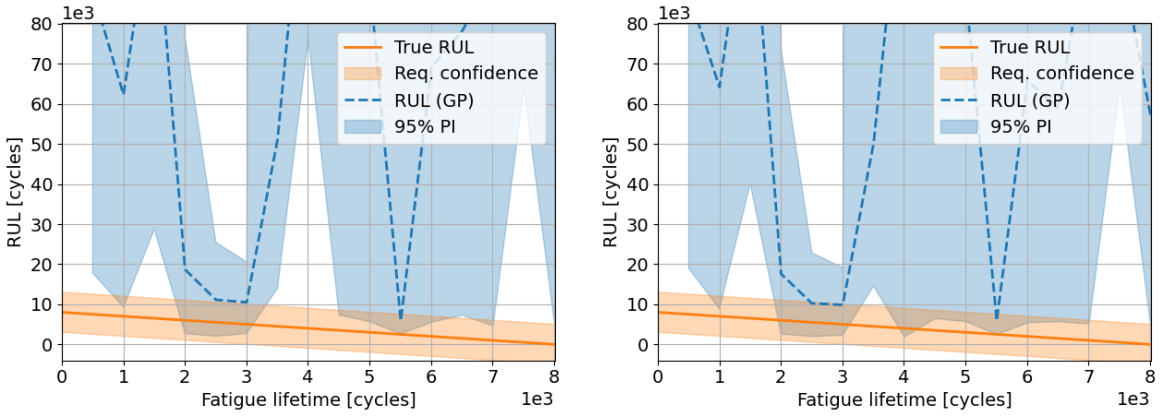
Figure B.17: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A006



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

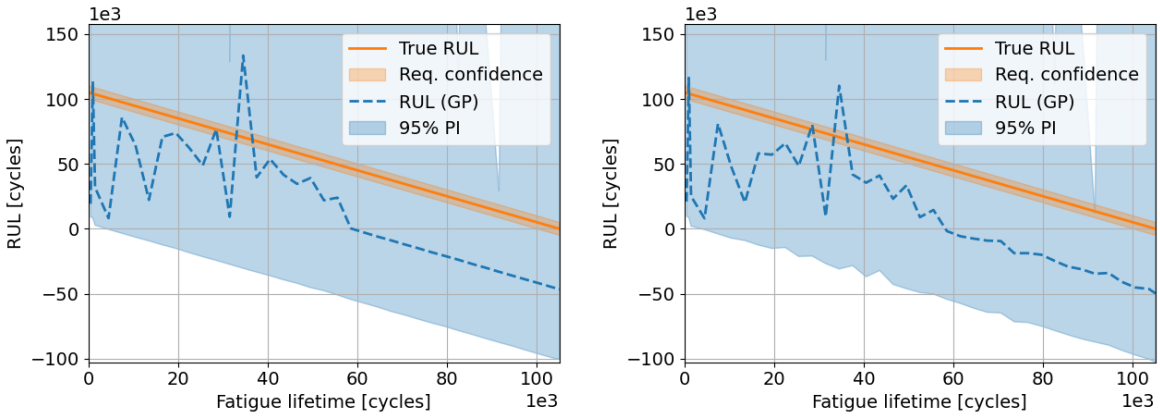
Figure B.18: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A007



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

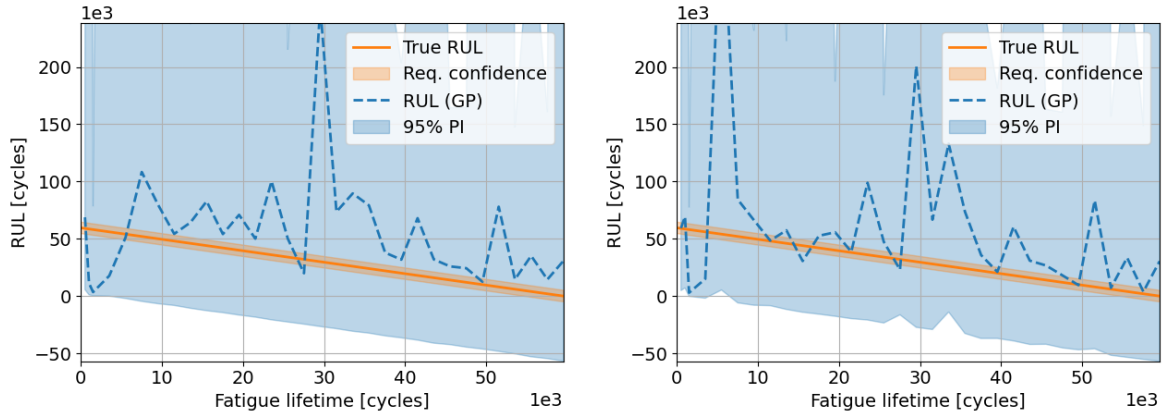
Figure B.19: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A010



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

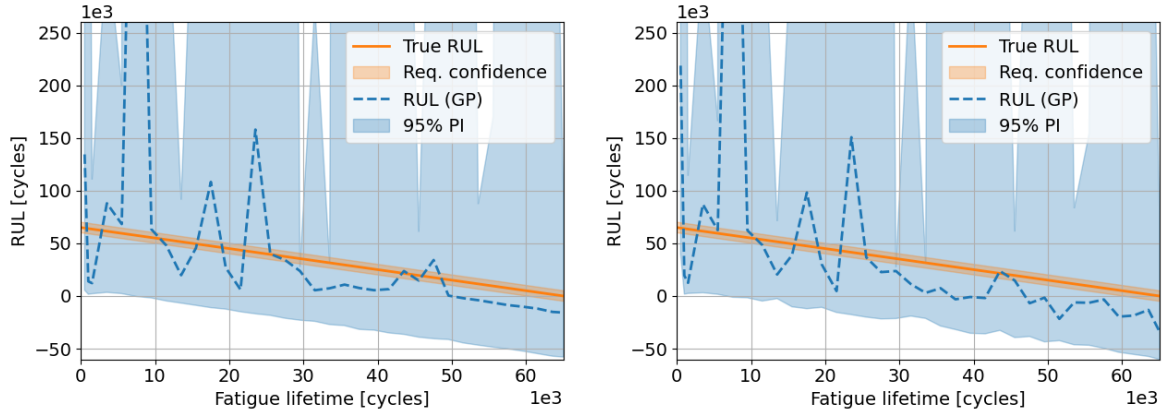
Figure B.20: RUL predictions by the GP regression with Ma3+lin kernels, trained on VAF data, tested on specimen A017



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.21: RUL predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A001



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.22: RUL predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A005

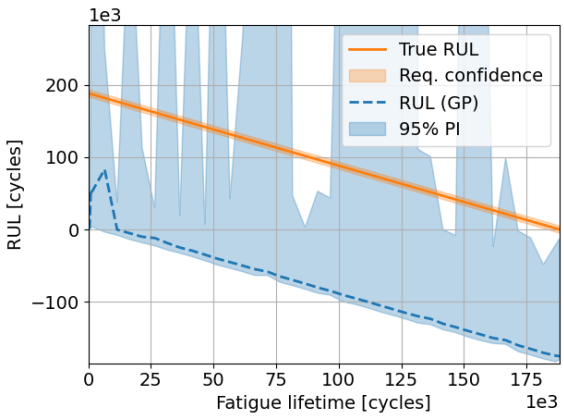
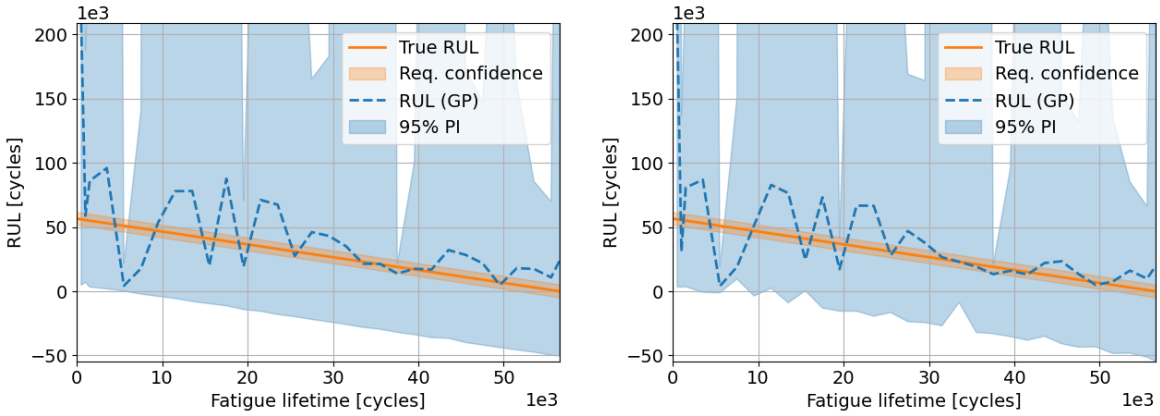


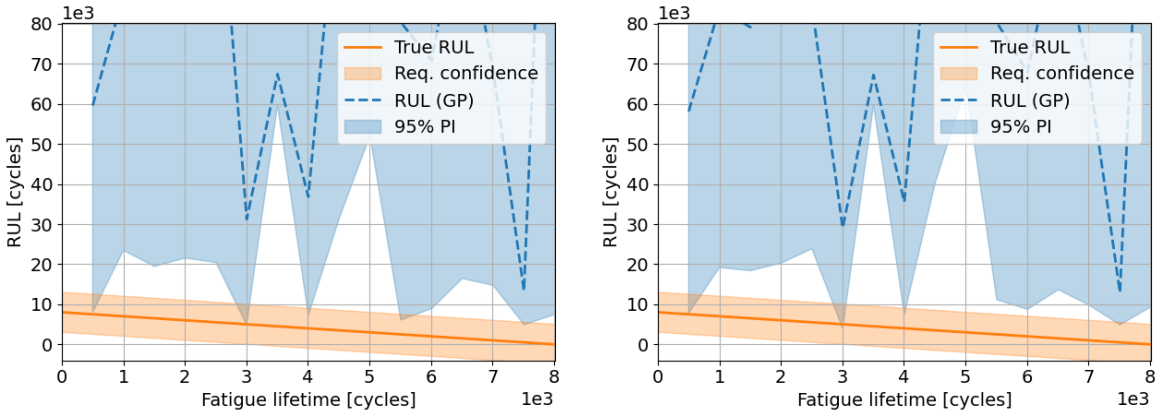
Figure B.23: Prediction from GP regression with correlation adjustment, Ma5+lin kernels, trained on VAF data, tested on specimen A006. The plain prediction can be found in section 5.2.2.



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

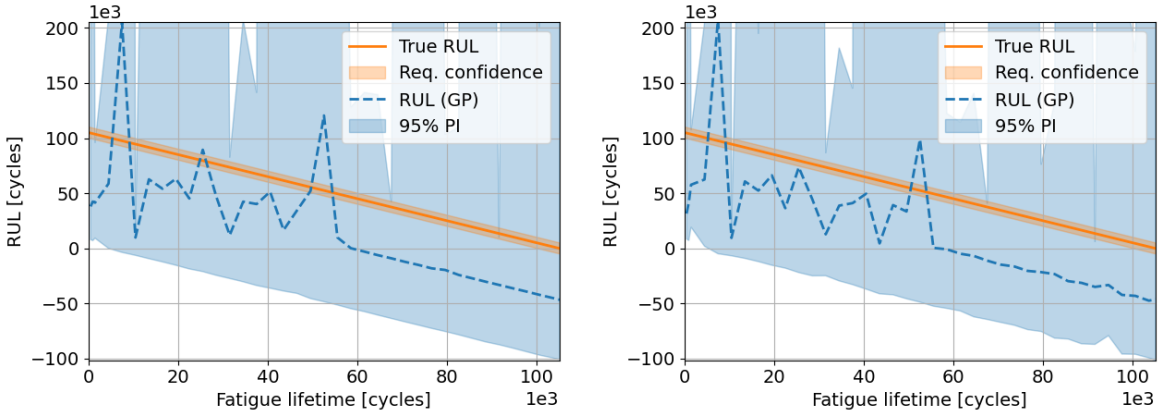
Figure B.24: RUL predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A007



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

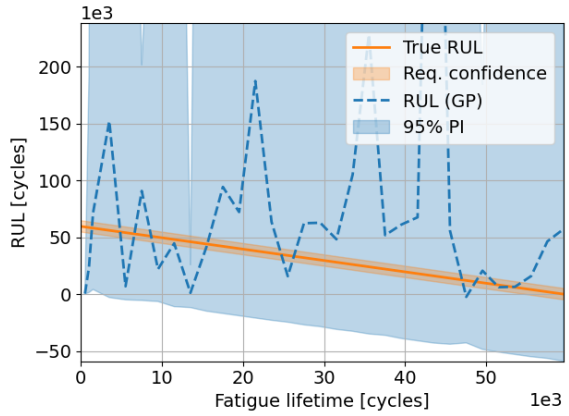
Figure B.25: RUL predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A010



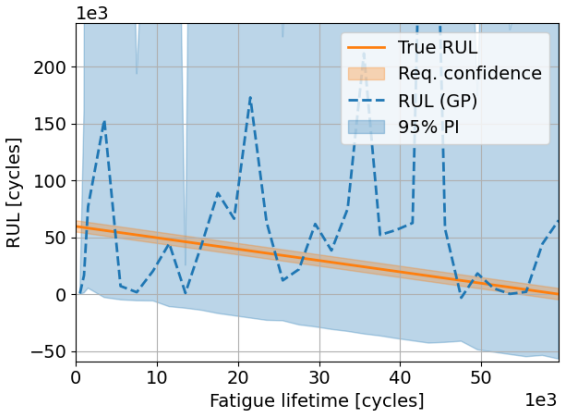
(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.26: RUL predictions by the GP regression with Ma5+lin kernels, trained on VAF data, tested on specimen A017

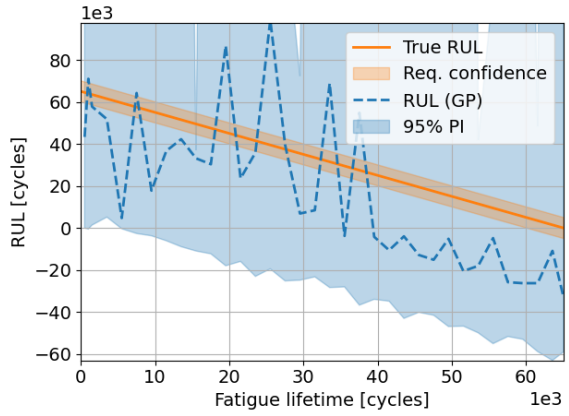


(a) Prediction from plain GP regression

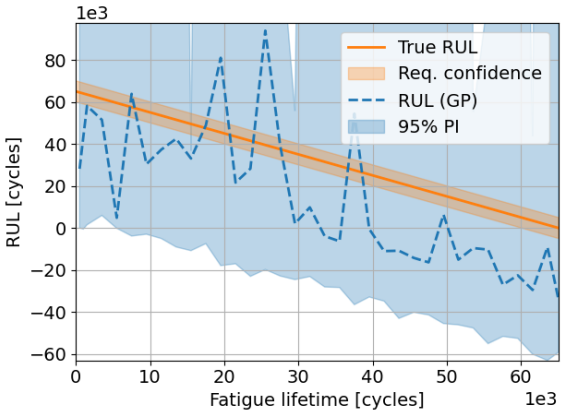


(b) Prediction from GP regression with correlation adjustment

Figure B.27: RUL predictions by the GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A001

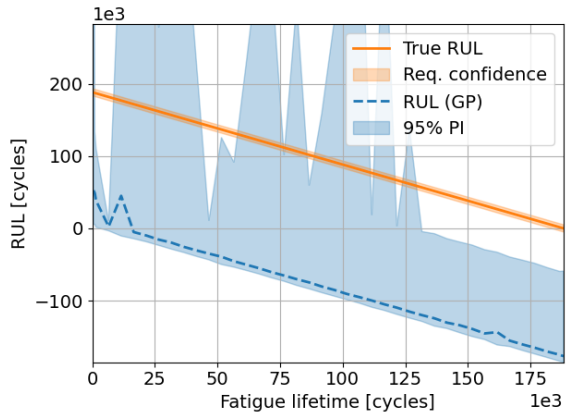


(a) Prediction from plain GP regression

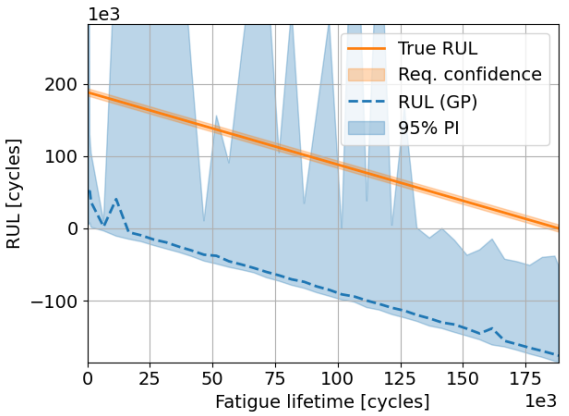


(b) Prediction from GP regression with correlation adjustment

Figure B.28: RUL predictions by the GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A005

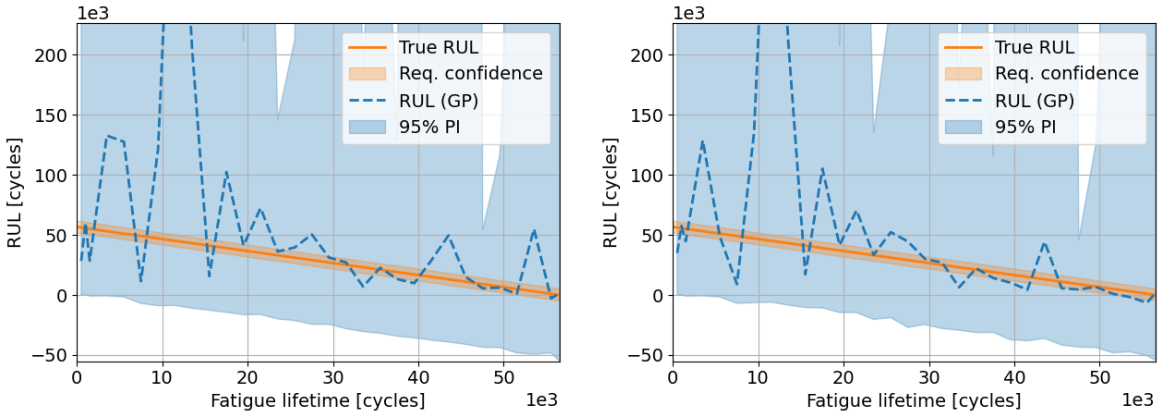


(a) Prediction from plain GP regression



(b) Prediction from GP regression with correlation adjustment

Figure B.29: RUL predictions by the GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A006



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.30: RUL predictions by the GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A007

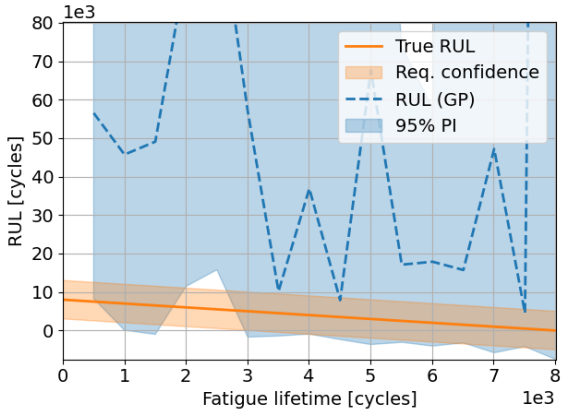
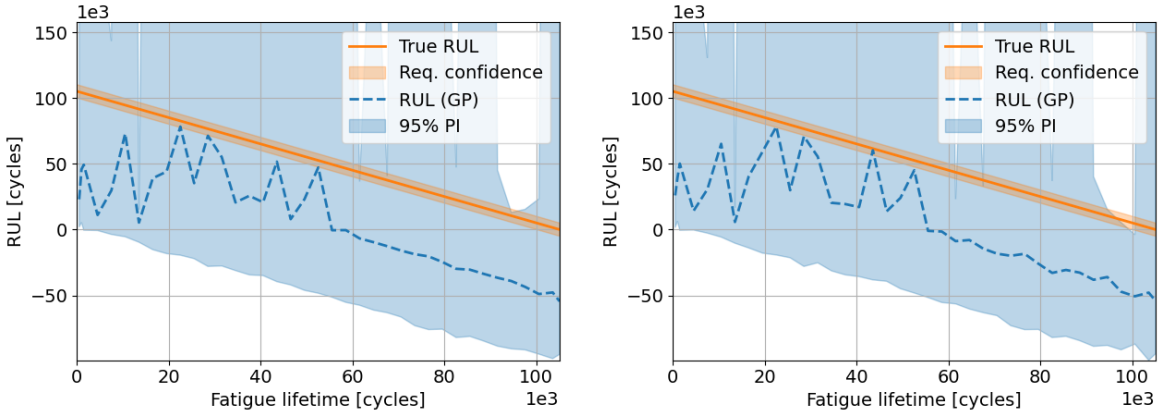


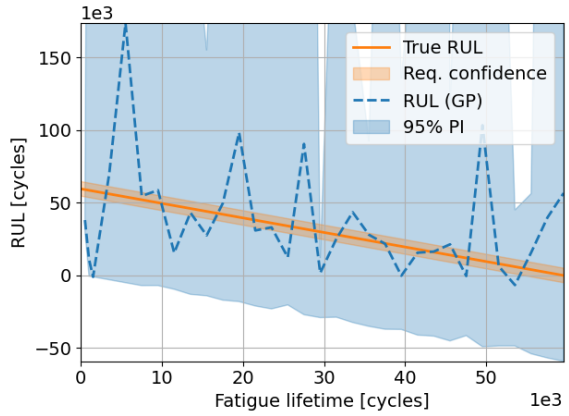
Figure B.31: Prediction from GP regression with correlation adjustment, Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A010. The plain prediction can be found in section 5.2.4.



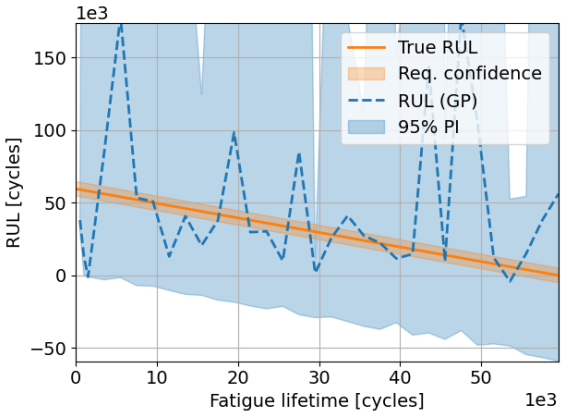
(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.32: RUL predictions by the GP regression with Ma3+lin kernels, trained on CAF and VAF data, tested on specimen A017

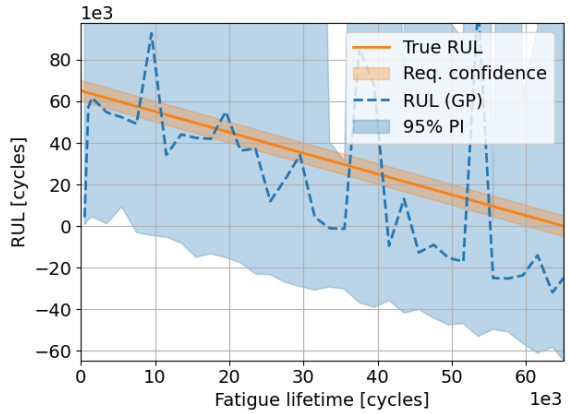


(a) Prediction from plain GP regression

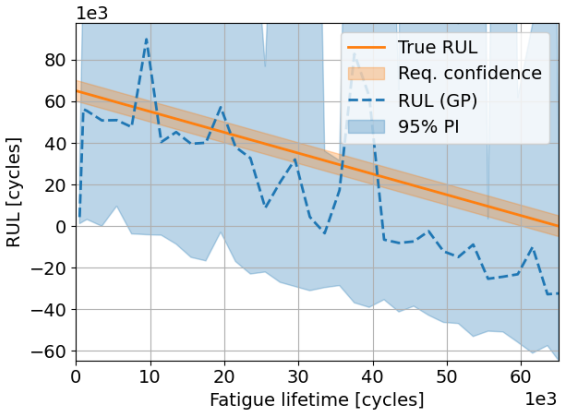


(b) Prediction from GP regression with correlation adjustment

Figure B.33: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A001

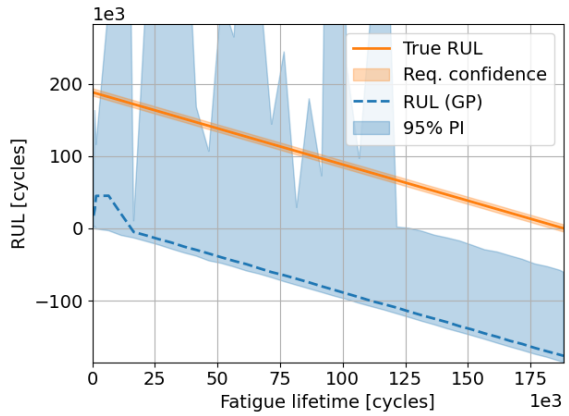


(a) Prediction from plain GP regression

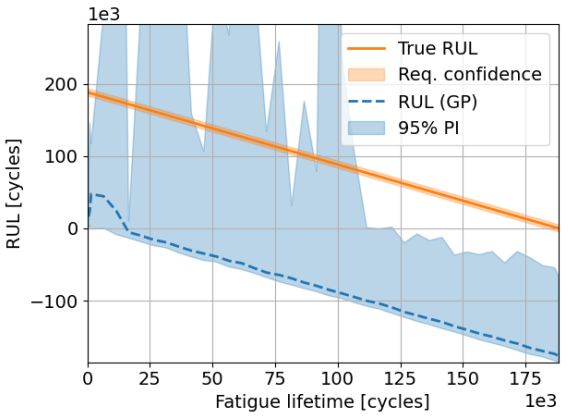


(b) Prediction from GP regression with correlation adjustment

Figure B.34: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A005

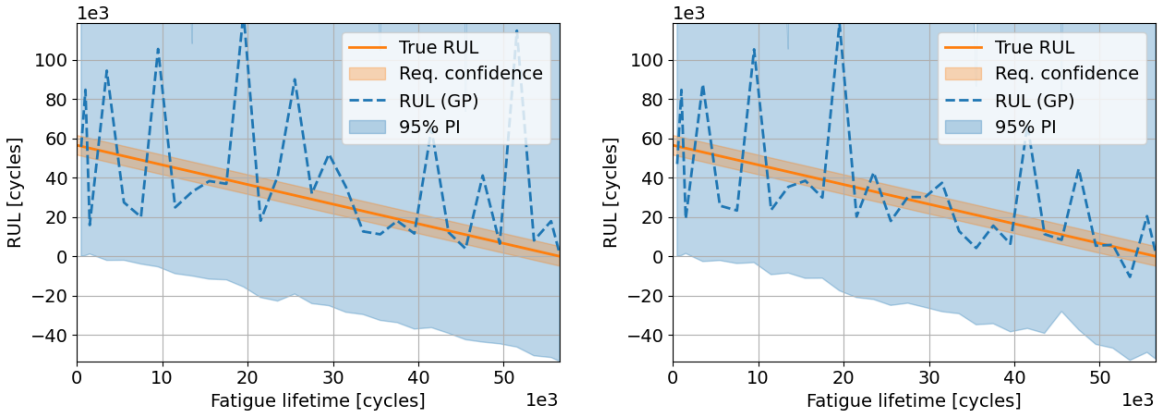


(a) Prediction from plain GP regression



(b) Prediction from GP regression with correlation adjustment

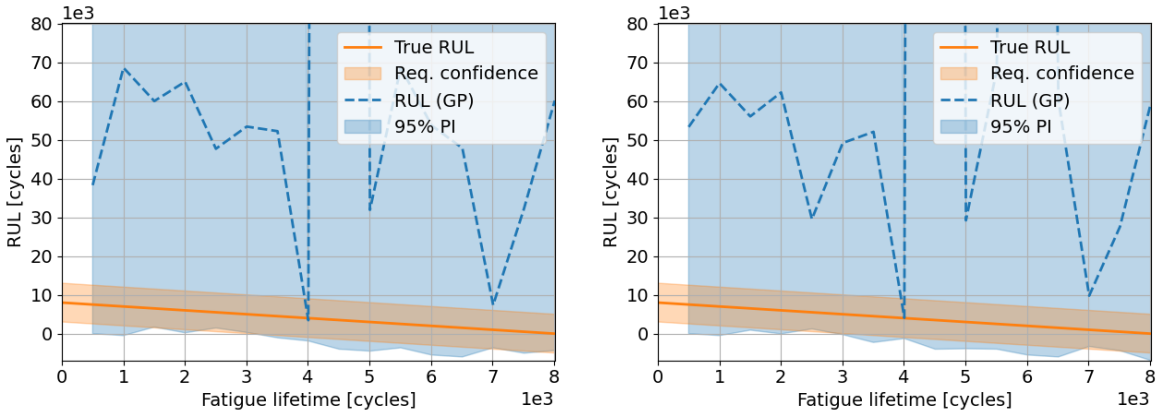
Figure B.35: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A006



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

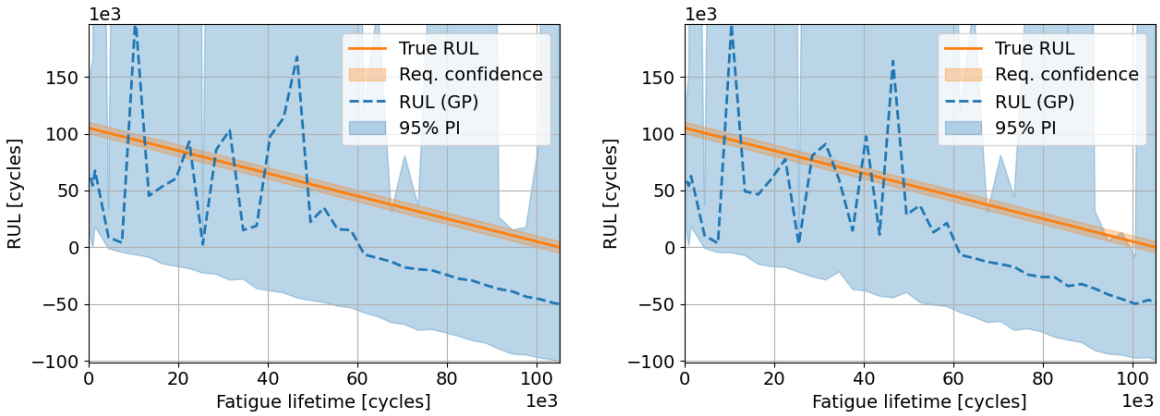
Figure B.36: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A007



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.37: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A010



(a) Prediction from plain GP regression

(b) Prediction from GP regression with correlation adjustment

Figure B.38: RUL predictions by the GP regression with Ma5+lin kernels, trained on CAF and VAF data, tested on specimen A017

B.2.3. Correlation adjustment

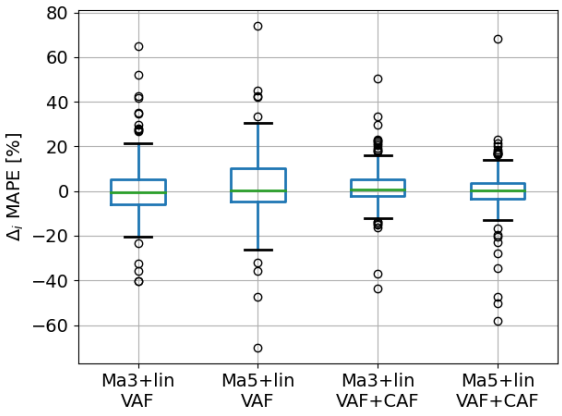


Figure B.39: Box plot of the change in MAPE (expected energy threshold versus actual failure energy) by applying the correlation adjustment to the threshold PDF. The difference is calculated by $MAPE_{original} - MAPE_{adjusted}$. The green line indicates the median, with the box encapsulating the 1st and 3rd quartiles Q_1 and Q_3. The whiskers extend up to $1.5(Q_3 - Q_1)$. Outliers are plotted as dots.

B.2.4. Performance metric tables

Table B.8: Performance metrics for the GP regression with Ma3+lin kernels, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	1.74e+09	294	-	9.34	-391	-	-
A005	4.81e+08	115	-	3.69	-74	-	1.1e+07
A006	2.96e+10	631	-	0.189	-984	-	5.14e+06
A007	3.33e+09	539	-	10.7	-807	4.42e+04	2.29e+07
A010	6.07e+09	3.4e+03	-	12.8	-4.47e+03	-	-
A017	2.03e+09	223	-	3.04	-275	5.76e+04	-

Table B.9: Performance metrics for the GP regression with Ma3+lin kernels with correlation adjustment, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	1.31e+09	243	-	9.23	-305	-	2.51e+06
A005	5.2e+08	145	-	3.79	-132	-	9.3e+06
A006	2.97e+10	631	-	0.182	-985	-	4.94e+06
A007	2.79e+09	492	-	10.9	-726	4.1e+04	2.08e+07
A010	5.74e+09	3.4e+03	-	13.5	-4.5e+03	4.15e+04	-
A017	2.2e+09	232	-	2.77	-288	5.66e+04	-

Table B.10: Performance metrics for the GP regression with Ma5+lin kernels, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	2.75e+09	177	-	8.03	-145	-	-
A005	1.33e+10	155	-	4.11	-85.9	1.76e+05	5.59e+06
A006	3e+10	630	-	0.111	-982	1.29e+05	-
A007	1.29e+09	120	-	17.3	-63.7	2.69e+04	4.29e+06
A010	8.56e+09	2.63e+03	-	2.09	-2.93e+03	-	-
A017	2.33e+09	229	-	2.38	-278	5.62e+04	-

Table B.11: Performance metrics for the GP regression with Ma5+lin kernels with correlation adjustment, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	4.84e+09	157	-	8.21	-95.6	-	-
A005	1.22e+10	170	-	3.96	-108	1.52e+05	5.34e+06
A006	3e+10	630	-	0.122	-982	1.29e+05	-
A007	1.21e+09	98	-	17.9	-27.8	2.54e+04	3.89e+06
A010	7.42e+09	2.53e+03	-	1.97	-2.85e+03	-	-
A017	2.39e+09	237	-	2.26	-292	5.68e+04	-

Table B.12: Performance metrics for the GP regression with Ma3+lin kernels, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	1.71e+10	343	-	4.12	-388	1.54e+05	-
A005	8.04e+08	158	-	4.7	-145	-	4.89e+05
A006	3e+10	631	-	0.107	-983	-	3.75e+05
A007	7.02e+09	173	-	12.2	-108	9.64e+04	1.05e+06
A010	4.26e+10	1.21e+03	-	19.4	-1.19e+03	-	-
A017	2.52e+09	245	-	2.21	-307	5.72e+04	-

Table B.13: Performance metrics for the GP regression with Ma3+lin kernels with correlation adjustment, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	1.58e+10	320	-	4.36	-352	-	-
A005	7.79e+08	153	-	4.8	-136	3.71e+04	4.95e+05
A006	2.99e+10	629	-	0.115	-979	-	2.12e+05
A007	7.17e+09	127	-	12.5	-31	1.06e+05	9.92e+05
A010	3.87e+10	1.15e+03	-	21.1	-1.09e+03	-	-
A017	2.54e+09	248	-	2.02	-312	5.76e+04	-

Table B.14: Performance metrics for the GP regression with Ma5+lin kernels, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	1.4e+09	156	-	7.46	-141	-	-
A005	9.7e+08	204	-	6.11	-237	4.39e+04	4.5e+05
A006	3.03e+10	632	-	0.0988	-984	-	-
A007	1.22e+09	203	-	9.62	-226	-	1.16e+06
A010	4.19e+11	6.39e+03	-	16.4	-6.55e+03	-	-
A017	2.9e+09	249	-	2.1	-310	-	-

Table B.15: Performance metrics for the GP regression with Ma5+lin kernels with correlation adjustment, trained on CAF and VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	2.81e+09	218	-	7.65	-235	-	-
A005	6.86e+08	180	-	6.29	-199	3.93e+04	-
A006	3.02e+10	630	-	0.0913	-980	-	-
A007	6.82e+08	145	-	10.4	-127	2.92e+04	1.12e+06
A010	4.52e+11	8.76e+03	-	16.2	-9.7e+03	-	-
A017	2.79e+09	246	-	2.05	-307	-	-

B.3. Recurrent neural network

B.3.1. Model architectures

Table B.16: Optimal RNN architectures and their corresponding estimated generalisation loss

Training set	Excluded specimen	Optimal model		
		Hidden nodes	Epochs	Est. gen. loss
CAF	A001	1	320	0.049
VAF	A001	1	160	0.06
	A005	16	640	0.039
	A006	1	90	0.05
	A007	16	250	0.052
	A010	4	110	0.036
	A017	2	130	0.068
CAF and VAF	A001	2	40	0.06
	A005	32	970	0.032
	A006	4	930	0.038
	A007	1	910	0.056
	A010	1	470	0.03
	A017	1	50	0.054

B.3.2. Sensitivity analysis

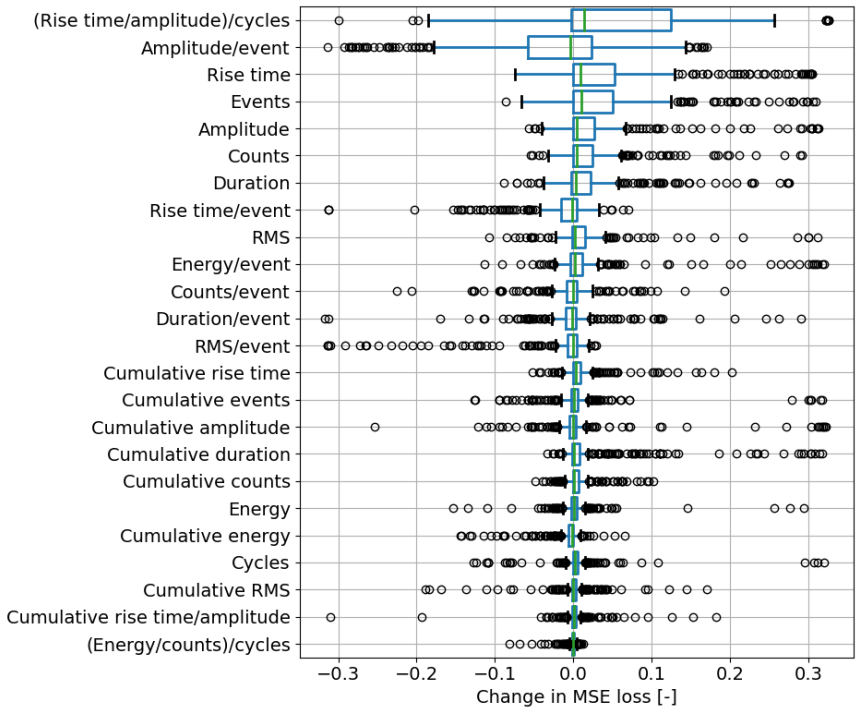


Figure B.40: Sensitivities of the MSE loss of the RNN, when trained on CAF data. The sensitivities are sorted by the width of the box encapsulating the 1st and 3rd quartiles. The green line indicates the median, with the box encapsulating the 1st and 3rd quartiles Q_1 and Q_3. The whiskers extend up to $1.5(Q_3 - Q_1)$. Outliers are plotted as dots.

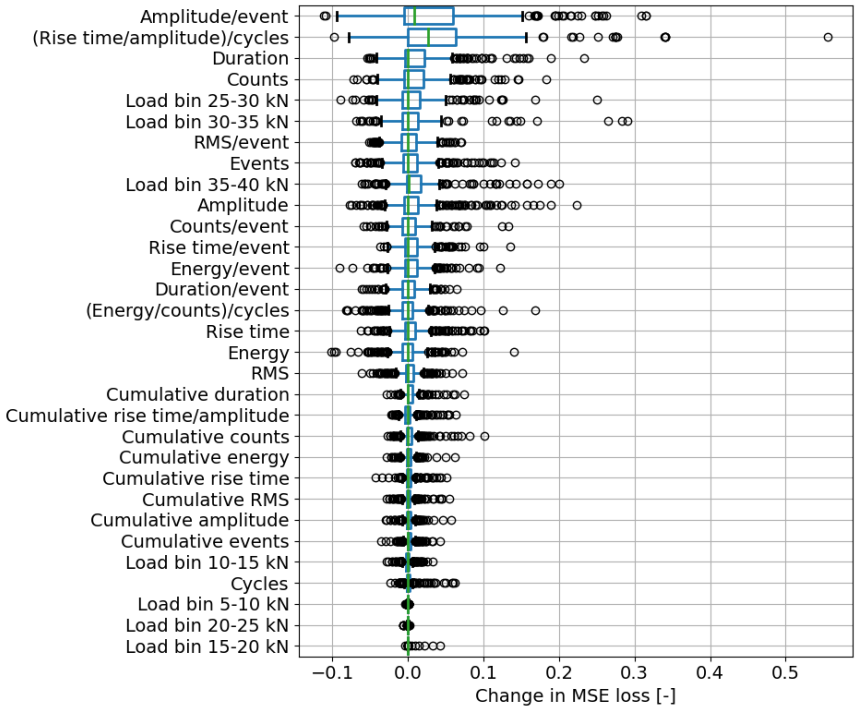


Figure B.41: Sensitivities of the MSE loss of the RNN, when trained on VAF data. The sensitivities are sorted by the width of the box encapsulating the 1st and 3rd quartiles.

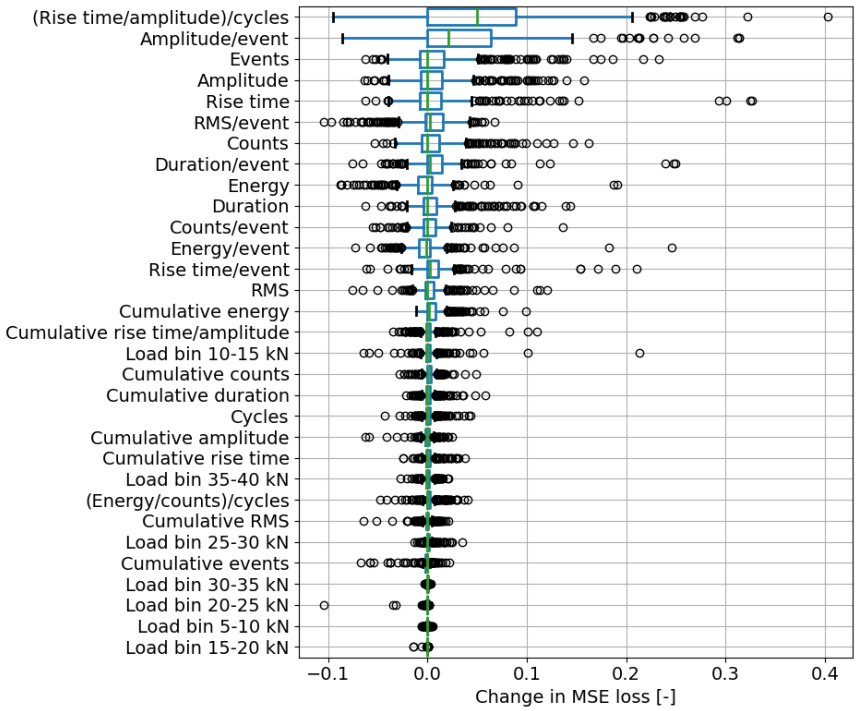
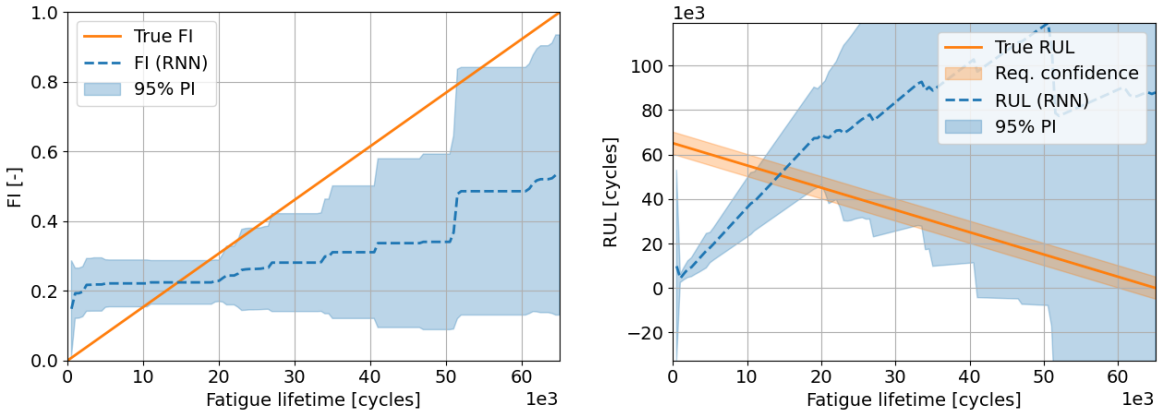


Figure B.42: Sensitivities of the MSE loss of the RNN, when trained on CAF and VAF data. The sensitivities are sorted by the width of the box encapsulating the 1st and 3rd quartiles.

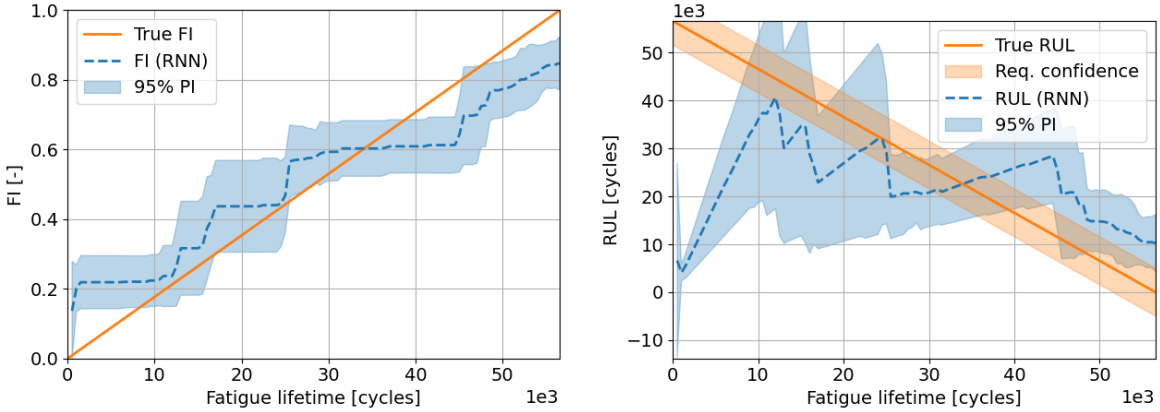
B.3.3. Failure index and remaining useful life predictions



(a) FI prediction

(b) RUL prediction

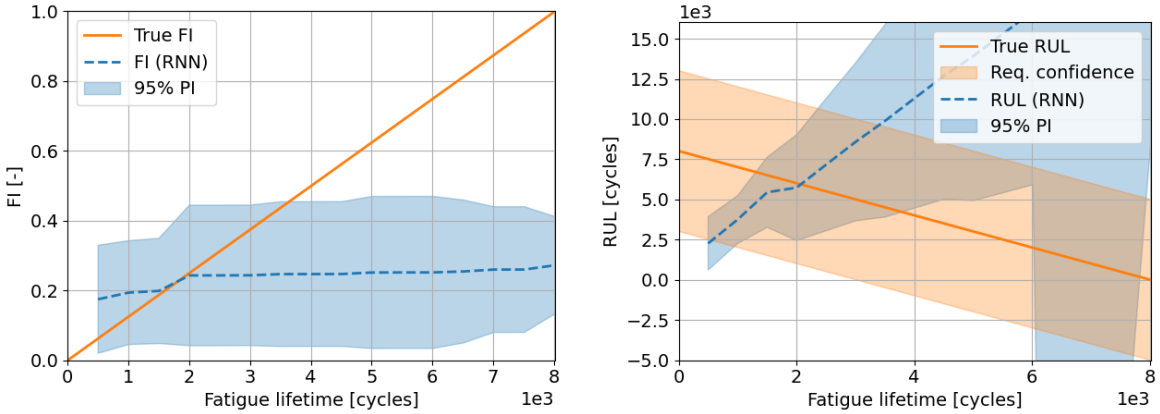
Figure B.43: RNN predictions for specimen A005, trained on CAF data



(a) FI prediction

(b) RUL prediction

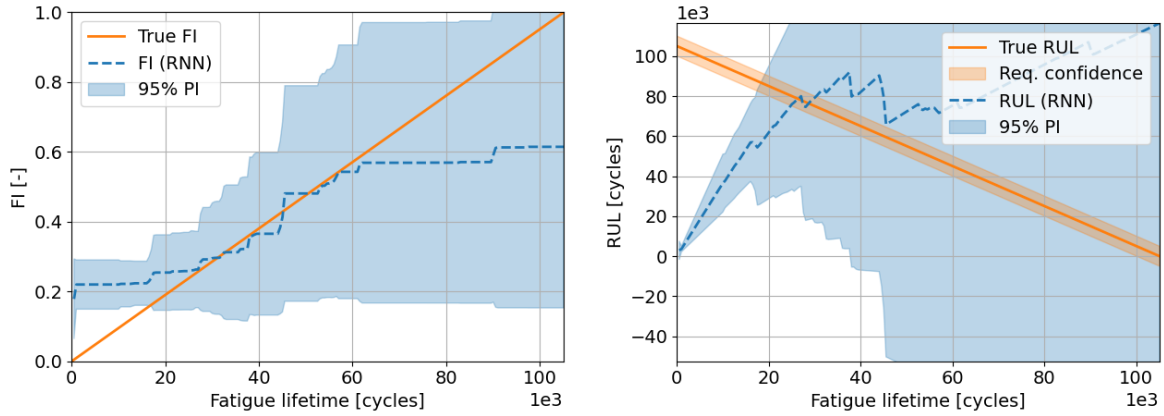
Figure B.44: RNN predictions for specimen A007, trained on CAF data



(a) FI prediction

(b) RUL prediction

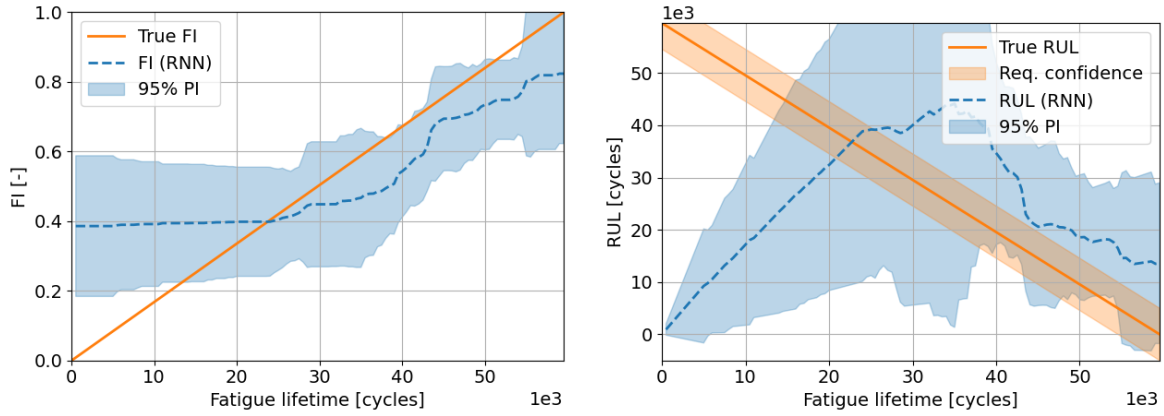
Figure B.45: RNN predictions for specimen A010, trained on CAF data



(a) FI prediction

(b) RUL prediction

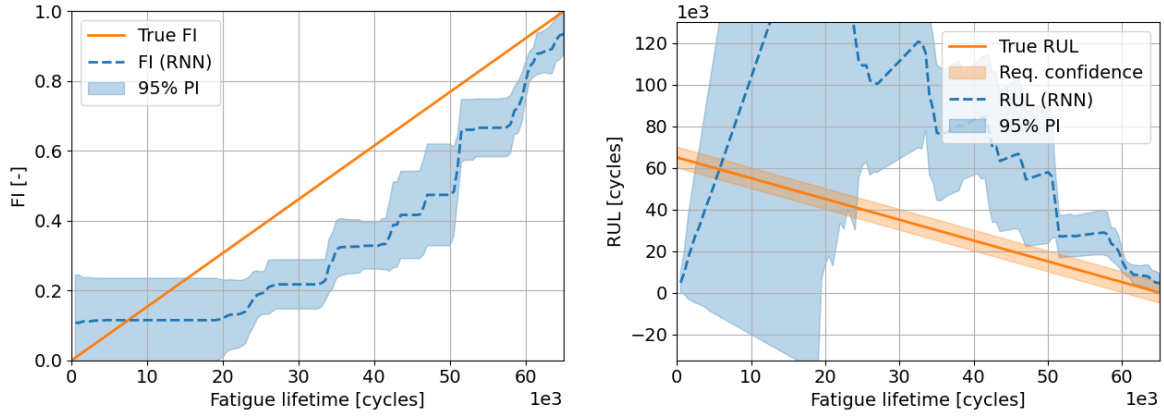
Figure B.46: RNN predictions for specimen A017, trained on CAF data



(a) FI prediction

(b) RUL prediction

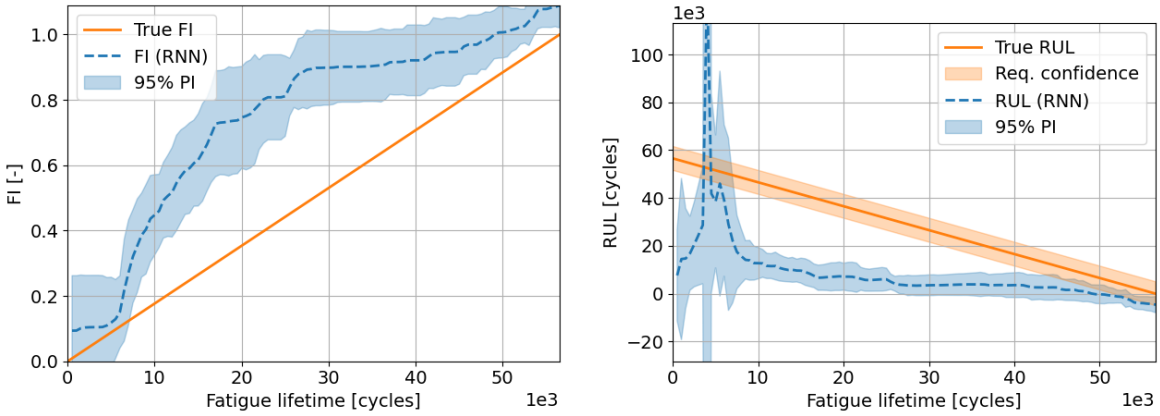
Figure B.47: RNN predictions for specimen A001, trained on VAF data



(a) FI prediction

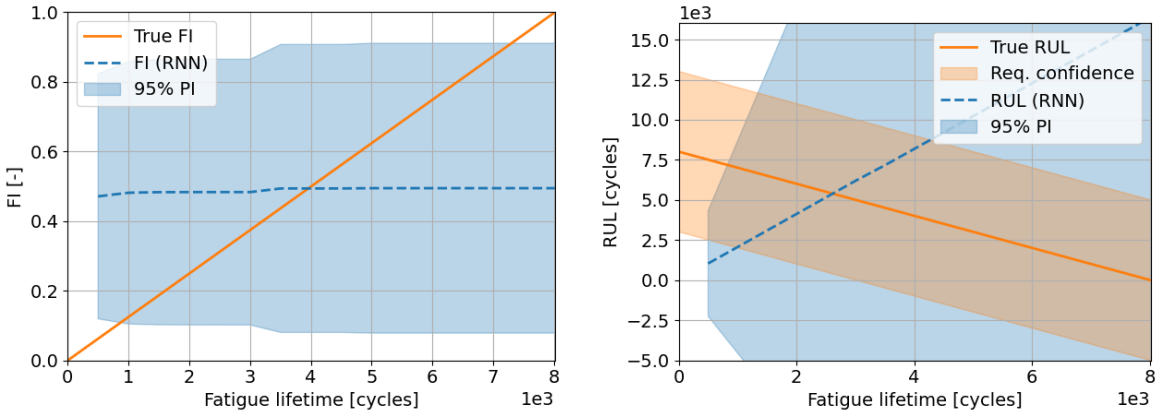
(b) RUL prediction

Figure B.48: RNN predictions for specimen A005, trained on VAF data



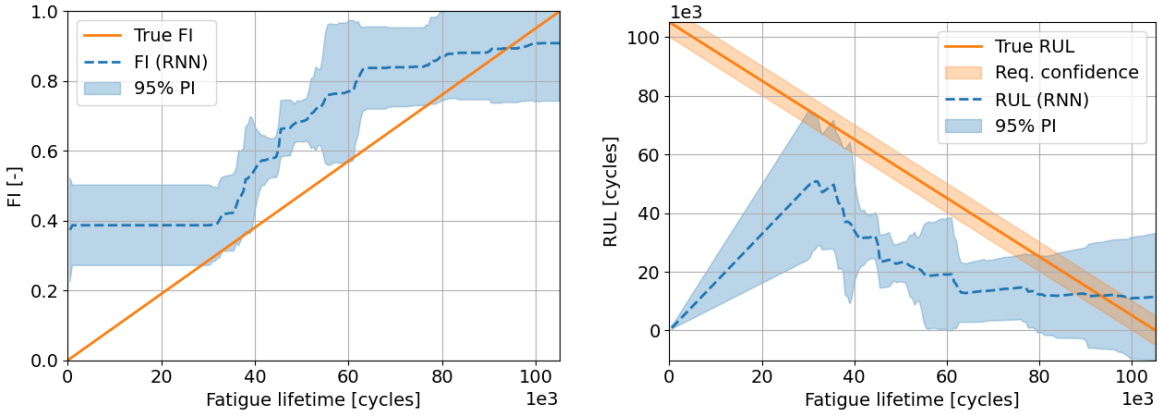
(a) FI prediction (b) RUL prediction

Figure B.49: RNN predictions for specimen A007, trained on VAF data



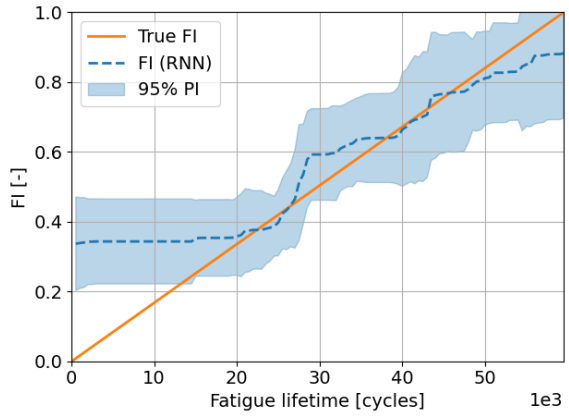
(a) FI prediction (b) RUL prediction

Figure B.50: RNN predictions for specimen A010, trained on VAF data

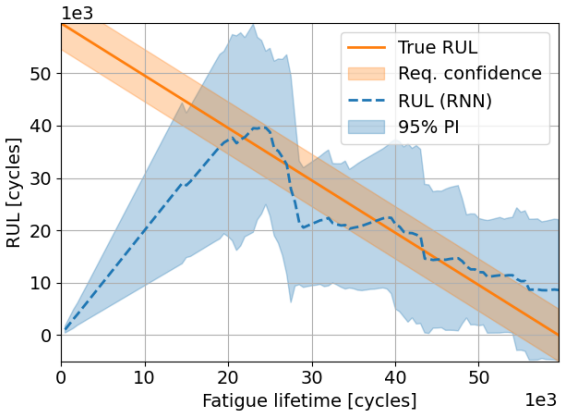


(a) FI prediction (b) RUL prediction

Figure B.51: RNN predictions for specimen A017, trained on VAF data

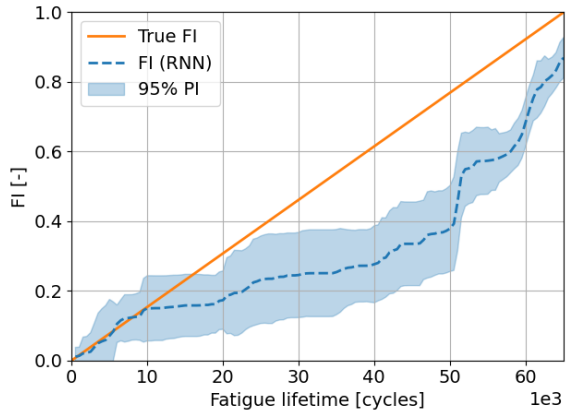


(a) FI prediction

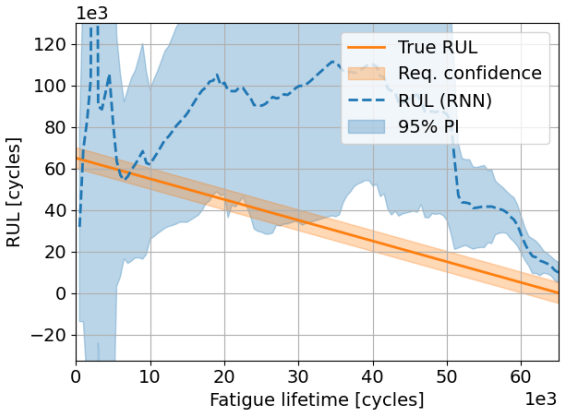


(b) RUL prediction

Figure B.52: RNN predictions for specimen A001, trained on CAF and VAF data

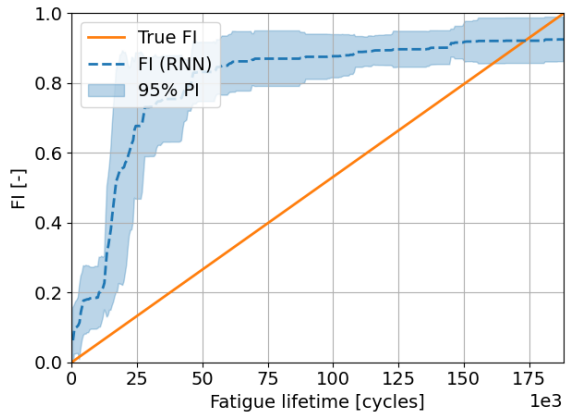


(a) FI prediction

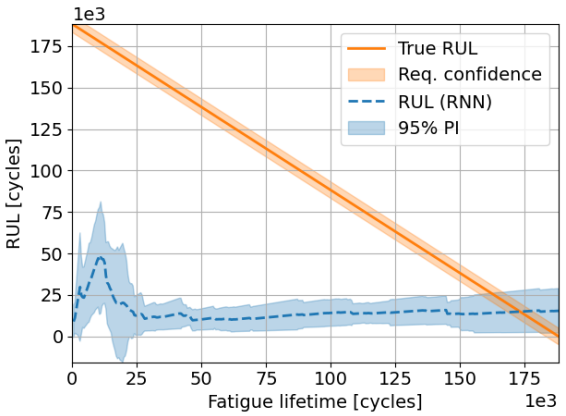


(b) RUL prediction

Figure B.53: RNN predictions for specimen A005, trained on CAF and VAF data

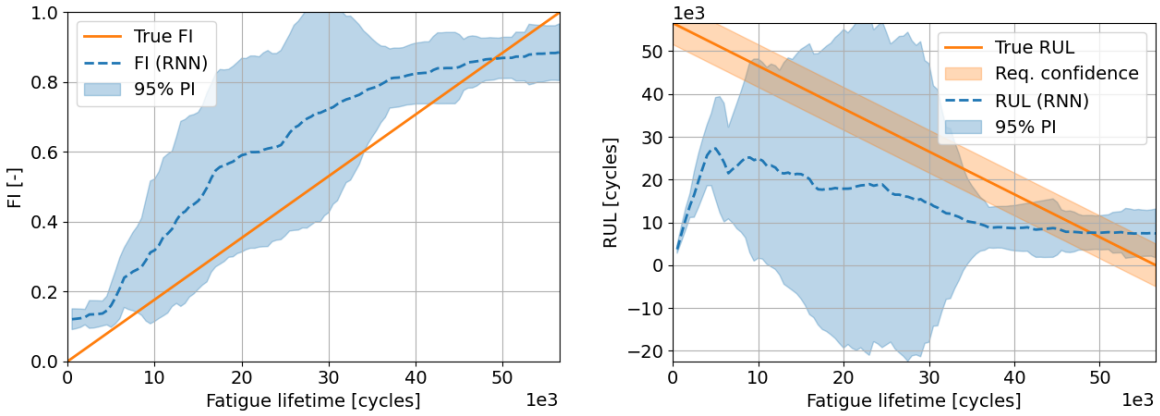


(a) FI prediction



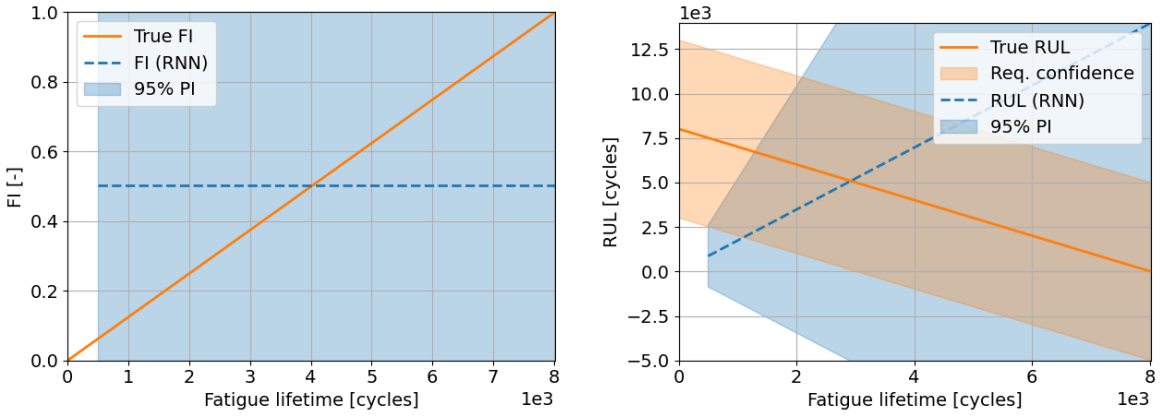
(b) RUL prediction

Figure B.54: RNN predictions for specimen A006, trained on CAF and VAF data



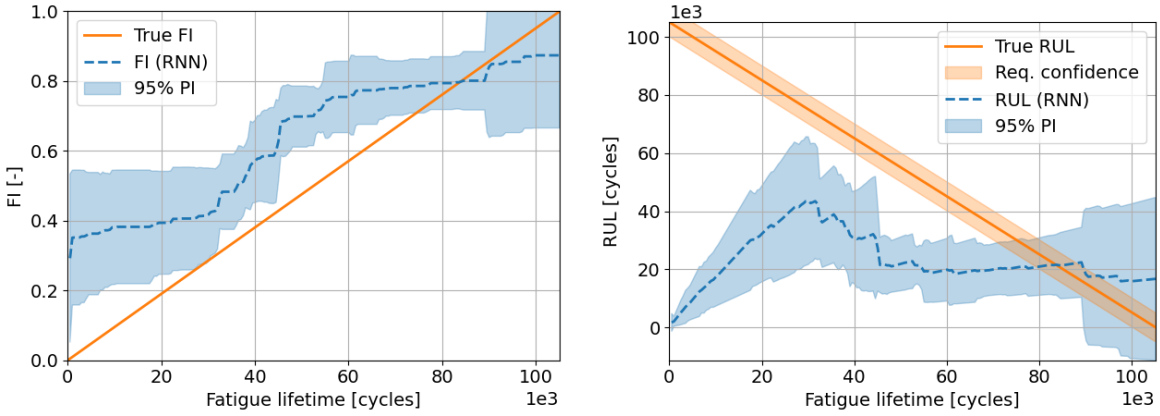
(a) FI prediction (b) RUL prediction

Figure B.55: RNN predictions for specimen A007, trained on CAF and VAF data



(a) FI prediction (b) RUL prediction

Figure B.56: RNN predictions for specimen A010, trained on CAF and VAF data



(a) FI prediction (b) RUL prediction

Figure B.57: RNN predictions for specimen A017, trained on CAF and VAF data

B.3.4. Performance metric tables

Table B.17: Means of four performance metrics for the RNN

Training data	MSE [cycles^2]	MAPE [%]	CBPM [%]	CRA [%]
CAF	4.06e+09	475	16.4	-697
VAF	2.72e+09	174	13	-134
CAF and VAF	2.57e+09	160	23.6	-121

Table B.18: Performance metrics for the RNN, trained on CAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	2.72e+08	66.1	-	40.8	3.02	2.04e+04	-
A005	3.77e+09	643	-	4.33	-1e+03	-	-
A006	1.63e+10	693	-	2.01	-1.17e+03	-	-
A007	2.75e+08	93.1	-	24.6	-41.8	2.17e+04	2.78e+04
A010	2.52e+08	849	-	22.3	-1.17e+03	-	-
A017	3.53e+09	505	-	4.27	-809	-	-

Table B.19: Performance metrics for the RNN, trained on VAF data

Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	5.13e+08	118	-	18.6	-74.6	2.43e+04	-
A005	4.33e+09	198	1001	6.93	-126	4.84e+04	1.25e+05
A006	9.09e+09	99.8	-	5.96	-14.2	8.21e+04	-
A007	5.97e+08	99.4	500	9.73	-22.7	2.5e+04	2.53e+05
A010	7.4e+07	453	-	22.3	-568	-	-
A017	1.69e+09	79.2	-	14.6	3	4.06e+04	-

Table B.20: Performance metrics for the RNN, trained on CAF and VAF data

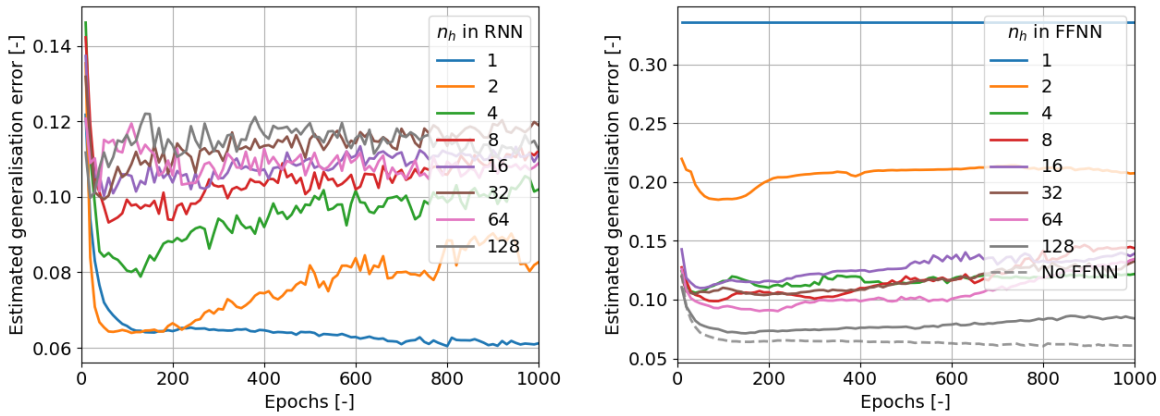
Specimen	MSE [cycles^2]	MAPE [%]	PH [cycles]	CBPM [%]	CRA [%]	Δ convergence [cycles]	PI convergence [cycles]
A001	3.95e+08	65.2	-	41.3	10.7	2.05e+04	-
A005	3.98e+09	252	-	1.53	-249	5.23e+04	4.71e+05
A006	9e+09	97.1	-	9.04	-9	8.22e+04	-
A007	3.2e+08	72.3	-	38	7.4	2.04e+04	-
A010	5.35e+07	381	-	34.4	-461	-	-
A017	1.7e+09	94	-	17.5	-26.3	3.99e+04	-

B.4. Case study

B.4.1. Architecture validation

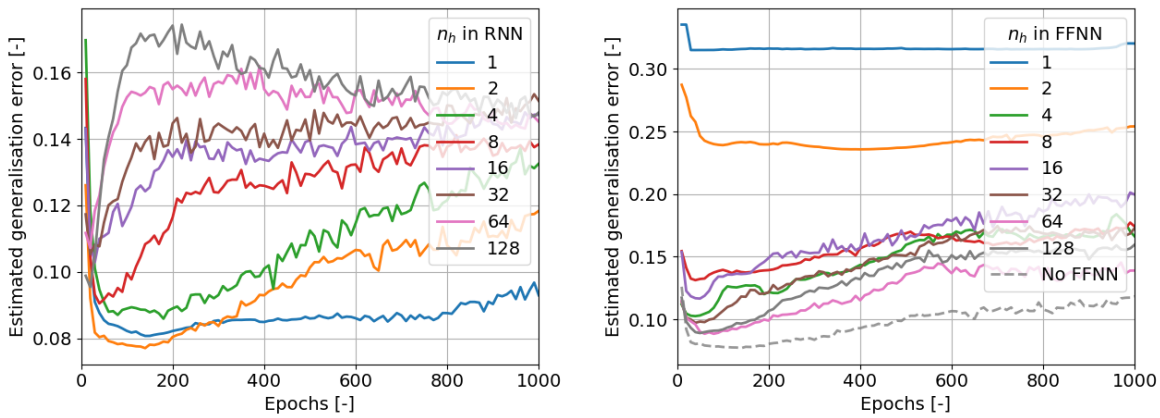
Table B.21: Optimal RNN architectures and their corresponding estimated generalisation loss for the GFRP data-set

Excluded specimen	Optimal model		
	Hidden nodes	Epochs	Est. gen. loss
6	1	110	0.078
7	1	800	0.060
8	2	140	0.077
9	1	60	0.079
10	2	510	0.060
11	1	80	0.075
12	1	90	0.077



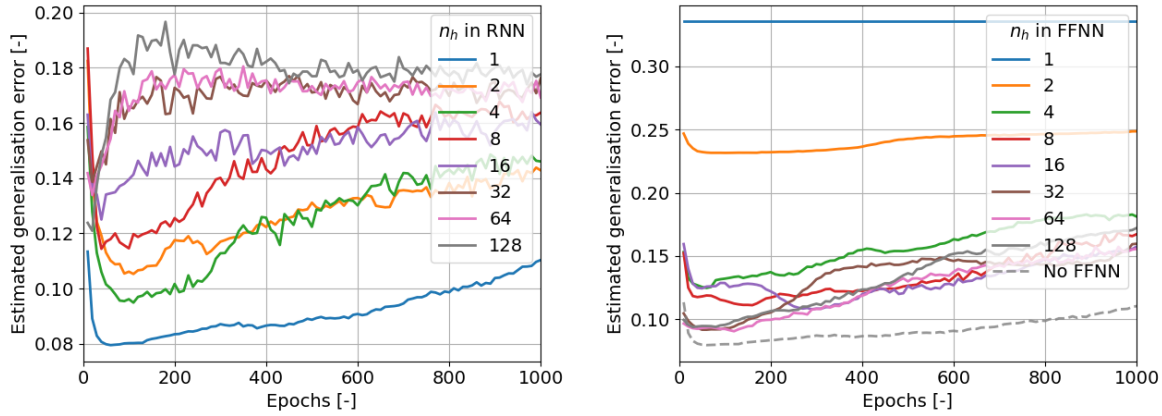
(a) RNN (b) RNN+FFNN

Figure B.58: Estimated generalisation errors for specimen 7 in the GFRP data-set



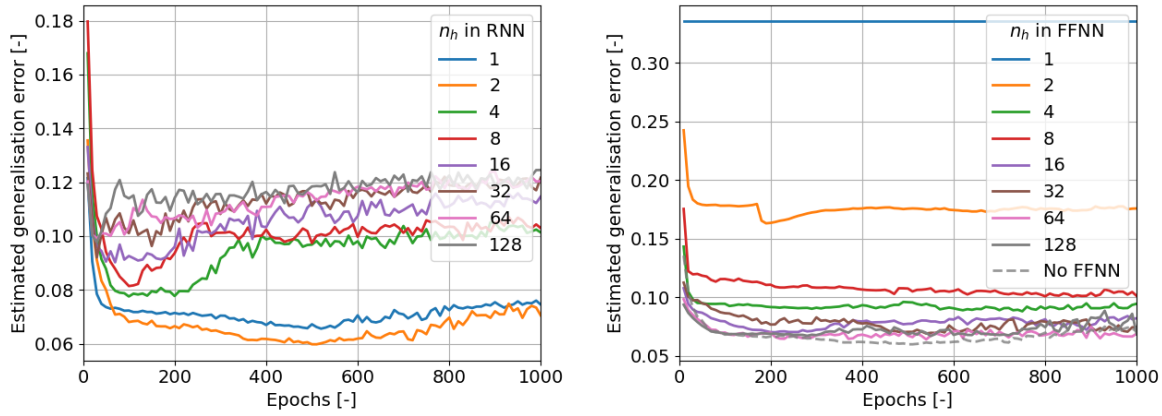
(a) RNN (b) RNN+FFNN

Figure B.59: Estimated generalisation errors for specimen 8 in the GFRP data-set



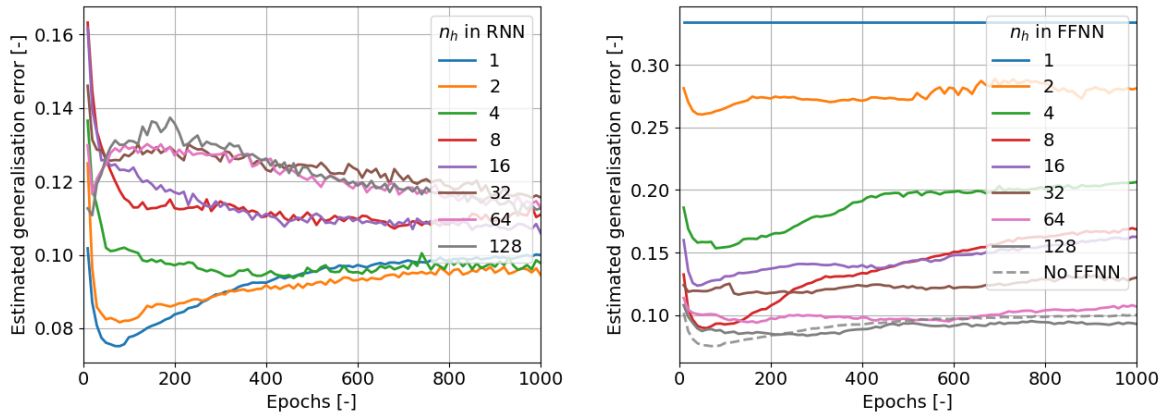
(a) RNN (b) RNN+FFNN

Figure B.60: Estimated generalisation errors for specimen 9 in the GFRP data-set



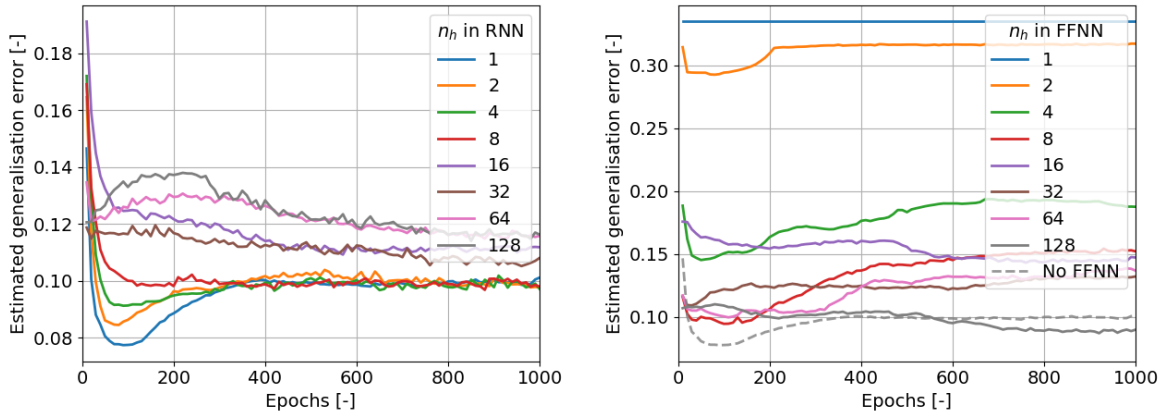
(a) RNN (b) RNN+FFNN

Figure B.61: Estimated generalisation errors for specimen 10 in the GFRP data-set



(a) RNN (b) RNN+FFNN

Figure B.62: Estimated generalisation errors for specimen 11 in the GFRP data-set

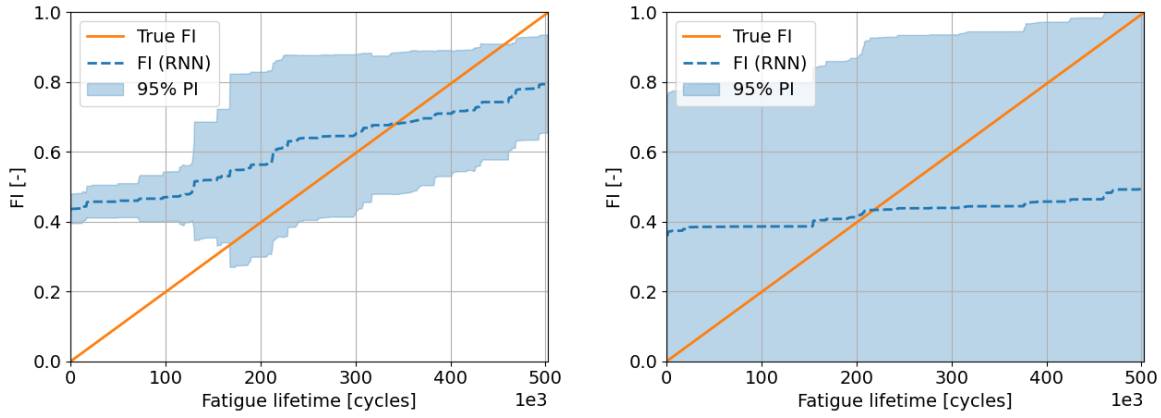


(a) RNN

(b) RNN+FFNN

Figure B.63: Estimated generalisation errors for specimen 12 in the GFRP data-set

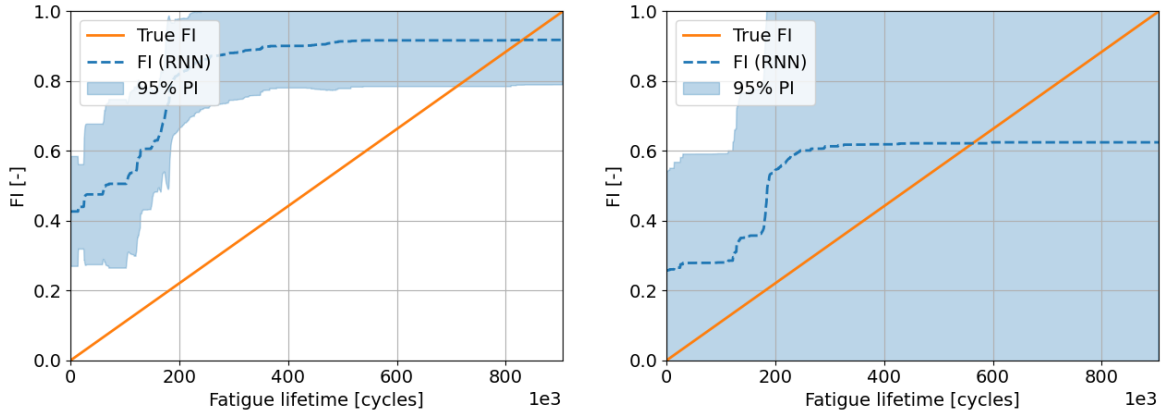
B.4.2. Failure index predictions



(a) FI prediction from the RNN

(b) FI prediction from the RNN+FFNN

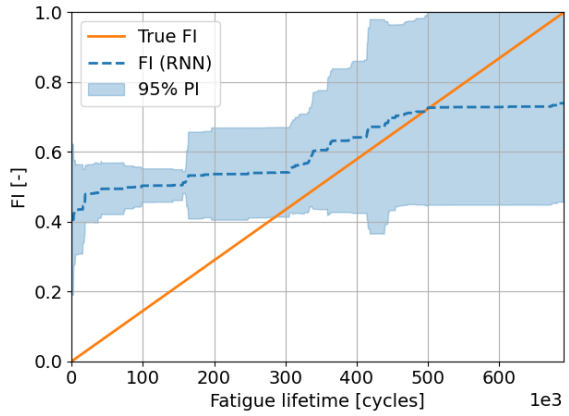
Figure B.64: FI predictions for specimen 6 in the GFRP data-set



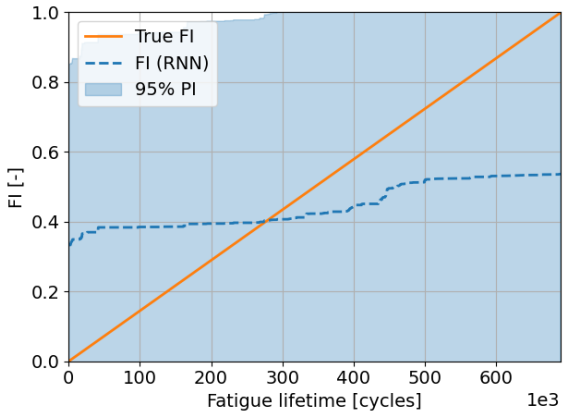
(a) FI prediction from the RNN

(b) FI prediction from the RNN+FFNN

Figure B.65: FI predictions for specimen 7 in the GFRP data-set

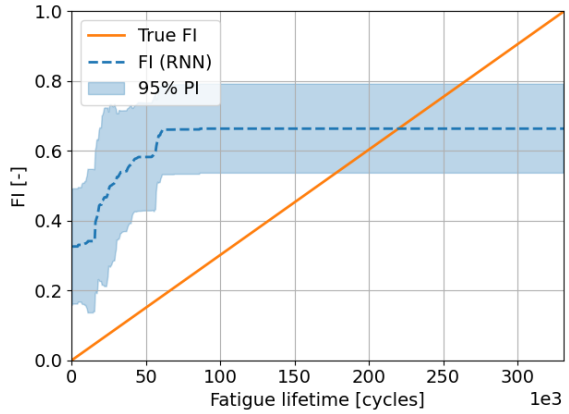


(a) FI prediction from the RNN

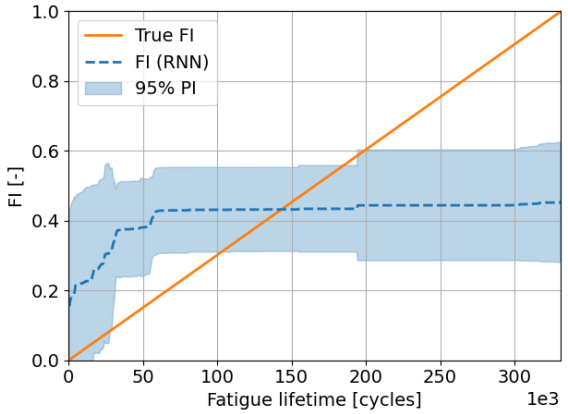


(b) FI prediction from the RNN+FFNN

Figure B.66: FI predictions for specimen 9 in the GFRP data-set

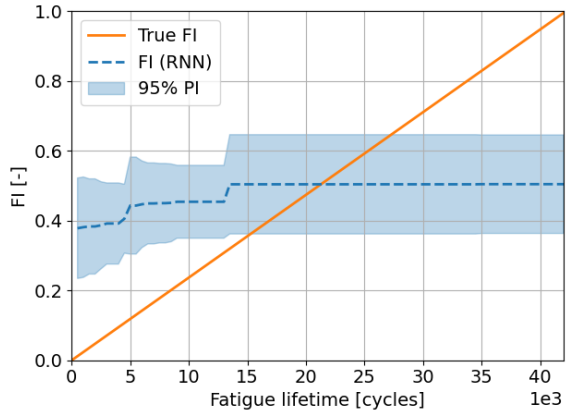


(a) FI prediction from the RNN

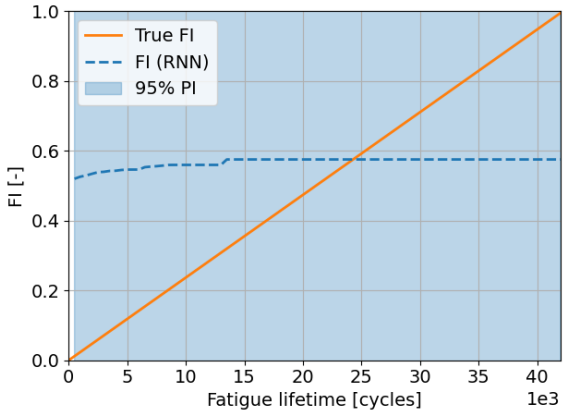


(b) FI prediction from the RNN+FFNN

Figure B.67: FI predictions for specimen 10 in the GFRP data-set



(a) FI prediction from the RNN



(b) FI prediction from the RNN+FFNN

Figure B.68: FI predictions for specimen 12 in the GFRP data-set