An end-to-end framework for Prognostics and Health Management
From raw data to maintenance scheduling

Komninos, P.

DOI
10.4233/uuid:6ed40814-ed05-4686-8414-b50fd31bbd0a

Publication date
2025

Document Version
Final published version

Citation (APA)
Komninos, P. (2025). *An end-to-end framework for Prognostics and Health Management: From raw data to maintenance scheduling*. [Dissertation (TU Delft), Delft University of Technology]. TU Deflt. https://doi.org/10.4233/uuid:6ed40814-ed05-4686-8414-b50fd31bbd0a

An end-to-end framework for Prognostics and Health Management

From raw data to maintenance scheduling

Panagiotis Komninos

An end-to-end framework for Prognostics and Health Management

From raw data to maintenance scheduling

An end-to-end framework for Prognostics and Health Management

From raw data to maintenance scheduling

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Thursday, 13 November 2025 at 12:30

by

Panagiotis KOMNINOS

Master of Science in Mechanical Engineering & Aeronautics, University of Patras, Greece
born in Patras, Greece

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Prof. dr. C. Witteveen,	Chairperson
Prof. dr. R.C. Alderliesten,	Delft University of Technology, *promotor*
Dr. ir. D. Zarouchas,	Delft University of Technology, *promotor*

Independent members:

Prof. dr. ir. R. Benedictus,	Delft University of Technology
Prof. dr. S.J. Watson,	Delft University of Technology
Dr. F. Cadini,	Politecnico di Milano
Prof. dr. T. Loutas,	University of Patras
Prof. dr. ir. R. De Breuker,	Delft University of Technology, reserve member

Keywords: Prognostics & Health Management, Structural Health Monitoring, Condition-based Maintenance, Artificial Neural Networks, Interpretable AI, Multi-modal, Deep Reinforcement Learning, Unsupervised Learning

Printed by: Ipskamp printing

Front & Back: Cover design by Panagiotis Komninos, with AI-generated elements

ISBN 978-94-6384-868-8

The best way to predict the future is to invent it.

Alan Kay

CONTENTS

Summary xi

Samenvatting xiii

1. Introduction 1
- 1.1. The Prognostics and Health Management strategy 2
- 1.2. Problem statement 4
- 1.3. Research Plan 6

2. From raw high-dimensional data straight to RUL estimations 13
- 2.1. Introduction 14
- 2.2. Theory 17
  - 2.2.1. Transformers & Multi-head attention 17
  - 2.2.2. Vision transformers 20
- 2.3. Problem formulation and experimental setup 21
  - 2.3.1. Problem formulation 21
  - 2.3.2. Experimental setup 21
- 2.4. Model architecture and learning process 24
  - 2.4.1. Data augmentation 24
  - 2.4.2. Patching 24
  - 2.4.3. Windowing 26
  - 2.4.4. Training setup 26
  - 2.4.5. Interpretability: information flow 33
- 2.5. Experimental results and discussion 35
  - 2.5.1. Model performance, optimal and suboptimal RUL predictions 35
  - 2.5.2. Explaining the results - model interpretation 37
  - 2.5.3. The role of temporal attention 41
- 2.6. Conclusions and recommendations 43

3. Deep soft monotonic feature extraction and prognostics 53
- 3.1. Introduction 54
- 3.2. Related work in monotonic feature extraction and clustering 57
  - 3.2.1. Feature extraction for prognostic-related tasks in healthcare and engineering 57
  - 3.2.2. Multi-modal deep clustering 59
  - 3.2.3. Monotonic neural networks 59
- 3.3. Methodology 60
  - 3.3.1. Datasets and data shaping 60

3.3.2. Model architecture . . . 66
3.3.3. Training the DSMC model . . . 71
3.3.4. Bayesian optimization for hyperparameter tuning . . . 72
3.3.5. Prognostics . . . 74
3.4. Results . . . 76
3.4.1. Clustering results and survivability analysis of the MIMIC-III and C-MAPSS datasets . . . 76
3.4.2. Benchmarking . . . 79
3.4.3. Interpretabilty of DSMC model . . . 83
3.4.4. Towards multi-modal feature extraction and clustering: evaluating the F-MOC dataset . . . 89
3.4.5. Qualitative analysis of number of clusters hyperparameter . . . 89
3.5. Conclusion . . . 91

4. A Bayesian inference-based framework for modeling imperfect post-repair behavior of RUL **109**
4.1. Introduction . . . 110
4.2. Theory . . . 115
4.2.1. Bayesian inference . . . 115
4.2.2. Prognostics . . . 120
4.3. Methodolody . . . 120
4.3.1. Data shaping . . . 122
4.3.2. Feature extraction and prognostic models . . . 122
4.3.3. The repair model . . . 123
4.4. Case study . . . 128
4.4.1. Values of hyperparameters highly dependent on the dataset . . . 132
4.5. Outcomes of imperfect repair model . . . 133
4.5.1. Clustering assignments & prognostics . . . 133
4.5.2. Evaluation of repair model . . . 136
4.5.3. Extension to multiple repairs . . . 138
4.6. Conclusion . . . 140

5. A framework for post-prognosis decision-making under imperfect maintenance decisions **149**
5.1. Introduction . . . 150
5.2. Theoretical background for constructing the PPDM framework . . . 153
5.2.1. Markov Decision Process & deep Reinforcement Learning . . . 153
5.2.2. Action masking . . . 155
5.2.3. Uncertainty quantification in discrete domains . . . 156
5.2.4. Value of Information . . . 158
5.3. Methodolody . . . 159
5.3.1. Problem formulation . . . 159
5.3.2. PPDM task and MDP formulation . . . 162
5.3.3. The DRL model . . . 165
5.3.4. Constraints & action masking . . . 168
5.3.5. Mapping uncertainty with decision probabilities . . . 168

5.3.6. Deciding optimal time to acquire new information 169
5.4. Case study 172
5.5. PPDM outcomes 174
5.5.1. Performance of the RL agent 174
5.5.2. Scheduling maintenance actions 175
5.5.3. The role of VoI in choosing the optimal time to acquire new information 178
5.5.4. Sensitivity analysis based on the number of specimens 180
5.6. Discussion and Conclusion 181

6. Conclusions and Recommendations 191
6.1. Conclusions 191
6.2. Recommendations 192

A. Additional information ISTRUST 195

B. Additional information regarding DSMC and HSMM 197
B.1. Extended data 197
B.2. Implementation details of the F-MOC dataset 201
B.2.1. Experimental setup 201
B.2.2. Synchronization of acoustic emission and DIC data 202
B.3. HSMM definition and re-estimation process 203

C. Additional results of imperfect repair model 207

D. Additional information PPDM 209
D.1. Proximal Policy Optimization 209
D.2. Mutual Information theory 211
D.3. Proofs 211
D.4. Supplementary results 213

Acknowledgements 215

Curriculum Vitæ 217

List of Publications 219

List of Publications 219

SUMMARY

This dissertation introduces a comprehensive end-to-end framework related to Prognostics and Health Management (PHM) strategy, with the goal of utilizing raw sensor (placed on a structure's components) data to make maintenance decisions for extending the structure's lifecycle. The study addresses key challenges in PHM, such as dealing with high-dimensional, multi-modal data, extracting features, predicting Remaining Useful Life (RUL), managing uncertainty, modeling repair scenarios, and optimizing maintenance decisions. All these challenges are organized in phases. These phases are namely, i) data collection, ii) feature extraction, iii) Health Indicator (HI) construction, iv) prognostics, v) modeling of maintenance actions, and vi) Post-Prognosis Decision-Making (PPDM). Each phase of the PHM strategy is developed independently but integrated into a cohesive framework, ensuring modularity, transparency, and adaptability across diverse applications.

A major contribution to the framework is proving that neural networks, a preferable approach for handling complex data, can become interpretable, thus unveiling the black-box nature of such models. In this regard, the ISTRUST model, an interpretable Transformer-based architecture for predicting RUL directly from raw sequential image data of a structure under fatigue loads, is proposed. By leveraging attention mechanisms, the model captures critical spatiotemporal features of structural damage, offering insights into prediction accuracy and variability. The model's interpretability highlights the understanding of its predictions. Simultaneously, via this interpretation, it is shown that predicting RUL directly from high-dimensional raw data is challenging or even impossible, necessitating focusing on each phase of the PHM strategy separately instead of unifying the majority of the PHM strategy in one model.

Central to the framework is the development of the Deep Soft Monotonic Clustering (DSMC) model, designed to extract meaningful features and then construct HIs from multi-modal data. This model extracts monotonic features that are related to each component's degradation. Subsequently, it considers these features to perform monotonic clustering representing HIs and enables the integration of those HIs into prognostic models to predict RUL across diverse domains.

The dissertation further explores the impact of imperfect repairs on components, where repairs often leave the component in a state between fully restored and partially damaged. The health state of the component is measured via a stochastic recovery of the predicted RUL and a Bayesian inference-based model is employed to quantify this stochastic behavior. The imperfect repair (Bayesian) model can be also extended for multiple sequential repairs. This phase of the PHM strategy emphasizes the importance of understanding imperfect repair effectiveness to enhance maintenance strategies.

The final phase of the framework concerns PPDM. Given a set of maintenance actions, including replacements and imperfect repairs, and a set of operational conditions and constraints, PPDM is modeled as a Markov Decision Process and optimized via deep Reinforcement Learning. PPDM's ultimate target is to optimize the scheduling of maintenance actions proactively within a predefined horizon length.

Experimental validation is conducted on each of the proposed models. The ISTRUST model was validated on fatigue-loaded composite specimens, utilizing sequential raw image data taken by a camera. The DSMC model was tested using diverse datasets, including engineering and healthcare. The imperfect repair modeling was applied to tension-tension fatigue experiments on open-hole aluminum coupons, capturing stochastic recovery behavior after repairs. The same experiment was utilized for evaluating the PPDM framework.

Overall, a holistic end-to-end PHM framework lays the groundwork for advancing Condition-based Maintenance (CBM) strategies. By integrating advanced models for each phase of the PHM strategy, the research highlights practical opportunities for embedding PHM into CBM and encourages further innovation and refinement by other researchers in the field. Although the proposed PHM framework has been an initial attempt towards this direction, its capabilities can be further enhanced through improvements in data scalability, exploration of varied PPDM formulations, sensitivity analyses across PHM phases, and practical integration with CBM for broader system-level maintenance optimization.

SAMENVATTING

Dit proefschrift introduceert een uitgebreid end-to-end raamwerk met betrekking tot de Prognose- en Gezondheidsbeheerstrategie (Prognostics and Health Management, PHM), met als doel het gebruik van ruwe sensorgegevens (geplaatst op componenten van de constructie) om onderhoudsbeslissingen te nemen die de levensduur van de constructie verlengen. De studie behandelt belangrijke uitdagingen binnen PHM, zoals het omgaan met hoog-dimensionale, multi-modale data, het extraheren van kenmerken, het voorspellen van de resterende levensduur (Remaining Useful Life, RUL), het beheren van onzekerheid, het modelleren van herstelscenario's en het optimaliseren van onderhoudsbeslissingen. Al deze uitdagingen zijn georganiseerd in fasen, namelijk i) gegevensverzameling, ii) kenmerkextractie, iii) opbouw van Gezondheidsindicatoren (Health Indicators, HI), iv) prognose, v) modellering van onderhoudsacties, en vi) Besluitvorming na Prognose (Post-Prognosis Decision-Making, PPDM). Elke fase van de PHM-strategie wordt onafhankelijk ontwikkeld, maar geïntegreerd in een samenhangend raamwerk, dat modulariteit, transparantie en aanpasbaarheid waarborgt over uiteenlopende toepassingen heen.

Een belangrijke bijdrage aan het raamwerk is het aantonen dat neurale netwerken – een voorkeursmethode voor het verwerken van complexe data – interpreteerbaar kunnen worden gemaakt, waarmee de black-box-aard van dergelijke modellen wordt doorbroken. In dit opzicht wordt het ISTRUST-model voorgesteld: een interpreteerbare, op Transformers gebaseerde architectuur voor het rechtstreeks voorspellen van de resterende levensduur (RUL) op basis van ruwe sequentiële beeldgegevens. Door gebruik te maken van aandachtmechanismen legt het model kritieke spatio-temporele kenmerken van structurele schade vast, wat inzicht biedt in zowel de nauwkeurigheid als de variabiliteit van de voorspellingen. De interpreteerbaarheid van het model benadrukt het begrip van zijn voorspellingen. Tegelijkertijd wordt door deze interpretatie aangetoond dat het rechtstreeks voorspellen van RUL uit hoog-dimensionale ruwe data uitdagend of zelfs onmogelijk is, wat de noodzaak onderstreept om elke fase afzonderlijk aan te pakken in plaats van het merendeel van de PHM-strategie in één enkel model te verenigen.

Centraal in het raamwerk staat de ontwikkeling van het Deep Soft Monotonic Clustering (DSMC) model, dat is ontworpen voor het extraheren van betekenisvolle kenmerken en het vervolgens construeren van Gezondheidsindicatoren (HIs) uit multi-modale data. Dit model extraheert monotone kenmerken die verband houden met de degradatie van elk afzonderlijk component. Vervolgens gebruikt het deze kenmerken om monotone clustering uit te voeren, wat resulteert in representaties van HIs. Hierdoor kunnen deze HIs worden geïntegreerd in prognosemodellen om de RUL te voorspellen in uiteenlopende toepassingsdomeinen.

Het proefschrift onderzoekt verder de impact van imperfecte reparaties aan componenten, waarbij reparaties het component vaak achterlaten in een toestand tussen volledig hersteld en gedeeltelijk beschadigd. De gezondheidstoestand van het component wordt gemeten via een stochastisch herstel van de RUL, en een op Bayesiaanse inferentie gebaseerd model

wordt gebruikt om dit stochastische gedrag te kwantificeren. Het imperfecte reparatiemodel (Bayesiaans) kan tevens worden uitgebreid naar meerdere opeenvolgende reparaties. Deze fase van de PHM-strategie benadrukt het belang van inzicht in de effectiviteit van imperfecte reparaties om onderhoudsstrategieën te verbeteren.

De laatste fase van het raamwerk betreft de Post-Prognose Besluitvorming (Post-Prognosis Decision-Making, PPDM). Gezien een reeks onderhoudsacties, waaronder vervangingen en imperfecte reparaties, en een reeks operationele omstandigheden en beperkingen, wordt PPDM gemodelleerd als een Markov Decision Process en geoptimaliseerd via deep Reinforcement Learning. Het uiteindelijke doel van PPDM is het proactief optimaliseren van de planning van onderhoudsacties binnen een vooraf bepaalde tijdshorizon.

Experimentele validatie is uitgevoerd op elk van de voorgestelde modellen. Het ISTRUST-model werd gevalideerd op composiet proefstukker onder vermoeiingsbelasting, waarbij gebruik werd gemaakt van sequentiële ruwe beeldgegevens vastgelegd met een camera. Het DSMC-model werd getest met diverse datasets, waaronder die uit de techniek en de gezondheidszorg. Het modelleren van imperfecte reparaties werd toegepast op trek-trek vermoeiingsexperimenten met aluminium teststukken met een open gat, waarbij het stochastische herstelgedrag na reparaties werd vastgelegd. Datzelfde experiment werd tevens gebruikt voor de evaluatie van het PPDM-raamwerk.

Al met al vormt het holistische end-to-end PHM-raamwerk de basis voor het verder ontwikkelen van Condition-based Maintenance (CBM) strategieën. Door geavanceerde modellen te integreren voor elke fase van de PHM strategie, benadrukt het onderzoek praktische mogelijkheden om PHM in CBM te integreren en moedigt het verdere innovatie en verfijning aan door andere onderzoekers in het veld. Hoewel het voorgestelde PHM-raamwerk een eerste poging in deze richting is geweest, kunnen de mogelijkheden ervan verder worden verbeterd door verbeteringen in dataschaalbaarheid, verkenning van diverse PPDM-formuleringen, gevoeligheidsanalyses over de PHM-fasen heen, en praktische integratie met CBM voor bredere systeemgerichte onderhoudsoptimalisatie.

NOMENCLATURE

α	Tunable hyperparameter that weights the importance of the monotonic behavior
α_1	Lower bound of Uniform distribution for μ_{mean}
α_2	Lower bound of Uniform distribution for σ^2_{mean}
α_t	Action at time t
α_{mean}	Lower bound of Truncated Normal distribution
β	Hyperparameter that weights the contribution of the $Loss^{AE}$
β_1	Upper bound of Uniform distribution for μ_{mean}
β_2	Upper bound of Uniform distribution for σ^2_{mean}
β_{mean}	Upper bound of Truncated Normal distribution
γ	Discount factor
μ	Mean
μ_j^d	Cluster centroids per sample d drawn from dataset D
μ_j	A cluster center ranging from j=1 to j=K number of clusters
μ_{mean}	Mean of random variable of mean normalized RUL after repair
μ_{new}	Random variable of mean normalized RUL after repair
μ_{old}	Random variable of mean normalized RUL before repair
Ω	Finite set of observations
$\pi(.)$	Policy
σ^2	Variance
σ^2_{mean}	Mean of random variable of variance of RUL
σ^2_{new}	Random variable of variance of RUL after repair
σ^2_{old}	Random variable of variance of RUL before repair
σ^2_{var}	Variance of random variable of variance of RUL

τ	Temperature hyperparameter
$\mathbf{x}$	Sample of a square patch
Θ	Weights of NN
θ	Parameters of the function approximator
$A(s_t,\alpha_t)$	Advantage function
α	Degrees of freedom of the Student's t-distribution
A_{self}	Attention weight matrix
A_{token}	Attention weight token
$ABAO$	As-Bad-As-Old
AE	Autoencoder
$AGAN$	As-Good-As-New
$AUROC$	Area Under Receiver Operating Characteristic
b_v	Bias of one neuron
$batch^{AE}$	Batch size for training the AE model
$batch^{DSMC}$	Batch size for training the DSMC model
BO	Bayesian Optimization
$C\text{-}MAPSS$	NASA's Commercial Modular Aero-Propulsion System Simulation dataset
C_{action}	Cost of taking a specific action by the agent
CBM	Condition-based Maintenance
$CFRP$	Carbon Fiber Reinforced Polymer
CI	Confidence Intervals
$CNN3D$	3-dimensional CNN
CNN	Convolutional Neural Network
D	Dataset
d_{c_i}	Difference between the sequentially predicted labels c_i, and c_{i-1} at timesteps i and $i-1$
d_{model}	Size of one dimensional vector related to ISTRUST architecture
DIC	Digital Image Correlation

DL	Deep Learning
DQN	Deep Q-Networks
DRL	Deep Reinforcement Learning
$drop_{rate}$	Dropout rate
$DSMC$	Deep Soft Monotonic Clustering
EOL	End-of-Life
$epochs^{AE}$	Number of epochs for training the autoencoder
$epochs^{DSMC}$	Number of epochs for training the DSMC model
$F\text{-}MOC$	Fatigue Monitoring of Composites dataset
f_θ	Function approximation with parameters θ
FC	Fully Connected
$g(r_u)$	Activation function of the neuron of the previous layer u
GAE	Generalized Advantage Estimation
$GBDT$	Gradient Boosting Decision Tree
$H(.)$	Entropy
H	Height
H_P	Height of each patch
$H^{enc}_{in,acoustic}$	Hidden features extracted by the encoder related to acoustic emission signals
$H^{enc}_{in,image}$	Hidden features extracted by the encoder related to sequential images
H^{enc}_{in}	Hidden features extracted by the encoder
HI	Health Indicator
$HSMM$	Hidden-Semi Markov Models
$I(.)$	Mutual information function
ICU	Intensive Care Unit
$ISTRUST$	Interpretable Spatiotemporal TRansformer for Understanding STructures
K_{self}	Key matrix
K_{token}	Key token
KL	Kullback-Leibler

L_{avg}	Average trajectory length
$L_{horizon}$	Horizon length
L_{MAE}	MAE loss
L_{SupCR}	Supervised contrastive regression loss
L_{window}	Overlapping window length
$Loss^{AE}$	Autoencoder loss
$Loss^{DSMC}$	DSMC loss
$Loss^{tot}$	Total loss
lr	Learning rate
lr^{AE}	Learning rate for training the autoencoder
lr^{DSMC}	Learning rate for training the DSMC model
$LSTM$	Long Short-Term Memory
$LSTMCAE$	Long Short-Term Memory Convolutional Autoencoder
M	Number of models
m	Number of sequential generated episodes for training the VoI model
MAE	Mean Absolute Error
MC	Monte Carlo
$MCMC$	Markov Chain Monte Carlo
MDP	Markov Decision process
MI	Mutual Information
$MIMIC\text{-}III$	Medical Information Mart for Intensive Care III dataset
ML	Machine Learning
MLP	Multi-layer perceptron
MSE	Mean Squared Error
N	Number of specimens/components
N_{act}	Number of actions in the action space
n_{ep}	Current episode
n_{heads}	Number of attention heads

N_{obs}	Number of observations in the observation space
N_{traj}	Number of training trajectories
NLP	Natural Language Processing
NN	Neural Networks
$NUTS$	No U-Turn Sampler
O	Observation function
O_{out}^{dec}	Output layer of the decoder's monotonic module
O_{in}^{enc}	Input layer of the encoder's monotonic module
$P(.)$	Probability
p	Auxiliary target distribution
P_H	Number of vertical patches
P_t	Transition probability at time t
P_W	Number of horizontal patches
p_{thresh}	Probability threshold
PDF	Probability Density Function
PHM	Prognostics and Health Management
$PPDM$	Post-Prognosis Decision-Making
PPO	Proximal Policy Optimization
q	Soft assignment probability distribution
$Q^*(s_t,\alpha_t)$	Optimal action-value function given a state s_t and action α_t at time t
Q_{self}	Query matrix
Q_{token}	Query token
R	Recovery
r_t	Reward at time t
r_v	Neuron equation
r_{act}	Reward for taking action
r_{cost}	The cost of taking an action as a negative reward
r_{maint}	Negative reward for taking a maintenance action

R_{mean}	Mean of recovery
$r_{penalty}$	Negative reward if any RUL of a specimen drops below the predefined RUL threshold
$r_{success}$	Reward if the agent reached successfully the end of the episode
R_{var}	Variance of recovery
$ReLU$	Rectified Linear Unit activation function
RL	Reinforcement Learning
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
RUL	Remaining Useful Life
RUL_{thresh}	RUL threshold
S	Data points to skip per window
s_t	State at time t
$s_{0,nep}$	First available observation at episode n_{ep}
s_{skip}	Number of images to skip per window
s_{window}	Number of windows to skip
$SAPS\ III$	Simplified Acute Physiology Score III
SGD	Stochastic Gradient Descent
SHM	Structural Health Monitoring
$SOFA$	Sepsis-related Organ Failure Assessment score
SPE	Spatial Positional Encoding
$steps_{episodic}$	Total steps to finish an episode
SVR	Support Vector Regression
T	Number of stochastic forward passes in MC dropout
t	Time
$t\text{-}SNE$	t-distributed Stochastic Neighbor Embedding
t_{avg}	Average time
T_{img}	Number of sequential images per window

TPE	Temporal Positional Encoding
$TruncBNorm$	Truncated Bivariate Normal distribution
$TruncNorm$	Truncated Normal distribution
$U^{(i)}$	Total uncertainty at step i
$UMAP$	Uniform Manifold Approximation and Projection
$V^*(s_t)$	Optimal value function given a state s_t at time t
V_{token}	Value token
VoI	Value of Information
W	Width
W_P	Width of each patch
w_{uv}	Weights coming from layer u to neuron v
X'	Reconstructed input data space
X	Input data space
x	Sample drawn from a dataset D
X_{new}	Random variable of normalized RUL after repair
x_{new}	Sample drawn from X_{new}
X_{old}	Random variable of normalized RUL before repair
x_{old}	Sample drawn from X_{old}
Z'_{self}	Intermediate self-attended patches
Z	Embedding space
z	Embedding array belonging to the embedding space
z_{SE}	Spatial encoded embeddings

LIST OF FIGURES

1.1. Illustration of the PHM strategy. 3
1.2. Description of the entire process as implemented in this work; from acquiring sensor data to PPDM. 9

2.1. Transformer-encoder 19
2.2. Types of cracks 22
2.3. Three stage damage accumulation process in composite structures [61, 62]. 23
2.4. Process from raw sequential images to RUL prediction 25
2.5. Patching 25
2.6. Illustration of the patching 25
2.7. Sampling images from raw data 26
2.8. ISTRUST model's architecture 28
2.9. Convolutional embedding 29
2.10. Illustration of the convolutional embedding process 29
2.11. Custom kernel weight initialization 31
2.12. UMAP representation of the spatial encoded embeddings 32
2.13. Cross-validation of successful predictions of RUL 37
2.14. Cross validation of suboptimal predictions of RUL 38
2.15. Spatial attention weights of selected testing specimens at the EOL condition 39
2.16. Spatial and temporal evolution 40
2.17. Temporal attention weights 42

3.1. The concept of DSMC 60
3.2. Architecture of DSMC 68
3.3. The redesigned architecture of the model 71
3.4. Clustering results for each dataset 78
3.5. ROC curves after 1, 25, 50, and 80 hours after each patient's entrance to the ICU 81
3.6. RUL prediction of different prognostic models for one inner and one outlier trajectory 82
3.7. Comparison of RUL curves for C-MAPSS dataset 83
3.8. Time gradients (gradients of each encoder's output with respect to the time feature) 85
3.9. Z-space visualization for the C-MAPSS dataset before training the DSMC model 87
3.10. Z-space visualization for the MIMIC-III dataset before training the DSMC model 88

3.11. (**a**) Clustering results for the F-MOC dataset. (**b**) Z-space visualization for the F-MOC dataset after training the DSMC model. 90
3.12. Stochastic RUL predictions of first and second specimens 90
3.13. Qualitative analysis of the number of clusters hyperparameter for each dataset 91

4.1. Maintenance strategies and actions 111
4.2. Illustration of the MCMC process 118
4.3. Illustration of the entire process from raw data to imperfect repairs 122
4.4. Graphical representation of the random variables 124
4.5. Recovery distribution for single and multiple repairs 128
4.6. Experimental setup and the varying specimens' condition states 131
4.7. The convergence of train and validation losses 134
4.8. Leave-one-out cross-validation 135
4.9. Clustering assignments concerning the test specimens 135
4.10. Stochastic RUL predictions of testing specimens 6-11 136
4.11. Posterior distribution of the random variables 137
4.12. Posterior predictive distribution of the mean of recovery 137
4.13. Posterior predictive distributions of μ_{new} 139
4.14. Posterior predictive distributions of the mean of recovery for sequential repairs 140

5.1. Basic components for implementing the PPDM task 160
5.2. Illustration of the PPDM task 161
5.3. Illustration of the MDP 164
5.4. ANN architecture for both policy network and value function 167
5.5. Loop for deciding whether to acquire new information based on estimated VoI 170
5.6. ANN architecture of the surrogate model 172
5.7. Training performance of the RL agent with four components 175
5.8. Probabilities of taking each decision 176
5.9. Scheduling for the first episode 177
5.10. RUL behavior of each component during the scheduling of the first episode 178
5.11. Results related to VoI model 179
5.12. Number of next sequential episodes to ignore 180
5.13. Training performance of the RL agent with different number of components 181

A.1. Spatial and temporal evolution of testing specimens 1 and 2 195
A.2. Spatial and temporal evolution of testing specimens 3 and 4 196

B.1. Survivability rates for MIMIC-III dataset 198
B.2. Reliability curves for C-MAPSS dataset 199
B.3. The convergence of train and validation losses for each dataset 200
B.4. Two examples of image snapshots of one specimen 201
B.5. An example of a specimen's low-level features 202

C.1. Posterior predictive distributions of the random variable μ_{new} using non-informative prior distributions 207

D.1. RUL behavior of each component during the scheduling of day 36 within the horizon . . . 213

LIST OF TABLES

1.1. Phases and associated challenges of the PHM strategy per chapter 6

2.1. Hyperparameter tuning 35
2.2. Epochs for which the encoder and predictor states were taken 36
2.3. Parametric study of hyperparameters s_{window} and s_{skip} 42

3.1. List of variables extracted by the MIMIC-III dataset 62
3.2. List of variables extracted by the C-MAPSS dataset 63
3.3. The low-level features that are considered and extracted by the AMSY-6 Vallen Systeme GmbH 64
3.4. General characteristics of the F-MOC dataset 65
3.5. Hyperparameters' values for the LSTM and CNN3D layers 72
3.6. Hyperparameter search ranges and final values optimized by the Bayesian optimization algorithm for each dataset 74
3.7. Comparison table for the MIMIC-III dataset 80
3.8. RMS for each test engine 84

4.1. The necessary components for the Bayesian modeling of a repair model . . . 126
4.2. The necessary components for the Bayesian modeling of our proposed repair model 127
4.3. Test matrix of the experimental setup 129
4.4. The low-level features that are considered and extracted by the AMSY-6 Vallen Systeme GmbH 130
4.5. Hyperparameter search ranges and final values optimized by the Bayesian Optimization algorithm 133
4.6. Tuned hyperparameters for the initialization of the MCMC with the NUTS algorithm 133

5.1. Hyperparameters related to DRL utilizing PPO algorithm 167
5.2. Hyperparameters related to surrogate model related to VoI 171
5.3. Chosen hyperparameters related to PPDM for this case study 174
5.4. Characteristics of test specimens/components for this case study 174

B.1. Reproducibility of the training process 197

1

INTRODUCTION

With this introductory chapter, the main objectives of the thesis are described based on the existing critical barriers of the Prognostics and Health Management field. Then, a plan is presented to structure the proposed solution.

1.1. THE PROGNOSTICS AND HEALTH MANAGEMENT STRATEGY

In a world increasingly reliant on complex systems–from engineering to healthcare (healthcare devices, patients)–the quest for enhanced reliability and performance has led to the rise of the Prognostics and Health Management (PHM) strategy. The discipline of PHM has emerged as a key enabler in ensuring the continuous and optimal functioning of these systems by providing early insights into their health status and predicting potential failures [1]. The term 'PHM' encompasses a set of multidisciplinary approaches and tools designed to monitor, assess, and manage the health of systems and their components in real-time [2]. By integrating data-driven analytics, advanced sensors, Machine Learning (ML), and domain knowledge, PHM offers a proactive means of identifying anomalies [3–5], predicting impending failures [6–8], and prescribing timely maintenance or remedial actions [9–11]. As such, PHM has the potential to revolutionize the way industries approach maintenance, improving system utilization, reducing downtime, and minimizing life cycle costs.

One of the central pillars of PHM that has garnered increasing attention in recent years is the estimation of Remaining Useful Life (RUL). While PHM encompasses a broad spectrum of techniques for monitoring and assessing system health, RUL estimation stands out as a critical phase in the quest for enhanced system reliability and performance optimization. The concept of RUL can be succinctly defined as the remaining time a system can be expected to operate within acceptable performance and reliability limits before experiencing failure or degradation beyond tolerable levels.

When focusing on systems such as structures, the PHM strategy promises to revolutionize how we monitor and maintain them, aiming not just to react to failures, but to predict and prevent them. Proactively preventing failures is the heart of all maintenance strategies. Condition-based maintenance (CBM) is a maintenance strategy that monitors the real-time condition of systems to determine when maintenance should be performed. Unlike traditional time-based or reactive maintenance approaches, which rely on fixed maintenance schedules, CBM focuses on identifying early signs of malfunction (as far as the placed sensors allow for this detection) or degradation through continuous assessments. This is typically achieved by analyzing data from sensors and operational parameters. The primary objective of CBM is to maximize the RUL of multiple components representing a structure while minimizing the risk of unexpected failures. By addressing maintenance needs based on the actual condition of the structures' components, CBM ensures that interventions are performed only when necessary, enhancing system reliability and efficiency while reducing downtime. However, implementing CBM effectively requires sophisticated methodologies to predict the RUL of systems and optimize maintenance actions.

It can be already observed that a strong correlation exists between CBM and PHM. In essence, PHM primarily focuses on the health and performance of systems, structures, individual components, or the interconnected components within a structure. CBM leverages PHM to plan maintenance activities across multiple systems. This relationship becomes clearer when examining the phases of the PHM strategy. As illustrated in Figure 1.1, given a set of structures (here the motors of an aircraft) forming a system (the aircraft), the PHM strategy begins with data collection from sensors placed on a system's components (here the motor blades), using one or more Structural Health Monitoring (SHM) techniques. The collected data are stored in a central unit (e.g., a computer) and processed to extract relevant features. These features can be used for diagnosis (e.g. damage detection and localization) or prognosis, such as predicting RUL of components. An intermediate step may involve constructing a Health Indicator (HI), which indicates the condition of each component. Diagnostics and prognostics can be performed using either the extracted features or HIs. Following the prognosis phase, the decision-making phase, known as Post-Prognosis Decision-Making (PPDM) [12], selects the most appropriate maintenance actions and time, based on the system's health condition and a set of predefined constraints. When maintaining multiple systems, PPDM can offer independent decisions for each component or a coordinated schedule for all components within each system, enabling a comprehensive maintenance plan across all systems (this last step is out of the scope of PPDM). Therefore, applying the PHM strategy to individual systems and planning their maintenance collectively forms the foundation of the CBM strategy.

Although Figure 1.1 represents an application from aviation, the same process is followed, given different systems, structures, and components. Additionally, it should be mentioned that the PHM strategy can be followed even when components and structures forming a system vary completely from each other.

It is worth highlighting that maintenance actions and PPDM phases are shown mid-way through the PHM strategy, as they can be considered only by CBM or PHM, or both. Although PHM aims at decision-making, thus requiring PPDM and a set of maintenance actions within its scope, these phases have primarily been associated with CBM independently. In such a case, PHM's output was the prognostics phase. Nevertheless, the inclusion of 'repair' as a maintenance action has introduced the need to model the component's behavior post-repair (in terms of RUL) before advancing to the PPDM step. This is because, in practice, repairs are rarely perfect; the component's condition typically falls somewhere between its pre-repair state and the fully restored, brand-new condition. Consequently, maintenance actions and, hence, PPDM are becoming integral phases within the PHM strategy, gaining increasing importance throughout the framework.

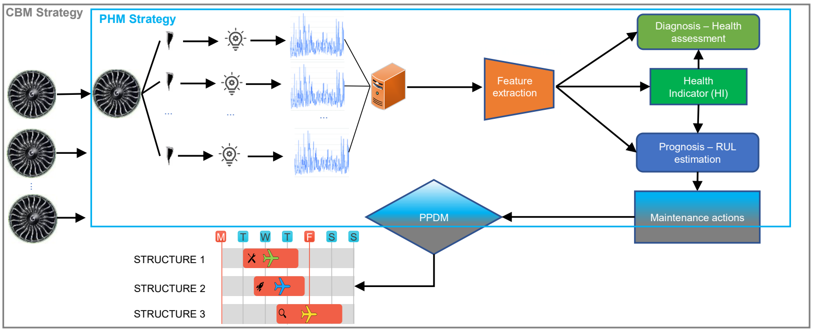


Figure 1.1.: Illustration of the PHM strategy.

1.2. PROBLEM STATEMENT

The era of digitalization has offered a vast amount (both, magnitude and diversity) of data that should be processed and monitored accordingly to produce insightful information about the component's current and future condition, being necessary steps for planning maintenance actions over systems. As the landscape of PHM evolves, rather than focusing on a single phase of the PHM strategy, there exist an increasing demand of a holistic and generalized framework. This means that it becomes apparent that each phase, ranging from data acquisition to PPDM, requires independent yet interconnected management. This approach acknowledges real-world complexities, where these phases may overlap with each other (signifying interconnected management). Nevertheless, a comprehensive framework has yet to be developed. This is largely due to several significant challenges faced by experts. One major obstacle is managing raw, multi-dimensional data and extracting relevant prognostic features to construct HIs. Additionally, there is considerable uncertainty in both the data and the models developed from them. Another complication arises from maintenance actions, particularly when imperfect repairs are involved, as these can influence the RUL behavior. Finally, coordinating and scheduling multiple maintenance actions within a given time horizon, while considering various operational and maintenance constraints, adds further complexity. The following paragraphs will discuss each of the aforementioned challenges in more detail.

One of the primary challenges in developing a holistic framework is managing raw multi-dimensional data, which often comes in large volumes from various sensors and monitoring systems. This data is typically noisy, heterogeneous, and multi-modal, making it difficult to preprocess, clean, and transform into a structured format suitable for analysis. Additionally, the complexity of relationships between different variables further complicates data handling, requiring advanced statistical and ML techniques to extract meaningful insights.

In order to process the raw data, it is important to identify relevant features that accurately reflect the system's health, which in turn is another major challenge. Prognostic-related features must be carefully extracted to develop meaningful HIs that can effectively predict the RUL of components. However, defining these features is often system-specific and requires domain expertise and advanced signal processing techniques before implementing any ML model. Poorly chosen features can lead to unreliable health assessments and inaccurate predictions.

Furthermore, the presence of uncertainty in both the collected data and the predictive models adds another layer of complexity. Sensor data can be noisy, incomplete, or affected by environmental conditions, leading to data scarcity or sparsity. Moreover, predictive models, whether data- or numerically-driven, are often approximations that cannot fully capture the true degradation process. This uncertainty can impact decision-making, making it difficult to achieve reliable and robust predictions for maintenance planning.

Maintenance activities, especially imperfect repairs, introduce another challenge in implementing holistic approaches regarding the PHM strategy. Unlike perfect repairs that restore a system to its initial condition, imperfect repairs only partially restore the system's health. This makes it difficult to predict RUL accurately, as the degradation trend does not reset but rather continues with altered dynamics. Properly incorporating the effects of maintenance actions into predictive models is crucial for realistic and effective maintenance planning.

Coordinating multiple maintenance actions within a specific time horizon while considering imperfect repair actions, operational constraints, cost, and resource availability is a complex optimization problem. Different components of a system may have varying degradation rates, failure risks, health recovery, and maintenance requirements, making it challenging to determine the optimal schedule. Additionally, the cumulative uncertainty from earlier phases of the PHM strategy, which is quantified during the prognostics phase, adds another layer of complexity to scheduling decisions. Developing an efficient maintenance schedule requires balancing reliability, cost, and operational efficiency while accounting for uncertainties in degradation and repair effects.

Finally, in PHM, data collected from sensors is often in the form of time series. When dealing with complex, multi-dimensional raw time-series data from various sensor systems, ML, particularly Deep Learning (DL), becomes essential. Developing Neural Network (NN) models allows the capture of intricate relationships within the data, enabling the extraction of simplified features and the construction of meaningful HIs that can be easily integrated into prognostic models. Additionally, during the PPDM phase, where different data sources, such as stochastic RUL, tabular data, and static data, must be combined for a set of components, NN presents a compelling option to improve decision-making algorithms. Nevertheless, the use of these powerful yet opaque black-box models raises concerns about their interpretability. This lack of transparency can reduce reliability of the entire PHM framework, making it a critical challenge that must be addressed in this thesis.

To sum up, given the points discussed above, we can summarize the challenges associated with building a PHM framework as follows:

1. Building a generalizable PHM framework for varying applications.
2. Extracting important information from raw multi-dimensional and multi-modal data.
3. Creating prognostic-related features to construct HIs.
4. Quantifying the introduced uncertainty due to noisy data, model imperfections, stochastic nature of RUL, imperfect repairs, and decision-making.
5. Modeling the impact of imperfect repairs on RUL's behavior.
6. Scheduling multiple maintenance actions within a time horizon considering imperfect repairs given RUL predictions, operational conditions, logistics, and a set of constraints.
7. Making each phase interpretable, especially the ones where NNs are considered.

These are the critical challenges associated with constructing a holistic PHM framework. Additional barriers, specific to each phase, are identified and discussed in the corresponding chapters.

1.3. RESEARCH PLAN

There are two main approaches to developing a holistic framework for the PHM strategy. The first approach involves combining phases together utilizing a unified model. For example, a single model could combine feature extraction, HI construction, and prognostics in one model potentially speeding up framework construction and deployment. However, this approach has two significant limitations. Firstly, it is of great challenge to optimize more than one phase sequentially end-to-end. Secondly, the resulting model can become too complex, making it less transparent and harder to improve. In chapter 2, the second limitation reflecting challenge 7 is faced whilst it is demonstrated that performance is not optimal when considering such a setup. This is left as an open question of whether a unified model can solve multiple phases at once.

The second approach is to address each phase separately, passing the output of one phase as input to the next. This is especially useful when pre-existing models have already been developed for varying phases, thus avoiding starting from scratch. Building on previous research, we can independently refine and connect these models to form a cohesive end-to-end PHM framework. This approach eliminates the need for entirely new models, simplifies the process, offers modularity, and makes the framework easier to adapt to other applications by having a clearer organization over each phase. Beyond the aforementioned advantages over the first approach, given the observations discussed in chapter 2, the second approach is preferred. Therefore, the subsequent chapters focusing on developing the framework are based on this approach.

In this regard, this thesis aims to develop a holistic PHM framework for the first time by overcoming the abovementioned challenges that have previously hindered its realization. Figure 1.2 illustrates the proposed framework in detail regarding each phase of the PHM strategy. Successfully building this framework requires detecting in which phases each challenge appears. Then, at each phase, the corresponding challenges should be resolved. In the next paragraphs, each phase and the associated challenges which should be resolved are presented in detail. Table 1.1 summarizes the following paragraphs for clarity.

Table 1.1.: Phases of the PHM strategy and the associated challenges for each phase per chapter. chapter 2 does not belong in this approach, hence is not shown.

Phase no.	Research field	Chapter no.	Challenge	1	2	3	4	5	6	7
1	Data acquisition (sensors)	-								
2	Feature extraction	3		✓	✓					✓
3	HI construction	3		✓		✓				✓
4	Prognostics	3, 4		✓			✓			
5	Imperfect repairs	4		✓			✓	✓		
6	PPDM	5		✓			✓		✓	✓

This thesis is focused on developing the required models for building the PHM framework, and these models are agnostic of the process of placing sensors on the examined components. In this regard, the **first phase**, concerning acquiring sensor data from a system's components, remains out of scope of this thesis. It is assumed that the examined structure already has sensors placed on it and that these sensors contain information about degradation. As such, any system that degrades over time can be considered an application of the framework.

The **second phase** of the PHM strategy focuses on extracting prognostic-related features given raw multi-dimensional and/or multi-modal sensor data facing challenge 2. These features can be then transformed into HIs, signaling the **third phase** and addressing challenge 3. The **fourth phase** concerns predicting RUL under uncertainty via a prognostic model given the constructed HIs. Uncertainty quantification inside the prognostic model faces a challenge 4. The development of phases 2, 3, and 4 are presented in chapter 3. Particularly, a novel NN-based hybrid model that combines data-driven approaches supported by physics-based information is introduced for extracting meaningful monotonic features. These features are then converted via deep clustering analysis into discretized HIs. In this chapter, it is additionally demonstrated that any prognostic model can be applied at phase 4 by comparing varying state-of-the-art models. The performance of the proposed models across these phases is validated through case studies on three distinct systems: one from healthcare and two from engineering (aerospace and mechanical). Each case study aims to face unique challenges. The healthcare system shows that the framework can be applied even in degrading systems outside the engineering field, which is associated with the challenge 1 in phases 2,3, and 4. One of the case studies regarding engineering systems primarily focuses on the model's interpretability in phases 2 and 3 (challenge 7) whilst the last case study further focused on challenge 2, especially regarding multi-modality.

The **fifth phase** concerns modeling one or more sequential imperfect repairs and how these repairs affect RUL predictions under uncertainty, facing challenge 5. This phase is extensively presented and discussed in chapter 4. The imperfect repair model is constructed in a way to be agnostic of the repair process. Therefore, it can be applied to any system subject to imperfect repairs, again addressing challenge 1. Quantifying and combining the uncertainty that comes from the prognostic model with the uncertainty introduced by the imperfect repair process answers challenge 4. The case study in this chapter focuses on an aluminum structure to simplify the experimental campaign, as repairs can be easily applied. Conducting experiments on more complex structures, such as composites, would be significantly more time-consuming, costly, and challenging to implement. However, incorporating such structures would follow the same approach, as the input to the imperfect repair model consists of RUL predictions generated by the models developed in previous phases, which have already been validated on more complex systems.

The **sixth phase**, which signals the final phase of the PHM strategy, focuses on the PPDM task. This task is modeled as a Markov Decision Process (MDP) and optimized via Deep Reinforcement Learning (DRL). Decisions are chosen between 'do nothing', 'perform an imperfect repair', or 'replace with a brand-new component'. The most complicated system is a multi-component one, where decisions regarding each component should be combined to extend the working life of the system. The PPDM phase is implemented and evaluated based on a multi-component system given RUL, a set of constraints, logistics, and operational conditions, which is associated with challenge 6. Additionally, the cumulative uncertainty quantified in the previous phases is decomposed based on its source (model or data) and propagated through the DRL model into the final decisions. This is represented as probabilities that indicate the model's confidence in each decision, enhancing interpretability and is directly associated with challenges 4 and 7. In this phase, the same case study is considered as in chapter 4.

In summary, by facing the associated challenges with each phase of the PHM strategy, a successful holistic PHM framework can be developed. This framework should be generalizable across different systems, therefore, it should be validated by diverse case studies representing various deteriorating systems. The specific choice of datasets is less important than ensuring they exhibit multi-diversity, multi-modality, and raw data characteristics. Combining phases together into unified models can be time-efficient while still being interpretable, hence it is the first option that is examined in chapter 2. After this chapter, phases are treated independently, but they are still interconnected. Furthermore, prognostic-related features should be developed capable of being transformed into appropriate HIs given raw multi-dimensional and multi-modal sensor data (chapter 3). Moreover, any existing prognostic model that incorporates uncertainty should be seamlessly integrable. Given RUL information, an imperfect repair model should be developed to capture how RUL evolves after a repair while also accounting for the uncertainties associated with both RUL predictions and the repair process itself (chapter 4). Lastly, the PPDM phase should incorporate RUL data, the dynamics of imperfect repairs, and uncertainty to make optimal decisions within a predefined time horizon (chapter 5). These decisions should also be interpretable, meaning they should reflect the decision-maker's level of confidence.

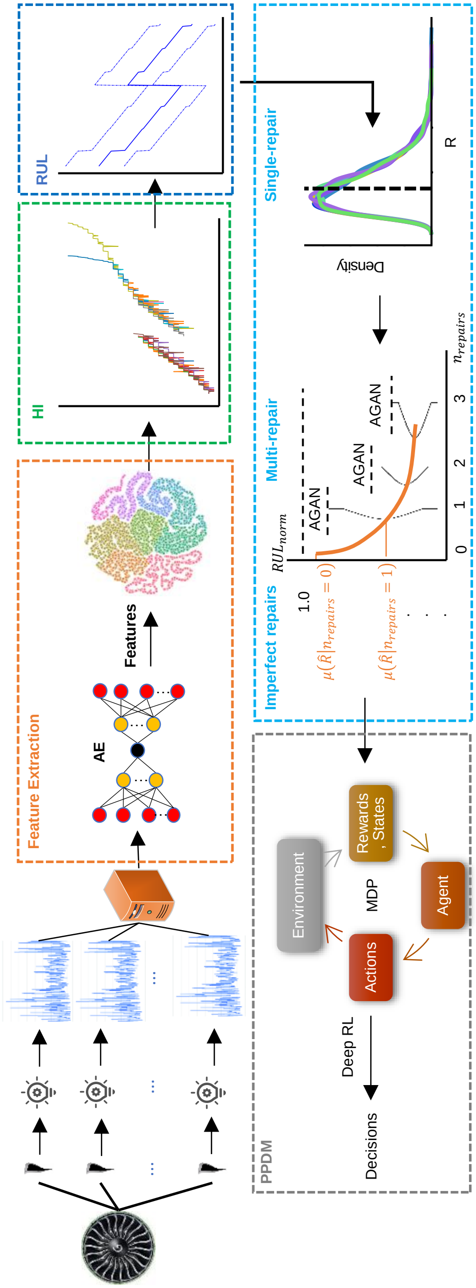


Figure 1.2.: Description of the entire process as implemented in this work; from acquiring sensor data to PPDM.

REFERENCES

[1] T. Xia, Y. Dong, L. Xiao, S. Du, E. Pan, and L. Xi. “Recent advances in prognostics and health management for advanced manufacturing paradigms”. In: *Reliability Engineering & System Safety* 178 (2018), pp. 255–268. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2018.06.021. URL: https://www.sciencedirect.com/science/article/pii/S095183201731459X.

[2] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel. “Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications”. In: *Mechanical Systems and Signal Processing* 42 (Jan. 2014), pp. 314–334. DOI: 10.1016/j.ymssp.2013.06.004.

[3] H. Li, W. Zheng, F. Tang, Y. Zhu, and J. Huang. “Few-shot time-series anomaly detection with unsupervised domain adaptation”. In: *Information Sciences* 649 (2023), p. 119610. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.119610. URL: https://www.sciencedirect.com/science/article/pii/S0020025523011957.

[4] C. Liu, Z. Yuan, B. Chen, H. Chen, and D. Peng. “Fuzzy granular anomaly detection using Markov random walk”. In: *Information Sciences* 646 (2023), p. 119400. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.119400. URL: https://www.sciencedirect.com/science/article/pii/S0020025523009854.

[5] L. Yan, C. Luo, and R. Shao. “Discrete log anomaly detection: A novel time-aware graph-based link prediction approach”. In: *Information Sciences* 647 (2023), p. 119576. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.119576. URL: https://www.sciencedirect.com/science/article/pii/S0020025523011611.

[6] Z. Wang, N. Liu, C. Chen, and Y. Guo. “Adaptive self-attention LSTM for RUL prediction of lithium-ion batteries”. In: *Information Sciences* 635 (2023), pp. 398–413. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.01.100. URL: https://www.sciencedirect.com/science/article/pii/S0020025523001007.

[7] C. Zhou, A. Hou, P. Dai, A. Li, Z. Zhang, Y. Mu, and L. Liu. “Risk factor refinement and ensemble deep learning methods on prediction of heart failure using real healthcare records”. In: *Information Sciences* 637 (2023), p. 118932. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.04.011. URL: https://www.sciencedirect.com/science/article/pii/S0020025523004991.

[8] A. Yan, W. Wang, C. Zhang, and H. Zhao. “A fault prediction method that uses improved case-based reasoning to continuously predict the status of a shaft furnace”. In: *Information Sciences* 259 (2014), pp. 269–281. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.04.025. URL: https://www.sciencedirect.com/science/article/pii/S0020025513003290.

[9] S. Hao, J. Zheng, J. Yang, H. Sun, Q. Zhang, L. Zhang, N. Jiang, and Y. Li. “Deep reinforce learning for joint optimization of condition-based maintenance and spare ordering”. In: *Information Sciences* 634 (2023), pp. 85–100. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.03.064. URL: https://www.sciencedirect.com/science/article/pii/S0020025523003572.

[10] P. Baraldi, F. Mangili, and E. Zio. “A belief function theory based approach to combining different representation of uncertainty in prognostics”. In: *Information Sciences* 303 (2015), pp. 134–149. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2014.12.051. URL: https://www.sciencedirect.com/science/article/pii/S0020025515000031.

[11] Y. Hu, X. Miao, Y. Si, E. Pan, and E. Zio. “Prognostics and health management: A review from the perspectives of design, development and decision”. In: *Reliability Engineering & System Safety* 217 (2022), p. 108063. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.108063. URL: https://www.sciencedirect.com/science/article/pii/S0951832021005652.

[12] A. Bousdekis, B. Magoutas, D. Apostolou, and G. Mentzas. “Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance”. In: *Journal of Intelligent Manufacturing* 29 (Aug. 2018). DOI: 10.1007/s10845-015-1179-5.

2

FROM RAW HIGH-DIMENSIONAL DATA STRAIGHT TO RUL ESTIMATIONS

In this chapter, a NN model is presented for predicting end-to-end RUL under uncertainty from raw sequential images of a composite structure. The proposed model can focus on the parts of the structure where damage is being propagated and correlate this damage with RUL, thus offering interpretability. This chapter highlights the importance of incorporating interpretability into NN models and emphasizes that skipping the feature extraction and HI construction phases should be avoided.

This chapter has been published in Composites Part B: Engineering **287**, 111863 (2024) [1].

Deep Learning advancements facilitate RUL forecasting by extracting features from diverse data formats such as time series, images, or sequences thereof, in one, two, or three dimensions, respectively. Yet, predicting RUL from image sequences often relies heavily on resource-intensive techniques like Digital Image Correlation (DIC), complicating data acquisition. To address challenges with high-dimensional data and unreliable models, this study introduces ISTRUST, an innovative Transformer-based architecture. ISTRUST (Interpretable Spatiotemporal TRansformer for Understanding STructures) tackles the dual challenges posed by high-dimensional data and the black-box nature of existing models. Leveraging Transformers' attention mechanism, ISTRUST breaks down the spatiotemporal domain, effectively realizing interpretable RUL predictions under uncertainty using only sparse raw image sequences as input. Evaluated on fatigue-loaded composite samples showcasing crack propagation, ISTRUST interprets the relation between cracks and RUL via the attention mechanism. The results substantiate its capacity to interpret and clarify instances in which predictions may exhibit variability in accuracy. Through the attention mechanism, a strong correlation between the model's spatiotemporal focus and the RUL predictions is established, making it, to the best of our knowledge, the first model to provide interpretable stochastic RUL predictions directly from sequential images of this nature.

2.1. INTRODUCTION

In today's rapidly evolving technological landscape, the seamless operation of critical structures is not only a matter of economic significance but also a concern for safety, sustainability, and efficiency. Under the expansive umbrella of PHM, the estimation of RUL has seen significant advancements, particularly in the context of data-driven approaches. While it has been widely applied to univariate or multivariate time-series data [2, 3], such as vibration-based signals, and image data [4, 5], it is important to acknowledge that its application to sequential images, which represents high-dimensional time-series data, is still in its nascent stages.

In the realm of univariate and multivariate time series data, RUL estimation has demonstrated remarkable success across various industries. This approach typically involves analyzing historical sensor measurements and operational data to predict when a system or component is likely to reach a predefined End-of-Life (EOL) threshold. Numerous algorithms and models, ranging from classical statistical [6–8] and numerical [2, 3] methods to sophisticated ML techniques [9–14], have been developed to handle this type of data. These approaches have proven invaluable in scenarios where data is collected over time, making them suitable for applications like SHM, predictive maintenance, and failure prognosis. In the context of crack propagation, this type of data often relates to crack growth extracted by measuring the structure's strains. Existing works considered post-processed strain data as input to predict RUL end-to-end utilizing different DL models, such as recurrent neural networks (RNN) and convolutional neural networks (CNN) [15, 16]. Despite the simplification of the process of predicting RUL through reduced memory requirements and the construction of simple DL models, these methods may lead to a loss of detailed information. This is mainly because the raw images contain complex patterns that do not directly reveal useful information. As a result, extracting meaningful data, such as in DIC, requires intensive computational processing.

Similarly, the extension of RUL estimation to image data, where predictive models are trained to make RUL predictions based on a single image or snapshot, has shown promise in diverse fields. Applications range from predicting the remaining lifespan of critical components in manufacturing machinery [4, 5] to assessing the structural integrity of infrastructure components such as concrete structures [17]. With the emerging field of DL, CNN, has played a pivotal role in enabling RUL predictions from images, revolutionizing the way assets are monitored and managed.

However, when dealing with video data, such as sequences of images, the landscape becomes more intricate. For RUL estimation, these data mainly consist of video frames, reflecting the damage of the examined structure and how it progresses. While the potential of such data in RUL estimation is evident [18–20], the complexity and dimensionality of this information pose unique challenges. Extracting meaningful features, handling spatiotemporal dependencies, and building models capable of predicting RUL accurately from frames remain active areas of research. Consequently, NNs, with their capacity to automatically learn hierarchical representations from raw data, have shown remarkable capabilities in handling the challenges of two- or three-dimensional data. State-of-the-art approaches include a combination of RNN, such as Long Short-Term Memory (LSTM), with CNN to effectively capture the temporal and spatial features, respectively [21, 22].

Very recently, the LSTM-CNN approach started being replaced by tranformers [18–20, 23–27], an advanced technique initially proposed in Natural Language Processing (NLP) that has now been applied in engineering applications as well [18–20, 28–34]. In addition to the success of transformers in NLP [23], advancements have enabled their application in computer vision. Dosovitskiy et al. [24] proposed vision transformers that split images into patches and forward them as a sequence of linear embeddings to a transformer-encoder. Subsequent developments incorporated the ability to analyze a sequence of images by separating the attention mechanism into a spatial and temporal domain [25–27]. However, these studies either encountered challenges in fully separating the temporal and spatial domains throughout their architecture [25, 26] or struggled to reduce the network's depth [27].

The attention mechanism, the main building block of transformers, has given remarkable outcomes in a variety of engineering studies [18–20, 30–34]. Concerning RUL [18–20, 28–30], however, the aforementioned works are one-dimensional data that are easier to handle. Nevertheless, transformers are gaining increasing popularity in volumetric data as well. From video recognition [31] and object detection [32], to semantic segmentation [33] and damage detection [34], they enable the development of models with state-of-the-art performance due to their capacity for capturing important information of the complex input data. More importantly, they have the ability to interpret the results and visualize how the input-output pairs are correlated via the attention weights [35], which is a huge step towards unfolding the black-box barriers of DL and being advantageous over the typical LSTM-CNN techniques.

In spite of the substantial progress achieved by transformers in various engineering applications, their utilization in RUL prediction with raw sequential image data as the input remains largely unexplored. This is primarily attributed to the inherent limitations of raw images, which do not provide adequate or accurate information for reliable predictions. Mitigating this requires the application of extensive image processing techniques, notably the incorporation of DIC to capture pertinent features [36, 37]. Additionally, surrogate modeling is employed to augment interpretability, however, it is noteworthy that it may inadvertently oversimplify the complex analysis of structures [38]. These approaches can be correspondingly time and computationally inefficient, and there is no evident connection between the sequence of images and the predictions of RUL [38]. This makes the idea of developing a single model that can handle everything from raw sequential image data to RUL prediction end-to-end by harnessing the interpretability inherent to the attention mechanism.

Although limited work exists on tracking crack damage via image-based techniques, DL has been considered on other brittle structures. Pantoja-Rosero et al. [39] proposed a CNN model for detecting crack damage on images of earthquake-affected urban scenes accompanied by crack annotations. Another work considered DL for pixel-level crack segmentation on masonry surfaces [40]. This work presented a CNN model for crack detection on patch- and pixel-level images. By training with different network architectures and utilizing transfer learning the model succeeded in accurate crack classification and segmentation. A limitation of these implementations is inherent in the selected DL architecture, which relies on CNNs. This reliance may result in a potential loss of precision in crack detection and segmentation, particularly in discontinuous areas of the structure, as CNNs are highly dependent on neighboring pixel values. Furthermore, the aforementioned works, along with others in the literature [41–43], do not address the task of end-to-end crack damage tracking and RUL prediction using sequential image data as input.

While Physics-informed ML techniques are available [44, 45], they tend to be highly task-specific. The objective of this research is to develop an ML model that achieves both interpretability and generalizability across various applications. The ISTRUST model possesses the capability to discern scenarios in which predictions may be suboptimal and when they are poised to demonstrate efficacy. Its principal objective is to offer insights and comprehension regarding the quality of predictions, thereby achieving an unparalleled level of reliability and interpretation within the context of black-box models. The interpretability is achieved by decomposing the spatiotemporal continuum into spatial and temporal dimensions. Therefore, the spatial attention weights show the correlation between the cracks of each image separately and the predicted RUL, while the temporal attention weights indicate the contribution of each frame to the corresponding RUL. To the best of the authors' knowledge, this work is the first to provide interpretable RUL predictions directly from sequences of raw images and the model's performance is demonstrated with an experimental dataset acquired by composite samples that are under fatigue loads with visible cracks that propagate with time [46]. Finally, since RUL is a random variable, the proposed ISTRUST model considers the uncertainty of the predictions by integrating the Monte Carlo (MC) dropout technique [47] into the overall process.

In summary, the contribution and novelty of the current study to the corresponding research areas is highlighted as follows:

- The model, which consists of a combination of CNN and self-attention layers, performs RUL prediction under uncertainty given unprocessed raw sequential images as input of composite samples that is under fatigue loads.

- This approach contradicts typical models by unveiling the black-box barrier through its integrated interpretation. The model's interpretability is further improved by the effective decomposition of the spatiotemporal domain.

- Interpretability has been achieved by utilizing only one transformer-encoder layer for each domain, deviating from the conventional approach of employing `5-10` layers in vision transformers.

- While it is acknowledged that the predictive performance of the model may not be optimal in certain specimens, it is imperative to note that our method offers a rational and coherent explanation for the underlying phenomena contributing to this observation, paving the way for visionary and innovative concepts for further improving its efficacy.

- We incorporate supervised contrastive learning to compel the model to filter out irrelevant information, such as the small variations in the characteristics between each specimen, and the projected data augmentation technique [48–50] that helps in the training process. Moreover, we validate the model's performance to effectively filter out irrelevant information via a Uniform Manifold Approximation and Projection (UMAP) representation [51]. Utilizing contrastive learning we managed to train a vision transformer-inspired architecture on limited data.

This chapter is consequently organized as follows: section 2.2 describes the methodology from vanilla NN to transformers, self-attention, and Monte Carlo (MC) dropout for uncertainty quantification, along with the data acquisition strategy, experimental setup, and damage propagation phenomena in composite materials. In section 2.4, the architecture of the model is presented, visualized, and explained, accompanied by several training strategies to tackle the limited size of the dataset. Finally, the results coming after the model training are shown in section 2.5, followed by the conclusions in section 2.6.

2.2. THEORY

2.2.1. TRANSFORMERS & MULTI-HEAD ATTENTION

NN is a common tool used in ML to learn patterns from data. An NN consists of an input layer, optional hidden layers, and an output layer. A Fully Connected (FC) layer consists of several neurons, which aggregate and process the output of the previous layers linearly via matrix multiplications. An NN with at least one hidden FC layer and a non-linear activation function for that hidden layer is called a multilayer perceptron (MLP). Prior to applying a forward propagation, the weights and biases are initialized; typically the weights are sampled from a normal or uniform distribution and the biases are zeroed. Using the backward propagation algorithm [52], where the error is back propagated through the NN layers, the gradients of the weights and biases with respect to the error can be calculated. Using these gradients and an optimizer like Stochastic Gradient Descent (SGD), the weights and biases can be updated accordingly.

MLPs are a very inefficient way of handling images as they require flattening the image to a one-dimensional vector, resulting in an input layer with a large size, and consequently a large amount of weights. Furthermore, MLPs lack the ability to learn a specific feature irrespective of its position within an image. Consequently, MLPs must repetitively learn the same feature for each location individually, thus lacking translational invariance. For an image $x \in \mathbb{R}^{KxHxW}$, where K is the number of features (e.g. 3 in the case of an RGB image), H is the height of the image in pixels and W is the width of the image in pixels, CNN solve the problem of a large number of weights and the translational invariance by utilizing a different aggregation technique, i.e. a filter or kernel called 2D convolution [53, 54].

To further improve the models' performances, the transformer architecture was proposed in [23]. By processing sequential data using self-attention, Neimark et al. [55] eliminated the need for RNNs or CNNs. Transformers consist of an encoder, which processes the input data, and a decoder, which generates the output data. The encoder and decoder consist of sequential interchanged self-attention and MLP layers and are connected through the encoder-decoder attention mechanism. Following the complexity of the original transformer, the work in [56] successfully simplified the architecture by utilizing only the transformer-encoder, resulting in reduced complexity and enabling improved interpretability. In this regard, the proposed architecture is considered as a fundamental stepstone for constructing the ISTRUST model.

Self-attention acts as the fundamental component of transformers, in which queries (Q), keys (K), and values (V) are computed based on the input data. The queries, keys, and values are obtained by means of a linear projection from a sequence of one-dimensional input vectors with size d_{model}. Subsequently, they all undergo a scaled dot-product attention (see also Figure 2.1b) [23]:

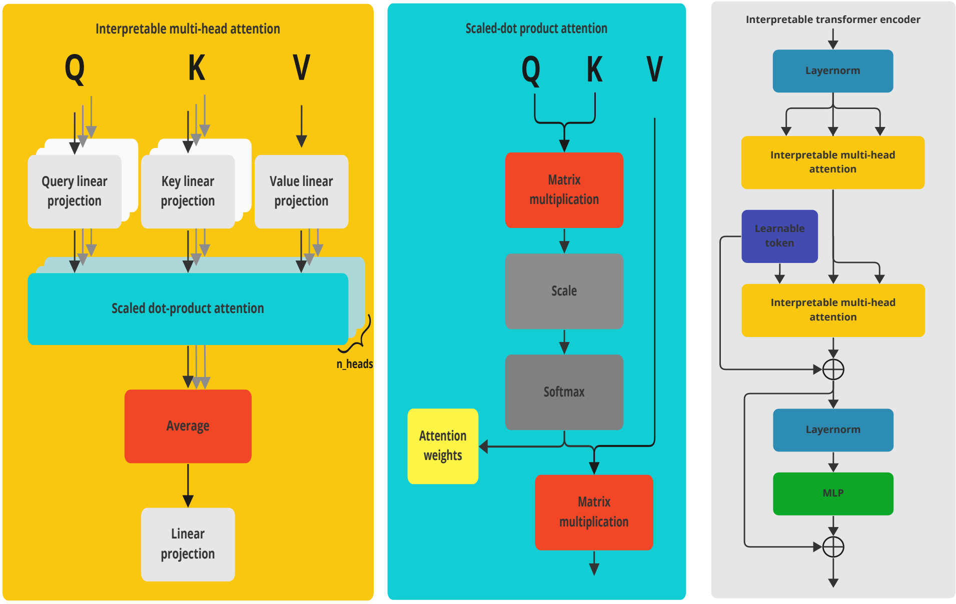
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V \tag{2.1}$$

where $\sqrt{d_{model}}$ is a scaling factor. Initially, the queries and keys are multiplied by the means of a dot product between each other. After scaling and applying the softmax function, the result is a matrix of attention weights. Finally, these attention weights are a matrix multiplication with the values, effectively collapsing the number of keys and the corresponding associated information, represented by the values.

Nevertheless, since the size of the queries and keys is typically in the order of a hundred, the computation of the attention weights is computationally expensive. Rather than downscaling d_{model}, which would sacrifice information, the queries and keys are split by performing multiple linear projections, resulting in multiple heads, as shown in Figure 2.1a. The resulting projected queries and keys have a size of $d_k = d_{model}/n_{heads}$, where n_{heads} represents the total number of heads. Typically, the values are also divided among the heads. However, this approach would compromise interpretability, as it would introduce an indeterminate level of relative importance among the heads. For that reason, a variant of multi-head attention called interpretable multi-head attention is implemented [57]. In interpretable multi-head attention, the values all undergo the same linear projection to achieve the projected values. Unlike the queries and keys, the values do not pose a computational bottleneck and can retain their original size of

d_{model}. Following the scaled-dot product attention in each head, the resulting attended outputs are averaged. Next, the averaged outputs are passed through a linear projection to finally obtain the attended vectors.

As depicted in Figure 2.1c, the interpretable transformer-encoder is distinguished from the typical architecture due to the interpretable multi-head attention mechanism. In our proposed architecture, there are two interpretable multi-head attention blocks. Similarly to the original transformer-encoder, the embedding vectors produced by computing the inputs (in our case, a sequence of images), are initially normalized using layer normalization [58] and then are consequently fed to the first interpretable multi-head attention block. Next, they are passed through the second multi-head attention block alongside a learnable token, which will be further described in section 2.2.2. This learnable token has additionally the role of a residual which is added to the embedding vectors produced by the multi-head attention. The final constructed embedding vectors are once again normalized using layer normalization, whereafter they are passed through an MLP. In the MLP block, there is no interaction or information transfer between the embedding vectors. Instead, they are transformed within their own dimension of d_{model}. After the MLP, there is another summation through a residual connection.



(a) Interpretable multi-head attention

(b) Scaled dot-product attention

(c) Interpretable transformer-encoder

Figure 2.1.: Transformer-encoder. (**a**) interpretable multi-head attention. (**b**) scaled-dot product attention. (**c**) fundamental part of the ISTRUST model, i.e. the interpretable transformer encoder

2.2.2. VISION TRANSFORMERS

Following the success in NLP, transformers are also being applied in computer vision tasks [24]. Vision transformers, as an alternative to traditional CNNs, have gained popularity due to their improved performance. Passing all pixels of an image into a vision transformer - which is based on the encoder part of the original transformer - to the self-attention mechanism would require computational power that is out of the limits of today's hardware. Therefore, the input image is divided into square patches. Since transformers usually take one-dimensional vectors as an input, the two-dimensional patches are either flattened and linearly projected or are fed into a CNN to obtain one-dimensional embedding vectors of size d_{model} [59]. Since the interpretable multi-head attention as shown in Figure 2.1a does not change the dimensionality of its input, the information present in the patches cannot be encoded into a lower dimensionality. This implies that the contribution of each patch to the outputs cannot be detected. One solution is to flatten each patch and stack those one-dimensional arrays together into one large vector. However, this can quickly become memory-inefficient, time-consuming, and the patches will lose the information about their pixels' relative positions. To address these issues and to keep the relative importance of each patch, another solution is to use a learnable parameter that contains information about their contribution to the architecture. For that reason, a learnable token (see also Figure 2.8) is concatenated to the embeddings, also called a CLS token in classification problems [24]. This learnable token is a one-dimensional vector of size d_{model} with learnable parameters that, similar to the weights in an NN, are updated through backpropagation. The learnable token is a crucial component of the vision transformer as it focuses on capturing the essential information within the image patches that is relevant to the desired output.

Unlike CNN, a transformer is inherently not aware of the location of a patch. In this regard, it was proposed in [23] to use a technique called positional encoding. This method involves adding a one-dimensional vector, dependent on the location, to the embedding vector. Various methods exist for calculating these positional encoding vectors, but a common approach involves using sine and cosine functions [23]. Considering the one-dimensional embeddings with their relative position encoded, they are fed into the transformer-encoder together with the learnable token. In the transformer-encoder, they undergo several interchanged self-attention and MLP layers. In the self-attention layer, the learnable token represented as a query searches for particular features, in our case RUL-related features, in the corresponding patches represented by the keys. The values, on the other hand, contain the information present in a patch that is ideally relevant to extract features with the queries using a dot-product.

Following the transformer-encoder, only the attended embedding vector originating from the learnable token is utilized for further processing, while the embedding vectors from the patches are discarded since their relevant information is encoded into the learnable token. Finally, the embedding vector originating from the learnable token is passed through another MLP with, as output dimension, the number of classes for classification problems or the number of outputs for regression problems.

To sum up, each input image is first divided into patches. Subsequently, patch embedding is applied followed by positional encoding. The transformer-encoder encodes the information pertinent to the specific problem into the learnable token using the interpretable multi-head attention. Following the transformer-encoder, the patch embeddings are discarded and the attended learnable token undergoes an MLP to achieve the final prediction. This is illustrated in Figure 2.8 which is further described in section 2.4

2.3. PROBLEM FORMULATION AND EXPERIMENTAL SETUP

2.3.1. PROBLEM FORMULATION

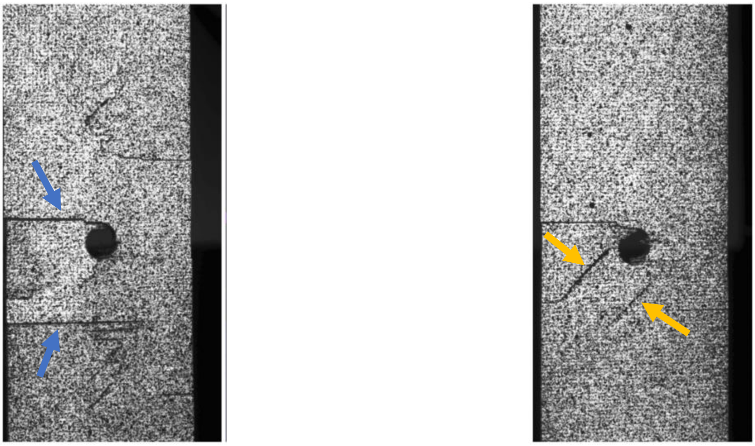
Composite structures exhibit crack growth under fatigue loading through complex mechanisms that differ from traditional metals. Cracks typically initiate at stress concentrations such as fiber-matrix interfaces, voids, or ply terminations. The growth occurs gradually as micro-cracks coalesce, often progressing along the fiber or matrix paths. The rate and direction of crack propagation are highly dependent on the composite layup, fiber orientation, and loading conditions. Delamination between layers is also a common mode of fatigue damage in laminated composites. RUL is closely connected to fatigue crack growth in composite structures because it estimates how much longer a component can perform its intended function before failure. As fatigue loading leads to progressive damage, such as crack initiation, growth, and delamination, these degradation mechanisms reduce the structural integrity over time. By monitoring crack development and damage accumulation, engineers can predict the RUL of a composite part.

To support this, data-driven approaches have emerged as powerful tools for estimating RUL based on observed damage evolution. Given input data comprising sequential images of a composite structure subjected to fatigue loads and exhibiting crack growth, the primary task is to predict the RUL of the structure and estimate the associated uncertainty by interpreting the model's focus within the spatial and temporal domains of the input data. In addition to executing a high-dimensional stochastic regression task, another task is enhancing the trustworthiness of the model's predictions by visualizing the areas of focus. This approach focuses on providing deeper insights into the model's prediction outcomes and improving reliability.

To achieve these tasks, an advanced DL technique based on vision transformers is proposed. The model is further refined to highlight and analyze critical regions and time frames that contribute most significantly to the prediction of RUL. The proposed method provides interpretable insights into the degradation process of the composite structure. Additionally, uncertainty quantification is performed using the MC dropout technique [47], which allows for the assessment of Confidence Intervals (CI) around the predicted RUL.

2.3.2. EXPERIMENTAL SETUP

The experimental data used in this paper is provided by a previous experimental study [13]. The material at hand is a unidirectional Prepreg tape Hexply® F6376CHTS(12K)-5-35. The laminate is manufactured using a hand lay-up of $[0/45/90/-45]_{2s}$ and is cured in an autoclave at a temperature of 180°C and pressure of $9\,bar$ for $120\,min$ as recommended by the manufacturer. The laminate is consequently cut to obtain specimens of $400\,mm$ x $45\,mm$ with an average thickness of $2.28\,mm$. Two examples of specimens can be found in Figure 2.2.



(a) Sample with multiple horizontal (blue arrows) transverse matrix cracks.

(b) Sample with multiple horizontal and diagonal (orange arrows) transverse matrix cracks.

Figure 2.2.: Types of cracks; horizontal-only (**a**) indicated with blue arrows and horizontal-diagonal (**b**) illustrated with orange arrows in the experimental samples.

Subsequently, the specimens are loaded under fatigue using a $100\,kN$ MTS fatigue controller and a fatigue machine, with an average fatigue load of $16.2\,kN$, a stress ratio of $R{=}0.1$, and a frequency of $10\,hz$. The experiments are paused every 500 cycles, during which the load is first allowed to go to the minimum load, after which the load is ramped up to its maximum over a duration of $1\,s$. The load then remains stationary at maximum for $2\,s$. In the middle of this interval, the pictures are taken using two 8-bit "Point Grey" cameras with "XENOPLAN 1.4/23" lenses, placed slightly left and slightly right in front of the specimen. Finally, the load is relieved to minimum over a duration of $1\,s$, after which the fatigue loading continues. More detailed information about the used materials and the experimental setup can be found in [60].

CRACK PROPAGATION MECHANISM AND RUL DEFINITION

During the fatigue test, several damage accumulation mechanisms were observed. Reifsnider et al.[61] proposed a three-stage process for damage accumulation, describing it as a multistate degradation phenomenon. Figure 2.3 illustrates the damage accumulation process in composite structures, while the damage observed specific to our data is shown in Figure 2.2. The process begins with transverse matrix cracking in layers with fibres transverse to the load, which have low stiffness and low stress in loading direction. In the second stage, the process follows the formation of debonding and delaminations at layer interfaces. In the final stage, fiber failures occur, leading to macroscopic failure, also known as EOL. It is important to note that the exact sequence depends on factors such as layup configuration, material properties, manufacturing defects, loading, and environment.

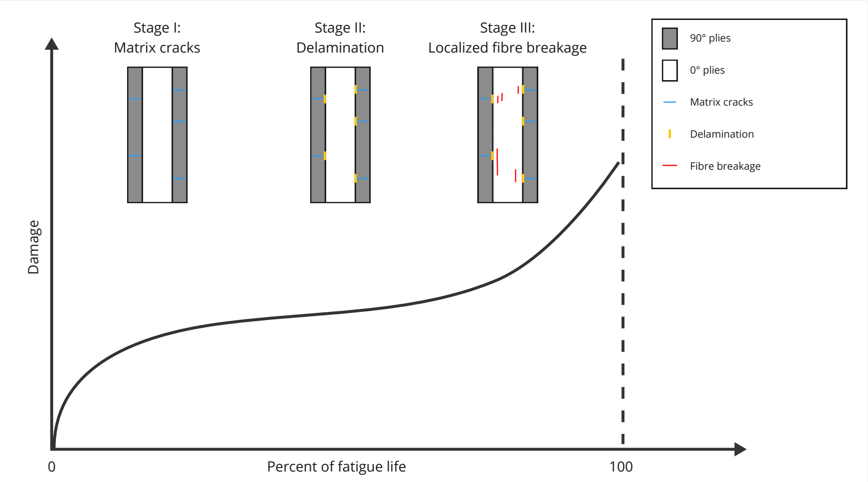


Figure 2.3.: Three stage damage accumulation process in composite structures [61, 62].

Following the fatigue test, the EOL is marked by the specimen breaking in two, rendering it incapable of bearing any further forces. At each timestep, the period remaining until reaching the EOL is referred to as the corresponding RUL at this specific timestep. Typically, RUL refers to a line at 45 degrees with a negative slope.

DATASET SETUP

Once the data is acquired, it undergoes a transformation process to prepare it for input into the ISTRUST model. Our data consists of six different specimens, each consisting of two views taken by the two cameras. Because the images are taken at fixed intervals, the larger the lifetime of the specimen, the more images will be taken. Hence, per specimen, 50 to 150 images are taken, depending on their corresponding total useful life. The raw data is processed as follows:

- For hyperparameter tuning, the dataset is split into a training set, consisting of four specimens, a single validation specimen, and a single testing specimen. The training set is consequently used to train the ISTRUST model, whilst the validation specimen is used to evaluate the performance of the model and adjust the hyperparameters accordingly via a technique called nested cross-validation [63].
- To evaluate the model with tuned hyperparameters, the validation specimen used for the hyperparameter tuning is discarded to avoid data leakage [64]. All specimens with the exception of the validation specimen are consequently used in cross-validation: the model is trained six times with the same hyperparameters,

where each specimen acts as a testing specimen at a certain training iteration. It is important to note that the testing specimens never influence the hyperparameters.

- Rather than using the two different camera views as features, they are split as if they were different specimens. This essentially doubles the length of the dataset, acting naturally as data augmentation and consequently reducing the risk of overfitting.
- The images are sampled from the raw data by utilizing the windowing technique (see section 2.4.3 for more details).
- The current RUL of a sample $\mathbf{x} \in \mathbb{R}^{TxHxW}$ is defined as the distance between the current time step and the EOL of the specimen which is measured at the moment of the latest picture in a specific array of images.

Based on the abovementioned processes, the training set consists of a range of 500-600 samples of 3 sequential images each, corresponding to an array of images, while the testing set contains a range of 90-130 samples with the same number of sequential images. The corresponding total number of training and validating samples depends on the specific specimen chosen as the testing set under the concepts of the cross-validation technique.

2.4. MODEL ARCHITECTURE AND LEARNING PROCESS

The general concept from creating the dataset to predicting RUL with interpretable capabilities is depicted in Figure 2.4.

2.4.1. DATA AUGMENTATION

In order to aid the ISTRUST model in learning useful representations rather than memorizing the data, an augmented training set is created by applying several data augmentation techniques. These techniques are a random resized crop, followed by a random horizontal and vertical flip, and a random rotation of $\pm 5^\circ$ is applied. These augmentations are applied multiple times on the input images x, resulting in an augmented training set that is multiple times larger than the original training set. Importantly, the exact same augmentations should be applied to each image in the temporal domain as well. Doing otherwise would confuse the temporal transformer-encoder since the images would no longer be spatially aligned in the temporal domain.

2.4.2. PATCHING

The input images $\mathbf{x} \in \mathbb{R}^{T_{img}xHxW}$ are split into square patches $\mathbf{x}' \in \mathbb{R}^{T_{img}xP_HxP_WxH_PxW_P}$, where the number of vertical patches P_H is H/H_P, the number of horizontal patches P_W is W/W_P, and H_P, W_P are the horizontal and vertical pixels of each patch, respectively, as shown in Figure 2.6. Following the patching, the spatial domain will refer to P_H and P_W rather than H and W. Note that the spatial domain can also be referred to as P in the case that P_H and P_W are flattened into a single dimension, where P is the total number of patches.

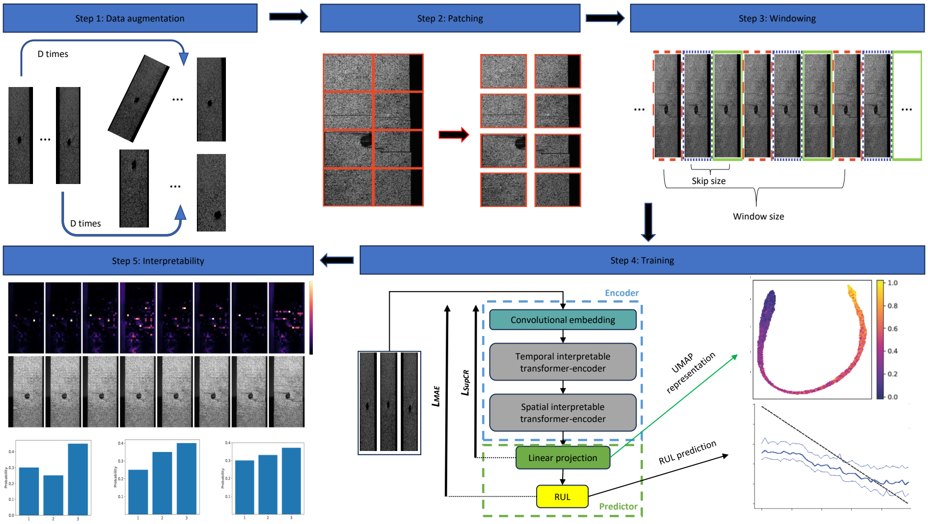


Figure 2.4.: The general concept concerning the task of predicting RUL under uncertainty from raw sequential images by interpreting the relation between RUL and images in the temporal and spatial domains.

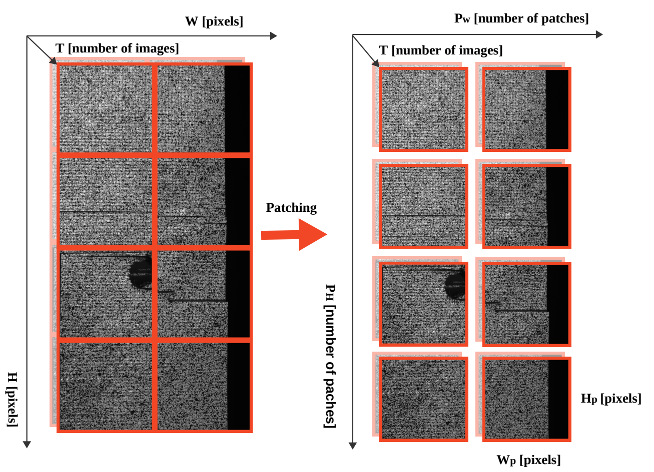


Figure 2.5.: Patching

Figure 2.6.: Illustration of the patching process. The overlapping images represent the temporal dimension T_{img}. For clarity, the image is only separated into a few patches, whilst, in practice, the image consists of significantly more patches, where this amount is limited by the hardware on which the model is trained.

2.4.3. WINDOWING

The images are sampled from the raw data by shifting a window with a size of s_{window} over the sequence of images. For both performance and overfitting reasons, a skip size s_{skip} was used, reducing the number of images in a window without reducing the desired window length, as illustrated in Figure 2.7. Therefore, a final input size of $\mathbb{R}^{T_{img} x H x W}$ is obtained, representing an array of images, where $T_{img} = \frac{s_{window} - 1}{s_{skip} + 1} + 1$ is the number of images, H the height, and W the width of the image. In this work, a skip size s_{skip} of 2 and a window size s_{window} of 7 are used.

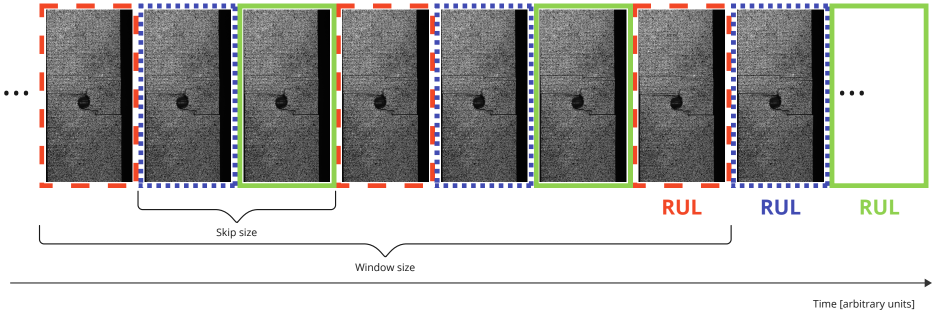


Figure 2.7.: Sampling images from raw data, for one specimen specifically. Three different samples are shown: the images of the first sample are circled with a red dashed line, the second with a blue dotted line, and the third with a green solid line. The horizontal axis represents the time domain, and consequently decreasing RUL of the specimen. The RUL of the sample is represented by the latest image of the corresponding color and is the distance between the current time step and the EOL of the examined specimen. In this work, a skip size s_{skip} of 2 and a window size s_{window} of 7 are used.

2.4.4. TRAINING SETUP

Since the task demands high-dimensional data as input representing sequences of images, a spatiotemporal domain prevails. In the custom vision transformer, a novel approach is introduced by decomposing the spatiotemporal domain into distinct domains, namely the temporal and spatial domains. This decomposition enables the capturing and analysis of the evolution in both domains separately, leading to improved understanding and interpretability in predicting the RUL from raw images.

The *temporal domain* within the custom vision transformer focuses on the manipulation and rearrangement of information along the T_{img} dimension, representing the temporal aspect of the data. This approach helps in identifying critical temporal patterns, visualizing their importance across multiple time steps, and leveraging them for accurate RUL predictions. By treating the spatial dimensions H and W as batch dimensions, i.e. dimensions where the individual samples remain isolated without exchanging any information, the temporal domain becomes a dedicated space for uncovering the temporal evolution of the input data.

In conjunction with the temporal domain, our custom vision transformer incorporates the *spatial domain*, which emphasizes operations within the spatial dimensions. By exclusively considering spatial relationships and patterns, valuable insights can be extracted into the spatial distribution of image features and their impact on RUL prediction. Through this spatial analysis, a deeper understanding of the material's health status and the location of the damage is provided.

The resulting model architecture used to predict the RUL from the raw images is illustrated in Figure 2.8. The temporal and spatial domains represent the first and second interpretable transformer-encoder, respectively. The rationale for prioritizing the decomposition of the temporal domain as a primary consideration, followed by the subsequent analysis of the spatial domain, stems from the inherent emphasis on identifying the most pertinent image and subsequently directing the focus towards specific regions within those images. The individual modules and submodules in the figure will be described in more detail throughout the following subsections. Unless explicitly stated otherwise, operations in one domain exclude operations in the other domain. In other words, when operations are performed in the spatial domain, the temporal domain acts solely as a batch dimension and vice versa.

Provided that the dataset is fairly small - only six specimens, with 50 to 150 images each - a model using multi-head attention like the proposed one has many difficulties learning useful patterns from the data rather than overfitting to the training set, compromising the performance on the testing one. The following techniques were applied to increase the ISTRUST model's performance.

CONVOLUTIONAL EMBEDDING

Following the patching layer, the convolutional embedding layer, shown in Figure 2.10, aims to encode the two-dimensional information of all patches into a one-dimensional embedding vector. It achieves this by performing four stepwise convolutional operations, followed by a batch normalization and GeLU activation function. The convolutional layers essentially reduce the size of the square patch size from shape $H_P x W_P x 1$, since the images are grayscale, to shape $1x1xd_{model}$. Consequently, these $1x1$ dimensions are discarded, with only the features remaining. Subsequent to the convolutional embedding the embedding vectors $z \in \mathbb{R}^{T_{img} x P_H x P_W x d_{model}}$ are obtained, having successfully reduced the 2D patches to a one-dimensional vector. It is important to note that, so far, there has been no information exchange or interaction between the patches in either the temporal or spatial domain. The convolutional embedding operates on each patch independently without considering the spatiotemporal relationships.

TEMPORAL POSITIONAL ENCODING

To produce the outputs of the temporal position encoding, i.e. the temporal positional encoded embeddings which should be fed to the temporal transformer-encoder, the original positional encoding equation using sine and cosine functions proposed in [23]

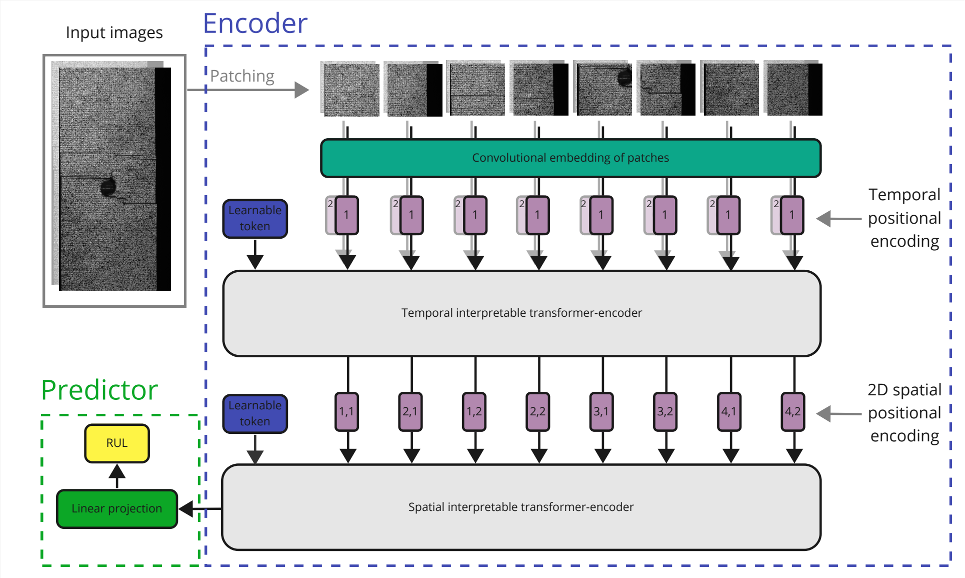


Figure 2.8.: ISTRUST model's architecture. Our proposed architecture is based on Vision transformer. The images are first divided into patches, followed by an embedding layer and temporal positional encoding. The patch embedding is followed by the temporal interpretable transformer-encoder, which is shown in more detail in Figure 2.1. The temporal interpretable tranformer-encoder is followed by 2D spatial positional encoding. Subsequently, the spatial encoded embeddings are passed through the spatial interpretable transformer-encoder. Finally, the attended learnable token is sent through an MLP to adjust the dimension for the output, i.e. the RUL predictions. Visualizing the temporal and spatial attention weights of the corresponding interpretable transformer-encoder offers valuable insights into the relation between RUL and the speed and size of the structure's cracks, respectively.

was used:

$$TPE(t, 2i) = sin\left(t \cdot e^{\left(2i \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.2}$$

$$TPE(t, 2i+1) = cos\left(t \cdot e^{\left((2i+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.3}$$

where t represents the moment in time and $t \in [0, T_{img})$, $2i$ represents an even index and $2i+1$ represents an odd index in the embedding vector where $\{2i, 2i+1\} \in [0, d_{model})$. These temporal positional encoding vectors TPE are added to the embeddings z to obtain the temporal positional encoded embeddings $z^{TPE} \in \mathbb{R}^{P_H x P_W x T_{img} x d_{model}}$.

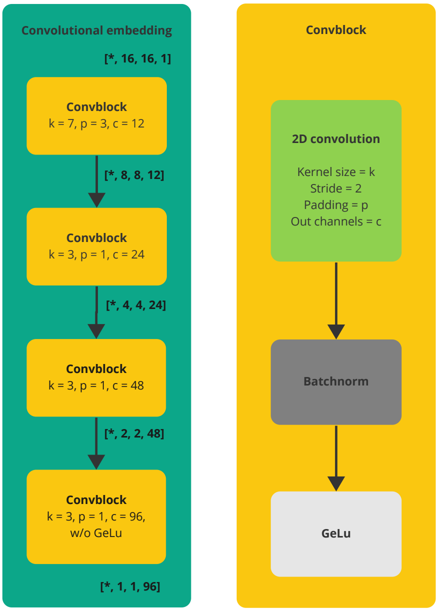


Figure 2.9.: Convolutional embedding

Figure 2.10.: Illustration of the convolutional embedding process. The convolutional embedding layer is shown on the left, where the Convblock is illustrated on the right. The shape of the patches is illustrated with square brackets, where $*$ represents an arbitrary number of leading dimensions acting as batch dimensions, which are being left out for clarity. This figure specifically illustrates the convolutional embedding for a 16x16 patch; a smaller patch size would require fewer Convblocks and vice-versa.

TEMPORAL INTERPRETABLE TRANSFORMER-ENCODER

Following the temporal positional encoding, the temporal positional encoding vectors z^{TPE} are passed through the temporal transformer-encoder where the spatial dimensions P_H and P_W act as batch dimensions. Consequently, information will only be transferred in the temporal domain and not the spatial domain. Traditional vision transformers have multiple self-attention layers being interchanged with MLPs, essentially putting multiple transformer-encoder blocks in series. To achieve interpretability, however, our novel interpretable transformer-encoder shown in Figure 2.1c consists of two attention mechanisms, namely the multi-head self-attention and the multi-head token-attention, followed by an MLP. The interpretable transformer-encoder block is applied only once in each domain.

After normalizing the temporal positional encoding vectors z^{TPE} using layer normalization, they undergo the multi-head self-attention layer. However, unlike the conventional vision transformers, the learnable token does not pass through the multi-head self-attention layer. The purpose of the self-attention layer is to facilitate information exchange among the patches. The self-attention layer is followed by the token-attention layer, which is a modified attention layer based on the self-attention in vision transformers [24]. The multi-head token-attention employs the interpretable multi-head attention similar to the self-attention layer (see section 2.2.1). Nevertheless, the queries, keys, and values do not come from the same sequence of embeddings. While the queries originate from the learnable token, the keys and values originate from the attended patches that come from the previous self-attention layer, hence the name. The objective of this layer is to encode the information from all embedded patches in the temporal domain into a single vector. As previously mentioned in the description of the interpretable transformer-encoder (see Figure 2.1c), the attention layers are finally followed by layer normalization and an MLP, which in turn, come after another residual connection giving the final temporal encoded embedding vector $z_{TE} \in \mathbb{R}^{P_H x P_W x d_{model}}$, representing the output of the temporal interpretable transformer-encoder.

SPATIAL POSITIONAL ENCODING

Since an image is two-dimensional, utilizing the previous positional encoding technique responsible for the temporal positional encoding (Equation 2.2) would require flattening the spatial domain $[P_H,P_W]$ to $[P]$ and, thus, not differentiating between the horizontal and vertical dimensions. Therefore, an alternative positional encoding technique is performed that incorporates both spatial dimensions, by reserving half of the embedding vector for the positional encoding in the height direction, and the other half for the width direction:

$$SPE(h,2i) = sin\left(h \cdot e^{\left(2i \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.4}$$

$$SPE(h,2i+1) = cos\left(h \cdot e^{\left((2i+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.5}$$

$$SPE(w,2j) = sin\left(w \cdot e^{\left(2j \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.6}$$

$$SPE(w,2j+1) = cos\left(w \cdot e^{\left((2j+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2.7}$$

where h and w represent the position in the height dimension P_h and width dimension P_W, respectively, with $h \in [0,P_H)$ and $w \in [0,P_W)$, 2i and 2j represent an even index, and 2i+1, 2j+1 represent an odd index in the embedding vector where $\{2i,2i+1\} \in [0,d_{model}/2)$ and $\{2j,2j+1\} \in [d_{model}/2,d_{model})$. Finally, the temporal positional encoding TPE is added to the temporal encoded embeddings z_{TE} to obtain the spatial positional encoded embeddings z_{TE}^{TPE}.

SPATIAL INTERPRETABLE TRANSFORMER-ENCODER

Before the spatial positional encoded embeddings $z_{TE}^{TPE} \in \mathbb{R}^{P_H x P_W x d_{model}}$ can be fed to the spatial interpretable transformer-encoder, the spatial domain has to be flattened to obtain $z_{TE,f}^{TPE} \in \mathbb{R}^{P,d_{model}}$. Consequently, the spatial interpretable transformer-encoder works entirely similar to the temporal counterpart, except that P_H and P_W -which were previously acting as batch dimensions- are no longer present, and that the T_{img}-dimension is replaced by the P-dimension. The corresponding output is consequently the spatial encoded embedding $z_{SE} \in \mathbb{R}^{d_{model}}$.

LINEAR PROJECTION

The spatial interpretable transformer-encoder is finally followed by a linear projection layer. It encodes the spatial encoded embedding z_{SE}, from $\mathbb{R}^{d_{model}}$ to $\mathbb{R}$ by means of a single linear projection layer followed by a ReLU activation function, giving the predicted RUL.

WEIGHT INITIALIZATION

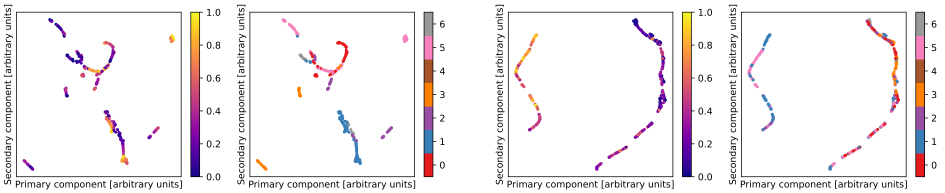
On each image, although damage can be detected diagonally, the majority is horizontal. This is because most cracks initiate in the 90° ply direction as it is orthogonal to the loading direction. Using the fact that most of these cracks are horizontal, the convolutional embedding layer can be engineered to capture these cracks. Typically, the weights of the kernels, also called filters, in any convolutional layer are sampled from either a uniform or normal distribution, in this case, a normal distribution, giving no preference to horizontal or vertical features, as shown in Figure 2.11a. In this case however, although still sampling from a normal distribution, no variance is allowed in the horizontal direction, thus forcing the kernels to initially filter out vertical features and capture only horizontal features as shown in Figure 2.11b, significantly speeding up the training process and consequently increasing the model's performance. Note that this only happens in the initial stages of the model's training and the kernels are not constrained, hence horizontal variance can still become detectable throughout the learning process.

(a) Regular kernel weight initialization (b) Custom kernel weight initialization

Figure 2.11.: Custom kernel weight initialization. (**a**) demonstrates the weights of CNN kernels, which are initialized by sampling from a normal distribution and allowing variance in all directions, whilst (**b**) visualizes the custom CNN weight initialization where no variance is allowed in the horizontal direction, but the weights are still being sampled from a normal distribution.

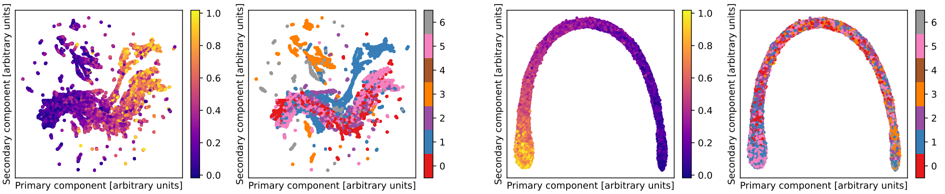
CONTRASTIVE LEARNING

Contrastive learning is a semi-supervised technique that aims to learn useful representations by contrasting similar and dissimilar pairs of data samples [49, 50]. In



(a) UMAP representation of z^{SE} for the training set trained directly on the RUL prediction using $\mathcal{L}_{MAE}$.

(b) UMAP representation of z^{SE} for the training set trained with contrastive learning using $\mathcal{L}_{SupCR}$.



(c) UMAP representation of z^{SE} for the augmented training set trained directly on the RUL prediction using $\mathcal{L}_{MAE}$.

(d) UMAP representation of z^{SE} for the augmented training set trained with contrastive learning using $\mathcal{L}_{SupCR}$.

Figure 2.12.: UMAP representation of the spatial encoded embeddings z^{SE}. In each subfigure, the left-hand image represents the primary and secondary component of the UMAP representation of z^{SE}, supplemented by the normalized ground truth RUL as a continuous colorbar on the right. The right-hand image also represents the primary and secondary components, supplemented by the sample number of the relevant specimen as a discrete colorbar on the right. In these images, it can be seen that the spatial encoded embeddings z_{SE} resulting from the spatial transformer-encoder part still contain information related to the relevant specimen that is unnecessary for the RUL prediction. This means that each specimen has its own location in the UMAP representation. Consequently, the final linear projection layer can overfit to this location in the UMAP representation, and thus to the relevant specimen and/or data augmentation. Therefore, it is desired to force samples with a similar RUL to the same location in the UMAP representation, which highlights the importance of utilizing contrastive learning.

this approach, the model is trained to distinguish between positive pairs, which are similar in some way, and negative pairs, which are dissimilar. The key idea is to maximize the similarity between positive pairs while minimizing the similarity between negative pairs. This is typically achieved by training the encoded embedding vector (the output of the tranformer-encoder), as depicted in Figure 2.8. By optimizing the encoder to pull similar samples closer together and push dissimilar samples apart, contrastive learning enables the ISTRUST model to capture important features and patterns that are relevant to the task at hand. This process allows the model to learn generalized representations.

To visualize the embeddings obtained from the encoder, the UMAP representation is employed. The UMAP representation provides a low-dimensional visualization of the encoded embedding vector in terms of primary and secondary components. When training the whole model, i.e. both the encoder and the predictor, directly with the Mean Absolute Error (MAE) loss function $\mathcal{L}_{MAE}$, it fails to filter out information related to the specific specimen, as shown in UMAP representation in Figure 2.12a and the specific data augmentation, as shown in Figure 2.12c. In these images, it can be seen that the spatial encoded embeddings z_{SE} resulting from the spatial transformer-encoder part still contain information related to the relevant specimen that is unnecessary for the RUL prediction. This means that each specimen has its own location in the UMAP representation. Consequently, the final linear projection layer can overfit to this location in the UMAP representation, and thus to the relevant specimen and/or data augmentation. Therefore, it is desired to force samples with a similar RUL to the same location in the UMAP representation, which highlights the importance of utilizing contrastive learning.

The work in [48] was the first to propose contrastive learning in supervised regression problems based on the work of [49, 50] in classification problems. However, they recommended the application of data augmentation at every training iteration twice, aiming to ensure convergence by always having two samples with the same RUL, also called positive samples. Because of the computational expense related to performing the data augmentation at every iteration, data augmentation is applied D times on the training set before the training process, creating the augmented training set. Using this technique, no significant difference was observed in the current implementation compared to sampling at every training iteration. Moreover, as stated previously, sampling at every iteration gives a large bottleneck in terms of computational performance. Thereby, the following loss function is used where we replaced the term $2N$ from the fundamental equation with the term N, hence:

$$\mathbb{L}_{SupCR}=-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N-1}\sum_{j=1,j\neq i}^{N} log\frac{e^{L^2\text{-}dist(z_{SE,i},z_{SE,j})/\tau)}}{\sum_{k=1}^{N}\mathbb{1}[k\neq i,\, L_{MAE}(\tilde{y}_i,\tilde{y}_k)\geq L_{MAE}(\tilde{y}_i,\tilde{y}_k)]\, e^{L^2\text{-}dist(z_{SE,i},z_{SE,k})/\tau)}} \tag{2.8}$$

where $\mathcal{L}_{SupCR}$ is the supervised contrastive regression loss, N is the batch size, L^2-$dist$ is the L^2 distance between the two input vectors, is the spatial encoded embedding, $\tilde{y}$ the corresponding ground truth RUL, $\mathbb{1}$ is true when the statement in brackets is satisfied and zero otherwise, τ is the temperature hyperparameter of the softmax function, and L_{MAE} is the MAE between the two input targets. Since the model fails to filter out specimen-related information (Figure 2.12a) rather than augmentation-related information (Figure 2.12c), sampling at every training iteration should be avoided as claimed in [48]. Note that this loss function is applied on z_{SE}, and thus remains untrained during the contrastive learning stage, resulting in the absence of utilizing the last layer responsible for the RUL prediction. To subsequently consider the final linear projection layer, it is trained in a second stage whilst fixing the weights of all layers obtained in the first stage via contrastive learning.

2.4.5. INTERPRETABILITY: INFORMATION flow

This section explains the information flow in both the temporal and the spatial transformer-encoder by analyzing the attention weights. For simplicity, a single attention head is assumed. At the end of the section, it will show that this can easily be extrapolated to the interpretable multi-head attention. Besides, since the temporal and spatial transformer-encoder work in the exact same way, only in a different domain, no distinction will be made between them during the following explanation of the information flow. The patches in the relevant domain will merely be referred to as patches, meaning that either all of them are in the temporal domain T_{img} where the spatial domain acts as a batch dimension and does not contribute to any mathematical operations in any way, or they are in the spatial domain P.

The transfer of information in the self-attention mechanism can be interpreted by analyzing the attention weights computed in self-attention and token-attention. In self-attention, the attention weights are calculated by the means of a matrix multiplication between the query Q_{self} and key K_{self} matrices, followed by a softmax operation, as expressed in Equation 2.1. This computes the correlation between each query and all keys, representing the relationship between the patches. The resulting attention weights, represented by the matrix A_{self}, indicate how much information is transferred between patches. The actual information transfer happens when these attention weights are multiplied by the values, resulting in the intermediate self-attended patches Z'_{self}.

Similarly, in the token-attention, there is only one query originating from the learnable token. The attention weights are computed by attending the query Q_{token} to the attended patches K_{token} originating from the self-attention. The resulting attention weights A_{token} determine the importance of the attended patches for the RUL prediction. Multiplying these attention weights with the values V_{token} originating from the attended patches yields the encoded embedding.

Likewise, by tracing back the information present in the encoded embedding through the attention weights, it can accurately be determined how much information was taken from each patch to obtain the encoded embedding. Knowing that V_{token} originates from Z'_{self}, the flow of information in the entire interpretable transformer-encoder is represented by matrix $\mathbf{A}$:

$$\mathbf{A} = A_{token} A_{self} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n_{keys}} \end{bmatrix}_{token} \begin{bmatrix} a_{1,1} & \cdots & a_{1,n_{keys}} \\ \vdots & \ddots & \vdots \\ a_{n_{queries},1} & \cdots & a_{n_{queries},n_{keys}} \end{bmatrix}_{self} = \begin{bmatrix} \tilde{a}_1 & \cdots & \tilde{a}_n \end{bmatrix} \tag{2.9}$$

where $\mathbf{A}$ represents the matrix containing how much information has been taken from each patch to arrive at the encoded embedding and $\tilde{a}_i$ represents how much information has been taken from the i^{th} patch.

As can be seen in Figure 2.1a, the attention weights in the multi-head attention are averaged before being multiplied by the values. The same can thus be done for the attention weights in both the self-attention and the token-attention. It is important that these attention weights are averaged before applying Equation 2.9 since the head i of the self-attention unmistakably has no correlation with the head i of the token-attention. It can thus be concluded that it is possible to interpret the information flow of both the temporal and spatial domains separately. This explanation is achieved by reducing the attention weights in the multi-head self- and token-attention such that each patch has a single weight in the relevant domain. The resulting attention weights directly determine how much information from each patch contributes to the final prediction.

2.5. EXPERIMENTAL RESULTS AND DISCUSSION

To obtain the experimental results, the ISTRUST model was trained in PyTorch in a two-stage process on an Nvidia RTX4080 16GB GPU. Because estimating the time complexity theoretically is challenging, it was approximated by performing forward passes with different image heights, widths, frame lengths, layers, number of heads in multi-head attention, and dimensions of the encoded embeddings. The elapsed time to perform a forward pass with $1\times$, $2\times$, $4\times$, and $8\times$ of the above sizes (all of them were increased simultaneously) was $0.21, 0.41, 0.96, 2.69$ seconds, respectively. This means that the time is increasing somewhere between linearly and quadratically with these sizes. In particular, the time increases approximately by a factor of 1.2, since $\frac{2.69}{0.96}/\frac{0.96}{0.41}=\frac{0.96}{0.41}/\frac{0.41}{0.21}\simeq 1.2$. Consequently, the time complexity is estimated approximately to be $O(n^{1.2})$.

Before evaluating the model's performance, it was fine-tuned the hyperparameters by training on the training dataset and adjusted the hyperparameters according to the performance on the validation dataset. The resulting hyperparameters are given in Table 2.1. During the first stage, we trained the encoder, which encompasses the entire model as outlined in Figure 2.8, with the exception of the final linear projection layer. The first stage of training was conducted employing contrastive learning, utilizing $\mathcal{L}_{SupCR}$, and SGD with a momentum of 0.9 on the augmented training set. The weights were initialized according to the guidelines provided in [65]. The first stage training process was approximately 30 to 40 minutes per fold, depending on the rate of convergence of the specific fold in the cross-validation process.

Table 2.1.: Values of the parameters resulting from the hyperparameter tuning on the validation set. All values are dimensionless.

Corresponding	Parameter	Values	Parameter	Values for $\mathcal{L}_{SupCR}$	Values for $\mathcal{L}_{MAE}$
Dataset	s_{skip}	2	Epochs	30	6
	s_{window}	7	Batch size	32	32
	T_{img}	3	Learning rate embedder	2.00×10^{-2}	0
	H	640	Learning rate predictor	-	1.00×10^{-3}
	W	320	Dropout model	0.3	0.3
ISTRUST model	d_{model}	96	τ	2	-
	d_k	16			
	$H_P = W_P$	16			
	P_H	40			
	P_W	20			
	P	800			
	n_{heads}	6			

2.5.1. MODEL PERFORMANCE, OPTIMAL AND SUBOPTIMAL RUL PREDICTIONS

Through experimentation on the validation set, it was discovered that the optimal encoder performance following the contrastive learning stage was attained by utilizing the encoder's state at the lowest non-augmented training loss. The resulting epochs for which the embedder states were taken are given in Table 2.2. The UMAP representation

Table 2.2.: Epochs for which the encoder and predictor states were taken. The epochs concerning the encoder correspond to the contrastive learning stage, while the epochs regarding the predictor represent the second training stage, i.e. the learning of the final linear projection layer. The total losses in the rightmost column are calculated using the corresponding specimens in each fold.

Specimen	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	Total
Epochs (encoder)	27	9	25	8	17	6	-
Epochs (predictor)	5	5	5	5	5	5	-
Training $\mathcal{L}_{MAE}$	0.13	0.15	0.15	0.16	0.14	0.17	0.15
Testing $\mathcal{L}_{MAE}$	0.14	0.41	0.15	0.31	0.29	0.21	0.26

of the augmented training set using contrastive learning is depicted in Figure 2.12b, accompanied by the non-augmented training set in Figure 2.12d, which was not included in the training set during the contrastive learning stage. These representations reveal that the encoder is no longer discriminating between the specimens based on the applied augmentation or the relevant specimen, with the primary variation observed in the UMAP representation relating to the ground truth RUL. This serves as the initial validation of the ISTRUST model's performance, indicating that the contrastive learning was successful since the model no longer overfits the specific specimen or augmentation and thus correctly filters out spurious information.

Following the contrastive learning stage, the state of the encoder was frozen and solely the predictor was trained using SGD with a momentum of 0.9 and $\mathcal{L}_{MAE}$ on the non-augmented training set for six epochs, with an observed training time of fewer than two minutes. The rationale for choosing such a small number of epochs is that the second learning stage is a much simpler process than the first stage of training. Consequently, during the hyperparameter tuning, the regression layer that exists in the second stage starts to overfit after some epochs, due to the high dimensionality of the spatial encoded embedding vector. Hence, the training is stopped after six epochs for optimal performance.

To ensure robustness in our testing, we utilized cross-validation with a total of six specimens. The RUL predictions under uncertainty utilizing MC dropout (95% CI) and the associated loss curves for specimens classified as successful can be found in Figure 2.13, while Figure 2.14 contains the corresponding results for specimens with suboptimal performance. Discrimination between optimal and suboptimal performances was guided by two heuristics: (i) whether RUL showed a rational decrease as it approached the EOL, and (ii) whether the spatial attention focused on damaged or noisy areas. The resulting losses are given in Table 2.2. Notably, across all specimens, the RUL profile exhibited a transitional pattern, characterized by an initial phase with an almost stable, flat slope, followed by a distinct transition to a steeper slope, signifying a sudden change in the model's processing of the input. This observation alongside the optimal and suboptimal performances will be explained through the interpretation of the ISTRUST model, i.e. via the attention weights, in the following subsection. Finally, it is worth noting that the corresponding uncertainties originate from the model itself rather than the data. This explains their persistence even close to the EOL condition where

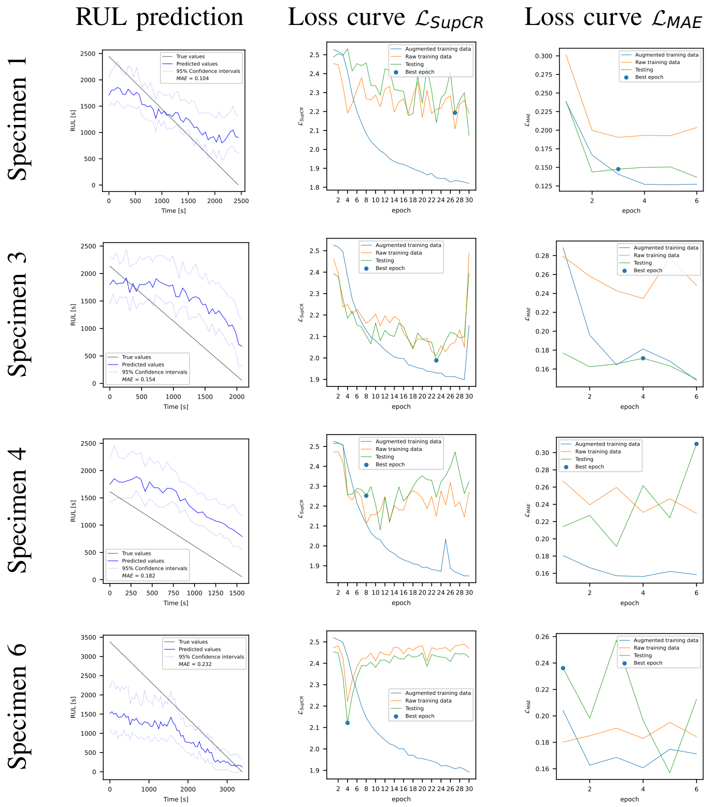


Figure 2.13.: Cross-validation of successful predictions of RUL. Across each specimen, a different fold of the cross-validation is given. The RUL predictions on the testing specimens are provided in the subfigures to the left-most part of the figure. The rest of the subfigures represent the loss curves.

they should have been neglected.

2.5.2. EXPLAINING THE RESULTS - MODEL INTERPRETATION

To explain the successful and suboptimal predictions, both the temporal and spatial attention weights originating from the multi-head self-attention and the multi-head token-attention were merged using Equation 2.9, ensuring that each patch in the relevant domain has only one weight. The temporal attention weights, obtained utilizing the same equation, were further reduced by averaging over the spatial domain, with each temporal attention weight being weighted according to the corresponding spatial attention weight. It is worth noting that all the displayed attention weights correspond to testing specimens that were not part of the training sets in their respective folds.

The spatial attention weights at the EOL for three selected specimens are presented in Figure 2.15. In Figure 2.15a and Figure 2.15b, the model successfully focuses on

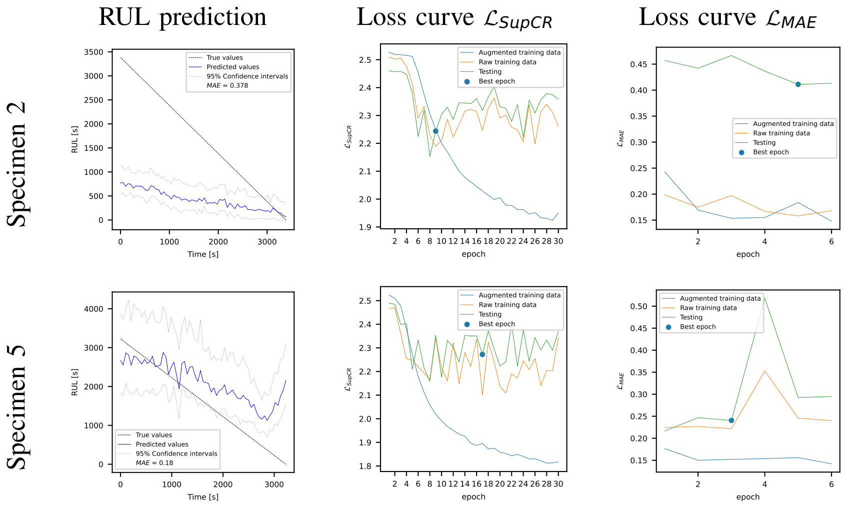
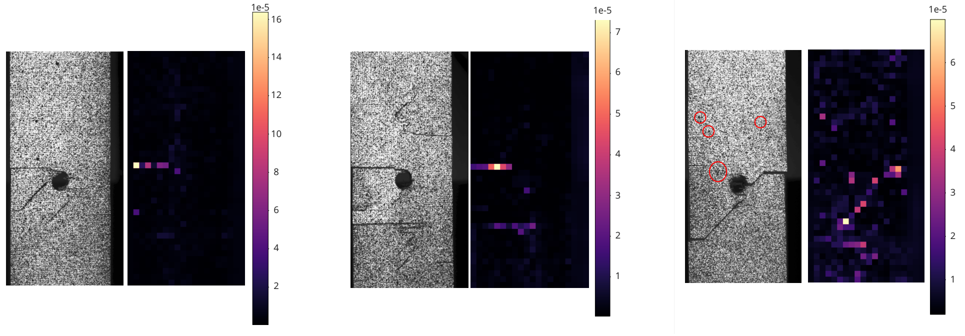


Figure 2.14.: Cross validation of suboptimal predictions of RUL alongside the loss curves depicted row-wisely for each specimen.

locations with damage near the EOL, which validates the desired results for specimens 3 and 6. In Figure 2.16, the spatial and temporal focus evolution of the ISTRUST model throughout the entire life of specimens 4 and 5 are depicted. In Figure 2.16a, it is evident that initially, the model rationally fails to capture the minor damage present in the specimens and instead focuses on spurious parts at random locations as the damage still remains indistinguishable. Consequently, the initial RUL values for each specimen exhibit a relatively consistent range, indicating that the model primarily relies on the average RUL derived from the entire dataset during its initial predictions. Nevertheless, as the damage accumulates, cracks start being detected by the model successfully, which focuses only on the important parts of the image, leading to an accurate decrease in the predicted RUL. Similar observations are present for specimens 1, 3, and 6 shown in Figure A.1 and Figure A.2. For these specimens there was significant cracks present, resulting in a successful RUL prediction. Furthermore, an examination of the temporal attention weights in the same figure reveals that the ISTRUST model reasonably prioritizes the latter images over the earlier ones. Nevertheless, the model does not entirely discard the initial ones. This can be attributed to the model's need for the earlier images to estimate the speed of damage accumulation while relying on the last image to assess the severity of damage in the current state.

In order to explain the suboptimal predictions of specimens 2 and 5, the spatial attention weights of specimen 5 at the EOL are shown in Figure 2.15c. It can be acknowledged that for this specimen, the model also captures the cracks, despite the less prevalent damage and the relatively poorer RUL predictions. This is because of two reasons. Firstly, due to the attention weights that focus on some spurious parts, circled in red. These are typically black dots, which distract the spatial attention from



(a) Spatial attention map of selected successful predictions for specimen 3

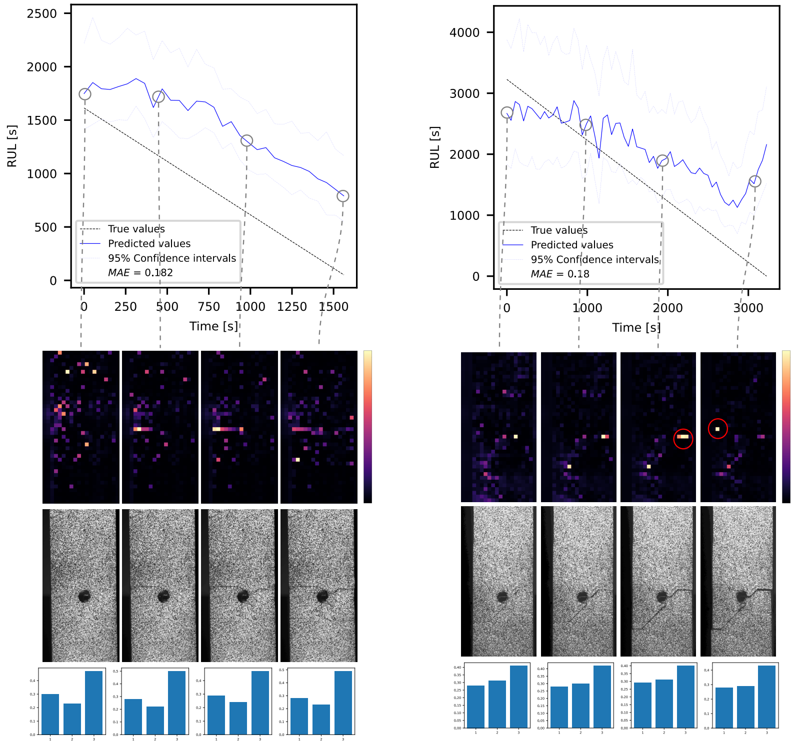
(b) Spatial attention map of selected successful predictions for specimen 6

(c) Spatial attention map of selected suboptimal predictions for specimen 5. The red circles indicate artifacts like black dots correlating to the spatial attention weights.

Figure 2.15.: Spatial attention weights of selected testing specimens at the EOL condition. The displayed colormaps indicate the spatial attention weights, accompanied by the input image on their left. The color within the colormap, along with the accompanying colorbar on the right, indicates the absolute magnitude of the attention weights. Subfigures (**a**) and (**b**) depict the spatial attention weights of specimens 2, 5 for which the results were deemed successful. Figure (**c**) shows the spatial attention weights of specimen 3, which is classified as suboptimal.

the actual damage. In this case, the model gratuitously focuses on additional parts of the image, which confuses its RUL estimation. This can be efficiently seen in Figure 2.16b in the last image representing a sample close to the EOL, where the model additionally considers some spurious parts as damage, thus violently changing its RUL estimation. Secondly, for specimen 5 specifically, significant cracks are observed at a 45-degree angle, a phenomenon not present in the other specimens. While the model does detect these cracks, it does not effectively correlate them with a reduction in the RUL curve. This is because similar cracks were not encountered during the training phase, leading to an unexpected shift in the RUL trajectory. Nevertheless, the RUL predictions in Figure 2.14 still exhibit a negative slope in general (with the only exception being the specimen's 5 latest RUL predictions), specifying that the ISTRUST model indeed captured, but underestimated the extent of the damage. Similar observations can be noted for specimen 2 in Figure A.1b. Consequently, it can be concluded that there is a noticeable correlation between the accuracy of the spatial attention weights, the severity of the visible damage, and the accuracy of the RUL prediction.

It should be acknowledged that even though the model managed to capture the majority of the damage, it only captures the corresponding one that is relevant for



(a) Specimen 4 (b) Specimen 5

Figure 2.16.: Evolution of spatial and the corresponding temporal attention weights of testing specimens 4 (a) and 5 (b). Each figure consists of three parts. The top graph represents the predicted RUL for each specimen. The colormaps displayed beneath the graph illustrate the changes in spatial attention weights over time. Additionally, for each attention map, the last input image can be found below each attention map. The color within the colormap, along with the accompanying colorbar on the right, indicates the relative magnitude of the attention weights. The bottom graph corresponds to the temporal attention weights of each sequence of images (here, only the last and most important image of each sequence is shown).

the RUL prediction. As a result, *the proposed ISTRUST model can currently not be accurately used as an anomaly detection method.* However, the shown results are promising, and it should be further investigated whether the attention weights can be leveraged for anomaly detection by modifying the training setup, the size of the dataset, or the model's architecture.

Lastly, the proposed model encounters the primary limitations associated with the DIC technique and the camera systems. Regarding DIC analysis, the primary limitation encompasses the extended computational time required for post-processing DIC data to extract strain fields, which hinders the application of this technique in real-time scenarios. Even upon extracting the strain field, post-processing fails to accurately identify high-damage areas due to the inability to measure substantial deformations and the corresponding strains [66]. Furthermore, the quality of the speckle pattern applied to the surface and how well it is captured by the camera can limit the feature extraction accuracy. Applying the model directly to raw images mitigates the risk that an inappropriate speckle pattern may introduce.

2.5.3. THE ROLE OF TEMPORAL ATTENTION

The role of spatial attention is well understood; however, the interest in temporal attention is less evident. To address this, a parametric study was conducted focusing on the key hyperparameters associated with temporal attention, specifically skip size (s_{skip}) and window size (s_{window}). Due to extensive memory requirements, the values were selected to ensure that the number of images (T_{img}) did not exceed three. Figure 2.17 illustrates examples of temporal attention weights and their behavior with varying hyperparameter values when testing specimen 1. Notably, when T_{img}=3, the final image in the sequence contributes more to RUL predictions than the preceding images. Conversely, when T_{img}=2, both images hold nearly equal importance for RUL prediction. This observation can be attributed to the fact that a sequence of at least two images allows the model to capture the crack propagation speed, which is closely related to RUL. This finding underscores the importance of temporal attention weights and utilizing sequential images rather than a single image as input.

Finally, for the sake of comprehensiveness, the impact of these hyperparameters on prognostics is presented in Table 2.3. It is evident that the prognostics are significantly influenced by these hyperparameters. When more images are skipped, the model's efficiency drops significantly. Having small or large window sizes negatively affects the model's performance as well.

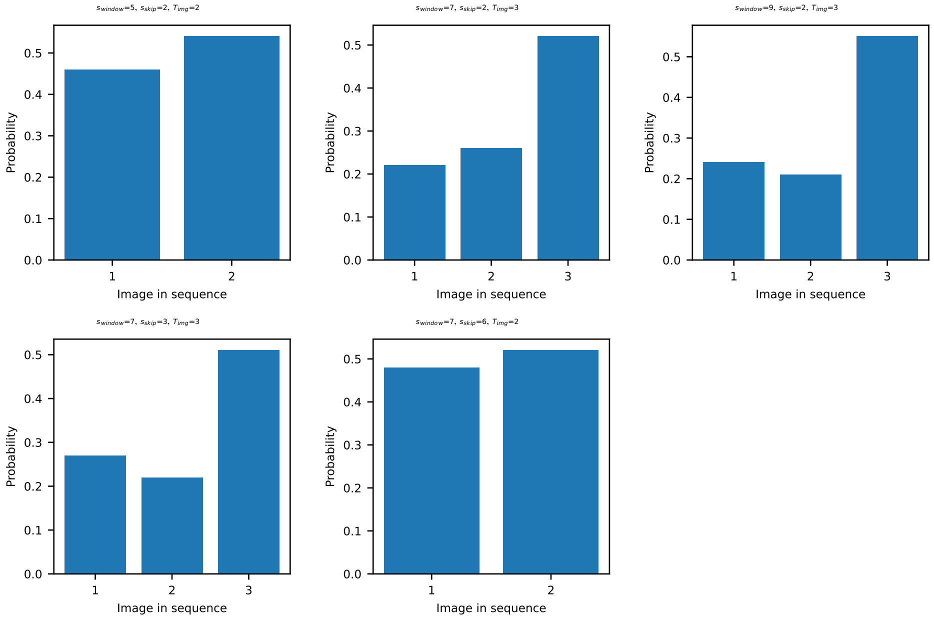


Figure 2.17.: Temporal attention weights when testing specimen 1 with varying hyperparameter values that affect the time domain.

Table 2.3.: Parametric study of the hyperparameters s_{window} and s_{skip} and their effects on prognostics using the corresponding specimens in each fold. The data shows that when more images are skipped, the model's efficiency drops significantly. Having small or large window sizes negatively affects the model's performance as well.

Examined values	Specimen	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	Total
s_{window}=5, s_{skip}=2, T_{img}=2	Epochs (encoder)	27	17	17	16	22	28	-
	Epochs (predictor)	6	2	4	5	5	5	-
	Training $\mathcal{L}_{MAE}$	0.17	0.14	0.17	0.20	0.20	0.21	0.18
	Testing $\mathcal{L}_{MAE}$	0.16	0.42	0.19	0.32	0.30	0.31	0.28
s_{window}=7, s_{skip}=2, T_{img}=3	Epochs (encoder)	27	9	25	8	17	6	-
	Epochs (predictor)	5	5	5	5	5	5	-
	Training $\mathcal{L}_{MAE}$	0.13	0.15	0.15	0.16	0.14	0.17	**0.15**
	Testing $\mathcal{L}_{MAE}$	0.14	0.41	0.15	0.31	0.29	0.21	**0.26**
s_{window}=9, s_{skip}=2, T_{img}=3	Epochs (encoder)	27	23	20	12	27	7	-
	Epochs (predictor)	3	2	6	1	5	6	-
	Training $\mathcal{L}_{MAE}$	0.25	0.18	0.21	0.22	0.20	0.23	0.22
	Testing $\mathcal{L}_{MAE}$	0.29	0.42	0.20	0.32	0.29	0.21	0.28
s_{window}=7, s_{skip}=3, T_{img}=3	Epochs (predictor)	5	3	5	2	5	6	-
	Training $\mathcal{L}_{MAE}$	0.24	0.17	0.19	0.18	0.20	0.21	0.20
	Testing $\mathcal{L}_{MAE}$	0.25	0.43	0.17	0.31	0.32	0.31	0.30
s_{window}=7, s_{skip}=6, T_{img}=2	Epochs (predictor)	3	2	6	1	5	6	-
	Training $\mathcal{L}_{MAE}$	0.28	0.22	0.26	0.27	0.23	0.28	0.26
	Testing $\mathcal{L}_{MAE}$	0.26	0.47	0.28	0.34	0.31	0.31	0.33

2.6. CONCLUSIONS AND RECOMMENDATIONS

In this chapter, a novel model architecture - namely the ISTRUST model–based on vision transformers is proposed capable of predicting the RUL under uncertainty, given sequences of raw images as input, with a primary focus on interpretability. This model is evaluated on an experimental dataset acquired from composite samples that are under fatigue loads and visible cracks as damage propagates. To interpret the correlation between the input images taken from two cameras and the RUL predictions an innovative attention mechanism is proposed based on the decomposition of the spatiotemporal domain. By separating the temporal and spatial domains and leveraging the attention mechanism, the black-box commonly associated with DL architectures is circumvented, allowing for an interpretable AI prognostic model. The spatial and temporal attention weights demonstrated the model's ability to correctly prioritize patches with higher levels of damage in the spatial and temporal domain respectively. Besides, a noticeable correlation was observed between the accuracy of the spatial attention weights and the accuracy of the RUL prediction. Based on that, it was shown that despite the RUL prediction being initially a flat line for all specimens, as soon as the model focused on the damage, it started to drop at the anticipated negative slope.

Because of the limited data acquired from the experiment, a data augmentation technique is performed, thus increasing the risk of overfitting. In this regard, contrastive learning is utilized to help the ISTRUST model distinguish the important information responsible for crack propagation, which in turn affects the RUL, and filters out the spurious one. The UMAP representation is responsible for visualizing the results of the contrastive learning. Furthermore, the stochasticity of the RUL is naturally included in our ISTRUST model via the MC dropout, offering a simple yet meaningful representation of the introduced uncertainty.

Despite the weight initialization being designed to assist the model in capturing horizontal cracks, it notably succeeded in identifying diagonal cracks during the learning process. As a result, the model demonstrates the potential for generalization to more complex crack shapes. However, in such scenarios, further testing with various weight initializations is necessary to determine the optimal configuration. It is imperative to acknowledge that the naturally occurring cracks within the internal sections of the structure are detectable by the cameras only when they are detected on the surface. Consequently, in scenarios where the specimens are black, such as with carbon fiber composites, cracks may be difficult to detect visually. In these cases, it would be advisable to apply a white painted surface to the specimens before using the model to enhance crack visibility and ensure accurate analysis. This condition is critical for the application of this model and warrants further investigation in the future.

One key limitation of the present work is the lack of comprehensive data that captures the full range of failure mechanisms inherent to composite materials. The fatigue life of composites is highly dependent on the initiation, interaction, and propagation of cracks, which are intrinsically stochastic and can lead to significant variability in RUL predictions. In particular, specimen 5, with its major cracks oriented at 45 degrees, highlights the need for training data that includes diverse failure modes. Additional future work should address this by incorporating a wider variety of damage scenarios to improve the robustness of the model.

While it is recognized that the ISTRUST model's predictive performance may not be ideal for specific cases, it is essential to highlight that our approach provides a logical and cohesive explanation for the underlying factors influencing this observation. This lays the foundation for creative and forward-thinking ideas to enhance its effectiveness in the future. Additionally, it is recommended to apply our proposed architecture to larger datasets, hence the resolution of the model could be increased without the potential of overfitting. This is because higher resolution could avoid the distraction of the spatial attention weights to spurious features and allow the model to capture less extensive damage, thus further increasing the model's performance. Furthermore, because of the accurate spatial attention maps, it should be further investigated whether the attention weights can be leveraged for anomaly detection by modifying the training setup, the size of the dataset, or the model architecture.

Finally, as predictions depend entirely on raw data inputs, the ISTRUST model has the ability to identify situations where predictions may fall short or perform well. Consequently, it is reasonable to expect that the model's performance depends on the quality of the data provided. This demonstrates that in real-world scenarios with complex and noisy data, an end-to-end model that processes raw multi-dimensional time-series data to predict RUL, even if interpretable, may not provide sufficiently accurate predictions for reliable decision-making. In this regard, in the next chapters, we will follow a hierarchical approach by focusing on each phase of the PHM strategy independently.

REFERENCES

[1] P. Komninos, A. Verraest, N. Eleftheroglou, and D. Zarouchas. “Intelligent fatigue damage tracking and prognostics of composite structures utilizing raw images via interpretable deep learning”. In: *Composites Part B: Engineering* 287 (2024), p. 111863. ISSN: 1359-8368. DOI: https://doi.org/10.1016/j.compositesb.2024.111863.

[2] R. Thomopoulos, S. Destercke, B. Charnomordic, I. Johnson, and J. Abécassis. “An iterative approach to build relevant ontology-aware data-driven models”. In: *Information Sciences* 221 (2013), pp. 452–472. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2012.09.015. URL: https://www.sciencedirect.com/science/article/pii/S0020025512006081.

[3] T. Li, S. Wang, E. Zio, J. Shi, and Z. Ma. “A numerical approach for predicting the remaining useful life of an aviation hydraulic pump based on monitoring abrasive debris generation”. In: *Mechanical Systems and Signal Processing* 136 (2020), p. 106519. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2019.106519. URL: https://www.sciencedirect.com/science/article/pii/S088832701930740X.

[4] X. Xu, X. Li, W. Ming, and M. Chen. “A novel multi-scale CNN and attention mechanism method with multi-sensor signal for remaining useful life prediction”. In: *Computers & Industrial Engineering* 169 (2022), p. 108204. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2022.108204. URL: https://www.sciencedirect.com/science/article/pii/S0360835222002741.

[5] C.-G. Huang, H.-Z. Huang, Y.-F. Li, and W. Peng. “A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing”. In: *Journal of Manufacturing Systems* 61 (2021), pp. 757–772. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2021.03.012. URL: https://www.sciencedirect.com/science/article/pii/S0278612521000674.

[6] P. Baraldi, F. Mangili, and E. Zio. “A belief function theory based approach to combining different representation of uncertainty in prognostics”. In: *Information Sciences* 303 (2015), pp. 134–149. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2014.12.051. URL: https://www.sciencedirect.com/science/article/pii/S0020025515000031.

[7] Y. Yang, H. Chen, Y. Mi, C. Luo, S.-J. Horng, and T. Li. “Multi-label feature selection based on stable label relevance and label-specific features”. In: *Information Sciences* 648 (2023), p. 119525. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.119525. URL: https://www.sciencedirect.com/science/article/pii/S0020025523011106.

[8] N. Eleftheroglou and T. Loutas. “Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes”. In: *Structural Health Monitoring* 15.4 (2016), pp. 473–488. DOI: 10.1177/1475921716646579. eprint: https://doi.org/10.1177/1475921716646579. URL: https://doi.org/10.1177/1475921716646579.

[9] Z. Wang, N. Liu, C. Chen, and Y. Guo. “Adaptive self-attention LSTM for RUL prediction of lithium-ion batteries”. In: *Information Sciences* 635 (2023), pp. 398–413. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.01.100. URL: https://www.sciencedirect.com/science/article/pii/S0020025523001007.

[10] S. Behera, R. Misra, and A. Sillitti. “Multiscale deep bidirectional gated recurrent neural networks based prognostic method for complex non-linear degradation systems”. In: *Information Sciences* 554 (2021), pp. 120–144. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.12.032. URL: https://www.sciencedirect.com/science/article/pii/S0020025520311981.

[11] J. Deutsch and D. He. “Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.1 (2018), pp. 11–20. DOI: 10.1109/TSMC.2017.2697842.

[12] Y. Wang, D. Wang, X. Ye, Y. Wang, Y. Yin, and Y. Jin. “A tree ensemble-based two-stage model for advanced-stage colorectal cancer survival prediction”. In: *Information Sciences* 474 (2019), pp. 106–124. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.09.046. URL: https://www.sciencedirect.com/science/article/pii/S002002551830759X.

[13] N. Eleutheroglou, D. Zarouchas, and R. Benedictus. “An adaptive probabilistic data-driven methodology for prognosis of the fatigue life of composite structures”. In: *Composite Structures* 245 (2020), p. 112386. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2020.112386. URL: https://www.sciencedirect.com/science/article/pii/S0263822319347634.

[14] N. Eleutheroglou, G. Galanopoulos, and T. Loutas. “Similarity learning hidden semi-Markov model for adaptive prognostics of composite structures”. In: *Reliability Engineering & System Safety* 243 (2024), p. 109808. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2023.109808. URL: https://www.sciencedirect.com/science/article/pii/S0951832023007226.

[15] A. Akrim, C. Gogu, T. G. d. Nerville, P. Strähle, B. W. Pagou, M. Salaün, and R. Vingerhoeds. “A Framework for Generating Large Data Sets for Fatigue Damage Prognostic Problems”. In: *2022 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 2022, pp. 25–33. DOI: 10.1109/ICPHM53196.2022.9815692.

[16] A. Akrim, C. Gogu, R. Vingerhoeds, and M. Salaün. “Self-Supervised Learning for data scarcity in a fatigue damage prognostic problem”. In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105837. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.105837. URL: https://www.sciencedirect.com/science/article/pii/S0952197623000210.

[17] T.-K. Nguyen, Z. Ahmad, and J.-M. Kim. “A Deep-Learning-Based Health Indicator Constructor Using Kullback-Leibler Divergence for Predicting the Remaining Useful Life of Concrete Structures”. In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: 10.3390/s22103687. URL: https://www.mdpi.com/1424-8220/22/10/3687.

[18] A. Wahid, M. Yahya, J. G. Breslin, and M. A. Intizar. “Self-Attention Transformer-Based Architecture for Remaining Useful Life Estimation of Complex Machines”. In: *Procedia Computer Science* 217 (2023). 4th International Conference on Industry 4.0 and Smart Manufacturing, pp. 456–464. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.12.241. URL: https://www.sciencedirect.com/science/article/pii/S1877050922023195.

[19] Q. Li and Y. Yang. “Transformer Model for Remaining Useful Life Prediction of Aeroengine”. In: *Journal of Physics: Conference Series* 2171.1 (Jan. 2022), p. 012072. DOI: 10.1088/1742-6596/2171/1/012072. URL: https://dx.doi.org/10.1088/1742-6596/2171/1/012072.

[20] Y. Mo, Q. Wu, X. Li, and et al. “Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit”. In: *Journal of Intelligent Manufacturing* 32.8 (Oct. 2021), pp. 1997–2006. DOI: 10.1007/s10845-021-01750-x.

[21] C. Zhao, X. Huang, Y. Li, and M. Yousaf Iqbal. “A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction”. In: *Sensors* 20.24 (2020). ISSN: 1424-8220. DOI: 10.3390/s20247109. URL: https://www.mdpi.com/1424-8220/20/24/7109.

[22] Y. Zhou, M. Hefenbrock, Y. Huang, T. Riedel, and M. Beigl. “Automatic Remaining Useful Life Estimation Framework with Embedded Convolutional LSTM as the Backbone”. In: *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*. Ed. by Y. Dong, D. Mladenić, and C. Saunders. Cham: Springer International Publishing, 2021, pp. 461–477. ISBN: 978-3-030-67667-4.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper˙files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[25] G. Bertasius, H. Wang, and L. Torresani. “Is Space-Time Attention All You Need for Video Understanding?” In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 813–824. URL: https://proceedings.mlr.press/v139/bertasius21a.html.

[26] C. Plizzari, M. Cannici, and M. Matteucci. “Spatial Temporal Transformer Network for Skeleton-Based Action Recognition”. In: *Pattern Recognition. ICPR International Workshops and Challenges*. Ed. by A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani. Cham: Springer International Publishing, 2021, pp. 694–701. ISBN: 978-3-030-68796-0.

[27] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid. “ViViT: A Video Vision Transformer”. In: *CoRR* abs/2103.15691 (2021). arXiv: 2103.15691. URL: https://arxiv.org/abs/2103.15691.

[28] X. Li, J. Li, L. Zuo, L. Zhu, and H. T. Shen. “Domain Adaptive Remaining Useful Life Prediction With Transformer”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–13. DOI: 10.1109/TIM.2022.3200667.

[29] Z. Zhang, W. Song, and Q. Li. “Dual-Aspect Self-Attention Based on Transformer for Remaining Useful Life Prediction”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–11. DOI: 10.1109/TIM.2022.3160561.

[30] D. Chen, W. Hong, and X. Zhou. “Transformer Network for Remaining Useful Life Prediction of Lithium-Ion Batteries”. In: *IEEE Access* 10 (2022), pp. 19621–19628. DOI: 10.1109/ACCESS.2022.3151975.

[31] J. Guo, X. Nie, Y. Ma, K. Shaheed, I. Ullah, and Y. Yin. “Attention based consistent semantic learning for micro-video scene recognition”. In: *Information Sciences* 543 (2021), pp. 504–516. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.05.064. URL: https://www.sciencedirect.com/science/article/pii/S0020025520304758.

[32] L. Wei and G. Zong. “EGA-Net: Edge feature enhancement and global information attention network for RGB-D salient object detection”. In: *Information Sciences* 626 (2023), pp. 223–248. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2023.01.032. URL: https://www.sciencedirect.com/science/article/pii/S0020025523000324.

[33] C.-H. Hua, T. Huynh-The, S.-H. Bae, and S. Lee. “Cross-Attentional Bracket-shaped Convolutional Network for semantic image segmentation”. In: *Information Sciences* 539 (2020), pp. 277–294. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.06.023. URL: https://www.sciencedirect.com/science/article/pii/S0020025520306101.

[34] S. Liao, H. Liu, J. Yang, and Y. Ge. “A channel-spatial-temporal attention-based network for vibration-based damage detection”. In: *Information Sciences* 606 (2022), pp. 213–229. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.05.042. URL: https://www.sciencedirect.com/science/article/pii/S0020025522004686.

[35] J. Kim and J. Canny. *Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention*. 2017. arXiv: `1703.10631 [cs.CV]`.

[36] S. Boukhtache, K. Abdelouahab, F. Berry, B. Blaysat, M. Grédiac, and F. Sur. "When Deep Learning Meets Digital Image Correlation". In: *Optics and Lasers in Engineering* 136 (2021), p. 106308. ISSN: 0143-8166. DOI: https://doi.org/10.1016/j.optlaseng.2020.106308. URL: https://www.sciencedirect.com/science/article/pii/S0143816620306588.

[37] Y. Wang and J. Zhao. "DIC-Net: Upgrade the performance of traditional DIC with Hermite dataset and convolution neural network". In: *Optics and Lasers in Engineering* 160 (2023), p. 107278. ISSN: 0143-8166. DOI: https://doi.org/10.1016/j.optlaseng.2022.107278. URL: https://www.sciencedirect.com/science/article/pii/S0143816622003311.

[38] X. Cheng, S. Zhou, T. Xing, Y. Zhu, and S. Ma. "Solving digital image correlation with neural networks constrained by strain-displacement relations". In: *Optical Express* 31.3 (Jan. 2023), pp. 3865–3880. DOI: 10.1364/OE.475232.

[39] B. Pantoja-Rosero, D. Oner, M. Kozinski, R. Achanta, P. Fua, F. Perez-Cruz, and K. Beyer. "TOPO-Loss for continuity-preserving crack detection using deep learning". In: *Construction and Building Materials* 344 (2022), p. 128264. ISSN: 0950-0618. DOI: https://doi.org/10.1016/j.conbuildmat.2022.128264. URL: https://www.sciencedirect.com/science/article/pii/S0950061822019250.

[40] D. Dais, İ. E. Bal, E. Smyrou, and V. Sarhosis. "Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning". In: *Automation in Construction* 125 (2021), p. 103606. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2021.103606. URL: https://www.sciencedirect.com/science/article/pii/S0926580521000571.

[41] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung. "Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle". In: *Sensors* 18.6 (2018). ISSN: 1424-8220. DOI: 10.3390/s18061881. URL: https://www.mdpi.com/1424-8220/18/6/1881.

[42] M. M. Khani, S. Vahidnia, L. Ghasemzadeh, Y. E. Ozturk, M. Yuvalaklioglu, S. Akin, and N. K. Ure. "Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines". In: *Structural Health Monitoring* 19.5 (2020), pp. 1440–1452. DOI: 10.1177/1475921719883202. eprint: https://doi.org/10.1177/1475921719883202. URL: https://doi.org/10.1177/1475921719883202.

[43] A. Zhang, K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li, and B. Li. "Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet". In: *Journal of Computing in Civil Engineering* 32.5 (2018), p. 04018041. DOI: 10.1061/(ASCE)CP.1943-5487.0000775. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CP.1943-5487.0000775. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0000775.

[44] S.-P. Zhu, L. Wang, C. Luo, J. Correia, A. Jesus, F. Berto, and Q. Wang. “Physics-informed machine learning and its structural integrity applications: state of the art”. In: *Philosophical Transactions of the Royal Society A* 381 (Sept. 2023). DOI: 10.1098/rsta.2022.0406.

[45] L.-K. Song, X.-Q. Li, S.-P. Zhu, and Y.-S. Choy. “Cascade ensemble learning for multi-level reliability evaluation”. In: *Aerospace Science and Technology* 148 (2024), p. 109101. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2024.109101. URL: https://www.sciencedirect.com/science/article/pii/S1270963824002347.

[46] N. Eleutheroglou. “Adaptive prognostics: a reliable RUL approach”. In: *Proceedings of the Annual Conference of the PHM Society 2023*. Annual Conference of the PHM Society, 2023. DOI: https://doi.org/10.36001/phmconf.2023.v15i1.3495. URL: https://papers.phmsociety.org/index.php/phmconf/article/view/3495.

[47] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: https://proceedings.mlr.press/v48/gal16.html.

[48] K. Zha, P. Cao, Y. Yang, and D. Katabi. “Supervised Contrastive Regression”. Manuscript under review. 2022. arXiv: 2210.01189 [cs.LG].

[49] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. “Supervised Contrastive Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 18661–18673. URL: https://proceedings.neurips.cc/paper˙files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[50] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

[51] L. McInnes, J. Healy, N. Saul, and L. Großberger. “UMAP: Uniform Manifold Approximation and Projection”. In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861. URL: https://doi.org/10.21105/joss.00861.

[52] M. Buscema. “Back Propagation Neural Networks”. In: *Substance Use & Misuse* 33.2 (1998). PMID: 9516725, pp. 233–270. DOI: 10.3109/10826089809115863. eprint: https://doi.org/10.3109/10826089809115863. URL: https://doi.org/10.3109/10826089809115863.

[53] Q. Wang, B. Zhao, H. Ma, and et al. “A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion”. In: *Journal of Mechanical Science and Technology* 33.6 (June 2019), pp. 2561–2571. DOI: 10.1007/s12206-019-0504-x.

[54] P. Ding, X. Liu, H. Li, Z. Huang, K. Zhang, L. Shao, and O. Abedinia. “Useful life prediction based on wavelet packet decomposition and two-dimensional convolutional neural network for lithium-ion batteries”. In: *Renewable and Sustainable Energy Reviews* 148 (2021), p. 111287. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2021.111287. URL: https://www.sciencedirect.com/science/article/pii/S1364032121005748.

[55] D. Neimark, O. Bar, M. Zohar, and D. Asselmann. “Video Transformer Network”. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 3156–3165. DOI: 10.1109/ICCVW54120.2021.00355.

[56] J. Devlin, M. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[57] B. Lim, S. Arık, N. Loeff, and T. Pfister. “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL: https://www.sciencedirect.com/science/article/pii/S0169207021000637.

[58] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. “On Layer Normalization in the Transformer Architecture”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 10524–10533. URL: https://proceedings.mlr.press/v119/xiong20b.html.

[59] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. “CvT: Introducing Convolutions to Vision Transformers”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 22–31. DOI: 10.1109/ICCV48922.2021.00009.

[60] N. Eleftheroglou. *Adaptive prognostics for remaining useful life of composite structures*. Oct. 2020. URL: https://repository.tudelft.nl/islandora/object/uuid:538558fb-ac9a-414d-8a59-4b523d8ff74c?collection=research.

[61] K. Reifsnider and A. Talug. “Analysis of fatigue damage in composite laminates”. In: *International Journal of Fatigue* 2.1 (1980), pp. 3–11. ISSN: 0142-1123. DOI: https://doi.org/10.1016/0142-1123(80)90022-5. URL: https://www.sciencedirect.com/science/article/pii/0142112380900225.

[62] X. Li, J. Kupski, S. Teixeira De Freitas, R. Benedictus, and D. Zarouchas. “Unfolding the early fatigue damage process for CFRP cross-ply laminates”. In: *International Journal of Fatigue* 140 (2020), p. 105820. ISSN: 0142-1123. DOI: https://doi.org/10.1016/j.ijfatigue.2020.105820. URL: https://www.sciencedirect.com/science/article/pii/S0142112320303510.

[63] G. C. Cawley and N. L. Talbot. “On over-fitting in model selection and subsequent selection bias in performance evaluation”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 2079–2107.

[64] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. “Leakage in Data Mining: Formulation, Detection, and Avoidance”. In: *ACM Trans. Knowl. Discov. Data* 6.4 (Dec. 2012). ISSN: 1556-4681. DOI: 10.1145/2382577.2382579. URL: https://doi.org/10.1145/2382577.2382579.

[65] X. S. Huang, F. Perez, J. Ba, and M. Volkovs. “Improving Transformer Optimization Through Better Initialization”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 4475–4483. URL: https://proceedings.mlr.press/v119/huang20f.html.

[66] N. Eleftheroglou, D. Zarouchas, T. Loutas, R. Alderliesten, and R. Benedictus. “Structural health monitoring data fusion for in-situ life prognosis of composite structures”. In: *Reliability Engineering & System Safety* 178 (2018), pp. 40–54. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2018.04.031. URL: https://www.sciencedirect.com/science/article/pii/S0951832017306737.

3

DEEP SOFT MONOTONIC FEATURE EXTRACTION AND PROGNOSTICS

This chapter proposes a novel NN model for performing monotonic feature extraction and HI construction related to degenerative phenomena. The constructed HIs are then fed into a prognostic model to estimate RUL under uncertainty. Additionally, the interpretability of the NN model is shown. The evaluation of this model was applied to three datasets with unique difficulties from different scientific fields, including healthcare and engineering, showcasing generalizability and robustness.

This chapter is under second revision in Data-Centric Engineering (2025) **2632-6736**.

3.1. INTRODUCTION

In today's interconnected world, the significance of comprehending and addressing degenerative phenomena across diverse domains remains crucial to the advancement of humanity as a whole. The ability to anticipate and proactively address this deterioration has far-reaching implications for enhancing safety, optimizing resource allocation, and improving the overall quality of life [1]. In particular, for healthcare and engineering applications, understanding and mitigating deterioration becomes even more crucial. The recognition and understanding of deterioration in medical applications, defined as clinical deterioration, holds paramount importance as it enables timely intervention and proactive management, ultimately safeguarding patient well-being and improving healthcare outcomes [2, 3]. Simultaneously, the significance of detecting and addressing deterioration in engineering applications cannot be overstated, as it facilitates proactive maintenance and optimization of operations, leading to enhanced productivity, cost-efficiency, and reliability [4, 5].

Advancements in technology, such as the Internet of Things and data analytics, have further facilitated the monitoring and management of deteriorating systems. Prognostics -a discipline that strives to anticipate the future behavior of systems based on their current conditions- is the principal component of understanding and predicting future deterioration. By harnessing the power of data-driven insights, prognostic models enable the identification of early warning signs, predict critical events, and facilitate real-time decision-making to prevent failures. However, the intrinsic challenges associated with this discipline, including the unsupervised nature of the task, constraints imposed by limited data availability, and the complex nature of understanding deterioration, collectively serve as significant impediments that must be addressed to achieve accurate and predictive outcomes. Thereby, the extraction of features from raw data represents a crucial intermediate step with the potential to facilitate the development of an efficient prognostic framework. Effective feature extraction simplifies the complexity of raw sensor data by reducing noise and identifying the most relevant signals for predicting deterioration. Additionally, it enhances interpretability and reduces computational complexity, thus simplifying the process of constructing prognostic models.

Currently, most prognostic frameworks are designed for specific fields, making them less adaptable to different areas of study. Yet, an increasing demand exists for versatile models capable of functioning across diverse disciplines, driven by the interconnectedness prevalent in modern times. Whether the focus is on healthcare, engineering, or other sectors, systems often overlap. Creating versatile prognostic models that can handle these interdisciplinary challenges is essential. It can provide valuable insights, enhance decision-making, and contribute to greater efficiency and resilience in diverse fields. Moreover, there exists an imperative demand for these models to exhibit user-friendliness, enabling individuals who lack expertise in the specific field to employ them without necessitating the creation of individual models on each occasion. This would save both valuable time and resources, making the development of adaptable and accessible models a significant step in addressing complex real-world issues. Clustering techniques have the potential to serve as a solution to this multifaceted problem by enabling the identification of common patterns and behaviors across different domains.

Despite the emerging contribution of ML and DL models concerning predicting degenerative phenomena via feature extraction and clustering techniques to medical and engineering applications, they come up with significant barriers to being easily applicable, adaptable, and transferable to diverse domains. Firstly, it is challenging to extract informative features from noisy raw data (data sparsity) under limited availability (data scarcity) and in an unsupervised manner. On the one hand, the complexity and heterogeneity of medical datasets pose a significant challenge in extracting actionable knowledge from data [6, 7]. On the other hand, engineering systems often consist of datasets with diverse parameters such as vibration patterns, temperature fluctuations, and acoustic emissions. The complexity of those systems, coupled with the vast amounts of data generated by sensors and monitoring devices, presents a significant challenge in extracting relevant information for prognostics.

Secondly, one of the primary challenges encountered in the clustering process within the context of comprehending the pattern of system deterioration for prognostic applications lies in the extensive preprocessing steps necessary prior to feeding the data into a clustering model [8]. These preprocessing steps involve the arduous tasks of noise removal, exclusion of irrelevant information, data fusion, and dimensionality reduction, requiring not only considerable time and computational resources, but also domain knowledge [9]. This barrier restricts the generalizability of the model to other domains, necessitating a similar exhaustive preprocessing effort and another training process for the new task.

Thirdly, current feature extraction models are being developed jointly with the selected prognostic algorithm. Consequently, their efficiency in extracting relevant features is not guaranteed if used with a different prognostic algorithm. In essence, feature extraction is not agnostic to the underlying prognostic algorithm, thus significantly constraining the model's generalizability.

Finally, current feature extraction models predominantly rely on unimodal inputs [10–12], overlooking the immense potential that multi-modal data fusion holds for prognostic-related tasks. Healthcare and engineering benefit substantially from the integration of diverse data streams encompassing clinical, laboratory, and demographic data (healthcare) [13], and a mix of time-series and image sensory data (engineering)[14]. However, integrating diverse data modalities often requires addressing issues related to data heterogeneity, varying scales, disparate formats, and inherent noise across different sources. Additionally, it requires specialized expertise, and robust methodologies for alignment and fusion ensuring harmonization among varying data sources [15–17].

The aforementioned challenges are not only detected in healthcare and engineering fields, but they are actively limiting the development of such models to any scientific field related to degenerative phenomena. Hence, creating robust models capable of extracting meaningful patterns and features automatically from any data source is essential for improved performance, generalizability, and robustness. In this regard, the current work introduces the DSMC model based on NN as a fundamental process for feature extraction via clustering analysis in the generic context of deteriorating systems and is showcased on multidisciplinary fields including healthcare and engineering. The proposed DSMC model is generalizable and agnostic of the chosen prognostic model developed after the clustering process, hence exhibiting promising potential for broader application across various domains beyond those examined in this study. Notably, the novelty of the model lies in its unique capability to extract prognostic-related features, i.e. increasing monotonic features as time increases, directly from raw and multi-modal data, in an unsupervised and end-to-end manner. The selection of prognostic-related features in the proposed approach aims to capture partial (soft) monotonicity rather than complete (hard) monotonicity. This choice is made to incorporatepotential occurrence of oscillations in the degradation trajectory of the analyzed system. As a result, the DSMC model has the capability to identify certain timestamps within a given trajectory where a substantial recovery may arise, thereby reflecting real-world systems and enabling a certain level of data comprehension.

The proposed model is applied to three carefully selected datasets from distinct scientific domains including healthcare and engineering, both of significant importance to humanity. The first dataset, known as Medical Information Mart for Intensive Care III (MIMIC-III) [18], pertains to the field of healthcare and encompasses numerous subsets representing diverse life-threatening conditions. For the purpose of this study, the sepsis subset within the MIMIC-III dataset was specifically chosen, given its intricate syndrome nature, substantial healthcare costs, and high mortality rates. In particular, sepsis contributes to 6% of hospitalizations and 35% of in-hospital deaths [19] (approximately 30% of patients do not survive longer than 6 months [20]) and corresponds to more than US$27 billion aannually in the USA [21]. The diverse multi-modal characteristics inherent in this dataset serve as a testament to the challenges encountered by our model in handling and effectively leveraging multiple modes of information. It consists of vital signs, treated as one-dimensional time-series data, and laboratory and demographic data, treated as tabular data.

The second dataset employed in this study is NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), which is associated with the engineering domain. The utilization of this dataset facilitates the development of prognostic models, contributing to technological advancements, enhanced safety measures, minimized costs, and reduced environmental impact [22]. This dataset consists of multivariate time-series data and is a proper candidate for comparing the outcomes with different standard techniques.

While these two datasets are recognized for highlighting not only the presence of multi-modality (time-series, static data) but also demonstrating the generalizable and robust nature of our model, an additional validation to ascertain the model's proficiency in handling multi-modal information involves the selection of a third dataset from the engineering domain. This dataset concerns an experimental campaign of a structure under fatigue loading [23] and comprises one-, two-, and three-dimensional data concurrently (time-series and sequences of images), thereby surpassing the complexity of the C-MAPSS dataset. For the remainder of this paper, the third dataset will be named F-MOC (which stands for Fatigue Monitoring of Composites). It is worth mentioning that the F-MOC dataset includes real data from the engineering field, unlike the C-MAPSS dataset, which consists of simulated data.

Additionally, the objectives of this work are:

- Extracting soft monotonic features from raw data that could be directly fed as input to any prognostic model.
- The soft monotonic feature extraction method should adeptly handle multi-modal data.
- Application in multidisciplinary domains including degenerative phenomena in a fully automatic fashion. Extensive preprocessing of the data should be avoided, thus a similar architecture can be reproduced, enabling generalizability.
- Interpretability of the proposed model to effectively understand its learning process and predicting capabilities.

3.2. RELATED WORK IN MONOTONIC FEATURE EXTRACTION AND CLUSTERING

3.2.1. FEATURE EXTRACTION FOR PROGNOSTIC-RELATED TASKS IN HEALTHCARE AND ENGINEERING

In the field of healthcare, prognostics play a pivotal role in enhancing patient care, optimizing treatment strategies, and allocating healthcare resources effectively [24, 25]. In-hospital clinical deterioration may relate to existing or emerging diseases, or a complication of the health care provided. Undoubtedly, by utilizing data-driven techniques, mainly through ML and its subfield, DL, prognostic models can identify new possible biomarkers [26], detect anomalies [27, 28], predict disease progression [29–35] and enable personalized treatment plans [36–38]. Additionally, prognostics also hold immense value in the domain of engineering systems [39–42] as they play a vital role in enhancing system performance and optimizing maintenance strategies.

Prognostic algorithms have great potential, but they face significant challenges, such as working with unlabeled data, limited data availability, and understanding deterioration complexities. Overcoming these obstacles is crucial for accurate predictions. Extracting features from raw data is a key step that could help improve prognostic algorithms in terms of accuracy and robustness. Having extracted features simplifies the available data, makes it easier to understand, and reduces complexity.

A comprehensive literature review on state-of-the-art feature extraction techniques for prognostics using DL in healthcare and engineering reveals significant advancements and diverse methodologies. In healthcare, DL models such as CNN [43, 44] and RNN [45, 46] have been extensively utilized to extract complex features from medical images and time-series data, respectively, aiding in the prediction of disease progression and patient outcomes. Similarly, in engineering, DL techniques are employed to analyze sensor data and identify critical patterns indicative of system health and impending failures. Hybrid models combining CNNs with LSTM networks have shown remarkable performance and have been currently the state-of-the-art approaches in capturing both spatial and temporal dependencies, enhancing prognostic accuracy [47, 48]. Autoencoders, known for their capability to learn efficient representations of data, can be combined with these approaches to further refine feature extraction by reducing dimensionality and denoising input data, thereby improving the robustness and accuracy of prognostic models [49–51]. Additionally, attention mechanisms are increasingly being integrated to refine feature extraction and improve model generalization across different datasets [52, 53]. These advancements underscore the critical role of DL in transforming prognostics by providing robust, data-driven insights in both healthcare and engineering domains.

Clustering techniques can offer valuable insights and improve feature extraction in prognostics by grouping similar data points, thereby uncovering inherent structures within the data. This unsupervised learning approach enables the identification of patterns and anomalies that might not be evident through traditional methods. Clustering models have significant potential as feature extractors, serving as a crucial preliminary step preceding the prognostic phase. Model-agnostic feature extraction methods that utilize data-driven clustering can categorize and extract relevant information allowing the development of adaptable and accessible prognostic tools that transcend disciplinary boundaries [54].

In the context of prognostics, clustering serves as a powerful tool for reducing dimensionality, enhancing interpretability, and improving the accuracy of prognostic models [55]. Considering healthcare, this process can unravel patient subgroups with distinct disease trajectories [56], thus enabling tailored interventions and personalized healthcare delivery. For instance, predicting the mortality rate of patients afflicted with sepsis, a life-threatening condition arising from the body's response to infection, can be achieved through the utilization of clustering and prognostic methodologies [57]. By identifying clusters within the patients' population, it becomes possible to uncover distinct patterns and subgroups that may have different mortality risks. The unique trends and characteristics observed by the clustering analysis can facilitate the development of prognostic models that are able to predict mortality rates with more accurate risk stratification.

Similarly, in the field of engineering systems, clustering techniques can reveal inherent patterns and relationships within sensor data [58], enabling the identification of distinct operational regimes [59, 60], detection of anomalies [61, 62], and optimization of maintenance strategies [63, 64]. By integrating clustering techniques into the prognostics workflow, researchers or human experts can leverage the underlying pattern within engineering data, extract representative features, and develop accurate prognostic models easily transferable to varying engineering applications [65].

3.2.2. MULTI-MODAL DEEP CLUSTERING

Numerous clustering techniques have been proposed in the literature in the past decades. Deep clustering, an extension of the typical clustering algorithms for tasks with increased data complexity utilizing NN, has shown promising results in fusing multi-sensory data and learning useful and interpretable representations. For instance, Xu et al. [66] proposed a unified framework based on NN with disentangled representation learning that learns interpretable representations by performing multi-view clustering, thus achieving multi-view information fusion without requiring label supervision. This was accomplished by constructing multiple autoencoders (AE) for handling each unique kind of information. Then, the embeddings were fused in the disentangled representation phase to keep the meaningful information for clustering. Dimitri et al. [67] aimed at deep multi-modal image fusion and clustering with an application in neuroimaging. The key novelty was the combination of deep AE for creating embeddings that were combined with other demographic data extracted from typical ML techniques to cluster the examined patients into subgroups based on the severity of brain damage. Finally, an algorithm that simultaneously learns feature representations and cluster assignments based on NN was suggested by Xie et al. [68] and evaluated on two image-related and one text-related public datasets.

The current study incorporates multi-modal data by combining either time-series and static data or a combination of time-series, static data, and sequential images.

3.2.3. MONOTONIC NEURAL NETWORKS

The literature on integrating monotonicity within the layers of an NN has a longstanding history. However, this field came up with significant barriers due to the significant constraints imposed on the parameter space. This resulted in the optimization process being prone to converging towards local optima [69]. Nevertheless, the fundamental research in [70] demonstrated that the universal approximation capabilities of an NN remain valid under the condition of constraining weights to be positive by leveraging an unconstrained continuous function. Subsequently, building upon this proposition, researchers in [71] illustrated that the backpropagation algorithm retains its functionality when unconstrained weights are transformed into their exponential counterparts, and all activation functions are assumed to be positive across their entire domain. Activation functions such as the Rectified Linear Unit (ReLU), Sigmoid, or Softmax have been identified as suitable candidates for this purpose since are positive everywhere. Consequently, weights can assume any real value while their exponential transformation ensures their confinement within the positive domain. Subsequently, numerous studies have adopted similar methodologies to enforce monotonicity within their NN architectures without constraining the exploration space of parameters [72–74].

3.3. METHODOLOGY

In this section, the methodology of the current study is presented. Figure 3.1 shows the general concept of this work, beginning from the correct shaping of the data where the time feature is inserted and the technique of sliding windows (similarly to chapter 2) is applied to the time-series signals. Next, the model construction takes place where the AE extracts the monotonic features to the output of the Z-space used to perform clustering analysis. The training of this model is performed iteratively by utilizing the Bayesian optimization for the tuning of the hyperparameters Finally, the prognostic algorithms are applied to estimate the reliability and survivability curves.

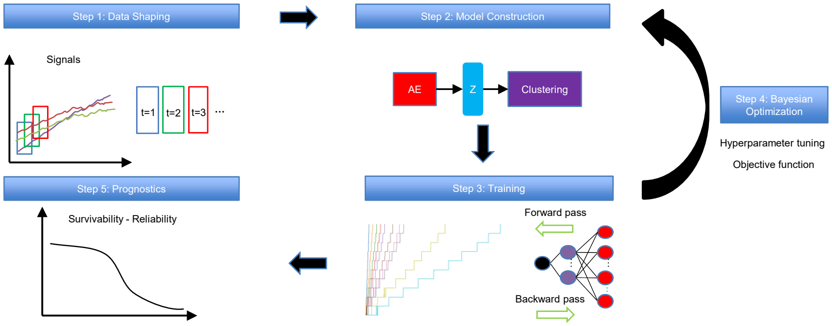


Figure 3.1.: The concept of the proposed methodology.

3.3.1. DATASETS AND DATA SHAPING

The purpose of this study is to present a generalized monotonic clustering model that can be applied in multidisciplinary domains, can identify deterioration in systems, and prepare monotonic features ready to be fed to any prognostic model in an unsupervised manner with limited training data (5-90 trajectories, depending on the dataset). In this regard, two publicly available datasets are examined from entirely different scientific fields. Additionally, a third dataset representing an experimental case study has been chosen for this work. Those datasets were carefully chosen due to their unique characteristics, difficulties, and contributions to healthcare and engineering. These difficulties are included in the objectives mentioned in section 3.1. Each of these datasets achieves one or more of the desired objectives. Particularly, the soft monotonic feature extraction outcomes and the model's generalizability and interpretability are showcased by all of the datasets. The model's prognostic algorithm agnosticism is established via the C-MAPSS dataset by demonstrating consistent prognostic outcomes across three distinct ML prognostic models. The evaluation of the proposed model's performance across multidisciplinary domains is illustrated by utilizing the MIMIC-III dataset. This evaluation involves a comparative analysis of survivability probabilities using various healthcare scoring systems. The handling of multi-modal data for soft monotonic feature extraction is mainly validated by the F-MOC dataset. Regarding interpretability, in detail, the flow of the time gradients is illustrated to validate the acceptance of the proposed technique concerning soft monotonicity. Simultaneously, the extracted hidden features are depicted before and after training the DSMC model to justify their role in both capturing the time constraint and performing appropriate clustering.

The MIMIC-III database is a publicly available and widely used database that incorporates patient information from patients hospitalized and stayed in an Intensive Care Unit (ICU) at Beth Israel Deaconess Medical Center (Bowers, Massachusetts, USA) between 2001 and 2012. It contains data about patients' demographics, vital signs, lab tests, and treatment assignments. From these data, focus was given on adult patients fulfilling the international consensus Sepsis-3 criteria [75] who passed away from sepsis and stayed at the ICU for more than 10 hours. Hence, patients who stayed 9 hours or less were excluded, as proposed in a previous work [76], due to potentially unreliable measurements. Thus, data from 62 patients that included non-missing values for demographics, vital signs, and lab tests were extracted. This is the total number of patients who met with a death event, showcasing a challenging dataset in terms of data scarcity. Vital signs contain the time-series data and demographics, and lab tests contain the supplementary data with the abovementioned time feature included. The unique challenge of this dataset is that it includes both time-series and static input data that should be combined effectively to cluster the severity of the sepsis in terms of mortality rate in an unsupervised manner with a soft monotonic behavior.

Table 3.1 summarizes the list of patients' statistics (mean, standard deviation, maximum, minimum, and mode). Two features related to demographics can be identified; the patient's gender and age. Additionally, similarly to a previous study [76], 15 lab test features are included. From these samples, we kept 52 for training and 10 for testing. To cover patients in the test set from the entire range of staying hours in the ICU, the data were sorted based on these staying hours, and 1 sample every 6 was excluded from the training set. Each sample contains 7 input time-series features representing the vital signs and is divided into windows with L_{window}=10 hours and step size S=1 hour, thus creating overlapping windows with 90% overlap. Those samples were normalized feature-wisely to the range [0,1] with min-max normalization according to the training samples. Then, the same statistical values were applied to the testing ones to avoid data leakage. It is noteworthy that any other required preprocessing step does not exist.

Table 3.1.: List of variables extracted by the MIMIC-III dataset. Vital signs represent the time-series inputs. Demographics and lab tests represent the supplementary data. The time feature was excluded from this table.

Category	Name	Mean	Std.	Max	Min	Mode	Unit
Demographics	Age	72.21	32.68	100	28	83	years
	Gender	44% Female	-	-	-	-	-
Lab test	Anion gap	13.55	1.13	21.0	13.0	13.35	mEq/L
	Bicarbonate	25.65	5.27	31.23	20.13	25.65	mEq/L
	Bilirubin	3.36	6.41	16.26	1.05	3.36	mg/dL
	Creatinine	2.24	1.78	8.60	0.60	1.5	mg/dL
	Chloride	105.20	6.55	121.0	86.00	104.0	mEq/L
	Glucose	145.03	62.26	376.0	62.00	134.0	mg/dL
	Hematocrit	32.59	5.90	49.0	22.30	27.5	%
	Hemoglobin	10.49	2.16	15.90	5.40	8.5	g/dL
	Lactate	2.44	2.14	3.12	1.75	2.44	mmol/L
	Platelet	235.05	155.28	311.12	178.13	235.0	$10^3/\mu L$
	Potassium	4.23	0.90	7.60	2.10	4.08	mEq/L
	PT	18.49	5.88	39.30	12.0	17.76	seconds
	Sodium	138.14	4.83	152.0	122.00	138.84	mEq/L
	BUN	36.97	20.71	108.0	11.00	29.85	mg/dL
	WBC	14.47	11.70	60.20	0.40	11.23	$10^3/\mu L$
Vital signs	Heart Rate	89.11	18.62	179.67	0.0	80.0	bpm
	Arterial BP [Systolic]	115.86	23.58	240.0	0.0	0.0	mmHg
	Arterial BP [Diastolic]	58.50	13.61	151.0	0.0	0.0	mmHg
	Respiratory Rate	21.56	7.40	75.11	0.0	0.0	bpm
	Temperature	37.46	0.92	41.15	34.6	37.0	°C
	SpO2	97.08	5.11	100.0	89.13	100.0	%
	GCS Total	10.44	3.67	15.0	3.0	15.0	-

The second examined dataset is NASA's C-MAPSS dataset which concerns propulsion systems (engines) and represents engineering applications with multivariate time-series sensor data. The C-MAPSS tool is responsible for generating this dataset. This tool models various engine fleet deterioration occurrences from an initial condition (baseline) to the point of failure, concerning the training data and a time period prior to the EOL in the test data. Each time series comes from a different engine thus the data can be considered from a fleet of engines of the same type. There are three operational settings (altitude, Mach number, and throttle resolver angle) that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise. Each engine operates normally at the start of each time series and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends prior to system failure. Since our focus is on systems that reach the EOL, we consider only the training set as our dataset to be split into train/test samples.

The subset named FD001 is used without excluding any of the sensor signals. The first two columns contain each engine's ID and deterioration time steps, the next three columns include the three engine's operational conditions and the rest 21 columns carry the sensor signals. Only the raw sensory information was kept, thus excluding the first 2 columns. The remaining signals can give an increasing, decreasing, or constant trend during the engine's deterioration which makes it tricky for the model to effectively cluster the severity of the damage in an unsupervised manner. Table 3.2 summarizes the mean, standard deviation, maximum, minimum, and mode, of each sensor.

Table 3.2.: List of variables extracted by the C-MAPSS dataset. The time feature was excluded from this table.

Name	Mean	Std.	Max	Min	Mode	Unit
Sensor 1 (Operational setting 1)	-0.0	0.0	0.01	-0.01	-0.0	*kft.*
Sensor 2 (Operational setting 2)	0.0	0.0	0.0	-0.0	-0.0	Mach
Sensor 3 (Operational setting 3)	100.0	0.0	100.0	100.0	100.0	°
Sensor 4	518.67	0.0	518.67	518.67	518.67	°*R*
Sensor 5	642.68	0.5	644.53	641.21	642.5	°*R*
Sensor 6	1590.52	6.12	1616.91	1571.04	1589.7	°*R*
Sensor 7	1408.89	9.0	1441.49	1382.25	1400.6	°*R*
Sensor 8	14.62	0.0	14.62	14.62	14.62	psia
Sensor 9	21.61	0.0	21.61	21.6	21.61	psia
Sensor 10	553.37	0.89	556.06	549.85	554.36	psia
Sensor 11	2388.1	0.07	2388.56	2387.9	2388.11	rpm
Sensor 12	9065.15	22.07	9244.59	9021.73	9046.19	rpm
Sensor 13	1.3	0.0	1.3	1.3	1.3	-
Sensor 14	47.54	0.27	48.53	46.85	47.47	psia
Sensor 15	521.41	0.74	523.38	518.69	521.66	pps/psi
Sensor 16	2388.1	0.07	2388.56	2387.88	2388.1	rpm
Sensor 17	8143.73	19.03	8293.72	8099.94	8138.62	rpm
Sensor 18	8.44	0.04	8.58	8.32	8.42	-
Sensor 19	0.03	0.0	0.03	0.03	0.03	-
Sensor 20	393.2	1.55	400.0	388.0	393.0	-
Sensor 21	2388.0	0.0	2388.0	2388.0	2388.0	rpm
Sensor 22	100.0	0.0	100.0	100.0	100.0	rpm
Sensor 23	38.82	0.18	39.43	38.14	38.86	lbm/s
Sensor 24	23.29	0.11	23.62	22.89	23.42	lbm/s

In subset FD001, there are 100 samples where the fault grows in magnitude until system failure. These samples are split into 90 training and 10 test trajectories. To test varying trajectory lengths the data were sorted and 1 sample every 10 was excluded from the training set and kept in the testing set. Each sample contains 21 input time-series representing the sensors and they are divided into overlapping windows with $L_{window}=10$ cycles and step size $S=1$ cycle (90% overlap). Similarly to MIMIC-III dataset, the data were normalized using min-max normalization according to the training samples to the range $[0,1]$, feature-wisely, and the same statistical values were applied to the testing samples. In this dataset, there are no other supplementary data besides the time feature.

The third dataset is the F-MOC dataset, that is, an experimental campaign developed in [23]. This experiment investigates the fatigue behavior of a unidirectional prepreg tape Hexply® F6376CHTS(12K)-5-35 laminate. The laminate is first manufactured and then cured in an autoclave as per manufacturer recommendations and specimens of standardized dimensions are obtained. Fatigue loading is applied using a Mechanical Testing System (MTS) controller on a bench fatigue machine. Images during pause intervals were captured using specialized cameras. The loading protocol involves cyclic loading with specified intervals and load transitions to analyze the laminate's fatigue behavior under varying stress conditions. The sensor system consists of two cameras for capturing image data and an acoustic emission system with a sampling rate of 2 MHz. The measurements were taken until the specimen's failure point. The acoustic emission low-level features were extracted by an AMSY-6 Vallen Systeme GmbH. From these features, the ones summarized in Table 3.3 were considered. The threshold value is defined at $50\ dB$, i.e. the acoustic emission signals that have an amplitude less than $50\ dB$ ($\approx 3.16\ \mu V$) were discarded.

Table 3.3.: The low-level features that are considered and extracted by the AMSY-6 Vallen Systeme GmbH.

Feature name	Unit	Description
Threshold	Decibel [dB]	Values below this threshold are discarded.
Amplitude	Volts [V]	The amplitude of the corresponding signal.
Duration	Seconds [s]	The duration that a signal constantly remains above the threshold.
Energy	$10^{-14}V^2s$ [eu].	Energy is the integral of the squared acoustic emission-signal over time
Counts	-	The number of positive threshold crossings of a hit.
Hit time	Seconds [s]	The absolute time when a hit is above the threshold.
Rise time	Seconds [s]	The time between the first threshold crossing and the maximum amplitude.

In this dataset, there are 7 trajectories of acoustic emission and image data representing 7 specimens, respectively. The lifetime, the number of images used, and the size of the acoustic emission and image data of each specimen are summarized in Table 3.4. To overcome memory issues, we kept only the data from the first camera and discarded the rest. Furthermore, we scaled down the image dimensions from a resolution of $[2048\times1024]$ to $[128\times64]$ via an average pooling filter. The synchronization process of the images and acoustic emission data is described in Appendix B.2.2. According to this process, it was chosen $L_{window}=6$ images with $S=3$ images. The corresponding variables for the acoustic emission are calculated based on the synchronization. Similarly to the previous datasets, the only preprocessing step is normalization to the range $[0,1]$. For the grey-scale image, this normalization is simply a division with the value 255 which corresponds to a pixel with a white color. For the acoustic emission signals, each sample was normalized feature-wisely, similarly to the two aforementioned datasets.

There are three types of data examined in this study; time series, static data, and time frames. Each trajectory is split into short overlapping windows of length L_{window} and step size S. Since the F-MOC dataset requires synchronization of the inputs as there is a conflict between active (DIC) and passive (acoustic emission) testing methods, the value of L_{window} is determined based on the sampling rate of the DIC process, which is 50 seconds. More details about synchronization can be found in Appendix B.2.2. Following the synchronization process, a configuration was established where six sequential images were integrated into a time frame, and concurrently, each acoustic emission signal was standardized to a length of 300 seconds. This pairing of a time

Table 3.4.: General characteristics of the F-MOC dataset.

Name	No. kept images	Lifetime (s)	Size of data (GB)	
			DIC	Acoustic
Specimen 1	1011	56,520	1.97	0.75
Specimen 2	168	14,380	0.39	0.21
Specimen 3	1073	59,600	2.09	0.79
Specimen 4	846	48,250	1.65	0.77
Specimen 5	480	29,950	0.96	0.41
Specimen 6	1257	68,810	2.45	0.97
Specimen 7	1384	75,160	2.70	1.08
Total			12.21	4.98

frame and its corresponding acoustic signal signifies a single window. The determination of subsequent windows involved an overlapping scheme, with the fourth image of the preceding window aligning with the first image of the succeeding window (thus, $S=3$). Consequently, the initial window comprised images 1-6, while the subsequent window included images 4-9, and so forth. Based on the synchronization process, a similar overlap is applied to the acoustic emission signals. The same procedure of overlapping windows is followed for the C-MAPSS and MIMIC-III datasets, without the synchronization step as only 1-dimensional signals exist. It is noteworthy that the hyperparameters L_{window} and S of the C-MAPSS and MIMIC-III datasets are not required to match those within the F-MOC dataset.

For each dataset, the percentage of overlapping can be calculated by the formula $(L_{window}-S)/(L_{window})100\%$. Depending on the position of the window into the trajectory, a value is assigned to the time feature. These values can be defined in any range with the only constraint of increasing monotonically with the constructed windows of the corresponding trajectory. A straightforward approach to defining the range of time feature values is to simply count the current number of windows that have been constructed and assign that value to t starting from $t=0$ for the first window of the trajectory, $t=1$ for the second, and so on. Unfortunately, this setup may give an unbalanced learning process if the trajectory lengths vary seriously. To mitigate this pitfall, an alternative approach is to calculate the average trajectory length L_{avg} given all the lengths of the training trajectories and then, apply a linear spacing for t depending on each trajectory length. In this regard, for each trajectory, the time feature is bounded in the range of $[0,L_{avg}]$. The intermediate values are then linearly spaced between those extremes according to the current trajectory length. In detail, for each trajectory of length L the time feature array is constructed to be $t=[0,\frac{L_{avg}}{L},\frac{2L_{avg}}{L},...,L_{avg}]$.

In summary, each sample should contain a window consisting of time series data and the corresponding scalar value of the time feature which constitutes one feature of the static data. Additional static data are considered for the MIMIC-III dataset, i.e.

demographic and lab test data, defined as supplementary data. The time series data are handled via LSTM layers while the time feature and the supplementary data are inserted in intermediate FC layers of the NN architecture and will be extensively discussed in the next section. The F-MOC dataset consists of overlapping windows of synchronized time-series and frames (3-dimensional data). In such case, the LSTM layers are extended to a combination of LSTM and 3-dimensional CNN (CNN3D) layers (see section 3.3.2 and Figure 3.3).

3.3.2. MODEL ARCHITECTURE

The concept of employing end-to-end feature extraction and then clustering utilizing a raw input data space of dimension X requires a transformation of those inputs into an D-dimensional embedding space Z^D, where D is typically much lower than the dimension of X, with a nonlinear mapping $f_\theta(X)=Z$, where f is a function approximator and θ its parameters. The extraction of valuable information from raw data entails the utilization of a complex function that involves intricate mathematical operations. ANN naturally emerges as a suitable choice for this purpose due to their theoretical function approximation properties and their demonstrated feature learning capabilities [77]. In the context of deteriorating systems, the input data typically comprises trajectories, predominantly in the form of time series. The first layer of the ANN should be responsible for extracting time-related information, thereby making the LSTM [78], that is, a recurrent layer an appropriate candidate. Subsequently, the remaining layers can consist of stacked Fully Connected (FC) layers. Notably, supplementary data, i.e. non-sequential data that give additional unique information for each sample, can be introduced into one of these FC layers, enabling their integration into the DSMC model and tackling the unique challenges of multi-modal data.

The DSMC model is ultimately a modified encoder that simultaneously extracts prognostic-related features and clusters those features accordingly. Training the DSMC model requires a two-stage training process; a pre-training of the encoder via a deep AE, and a following training process of the encoder that enables its output, namely, the Z-space, to assign cluster labels to the incoming input data. In the first stage, the AE setup is shown at a high level in Figure 3.2a. It consists of two modules stacked with multiple layers each. The first module, the feature extractor module, hiddenly extracts information from the input data via a set of LSTM layers, as shown in Figure 3.2b. It is possible to cover any kind of sequential data (time series, frames, etc.) by adapting the first layer(s) of the feature extractor module according to the examined domain. For instance, if the input contains 1-dimensional time series data then a typical LSTM layer is enough to capture the temporal hidden features of the sequence. If the input contains image sequences, such as sequential CT scans [79] or time-resolved segmentations [80], then a Convolutional LSTM or a 3-dimensional CNN (CNN3D) could be used. Importantly, a mix of 1, 2, and 3-dimensional sequential data could be provided at once by combining the abovementioned types of layers inside the feature extractor module. This efficiently enables the applicability of the proposed model to any kind of sequential input data.

The second module, referred to as the monotonic module, comprises a stacked configuration of FC layers with an incorporated monotonic modification, serving as a pivotal factor in extracting monotonic-related features. To establish this monotonicity it is important to apply a hard constraint between the time feature t and the output z. For each sample i, we want the gradients of z_i with respect to t_i to be non-negative, i.e. $\frac{\partial z_i}{\partial t_i} \geq 0, \forall x_i \in X$. For an MLP, it is proven in [71] that the output is increasing (decreasing) monotonically with respect to input, if and only if the weights and the activation functions of input, output, and intermediate layers are always increasing (decreasing). The corresponding biases can have any real value since they do not affect the outputs' gradients. In this regard, by employing an exponential operation on the weights of each neuron in every layer, ranging from the input layer to the output layer of the monotonic module (see Figure 3.2a), we ensure the desired monotonicity. This approach to enforcing monotonous constraints allows the weights to assume any real value during the learning process, without imposing any limitations on the weight space. Since these constraints are applied inside the structure of the network during the forward pass, the typical backpropagation algorithm can be used by satisfying the converging properties of the ANN. Consider the typical formulation of a neuron to be:

$$r_v = b_v + \sum_u w_{uv} g(r_u). \tag{3.1}$$

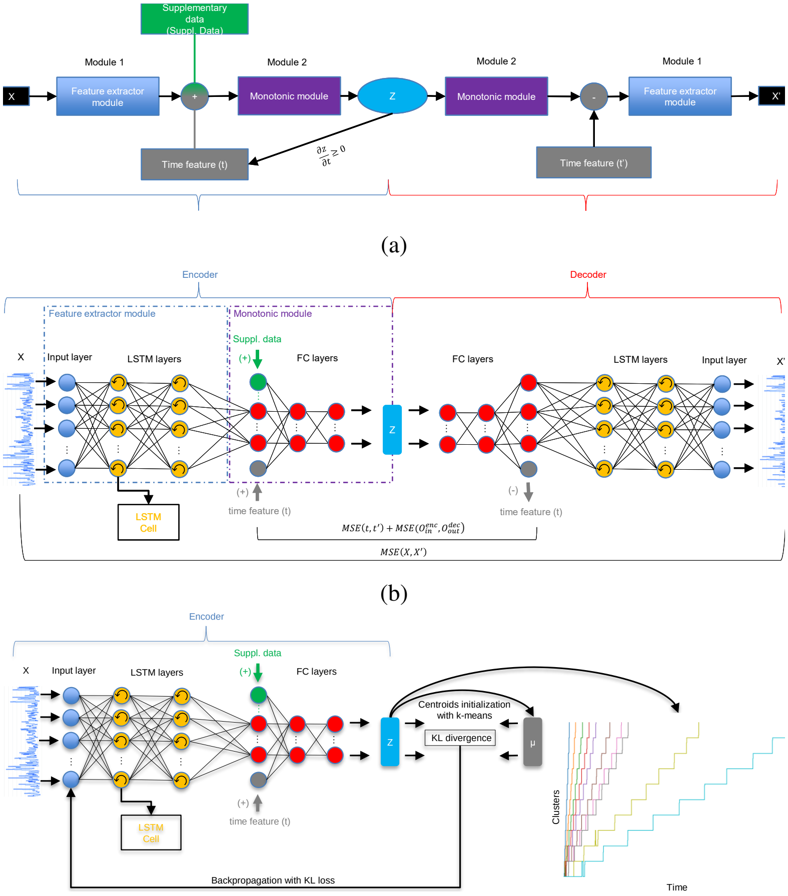
Where b is the bias, w the weights that come from neuron u of the previous layer and contribute to the current layer's neuron v, and $g(.)$ the activation function, hence $g(r_u)$ is the output of the neuron u that comes from the previous layer (i.e. the input of the current layer v). By applying the desired non-negative monotonic constraints to the neurons we have:

$$r_v = b_v + \sum_u e^{w_{uv}} g(r_u),\ g \geq 0,\ g' \geq 0\ everywhere. \tag{3.2}$$

The gradients with respect to the bias remain the same while the gradients concerning the weights used for backpropagation are converted to:

$$\frac{\partial Loss}{\partial w_{uv}} = g(r_u) e^{w_{uv}} \frac{\partial\, Loss}{\partial r_v}. \tag{3.3}$$

There are two crucial advantages of using this approach for achieving monotonicity. Firstly, monotonicity can be optionally applied in a sub-group of FC layers. Consequently, the rest of the ANN architecture which may contain other kinds of layers than FC could remain unchanged. Secondly, it is possible to have soft monotonicity between inputs and outputs, simply by applying an exponential operation only to the weights concerning the input which is under the examined constraint. For input variables such constraints are not required, thus the weights may remain unchanged to allow more flexibility. Both of the aforementioned attributes are desirable for our architecture since we need monotonic constraints only with respect to the time feature which, simultaneously, should be



(a)

(b)

(c)

Figure 3.2.: (**a**) Module-level architecture of the proposed AE model and (**b**) detailed architecture of the proposed AE model. (**c**) Detailed architecture of the DSMC model used for monotonic clustering.

inserted into an intermediate layer. As a result, monotonic relationships are exclusively attained within the monotonic module, specifically within the neurons influenced by the time feature for generating the output. This clarifies why, despite the existence of a hard monotonic constraint between Z and t, a soft monotonic behavior between Z and X is observed, ultimately leading to the desired soft monotonic clustering.

Except for the main contributing layers of the DSMC model shown in Figure 3.2b and Figure 3.2c, between each layer, Dropout, and parametric Batch-normalization (BN) layers are involved. An important observation is that the gradient outputs of BN layers can have a heavy impact on the monotonic constraints during backpropagation. To address this issue, the same exponential function was applied to the weights of each BN layer as applied similarly to the rest of the layers of the monotonic module described in Equation 3.2, without affecting the corresponding biases. After the final LSTM layer of the feature extractor, a flattening layer without any trainable parameters was applied to transform the array into 1-dimensional before inputting it to the FC layers. The activation function used after the LSTM layers corresponded to Tanh, while a Softplus function was used after every FC layer to ensure a positive monotonic increase. It should be noted that the FC layers that are applied before the monotonic module can be followed by any kind of activation function. The same layers are applied to the decoder.

Together, these two modules form both the encoder and decoder components. The decoder progressively increases the dimensionality through its layer-by-layer construction and is responsible for simultaneously reconstructing the input sequential data, the supplementary data, and the time feature. Once the data exits the monotonic module, the time feature is no longer required and is subsequently removed from the decoder. Given the vital role played by the supplementary data in the learning process, the AE is allowed to utilize this information implicitly, without any modifications.

Having the encoder pretrained via the AE setup, the deep clustering part which is based on [68] is taking place. In this approach, deep clustering seeks to cluster the input data points into K clusters by simultaneously learning the parameters θ of the ANN and the cluster centers $\{\mu_{\mathrm{j}} \in Z\}_{j=1}^{K}$. In this regard, each output $z \in Z^{D}$ of the encoder is fed to a k-means clustering algorithm for initializing the centroids $\mu_j^d, d \in [0,1,\ldots,D]$. The process of centroid initialization is applied only once for the entire training dataset. Subsequently, the encoder undergoes additional training with the objective of bringing the encoder output z and the corresponding centroid μ closer to each other. This is achieved through the computation of a soft assignment probability distribution q that establishes the relationship between them and by the utilization of an auxiliary target distribution p. By minimizing the Kullback-Leibler (KL) divergence between q and p, the goal is to make these distributions similar to each other. The probability distribution q corresponds to the Student's t-distribution which measures the similarity between z and μ as follows:

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2/\nu\right)^{-\frac{\nu+1}{2}}}{\sum_{j'}\left(1 + \|z_i - \mu_{j'}\|^2/\nu\right)^{-\frac{\nu+1}{2}}}. \tag{3.4}$$

Where ν is the degrees of freedom of the Student's t-distribution and in an unsupervised setting should be fixed to $\nu=1$. Similarly to [68], the target distribution is chosen to be:

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_{j'} q_{ij'}^2/\sum_i q_{ij'}}. \tag{3.5}$$

Then the loss function (KL-loss) for training the deep clustering is:

$$Loss^{DSMC} = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.6}$$

The primary concept behind this setup is to adopt a self-learning framework for the model, allowing it to autonomously learn the assignments to clusters with both high and low confidence. The model then focuses on enhancing the assignments that exhibit low confidence. The optimization proceeds by jointly optimizing the ANN's parameters θ and the cluster centroids μ_j using the Stochastic Gradient Descent algorithm with momentum and applying a standard backpropagation with respect to θ. The gradients are computed as:

$$\frac{\partial Loss^{DSMC}}{\partial z_i} = \frac{\nu+1}{\nu} \sum_j \left(1+\frac{\left\|z_i-\mu_j\right\|^2}{\nu}\right)^{-1} \times \left(p_{ij}-q_{ij}\right)\left(z_i-\mu_j\right). \tag{3.7}$$

$$\frac{\partial Loss^{DSMC}}{\partial \mu_j} = -\frac{\nu+1}{\nu} \sum_i \left(1+\frac{\left\|z_i-\mu_j\right\|^2}{\nu}\right)^{-1} \times \left(p_{ij}-q_{ij}\right)\left(z_i-\mu_j\right). \tag{3.8}$$

During the evaluation of the model, each sample x_i from the input data space X is transformed by the model into the embedding space z^i_d which in turn is assigned to a cluster as follows:

$$\text{cluster}^i = \max_{d \in D}\left(\arg\min_j\left(\left\{\mu^i_{jd}\right\}_{j=1}^{K} - z^i_d\right)\right). \tag{3.9}$$

where,

$$z^i_d = f_\theta(x_s), d \in [0, 1, \dots, D] \tag{3.10}$$

In the expression above, the inner operation is D-dimensional representing D cluster assignments for the same sample i. This should be reduced to one cluster assignment. To prioritize safety, the maximum assignment (outer operation) was selected as the final prediction. As a result, although there may be an overestimation of the deterioration, the approach significantly mitigates the risk of reaching the end of life (adopting a risk-averse policy [81, 82]).

ADAPTATION OF THE MODEL'S ARCHITECTURE FOR THE F-MOC DATASET

The general architecture of the model remains the same, with the only alternation being in the feature extractor module of the AE. Since there are both time-series (acoustic emission) and three-dimensional (sequences of images) data, the LSTM layers do not suffice to produce the inputs H^{enc}_{in} which are fed to the monotonic module. Consequently, a stack of CNN3D layers is added parallel to the LSTM layers from which hidden features $H^{enc}_{in,image}$ related to the sequential images are extracted and fused with the hidden features $H^{enc}_{in,acoustic}$ related to the acoustic emission signals. An alternative to the CNN3D layers could be a Convolutional LSTM layer [83], i.e. a combination of CNN and LSTM layers. However, due to the layer's increased computational power that emerged from its recurrent nature, the CNN3D layer remains the best option. Then, those features are concatenated and passed through an FC layer to produce the required dimensionality of H^{enc}_{in}.

This process is depicted in Figure 3.3. The feature extractor of the decoder performs the reverse process of the encoder, thus having the same number of layers and dimensions. This time the input of the encoder and the reconstructed input (output of decoder), X and X' respectively, is a set of sequential images representing a time frame and a window of acoustic emission data. Subsequent to minor adjustments in both the clustering process and the prognostic model, the number of clusters has been expanded from `10` to `30`, reflecting the increased trajectory lengths. This modification aims to enhance the model's capability to capture more extensive information pertaining to the progression of damage in the structure. Table 3.5 summarizes the hyperparameters related to LSTM and CNN3D. The hyperparameters of LSTM remain unchanged for all datasets except H^{enc}_{in} which is optimized by the Bayesian optimization per dataset.

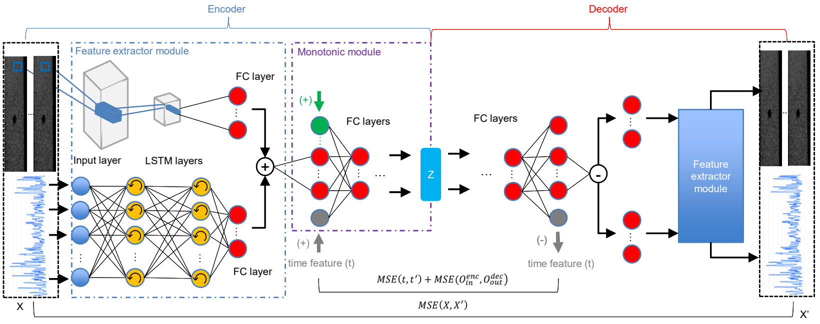


Figure 3.3.: The redesigned architecture of the model. Only the feature extractor module is affected. As before, the decoder still performs the reverse process of the encoder, thus the loss function remains the same.

3.3.3. TRAINING THE DSMC MODEL

To train the AE according to the proposed architecture, the typical reconstruction loss (original and reconstructed input) was modified with two additional terms; the reconstruction of time, and the reconstruction of the monotonic module. Consider the outcome of the input layer of the encoder's monotonic module and the outcome of the output layer of the decoder's monotonic module to be O^{enc}_{in} and O^{dec}_{out}. respectively.

Then, the loss function used for training the AE is given below:

Table 3.5.: Hyperparameters' values for the LSTM and CNN3D layers. The LSTM layers remain unchanged for all datasets with the corresponding values of H_{in}^{enc}.

NN type	Name	Value
LSTM	No.layers	2
	Hidden size (for both layers)	H_{in}^{enc}
	Bidirectional	No
CNN3D	No. layers	3
	Hidden channels (per layer)	[48, 96, 48]
	Kernel size	(3, 3, 3)
	Strides	(1, 2, 2)
	Padding	1

$$\mathrm{Loss}^{AE} = \mathrm{MSE}\left(X, X'\right) + \alpha \cdot \left[\mathrm{MSE}\left(t, t'\right) + \mathrm{MSE}\left(O_{\text{in}}^{\text{enc}}, O_{\text{out}}^{\text{dec}}\right)\right]. \tag{3.11}$$

Where $\mathrm{MSE}(.)$ is the Mean Squared Error, α is a tunable hyperparameter and X', t' are the reconstructed input and time, respectively. Ultimately, the hyperparameter α weights the importance that should be given to the monotonic behavior of the clustering. When the AE is trained, we keep only the encoder as a pretrained module, initialize the centroids and further train it using a combination of $Loss^{AE}$ and $Loss^{DSMC}$ as follows:

$$\mathrm{Loss} = \mathrm{Loss}^{DSMC} + \beta * \mathrm{Loss}^{AE}. \tag{3.12}$$

Where β is another tunable hyperparameter weighting the contribution of the $Loss^{AE}$. The reason for reusing the $Loss^{AE}$ for the clustering process is that we need to keep the soft monotonic nature of the embedding space, which would have gradually vanished otherwise.

3.3.4. BAYESIAN OPTIMIZATION FOR HYPERPARAMETER TUNING

Pre-training the encoder via the AE model and then further training it inside the DSMC model in an end-to-end manner poses significant challenges due to the unsupervised learning nature and the presence of multiple loss terms. The effectiveness of these models heavily relies on the setting of their hyperparameters. Manually tuning these hyperparameters can be a time-consuming and labor-intensive task, demanding substantial effort. In this regard, the Bayesian Optimization (BO) algorithm is chosen as the optimization algorithm for tuning the most important hyperparameters of the two models. BO requires a target function, namely the objective function, to be maximized during the optimization process. This framework is well-suited for ANN as it relaxes the constraint of solely relying on continuous loss functions for training purposes. Consequently, the ANN can be trained with its own continuous loss function and its hyperparameters could be tuned with a more efficient, non-continuous objective function.

Instead of utilizing the proposed $Loss$ (in its negative form, as the optimization process involves maximization), there is a flexibility to choose any function related to the clustering task that exhibits favorable maximization properties as the objective function for the BO algorithm. In the current study, through manual hyperparameter tuning, we observed substantial transitions from lower clusters to higher ones, while simultaneously witnessing a significant number of sequential values belonging to the ultimate cluster, particularly in larger trajectories. These observations are justified and come from the trade-off between the contribution of the time feature and the input time-series data to the cluster predictions. Therefore, the BO algorithm searches for hyperparameters generated by the chosen objective function to effectively mitigate these limitations. This function should contain a term that counts, for each trajectory, the number of cluster jumps and the sequential timesteps of staying to the ultimate cluster, only if it exceeds t_{avg}. However, it becomes evident that by utilizing an objective function incorporating the two aforementioned observations, the optimal solution entails clustering the trajectory in a balanced and fully monotonic manner with respect to the available labels. This implies giving utmost importance solely to the time feature while letting the DSMC model completely disregard the time-series input data. Consequently, it is necessary to introduce an additional term, assigned with a lower weight, to the objective function. This term (first part of the objective function, i.e. $0.6 * 1_{\mathrm{d_{c_i}} < 0}$) serves the purpose of promoting some backward transitions to previous clusters, ultimately facilitating the desired soft monotonicity. In this regard, the optimization task of the BO algorithm with the chosen objective function is given below:

$$\underset{\mathrm{h}}{\operatorname{argmax}} \left(\frac{\sum_{\mathrm{j}=0}^{\mathrm{N_{traj}}} \sum_{\mathrm{i}=1}^{\mathrm{L^{j}_{traj}}-1} \left[0.6 * 1_{\mathrm{d_{c_i}} < 0} - \left(1_{|\mathrm{d_{c_i}}| > 1} + 1_{\substack{\mathrm{d_{c_i}} = 0 \\ \mathrm{c_i} = \mathrm{K} - 1}} \right) \right]}{N_{traj}} \right). \tag{3.13}$$

where N_{traj} is the number of the training trajectories, $1_{condition}$ is equal to one if the condition is satisfied, $\mathrm{d_{c_i}}$ is the difference between the sequentially predicted labels c_i, and c_{i-1}, at timesteps i and $i-1$, respectively, which is given below:

$$\mathrm{d_{c_i}} = \mathrm{c_i} - \mathrm{c_{i-1}} \tag{3.14}$$

and $h = \{L_{window}, S, Z, H_{in}^{enc}, lr^{AE}, lr^{DSMC}, epochs^{AE}, epochs^{DSMC}, \alpha, \beta, drop_{rate}\}$ are the tunable hyperparameters that simultaneously maximized the objective function. Here, the hyperparameter H_{in}^{enc} corresponds to the number of neurons of the last hidden layer of the encoder's feature extractor module, and $drop_{rate}$ is the dropout rate. All of the hyperparameters h used in the Bayesian optimization plus four additional ones that were manually decided, including L_{window}, S, $batch^{AE}$, and $batch^{DSMC}$, are stored in Table 3.6 for each case study. These values were optimal after 100 iterations. The rationale for selecting these search ranges is grounded in trial-and-error, initially informed by the default values used in previous studies related to the MIMIC-III [84, 85] and C-MAPSS [86, 87] datasets. For the F-MOC dataset and the newly introduced in this work hyperparameters, only a trial-and-error approach was followed. Each range was deemed appropriate as long as the Bayesian optimization identified a hyperparameter that was not positioned near one of the extremes.

Table 3.6.: Hyperparameter search ranges and final values optimized by the Bayesian optimization algorithm for each dataset. Both the automatically and manually tuned hyperparameters are included.

Bayesian optimization	Hyperparameter	Search range	Optimized value		
			MIMIC-III	C-MAPSS	F-MOC
Yes	Z	[3, 32]	8	4	9
	H_{in}^{enc}	[32, 128]	116	123	48
	lr^{AE}	$[10^{-4}, 10^{-3}]$	$1.2*10^{-3}$	$5*10^{-4}$	$5*10^{-3}$
	lr^{DSMC}	$[5*10^{-5}, 10^{-3}]$	$6*10^{-4}$	$3*10^{-4}$	$9*10^{-4}$
	$epochs^{AE}$	[50, 200]	179	105	112
	$epochs^{DSMC}$	[10, 30]	26	23	17
	α	[0.7, 2.2]	1.795	0.772	2.0
	β	[0.01, 5.0]	1.720	2.756	0.964
	$drop_{rate}$	[0.1, 0.4]	0.2	0.3	0.14
No	L_{window}	-	10	10	6
	S	-	1	1	3
	$batch^{AE}$	-	128	32	128
	$batch^{DSMC}$	-	128	32	128

3.3.5. PROGNOSTICS

Although this study focuses on monotonic feature extraction and clustering to produce simple expressions that can be easily fed to any prognostic model to make predictions, for comprehensiveness, the Hidden Semi-Markov Model (HSMM) [88] was utilized for the prognostic task, For the developed HSMM, a Gaussian distribution is assumed for the observation process, the degradation process is non-parametric, and the number of hidden states is set to 8. For the estimation of the parameters of the model the Expectation-Maximization (EM) algorithm is applied. The E-M algorithm iterates between two steps: the E-step, where the expected values of the hidden states are computed based on the current model parameters using two auxiliary variables (the forward α and the backward variable β), and the M-step. During the M-step, the model parameters are updated to maximize the likelihood of the k observation sequences O (in this case, the extracted cluster trajectories from the DSMC model) by incorporating the expected hidden states obtained from the E-step. This iterative process continues until the convergence tolerance of 0.5 is met, therefore refining the parameter estimates and improving the model's fit to the data, based to the following equation:

$$
\begin{aligned}
L(\lambda, O^{(1:K)}) = \prod_{k=1}^{K} P(O^{(k)}|\lambda) \xrightarrow{L'=log(L)} \\
L'(\lambda, O^{(1:K)}) = \sum_{k=1}^{K} log(P(O^{(k)}|\lambda)) \\
\lambda^* = arg\max_{\lambda}\left(\sum_{k=1}^{K} log(P(O^{(k)}|\lambda))\right)
\end{aligned}
\tag{3.15}
$$

Once the model is trained, meaning that the degradation and observation processes are estimated, the Viterbi algorithm [89] is utilized to estimate the most likely sequence of hidden states that explains the observed data (trajectories). This process is referred to as decoding. It is worth noting that the Viterbi algorithm does not require the entire observation sequence to estimate the most likely sequence of hidden states; rather, the observation sequence up to time t is used to estimate the most likely state at time t. Thus, it provides the possibility to estimate the sequence in the testing phase, where at each time t, the observations up to t are used. The complete definition of the model, the parameter estimation and the decoding procedure can be found in [88] and it is also presented in Appendix B.3 for completeness.

The prognostic measure can then be applied, considering the estimated state sequence. [88] introduced a time-dependent prognostic measure as follows:

$$
\begin{aligned}
RUL_i^t = d_{i,i} \cdot \left(D_i(d-\tau) + \sum_{k=i+1}^{N-1} D_k(d) + \mathcal{N}(1,\epsilon) \right) \\
+ d_{i,i+1} \cdot \left(\sum_{k=i+1}^{N-1} D_k(d) + \mathcal{N}(1,\epsilon) \right)
\end{aligned}
\tag{3.16}
$$

$D_i(d)$ represents the Probability density function (PDF) evaluated in the probability of transition to the same state i. The variable τ is the time spent in the current state i. Therefore, the term $D_i(d-\tau)$ represents a shift in the pdf making this RUL expression time-dependent. Then, The variables $d_{i,i+1}$ and $d_{i,i}$ are defined as follows [90]:

$$d_{i,i+1} = P(d \le \tau | S_t = i) \tag{3.17}$$

$$d_{i,i} = 1 - d_{i,i+1} \tag{3.18}$$

The result of the prognostic measure is the pdf of RUL per time step. Therefore, the confidence intervals can easily be obtained by calculating the cumulative density function (CDF) and, later, choosing the confidence level, in this case, 95%.

3.4. RESULTS

In this section, firstly, the results concerning the MIMIC-III and C-MAPSS datasets are presented and compared. Next, we justify that the model's behavior is compatible with the aforementioned theory and interpret its behavior. Finally, the results of the F-MOC dataset, which represents a more complicated dataset, are discussed.

The DSMC model was trained on a single GPU (NVIDIA GeForce RTX 2080). The entire training process alongside the hyperparameter tuning via the Bayesian optimization is approximately 8 hours for the MIMIC-III and C-MAPSS datasets, whilst the computational time is increased substantially for the F-MOC dataset up to approximately 110 hours. This arises because BO involves conducting 100 iterations over the hyperparameter space, necessitating the training of the model anew in each iteration. Modifying the number of iterations has the potential to reduce computational effort, but it may concurrently result in a decrease in accuracy.

3.4.1. CLUSTERING RESULTS AND SURVIVABILITY ANALYSIS OF THE MIMIC-III AND C-MAPSS DATASETS

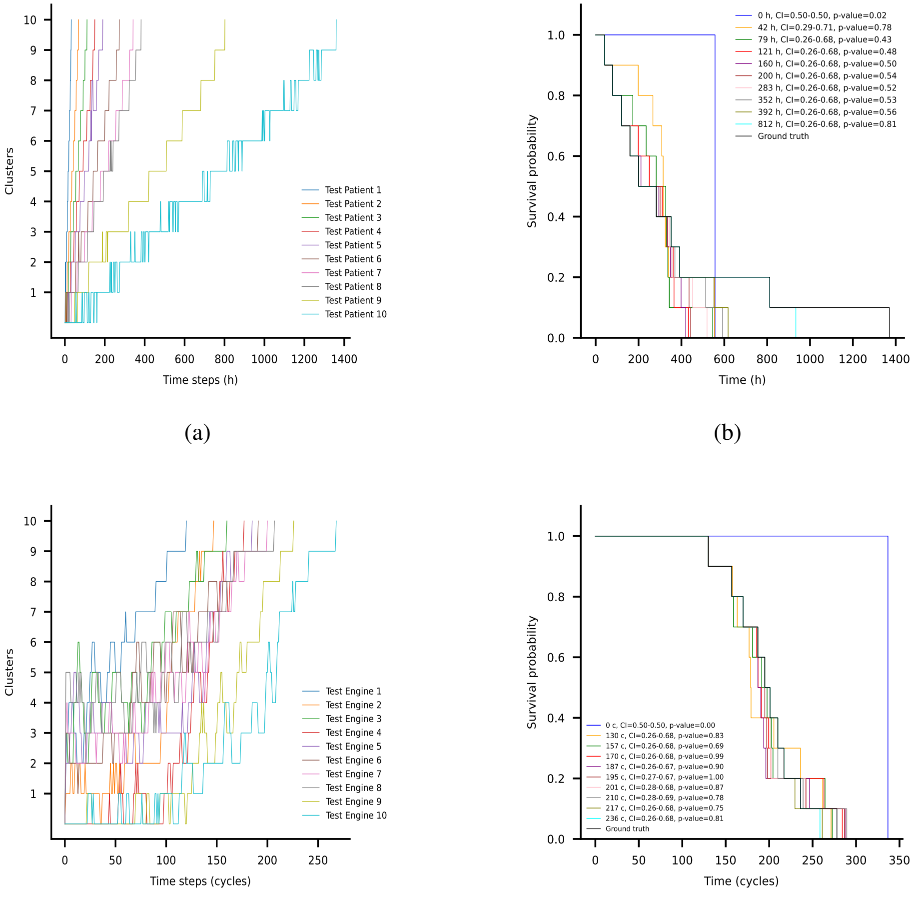
Running the DSMC model without BO took approximately 20 minutes for each of the first two datasets. The convergence of the training and validating loss corresponding to the reconstruction and time losses, respectively, is shown in the first four subfigures of Figure B.3. Running the training process with different weight initialization may produce a variety of loss values. Therefore, the reproducibility of the training process was presented via Table B.1 which depicts the mean and standard deviation of the training and validating losses. These statistics were produced by running the entire training process 10 times after initializing the NN's weights via a Uniform distribution.

After pretraining the encoder at the first stage of the DSMC model's training, the clustering process takes place, which is the second stage of training. The clustering results for each of the test trajectories are presented in Figure 3.4 for the two underlying datasets. It should be highlighted that a higher cluster prediction within the context of this study corresponds to a state that is more proximate to the most severe condition. This is due to the introduced monotonicity based on time which is ever-increasing, thus a higher cluster value is associated with a larger timestamp. Specifically, in the case of the MIMIC-III dataset, a higher cluster indicates a state that is closer to a 100% mortality rate. Similarly, for the C-MAPSS dataset, a higher cluster assignment signifies a state that is closer to the structure's EOL. In the context of both datasets, a deliberate selection of 10 clusters has been made, each associated with labels ranging from 0 to 9. This strategic choice has been made with the primary intention of elucidating the inherent soft monotonicity present within the cluster assignments. The rationale behind opting for precisely 10 clusters lies in the desire to avoid excessive complexity that would render the resulting clustering solution less comprehensible and interpretable. The introduction of a greater number of clusters would inadvertently introduce additional fluctuations that could potentially obscure the underlying patterns, diminishing the clarity of the analysis. Another cluster label equal to 10 is added at the end of each trajectory indicating the last data point of that trajectory. This is to explicitly provide a physical meaning to the last value representing the EOL and, by no means, is used during the testing phase where the last data points are unseen and unknown.

In relation to the MIMIC-III dataset, it is observed in Figure 3.4a that the labels assigned to the clusters exhibit a monotonic increase, with few exceptions in specific instances where there is an observed improvement in the patient's health. Given that the model is correctly trained, this decrease in cluster labels can be attributed solely to changes in the corresponding vital signs. From a data-centric perspective, the observed anomaly in the cluster labels can be attributed to irregularities in one or more of the provided input values corresponding to vital signs. Such anomalies may arise from either the presence of noisy data within the dataset or specific treatments administered to the patients. The latter scenario aligns with medical reasoning, as certain treatments can introduce anomalies in the monotonic behavior of the clusters. Consequently, by assuming that the input data is devoid of noise, we posit that a sudden decrease in cluster labels is closely associated with potential self-recovery or temporary improvement in a patient's condition resulting from a given treatment. However, establishing a direct correlation between the treatment and vital signs measurements is challenging. In preceding research [91], it was observed that the prompt initiation of antibiotic treatment could potentially result in significant adverse effects, including elevated mortality rates. Nevertheless, even if such a correlation could be identified, the accurate determination of the specific time delay required for the effects of the treatment to manifest in the patient under examination remains uncertain.

The prognostic results are presented by the Kaplan-Meier curve [92], shown in Figure 3.4b for the true survivability of the testing population and the corresponding 0h, 42h, 79h, 121h, 160h, 200h, 283h, 352h, 392h, and 812h predicted survival plots of the same population, respectively. These hours correspond to the time each patient stayed in the ICU until mortality. Hence, these curves indicate the prediction improvement towards the ground truth as more information is available with the elapsed time. Notably, except at 0h where no information is available and the predictions for all the patients are based only on the mean mortality of the training set, every other survival curve becomes increasingly more informative. Even though the right outlier is not predicted accurately, the prediction provided is mildly conservative and thus not harmful to the model's overall safety. Since these plots are interconnected based on time, the corresponding p-values (log-rank test) are expected to be high, thus being unable to reject the null hypothesis that these plots are similar. This phenomenon is occurring as expected, and for the sake of thoroughness, the p-values between the ground truth and each of the other plots are provided, respectively.

In contrast to the majority of related studies [93–96] that approach mortality prediction as a classification problem and assess their model's performance using receiver operating



(a) (b)

(c) (d)

Figure 3.4.: **a.** Clustering results for the MIMIC-III dataset. (**b**) Kaplan-Meier curves for the MIMIC-III dataset ('h' stands for 'hours'). (**c**) Clustering results for the C-MAPSS dataset. (**d**) Kaplan-Meier curves for the C-MAPSS dataset ('c' stands for 'cycles').

characteristic curves as diagnostic tools, we maintain its inherent nature and consider it as a regression process. Consequently, it is shown in Figure B.1 the survivability rates at 0%, 25%, 50%, and 75% of the corresponding trajectory lengths for 4 of the test patients, notably for patients 6, 7, 8, and 9. These curves alongside the corresponding reliability curves discussed below concerning the C-MAPSS dataset are derived from the utilization of the HSMM outcomes as described in [97].

Regarding the clustering outcomes obtained from the C-MAPSS dataset, Figure 3.4c reveals the presence of significant fluctuations in the assignments of clusters.

These fluctuations are observed in a reasonable manner. The presence of noisy sensor measurements within the C-MAPSS dataset, coupled with the challenge of effectively integrating sensor values that exhibit constant, monotonically increasing, and monotonically decreasing trends, accounts for the observed fluctuations in cluster assignments. Given that the model is trained in an unsupervised manner, it does not explicitly learn to disregard non-relative information as in supervised learning setups. Instead, it learns to leverage the entirety of the available information while adhering to the initial soft monotonic constraint in order to solve the task. As a result, the model's predictions should rationally include such fluctuations since, in some cases, the noisy raw data by no means reflect any actual increase in the system's health, which mirrors real-case scenarios.

Additionally, the prognostic results produced by HSMM are presented for the C-MAPSS dataset via the survival function estimated by the Kaplan-Meier method in Figure 3.4d. Similarly to the MIMIC-III dataset, the corresponding 0, 130, 157, 170, 187, 195, 201, 210, 217, and 236 cycles predicted survival plots of the testing population are presented representing the lifespan of each testing engine. Because there are no outliers in this dataset, the predictions are herein much closer to the ground truth. This finding provides evidence that the presence of various fluctuations in the clustering results does not hinder the prediction of reliability. On the contrary, the satisfactory outcomes can be attributed to the incorporation of soft monotonicity, which effectively captures the true information inherent in the input data rejecting the apparent measurement noise. Finally, in engineering, the effort is given to the reliability of the predictions, therefore the reliability curve extracted by the HSMM is utilized and shown in Figure B.2 for the 0%, 25%, 50%, and 75% of the corresponding trajectory lengths for 4 of the test engines (engines 6, 7, 8, and 9).

3.4.2. BENCHMARKING

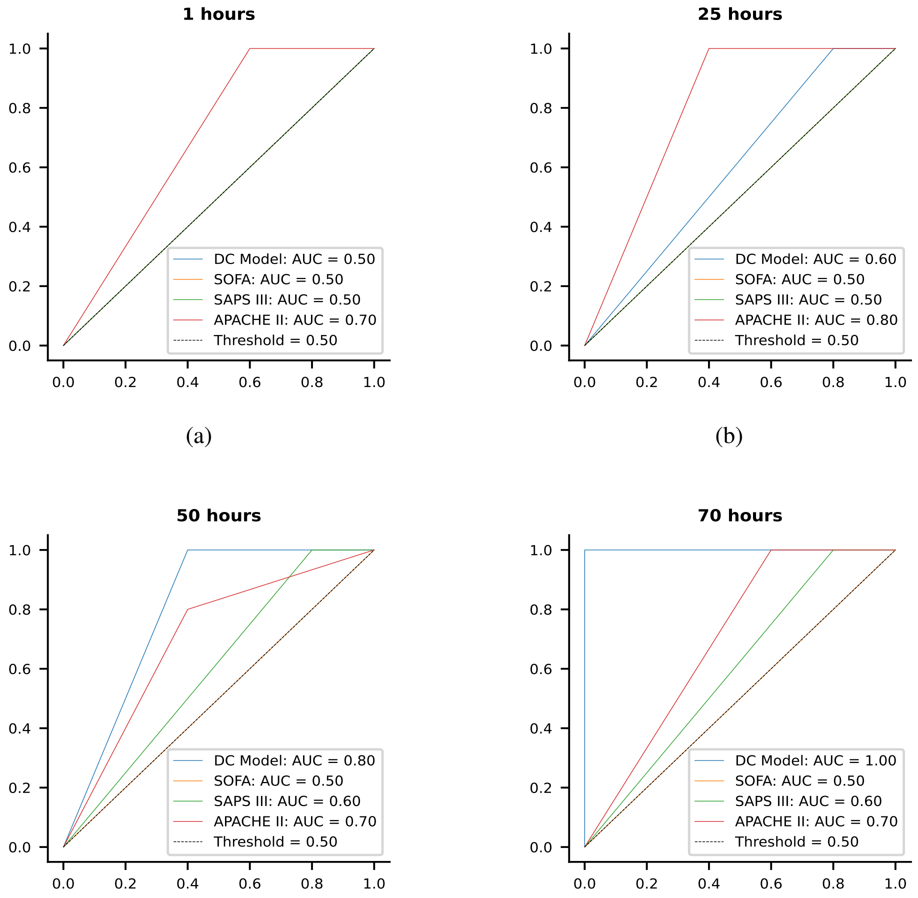
In this section, a comprehensive benchmarking analysis is provided to evaluate the a) performance, and b) the robustness of our proposed DSMC model, via the MIMIC-III and C-MAPSS datasets, correspondingly. Concerning the MIMIC-III dataset, comparing the results with other works is challenging since a part of the dataset is utilized accordingly for each specific case study. Indeed, as highlighted in [98], there exists a large heterogeneity in studies that makes it hard to compare and reproduce results. Additionally, in light of the author's comprehensive review and as mentioned in previous survey works [99, 100], it is evident that none of the prevailing methodologies has hitherto addressed the present dataset in the context of a regression paradigm. More importantly, to the best of the author knowledge, no prior work exists on feature extraction specifically related to prognostics, as existing methodologies primarily focus on directly predicting mortality rates [93–96, 101–103], which is not the primary objective of the DSMC model. As a corollary, it becomes imperative that the outcomes of the regression-oriented task be transmuted into a classification format solely for the purpose of benchmarking via well-known deterministic scoring systems. Consequently, the dataset has been streamlined into a classification framework, wherein patients are categorized based on the binary outcome of survival or mortality within a predetermined horizon. This horizon conventionally encompasses a span of 8 days (192 h), reflecting the survivability duration within the confines of the ICU for 50% of the examined population, thus having a perfectly balanced binary classification to evaluate.

The resultant classification outcomes are juxtaposed against the evaluations rendered by three widely recognized scoring systems routinely employed by healthcare practitioners, including Sepsis-related Organ Failure Assessment (SOFA) score [104], Simplified Acute Physiology Score (SAPS III) [105], and APACHE II [106] score. All of these scores can be used for an estimation of the risk of mortality during the ICU stay. Figure 3.5 shows for different timesteps (1, 25, 50, and 70 hours after each patient's entrance to the ICU) the comparison between the Receiver Operating Characteristic (ROC) curves of each benchmark with the DSMC model. Table 3.7 presents the corresponding Area Under Receiver Operating Characteristic (AUROC) scores alongside the precision, recall, and F1-score metrics. It becomes evident that the performance of the DSMC model exhibits challenges during the initial timesteps, primarily attributable to the insufficient data acquisition for generating precise predictions. In contrast, the APACHE II scoring system demonstrates a more natural aptitude in these early stages of prediction. Notably, both the SOFA and SAPS III scoring systems fail to yield substantial predictive value. This outcome can be attributed to the utilization of unprocessed and noisy data, factors that notably impede the efficacy of these deterministic scoring systems. Clearly, though, the DSMC model prognostic capabilities outperform those benchmarks by far when enough data are available, with AUROC, precision, recall, and F1-score being 0.80, 0.71, 1.0, and 0.83, respectively after 50 hours of stay in ICU. Impressively, after 70 hours of data acquisition, our model achieves the ultimate performance in all of the metrics, i.e. all of the patients are classified correctly, highlighting its superior performance.

Table 3.7.: Comparison table for the MIMIC-III dataset based on standard metrics between our proposed DSMC model and three widely used in healthcare scoring systems, including SOFA, SAPS III, and APACHE II. These metrics are calculated based on the 50 hours of stay in ICU.

Model	Metric			
	AUROC	Precision	Recall	F1
SOFA	0.50	0.0	0.0	0.0
SAPS III	0.60	0.56	1.0	0.71
APACHE II	0.70	0.67	0.80	0.73
DSMC	**0.80**	**0.71**	**1.0**	**0.83**

Furthermore, the robustness of the model is substantiated in the C-MAPSS dataset by demonstrating that the soft monotonic features can serve as viable inputs for any prognostic model, thereby yielding equivalent predictive performance. In pursuit of this validation, two additional prognostic models were deployed, derived from the realm of ML, namely Gradient Boosted Decision-Trees (GBDT) [107] and Support Vector



(a) (b)

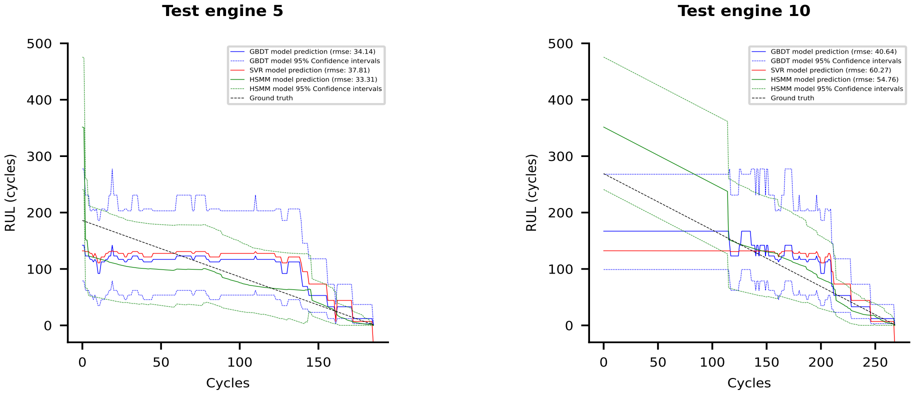
(c) (d)

Figure 3.5.: ROC curves after 1, 25, 50, and 80 hours after each patient's entrance to the ICU.

Regression (SVR) [108] algorithms. These models are developed utilizing Python's Scikit-learn package with default hyperparameters. The predictions correspond to the RUL, which signifies the critical time prior to the engine's failure.

Figure 3.6a and Figure 3.6b illustrate the comparative performance of each prognostic model when provided with the monotonic features as input, using two distinct testing samples. The first sample is derived from the average trajectory lengths, while the second originates from the right outliers, characterized by large trajectory lengths. These samples are namely the testing engines 5 and 10, respectively. In both figures, the mean predicted RUL is shown alongside the corresponding 95% confidence intervals for the HSMM and GBDT prognostic models. Due to the deterministic nature of the

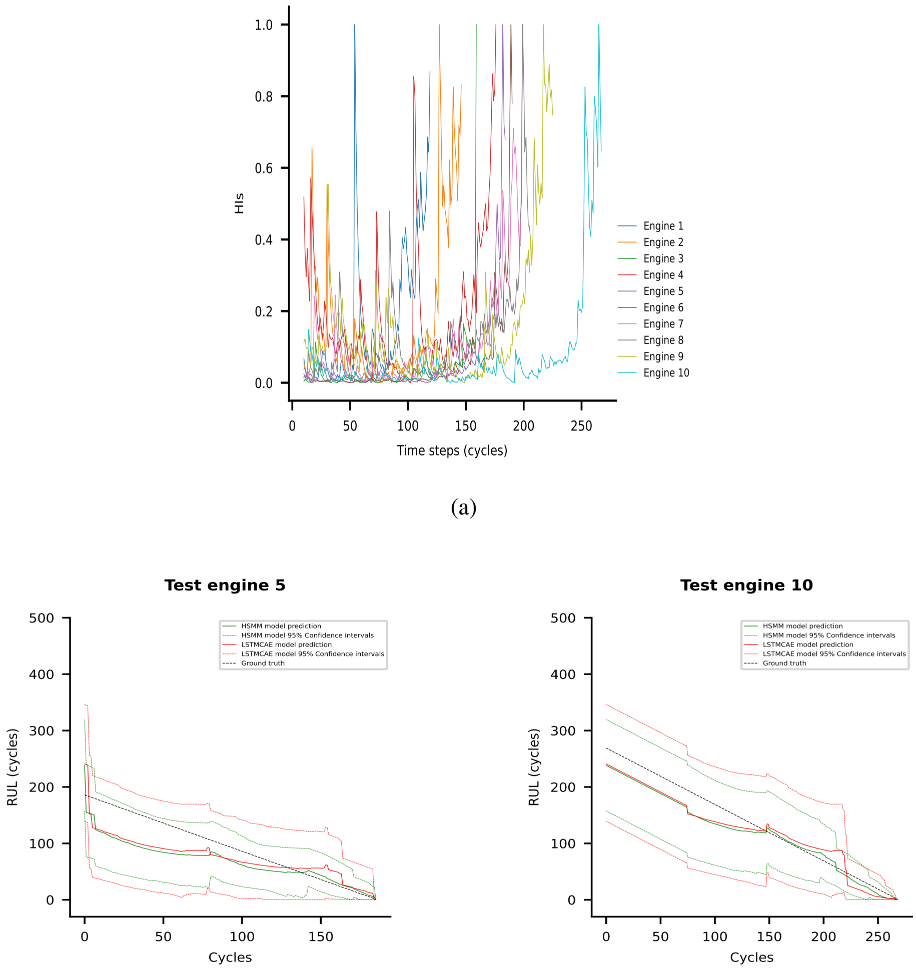
SVR model, only the mean values of RUL are available. Although Root Mean Squared (RMS) error exhibits variations among the prognostic models, it is worth noting that existing literature claims that the engine degradation is detected, on average and mostly concerning the engines with the highest lifetime, at approximately 100-150 operating cycles [109–114]. This assertion finds support in the clustering outcomes depicted in Figure 3.4c, where the cluster assignments for the engines with the larger trajectory lengths stay close to zero until a crucial point in the performed cycles is reached. This implies that it becomes possible to initiate meaningful predictions from that point onward. Consequently, the RUL's uncertainty and inconsistency are rationally large at initial timesteps, whilst all of the models converge towards the ground truth values as time progresses, enlightening that the cluster assignments, and hence, the DSMC model are agnostic of the chosen prognostic model.



(a) (b)

Figure 3.6.: RUL prediction of different prognostic models for one inner and one outlier trajectory. (**a**) Testing engine 5, whose length is close to average. (**b**) Testing engine 10, representing the right outlier.

Subsequently, except for validating DSMC's robustness, a comparative study has been performed against a state-of-the-art AE architecture, namely the long short-term memory convolutional autoencoder (LSTMCAE) [115], that outputs health indicators (HIs) to assess the engines' health status based on reconstruction errors. These HIs are shown in Figure 3.7a. To validate the performance of those two feature extraction models, the corresponding prognostics produced by HSMM are illustrated in Figure 3.7b and Figure 3.7c for the test engines 5 and 10, respectively. It is evident that the DSMC model slightly outperforms the LSTMCAE model and this can be further observed via the RMS over each test engine stored in Table 3.8. Especially for the test engines 1, 2, 3, and 8, DSMC significantly outperforms LSTMCAE.



(a)

(b)

(c)

Figure 3.7.: (**a**) Constructed HIs utilizing the LSTMCAE model.(**b**) Comparison of RUL curves between DSMC and LSTMCAE for test engine 5. (**c**) Comparison of RUL curves between DSMC and LSTMCAE for test engine 5.

3.4.3. INTERPRETABILTY OF DSMC MODEL

In this subsection, the interpretability of the developed DSMC model is illustrated, validated, and discussed. Achieving this involves the interpretation of the time gradients that should be strictly non-negative to perform a monotonic increase of the outcomes of z-space. Subsequently, the extracted features of z-space are depicted before and after the second stage of the training process that concerns clustering to demonstrate how the

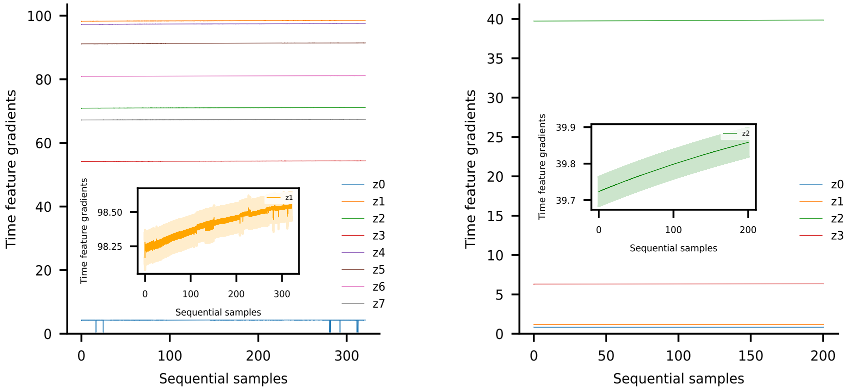
Table 3.8.: RMS for each test engine calculated by the HSMM model (ours) and the LSTMCAE model.

Engine no.	RMS (DSMC)	RMS (LSTMCAE)
1	15.78	36.83
2	34.64	44.93
3	19.84	35.42
4	49.89	50.00
5	30.39	31.66
6	29.14	29.80
7	36.65	41.40
8	52.77	64.25
9	20.57	22.48
10	22.59	22.61
Average	**31.23**	37.94

DSMC model learns to assign clusters accordingly.

TIME GRADIENTS flow

The proposed architecture introduces a key novelty and enables its application to various types of sequential data requiring a monotonic output. This novelty stems from the inclusion of a stringent monotonic constraint between the time feature and the encoder's output layer. As per this proposed constraint, it is required that the gradient of each output in the Z-space, concerning the time feature, strictly maintains a non-negative value. Indeed, Figure 3.8 provides a visual representation of these gradients for both datasets as the time variable progresses within the trajectories, effectively confirming their persistent non-negative nature. It should be noted that in this figure the samples of the entire training set are put in an increasing time order to sequentially reflect the output gradients with respect to time. Obviously, this was only applied for visualization purposes, after finishing the training process. To enhance clarity, a moving average window with a size of one is applied, and shaded areas around the mean values are generated using the standard deviation. Notably, the gradients exhibit strikingly similar trends with only shifts in their positions on the y-axis. This observation suggests the existence of a common underlying pattern or relationship between the time feature and the Z-space. The similarity in these gradients indicates that the network's output neurons respond (soft) monotonically with the time feature, revealing a shared and unified sensitivity of the Z-space to the time feature. Consequently, any potential decrease in cluster assignments is relatively reliant on the time-series inputs.



(a) (b)

Figure 3.8.: Time gradients (gradients of each encoder's output with respect to the time feature). (**a**) Time gradients for the MIMIC-III dataset. The time gradients for z_1 are zoomed in for clarity. (**b**) Time gradients for the C-MAPSS dataset. The time gradients for z_2 are zoomed in for clarity.

INTERPRETING Z-SPACE AND ITS RELATION TO SOFT MONOTONIC CLUSTERING

In certain instances, the utilization of k-means clustering alone proves adequate for generating meaningful clustering outcomes. This is particularly applicable to datasets that already exhibit indications of monotonicity within the input time-series data, thereby suggesting that the embedding space is inherently constructed to align with the desired monotonic behavior.

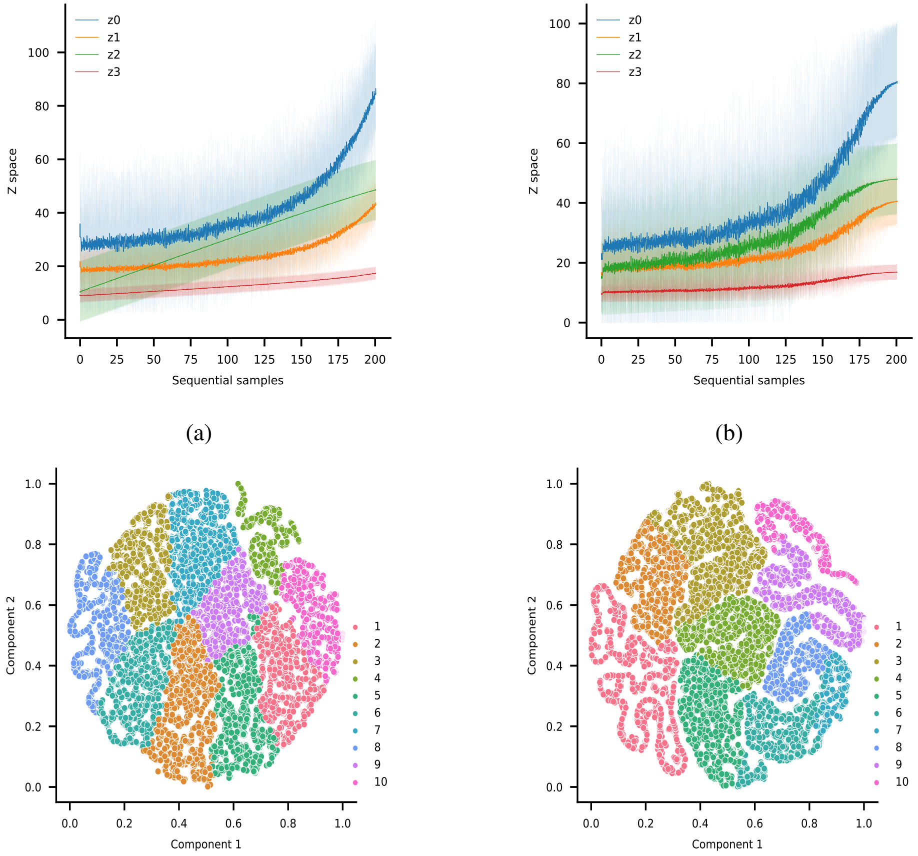
A notable example of this scenario is observed in the C-MAPSS dataset, where several sensors demonstrate strictly increasing or decreasing monotonic patterns. Consequently, the AE model is naturally compelled to create the Z-space in a manner that effectively incorporates these existing monotonic behaviors. Figure 3.9a and Figure 3.9b, which represent the results prior to and following the DSMC model's second stage of training, provide evidence supporting the assertion that visualizing the Z-space with samples arranged in a sequential manner based on their respective time (same ordering of the training set as in Figure 3.8) yields meaningful insights. To enhance visual clarity, the presented results are obtained using a moving average window with a window size of 30. The mean values are employed, and the shaded areas represent the standard deviation. It is evident that the encoder's predictions exhibit an increasing trend as the engine's damage severity progresses, even without the second stage of training. After training, the model manages to capture additional details, leading to improved feature extraction, particularly noticeable in the z_1 output.

Furthermore, although the trends of the outputs are increasing, there are several oscillations indicating the desired soft monotonic behavior. An important observation to note is that unlike the time gradients depicted in Figure 3.8, which are required to strictly maintain non-negative values for the application of the monotonic constraint, the Z-space is not subject to such constraints and can assume any real value. However, as expected, it is shown that there exists a soft monotonic increase within the Z-space as we progress toward the end of each trajectory length.

Moreover, Figure 3.9c and Figure 3.9d enlighten how this embedding space is grouped into the 10 clusters before and after the DSMC model's final stage of training, respectively, given a 2D space created by the method of t-distributed Stochastic Neighbor Embedding (t-SNE) [116]. The popularity of this method lies in its capacity to probabilistically uncover non-linear connections or similarities within the data. Evidently, in both figures, the model's cluster assignments correspond to adjacent regions, suggesting that the encoder's output following the AE model training can be employed directly for clustering purposes without proceeding to the next training stage of the DSMC model. It is important to note that the disparity between the two representations does not arise from training the DSMC model but rather from the inherently stochastic nature of the t-SNE method. Consequently, the precise positioning of each cluster in the 2D space is of lesser concern, as long as neighboring samples consistently exhibit the same cluster assignment, which seems to be the case for the proposed model.

For the C-MAPSS dataset, a notable contradiction becomes apparent when comparing the clustering results (Figure 3.4c) with the clustering performance of the DSMC model (Figure 3.8b and Figure 3.9). While the clustering results come up with several fluctuations, the AE model has successfully learned softly monotonic and distinguishable features even without the final training of the DSMC model. This occurrence can be attributed to the application of a soft monotonic constraint rather than a learnable monotonic behavior that would eliminate these fluctuations. For instance, in Figure 3.9a and Figure 3.9b, the feature z_2 demonstrates averagely monotonic behavior over time, but it occasionally exhibits temporary decreases, potentially resulting in lower cluster assignments. If the monotonic behavior was learnable by the model, it would be expected for z_2 to consistently exhibit a monotonic increase. However, this is not the case as it is necessary to capture potential instances of recovery that may be concealed within the input time-series data. Consequently, the DSMC model acknowledges the presence of these fluctuations as normal, yet still adeptly captures and groups them accurately.

While it can be argued that the DSMC model marginally enhances the clustering performance in the C-MAPSS dataset, its impact is transformative for the MIMIC-III dataset. This is primarily due to the absence of any inherent pre-existing monotonic behavior in the input time sequences of the MIMIC-III dataset. In such a case, the DSMC model proves to be a game-changer, as it effectively incorporates the necessary monotonicity to improve the clustering outcomes. Indeed, in Figure 3.10a, Figure 3.10b, and Figure 3.10c, Figure 3.10d sets, noticeable distinctions in the Z-space and cluster assignments are evident when comparing the results before and after finishing the second stage of DSMC model's training. In Figure 3.10a, it is observed that the encoder's

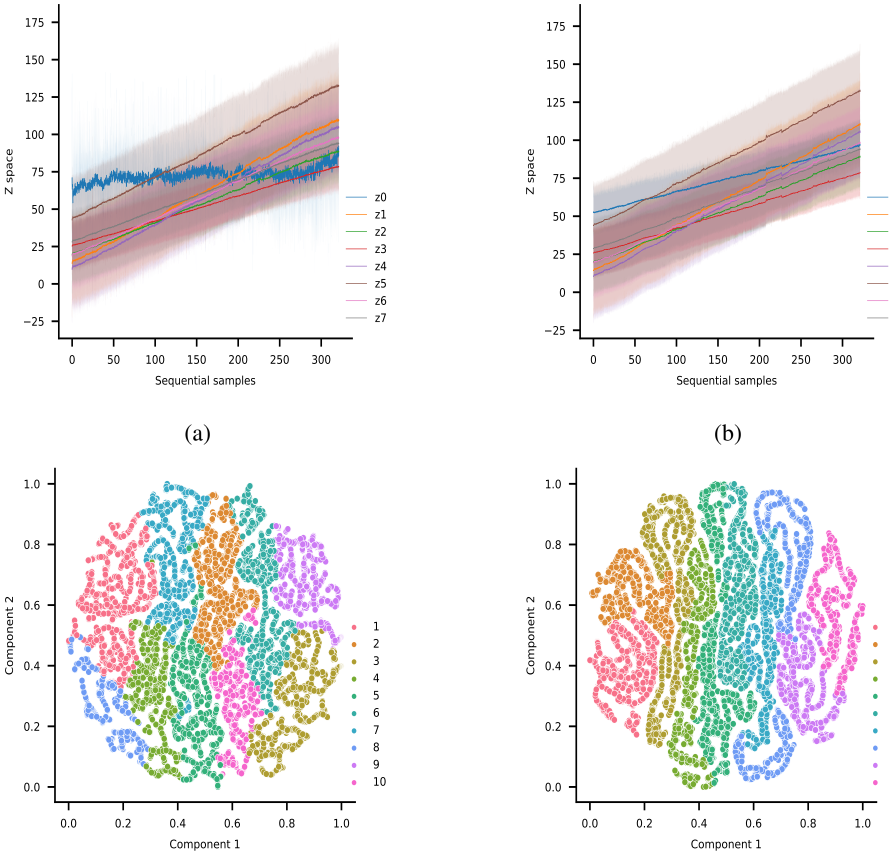


(a) (b)

(c) (d)

Figure 3.9.: (**a**) Z-space visualization for the C-MAPSS dataset before training the DSMC model. (**b**) Z-space visualization for the C-MAPSS dataset after training the DSMC model. (**c**) t-SNE graph with two principal components for the C-MAPSS dataset before training the DSMC model. (**d**) t-SNE graph with two principal components for the C-MAPSS dataset after training the DSMC model.

output z_0 exhibits an oscillated trend as the time feature increases. This suggests that the vital signs, supplementary data (demographic and laboratory data), or a combination of both may potentially confuse the encoder part of the AE model, which strives to adhere to the monotonic constraint between its outputs and the time feature while simultaneously incorporating the information provided by the inputs. This observation is further supported by the t-SNE graph presented in Figure 3.10c, where the cluster assignments appear to be more disordered, i.e. clusters are not homogenous and can be interrupted by others, compared to the C-MAPSS dataset. However, after finishing the entire training of the DSMC model, each of the encoder's outputs consistently and satisfactorily demonstrates an upward trend (Figure 3.10b)), while the cluster assignments exhibit greater similarity to their neighboring values (Figure 3.10d).



(a)

(b)

(c)

(d)

Figure 3.10.: (a) Z-space visualization for the MIMIC-III dataset before training the DSMC model. (b) Z-space visualization for the MIMIC-III dataset after training the DSMC model. (c) t-SNE graph for the MIMIC-III dataset before training the DSMC model. (d) t-SNE graph for the MIMIC-III dataset after training the DSMC model.

3.4.4. TOWARDS MULTI-MODAL FEATURE EXTRACTION AND CLUSTERING: EVALUATING THE F-MOC DATASET

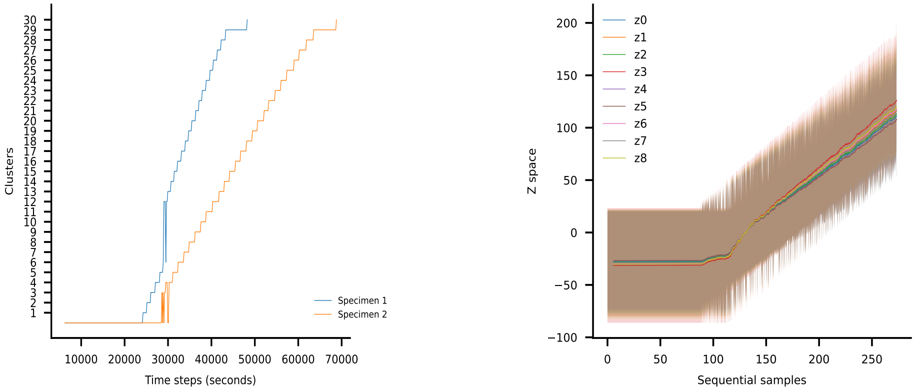
Given that the capabilities of the DSMC model have been thoroughly discussed in previous instances with the use of other datasets, the focus here is solely on evaluating and discussing the model's outputs and performance. Without the Bayesian optimization, the computational time required to train the DSMC model is approximately 1.1 hours.

The two test specimens under examination correspond to one chosen from the average and another from the right outliers in terms of specimens' lifespan in ascending order. Given the inherently challenging nature of this dataset, characterized by its multi-modal and high-dimensional data, an anticipated drop in performance is expected. However, surprisingly, the cluster assignments, as illustrated in Figure 3.11a, come up with a better monotonic behavior despite some potential fluctuations that are rationally detected due to the induced soft monotonicity. This improvement can be verified by the constructed z-space shown in Figure 3.11b. Clearly, the monotonicity of the extracted features surpasses that of the previous datasets, resulting in a more distinct monotonic clustering. This observation can be attributed to the continual accumulation of damage observed in the image data, making this type of data a perfect candidate for predicting degenerative phenomena. Hence, the limited fluctuations present can be solely associated with the occasional non-informative and noisy nature of the acoustic emission data [117] and, by no means, to an increase in the structure's health. At the same time, the constant values regarding z-space at the start of each trajectory affirm the corresponding flat region in the initial cluster assignments, indicating the absence of detected damage in the structure.

The reproducibility of the training process for this dataset is presented in Table B.1. The convergence of the training and validating loss corresponding to the reconstruction and time losses is depicted in Figure B.3e and Figure B.3f, respectively. Notably, our model adeptly captures the intricate dynamics underlying fatigue, enabling accurate prediction of RUL under uncertainty, as illustrated in Figure 3.12. Again, the HSMM prognostic model was chosen. However, drawing inspiration from [23], the present dataset employs 4 hidden states in the prognostic model, instead of 7. These states are selected to mirror diverse levels of damage accumulation within the composite structure's lifespan, encompassing phenomena such as matrix cracking, crack coupling, delamination, and fiber localized breaking [118].

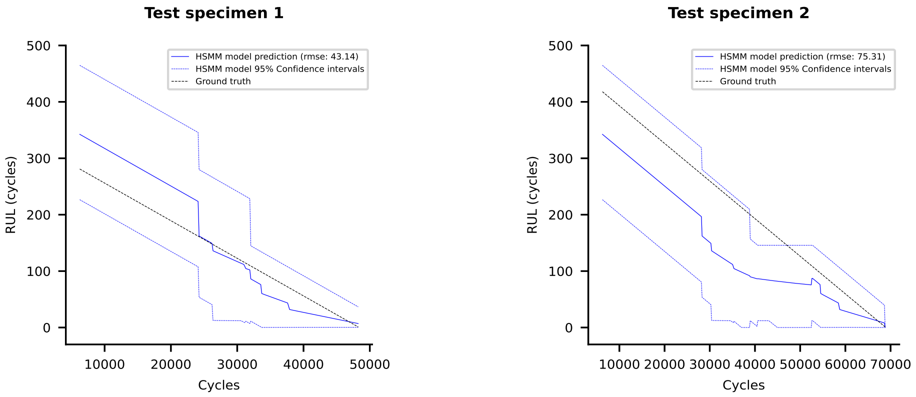
3.4.5. QUALITATIVE ANALYSIS OF NUMBER OF CLUSTERS HYPERPARAMETER

As already mentioned, the number of clusters was chosen to be relatively small for easier interpretation of the outcomes. Nevertheless, it is yet to be discussed how this number affects the accuracy of prognosis. In this regard, a qualitative analysis is performed for each dataset related to this hyperparameter. Particularly, the normalized RMS between



(a) (b)

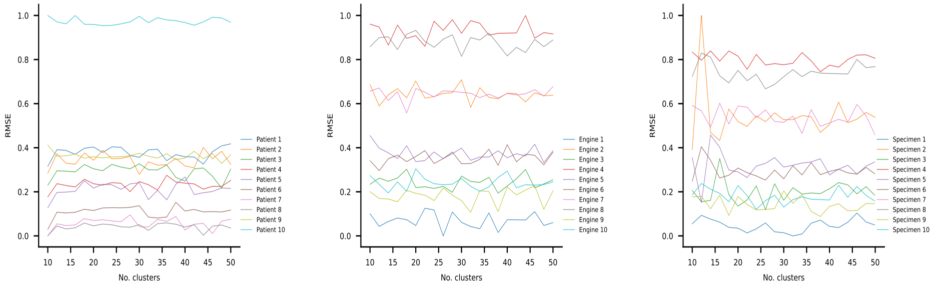
Figure 3.11.: (**a**) Clustering results for the F-MOC dataset. (**b**) Z-space visualization for the F-MOC dataset after training the DSMC model.



(a) (b)

Figure 3.12.: (**a**) Stochastic RUL predictions of the first test specimen. (**b**) Stochastic RUL predictions of the second test specimen.

the true and predicted RUL is calculated after training the DSMC and HSMM models with varying numbers of clusters ranging in `[10,50]` and taking every second value, i.e. 10, 12, and so on. The results of this analysis are depicted in Figure 3.13 for each dataset and each test trajectory correspondingly. Interestingly, increasing the number of clusters does not improve the accuracy of prognostics for the MIMIC-III and C-MAPSS datasets. However, as expected, for a small number of clusters the accuracy is lower concerning the F-MOC dataset which contains more complicated high-dimensional data. Nonetheless, having more than `20` clusters is enough for keeping a stable optimal performance.



(a) (b) (c)

Figure 3.13.: Qualitative analysis of the number of clusters hyperparameter for (**a**) MIMIC-III dataset. (**b**) C-MAPSS dataset. (**c**) F-MOC dataset.

3.5. CONCLUSION

The proposed DSMC model effectively achieves the objectives of this work through its distinctive architecture. By employing a two-stage training process and incorporating a monotonic constraint, the model successfully extracts soft monotonic features and simultaneously clusters them based on the system's level of deterioration. Unlike conventional approaches that train the model to strictly adhere to a monotonic behavior [74, 119, 120], the proposed model was compelled to establish a strict monotonic relationship between its outputs and the integrated time feature. Moreover, the ability to automatically fine-tune the hyperparameters through customizing the objective function of the BO algorithm not only streamlines the adaptation process but also enhances the model's overall performance and applicability across different domains.

Three datasets from distinct scientific domains were selected to evaluate the performance and capabilities of the DSMC model, including MIMIC-III, C-MAPSS, and F-MOC datasets. Concerning the MIMIC-III dataset, notably, the model exhibited impressive performance by providing useful soft monotonic features to the prognostic model, effectively surpassing the performance of SOFA, SAPS III, and APACHE II score systems, which are widely used in healthcare. The model's robustness is verified on the C-MAPSS dataset by showing similar prediction capabilities over three different prognostic models, including HSMM, GBDT, and SVR. This implies that the DSMC model produces expressive monotonic features that are highly correlated to the prognostic task, hence making it agnostic to the chosen prognostic algorithm. Furthermore, the interpretability of the model can be seen from those two datasets. In the context of the F-MOC dataset, our model's performance remained at high levels despite the inherent challenges posed by the multi-modality of the dataset and the intricate task of sensor fusion. Successfully navigating the complexities associated with multi-modal information, the model showcased its efficacy in feature extraction and subsequent clustering. This accomplishment is notable given the intricacies involved in integrating diverse sensory data sources, including one passive and one active testing technique, i.e. acoustic emission and image capturing, respectively.

Whilst the DSMC model is trained exclusively on trajectories that reach a definitive target, hence signaling their final value, running this in real-time makes the trajectory's last value remain unknown. Notably, if this last value occurs considerably later in terms of time compared to the training trajectories, it results in an extended duration within the penultimate cluster. This can be observed in Figure 3.4a, representing the right outlier of the MIMIC-III dataset (test patient 10). The same can be identified in Figure 3.4c concerning the right outlier of the C-MAPSS dataset (test engine 10). Nevertheless, this discrepancy does not pose a concern as it signifies an overestimation of risk, thus providing valuable support to human experts in making more cautious decisions concerning their corresponding field of expertise.

Despite the DSMC model having demonstrated its ability to capture potential recoveries within the soft monotonic clustering process, signifying a substantial correlation between the input sequences and the corresponding predictions, the specific mechanisms by which the input timestamps influence this monotonic behavior remain unclear. This aspect represents a crucial next step in this research, as gaining understanding of these relationships can yield valuable insights. For instance, in the context of sepsis threat, such an understanding could provide patterns illuminating how treatments impact a patient's current condition or identify the noisy sensors within the C-MAPSS dataset responsible for the observed fluctuations in the monotonic clustering. Additionally, it could be possible to correlate noisy measurements from the acoustic emission system or potential undetected damage in the structure via the cameras within the F-MOC dataset. Investigating and comprehending these underlying factors would enhance the physical understanding and practical application of the model, enabling better-informed decision-making and tailored interventions in real-world applications.

REFERENCES

[1] D. Hobbs. “Concrete deterioration: causes, diagnosis, and minimising risk”. In: *International Materials Reviews* 46.3 (Mar. 2001), pp. 117–144. ISSN: 0950-6608. DOI: 10.1179/095066001101528420. URL: https://doi.org/10.1179/095066001101528420%20http://www.tandfonline.com/doi/full/10.1179/095066001101528420.

[2] M. M. Churpek, T. C. Yuen, and D. P. Edelson. “Predicting clinical deterioration in the hospital: The impact of outcome selection”. In: *Resuscitation* 84.5 (May 2013), pp. 564–568. ISSN: 03009572. DOI: 10.1016/j.resuscitation.2012.09.024. URL: http://dx.doi.org/10.1016/j.resuscitation.2012.09.024%20https://linkinghub.elsevier.com/retrieve/pii/S0300957212008131.

[3] J. Malycha, S. Bacchi, and O. Redfern. “Artificial intelligence and clinical deterioration”. In: *Current Opinion in Critical Care* 28.3 (June 2022), pp. 315–321. ISSN: 1070-5295. DOI: 10.1097/MCC.0000000000000945. URL: https://journals.lww.com/co-criticalcare/Fulltext/2022/06000/Artificial˙intelligence˙and˙clinical˙deterioration.15.aspx%20https://journals.lww.com/10.1097/MCC.0000000000000945.

[4] H. Wang. “A survey of maintenance policies of deteriorating systems”. In: *European Journal of Operational Research* 139.3 (June 2002), pp. 469–489. ISSN: 03772217. DOI: 10.1016/S0377-2217(01)00197-7. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221701001977.

[5] D. M. Frangopol, S. Sabatino, and M. Soliman. “Maintenance and safety of deteriorating systems: a life-cycle perspective”. In: *Proceedings of the Second International Conference on Performance-based and Life-cycle Structural Engineering (PLSE 2015)* (Jan. 2015), pp. 48–57. DOI: 10.14264/uql.2016.1175. URL: http://espace.library.uq.edu.au/view/UQ:399362.

[6] N. Sapoval, A. Aghazadeh, M. G. Nute, D. A. Antunes, A. Balaji, R. Baraniuk, C. J. Barberan, R. Dannenfelser, C. Dun, M. Edrisi, R. A. Elworth, B. Kille, A. Kyrillidis, L. Nakhleh, C. R. Wolfe, Z. Yan, V. Yao, and T. J. Treangen. “Current progress and open challenges for applying deep learning across the biosciences”. In: *Nature Communications* 13.1 (2022), pp. 1–12. ISSN: 20411723. DOI: 10.1038/s41467-022-29268-7.

[7] M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, S. Ursprung, A. I. Aviles-Rivero, C. Etmann, C. McCague, L. Beer, J. R. Weir-McCall, Z. Teng, E. Gkrania-Klotsas, A. Ruggiero, A. Korhonen, E. Jefferson, E. Ako, G. Langs, G. Gozaliasl, G. Yang, H. Prosch, J. Preller, J. Stanczuk, J. Tang, J. Hofmanninger, J. Babar, L. E. Sánchez, M. Thillai, P. M. Gonzalez, P. Teare,

X. Zhu, M. Patel, C. Cafolla, H. Azadbakht, J. Jacob, J. Lowe, K. Zhang, K. Bradley, M. Wassin, M. Holzer, K. Ji, M. D. Ortet, T. Ai, N. Walton, P. Lio, S. Stranks, T. Shadbahr, W. Lin, Y. Zha, Z. Niu, J. H. F. Rudd, E. Sala, and C.-B. Schönlieb. “Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans”. In: *Nature Machine Intelligence* 3.3 (Mar. 2021), pp. 199–217. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00307-0. URL: https://www.nature.com/articles/s42256-021-00307-0.

[8] P. Agarwal, M. A. Alam, and R. Biswas. “Issues,Challenges and Tools of Clustering Algorithms”. In: *IJCSI International Journal of Computer Science Issues* 8.3 (Oct. 2011), pp. 523–528. URL: http://arxiv.org/abs/1110.2610.

[9] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects”. In: *Engineering Applications of Artificial Intelligence* 110.December 2021 (Apr. 2022), p. 104743. ISSN: 09521976. DOI: 10.1016/j.engappai.2022.104743. URL: https://doi.org/10.1016/j.engappai.2022.104743%20https://linkinghub.elsevier.com/retrieve/pii/S095219762200046X.

[10] X. Li, W. Zhang, and Q. Ding. “Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction”. In: *Reliability Engineering & System Safety* 182 (2019), pp. 208–218. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2018.11.011. URL: https://www.sciencedirect.com/science/article/pii/S0951832018308299.

[11] K. Deng, X. Zhang, Y. Cheng, Z. Zheng, F. Jiang, W. Liu, and J. Peng. “A Remaining Useful Life Prediction Method with Automatic Feature Extraction for Aircraft Engines”. In: *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, Aug. 2019, pp. 686–692. ISBN: 978-1-7281-2777-4. DOI: 10.1109/TrustCom/BigDataSE.2019.00097. URL: https://ieeexplore.ieee.org/document/8887318/.

[12] H. Zhao, H. Liu, Y. Jin, X. Dang, and W. Deng. “Feature Extraction for Data-Driven Remaining Useful Life Prediction of Rolling Bearings”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–10. ISSN: 0018-9456. DOI: 10.1109/TIM.2021.3059500. URL: https://ieeexplore.ieee.org/document/9354649/.

[13] M. Salvi, H. W. Loh, S. Seoni, P. D. Barua, S. García, F. Molinari, and U. R. Acharya. “Multi-modality approaches for medical support systems: A systematic review of the last decade”. In: *Information Fusion* 103 (2024), p. 102134. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.102134. URL: https://www.sciencedirect.com/science/article/pii/S1566253523004505.

[14] Z. Qiu, J. Martínez-Sánchez, P. Arias-Sánchez, and R. Rashdi. "External multi-modal imaging sensor calibration for sensor fusion: A review". In: *Information Fusion* 97 (2023), p. 101806. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101806. URL: https://www.sciencedirect.com/science/article/pii/S156625352300115X.

[15] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino. "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges". In: *Information Fusion* 35 (2017), pp. 68–80. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2016.09.005. URL: https://www.sciencedirect.com/science/article/pii/S156625351630077X.

[16] L. Zou, Z. Wang, J. Hu, and Q.-L. Han. "Moving horizon estimation meets multi-sensor information fusion: Development, opportunities and challenges". In: *Information Fusion* 60 (2020), pp. 1–10. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.01.009. URL: https://www.sciencedirect.com/science/article/pii/S1566253519310024.

[17] M. Nazarahari and H. Rouhani. "40 years of sensor fusion for orientation tracking via magnetic and inertial measurement units: Methods, lessons learned, and future challenges". In: *Information Fusion* 68 (2021), pp. 67–84. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.10.018. URL: https://www.sciencedirect.com/science/article/pii/S1566253520303997.

[18] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35. URL: https://ieeexplore.ieee.org/document/9158013/%20https://www.nature.com/articles/sdata201635.

[19] C. Rhee, R. Dantes, L. Epstein, D. J. Murphy, C. W. Seymour, T. J. Iwashyna, S. S. Kadri, D. C. Angus, R. L. Danner, A. E. Fiore, J. A. Jernigan, G. S. Martin, E. Septimus, D. K. Warren, A. Karcz, C. Chan, J. T. Menchaca, R. Wang, S. Gruber, and M. Klompas. "Incidence and Trends of Sepsis in US Hospitals Using Clinical vs Claims Data, 2009-2014". In: *JAMA* 318.13 (Oct. 2017), p. 1241. ISSN: 0098-7484. DOI: 10.1001/jama.2017.13836. URL: https://doi.org/10.1001/jama.2017.13836%20http://jama.jamanetwork.com/article.aspx?doi=10.1001/jama.2017.13836.

[20] T. G. Buchman, S. Q. Simpson, K. L. Sciarretta, K. P. Finne, N. Sowers, M. Collier, S. Chavan, I. Oke, M. E. Pennini, A. Santhosh, M. Wax, R. Woodbury, S. Chu, T. G. Merkeley, G. L. Disbrow, R. A. Bright, T. E. MaCurdy, and J. A. Kelman. "Sepsis Among Medicare Beneficiaries". In: *Critical Care Medicine* 48.3 (Mar. 2020), pp. 276–288. ISSN: 0090-3493. DOI: 10.1097/CCM.0000000000004224. URL: https://journals.lww.com/ccmjournal/Fulltext/2020/03000/Sepsis˙Among˙Medicare˙Beneficiaries˙˙1˙˙The.2.aspx%20http://journals.lww.com/10.1097/CCM.0000000000004224.

[21] H. Arefian, S. Heublein, A. Scherag, F. M. Brunkhorst, M. Z. Younis, O. Moerer, D. Fischer, and M. Hartmann. “Hospital-related cost of sepsis: A systematic review”. In: *Journal of Infection* 74.2 (Feb. 2017), pp. 107–117. ISSN: 01634453. DOI: 10.1016/j.jinf.2016.11.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0163445316302882.

[22] S. Vollert and A. Theissler. “Challenges of machine learning-based RUL prognosis: A review on NASA’s C-MAPSS data set”. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* 2021-Septe (Sept. 2021), pp. 1–8. ISSN: 19460759. DOI: 10.1109/ETFA45728.2021.9613682. URL: https://ieeexplore.ieee.org/document/9613682/.

[23] N. Eleutheroglou, D. Zarouchas, and R. Benedictus. “An adaptive probabilistic data-driven methodology for prognosis of the fatigue life of composite structures”. In: *Composite Structures* 245 (2020), p. 112386. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2020.112386. URL: https://www.sciencedirect.com/science/article/pii/S0263822319347634.

[24] Y. Jiang, Z. Zhang, W. Wang, W. Huang, C. Chen, S. Xi, M. U. Ahmad, Y. Ren, S. Sang, J. Xie, J. Y. Wang, W. Xiong, T. Li, Z. Han, Q. Yuan, Y. Xu, L. Xing, G. A. Poultsides, G. Li, and R. Li. “Biology-guided deep learning predicts prognosis and cancer immunotherapy response”. In: *Nature Communications* 14.1 (2023), pp. 1–16. ISSN: 20411723. DOI: 10.1038/s41467-023-40890-x.

[25] A. Ling and R. S. Huang. “Computationally predicting clinical drug combination efficacy with cancer cell line screens and independent drug action”. In: *Nature Communications* 11.1 (2020). ISSN: 20411723. DOI: 10.1038/s41467-020-19563-6. URL: http://dx.doi.org/10.1038/s41467-020-19563-6.

[26] J. Åkesson, S. Hojjati, S. Hellberg, J. Raffetseder, M. Khademi, R. Rynkowski, I. Kockum, C. Altafini, Z. Lubovac-Pilav, J. Mellergård, M. C. Jenmalm, F. Piehl, T. Olsson, J. Ernerudh, and M. Gustafsson. “Proteomics reveal biomarkers for diagnosis, disease activity and long-term disability outcomes in multiple sclerosis”. In: *Nature Communications* 14.1 (2023). ISSN: 20411723. DOI: 10.1038/s41467-023-42682-9.

[27] H. Shin, B. H. Choi, O. Shim, J. Kim, Y. Park, S. K. Cho, H. K. Kim, and Y. Choi. “Single test-based diagnosis of multiple cancer types using Exosome-SERS-AI for early stage cancers”. In: *Nature Communications* 14.1 (2023), pp. 1–10. ISSN: 20411723. DOI: 10.1038/s41467-023-37403-1.

[28] S. Lee, B. Jeong, M. Kim, R. Jang, W. Paik, J. Kang, W. J. Chung, G. S. Hong, and N. Kim. “Emergency triage of brain computed tomography via anomaly detection with a deep generative model”. In: *Nature Communications* 13.1 (2022), pp. 1–11. ISSN: 20411723. DOI: 10.1038/s41467-022-31808-0.

[29] W. Zhao, Z. Chen, P. Xie, J. Liu, S. Hou, L. Xu, Y. Qiu, D. Wu, J. Xiao, and K. He. “Multi-task oriented diffusion model for mortality prediction in shock patients with incomplete data”. In: *Information Fusion* (2023), p. 102207. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.102207. URL: https://www.sciencedirect.com/science/article/pii/S1566253523005237.

[30] F. Ali, S. El-Sappagh, S. M. R. Islam, D. Kwak, A. Ali, M. Imran, and K.-S. Kwak. “A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion”. In: *Information Fusion* 63 (2020), pp. 208–222. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.06.008. URL: https://www.sciencedirect.com/science/article/pii/S1566253520303055.

[31] Y. Zhong, C. Cai, T. Chen, H. Gui, J. Deng, M. Yang, B. Yu, Y. Song, T. Wang, X. Sun, J. Shi, Y. Chen, D. Xie, C. Chen, and Y. She. “PET/CT based cross-modal deep learning signature to predict occult nodal metastasis in lung cancer”. In: *Nature Communications* 14.1 (Nov. 2023), p. 7513. ISSN: 2041-1723. DOI: 10.1038/s41467-023-42811-4. URL: https://www.nature.com/articles/s41467-023-42811-4.

[32] X. Mei, Z. Liu, A. Singh, M. Lange, P. Boddu, J. Q. Gong, J. Lee, C. DeMarco, C. Cao, S. Platt, G. Sivakumar, B. Gross, M. Huang, J. Masseaux, S. Dua, A. Bernheim, M. Chung, T. Deyer, A. Jacobi, M. Padilla, Z. A. Fayad, and Y. Yang. “Interstitial lung disease diagnosis and prognosis using an AI system integrating longitudinal data”. In: *Nature Communications* 14.1 (2023), pp. 1–11. ISSN: 20411723. DOI: 10.1038/s41467-023-37720-5.

[33] H. Chao, H. Shan, F. Homayounieh, R. Singh, R. D. Khera, H. Guo, T. Su, G. Wang, M. K. Kalra, and P. Yan. “Deep learning predicts cardiovascular disease risks from lung cancer screening low dose computed tomography”. In: *Nature Communications* 12.1 (2021), pp. 1–10. ISSN: 20411723. DOI: 10.1038/s41467-021-23235-4. URL: http://dx.doi.org/10.1038/s41467-021-23235-4.

[34] J. Weiss, V. K. Raghu, D. Bontempi, D. C. Christiani, R. H. Mak, M. T. Lu, and H. J. Aerts. “Deep learning to estimate lung disease mortality from chest radiographs”. In: *Nature Communications* 14.1 (2023), pp. 1–10. ISSN: 20411723. DOI: 10.1038/s41467-023-37758-5.

[35] R. Li, W. Chen, M. Li, R. Wang, L. Zhao, Y. Lin, X. Chen, Y. Shang, X. Tu, D. Lin, X. Wu, Z. Lin, A. Xu, X. Wang, D. Wang, X. Zhang, M. Dongye, Y. Huang, C. Chen, Y. Zhu, C. Liu, Y. Hu, L. Zhao, H. Ouyang, M. Li, X. Li, and H. Lin. “LensAge index as a deep learning-based biological age for self-monitoring the risks of age-related diseases and mortality”. In: *Nature Communications* 14.1 (2023). ISSN: 20411723. DOI: 10.1038/s41467-023-42934-8.

[36] C. Habib, A. Makhoul, R. Darazi, and R. Couturier. “Health risk assessment and decision-making for patient monitoring and decision-support using Wireless Body Sensor Networks”. In: *Information Fusion* 47 (2019), pp. 10–22. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2018.06.008. URL: https://www.sciencedirect.com/science/article/pii/S156625351730790X.

[37] J. P. H. Verharen, J. W. de Jong, T. J. M. Roelofs, C. F. M. Huffels, R. van Zessen, M. C. M. Luijendijk, R. Hamelink, I. Willuhn, H. E. M. den Ouden, G. van der Plasse, R. A. H. Adan, and L. J. M. J. Vanderschuren. “A neuronal mechanism underlying decision-making deficits during hyperdopaminergic states”. In: *Nature Communications* 9.1 (Feb. 2018), p. 731. ISSN: 2041-1723. DOI:

10.1038/s41467-018-03087-1. URL: http://dx.doi.org/10.1038/s41467-018-03087-1%20https://www.nature.com/articles/s41467-018-03087-1.

[38] K. Nakamura, R. Kojima, E. Uchino, K. Ono, M. Yanagita, K. Murashita, K. Itoh, S. Nakaji, and Y. Okuno. “Health improvement framework for actionable treatment planning using a surrogate Bayesian model”. In: *Nature Communications* 12.1 (May 2021), p. 3088. ISSN: 2041-1723. DOI: 10.1038/s41467-021-23319-1. URL: http://dx.doi.org/10.1038/s41467-021-23319-1%20https://www.nature.com/articles/s41467-021-23319-1.

[39] J. Lu, R. Xiong, J. Tian, C. Wang, and F. Sun. “Deep learning to estimate lithium-ion battery state of health without additional degradation experiments”. In: *Nature Communications* 14.1 (2023), pp. 1–13. ISSN: 20411723. DOI: 10.1038/s41467-023-38458-w. URL: http://dx.doi.org/10.1038/s41467-023-38458-w.

[40] P. K. Jones, U. Stimming, and A. A. Lee. “Impedance-based forecasting of lithium-ion battery performance amid uneven usage”. In: *Nature Communications* 13.1 (2022). ISSN: 20411723. DOI: 10.1038/s41467-022-32422-w.

[41] C. Peng, Y. Chen, W. Gui, Z. Tang, and C. Li. “Remaining useful life prognosis of turbofan engines based on deep feature extraction and fusion”. In: *Scientific Reports* 12.1 (Apr. 2022), p. 6491. ISSN: 2045-2322. DOI: 10.1038/s41598-022-10191-2. URL: https://doi.org/10.1038/s41598-022-10191-2%20https://www.nature.com/articles/s41598-022-10191-2.

[42] M. Kerin, N. Hartono, and D. T. Pham. “Optimising remanufacturing decision-making using the bees algorithm in product digital twins”. In: *Scientific Reports* 13.1 (Jan. 2023), p. 701. ISSN: 2045-2322. DOI: 10.1038/s41598-023-27631-2. URL: https://doi.org/10.1038/s41598-023-27631-2%20https://www.nature.com/articles/s41598-023-27631-2.

[43] W. N. Ismail, M. M. Hassan, H. A. Alsalamah, and G. Fortino. “CNN-based health model for regular health factors analysis in internet-of-medical things environment”. In: *IEEE Access* 8 (2020), pp. 52541–52549.

[44] Y. Zhao, J. Xu, Q. Chen, *et al.* “Analysis of curative effect and prognostic factors of radiotherapy for esophageal cancer based on the CNN”. In: *Journal of healthcare engineering* 2021 (2021).

[45] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun. “Doctor ai: Predicting clinical events via recurrent neural networks”. In: *Machine learning for healthcare conference*. PMLR. 2016, pp. 301–318.

[46] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. D. Howell, C. Cui, G. S. Corrado, and J. Dean. “Scalable and accurate deep learning with electronic health records”. In: *npj Digital Medicine* 1.1 (May 2018), p. 18. ISSN: 2398-6352. DOI: 10.1038/s41746-018-0029-1. URL: https://doi.org/10.1038/s41746-018-0029-1.

[47] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.11.016. URL: https://www.sciencedirect.com/science/article/pii/S0888327017305988.

[48] R. Zhao, R. Yan, J. Wang, and K. Mao. "Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks". In: *Sensors* 17 (Jan. 2017), p. 273. DOI: 10.3390/s17020273.

[49] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu. "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data". In: *Mechanical Systems and Signal Processing* 72-73 (2016), pp. 303–315. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2015.10.025. URL: https://www.sciencedirect.com/science/article/pii/S0888327015004859.

[50] T. Junbo, L. Weining, A. Juneng, and W. Xueqian. "Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder". In: *The 27th Chinese Control and Decision Conference (2015 CCDC)*. 2015, pp. 4608–4613. DOI: 10.1109/CCDC.2015.7162738.

[51] W. Lu, X. Wang, C. Yang, and T. Zhang. "A novel feature extraction method using deep neural network for rolling bearing fault diagnosis". In: *The 27th Chinese Control and Decision Conference (2015 CCDC)*. 2015, pp. 2427–2431. DOI: 10.1109/CCDC.2015.7162328.

[52] X. Li, W. Zhang, and Q. Ding. "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism". In: *Signal Processing* 161 (2019), pp. 136–154. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2019.03.019. URL: https://www.sciencedirect.com/science/article/pii/S0165168419301124.

[53] Y. Chen, G. Peng, Z. Zhu, and S. Li. "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction". In: *Applied Soft Computing* 86 (2020), p. 105919. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105919. URL: https://www.sciencedirect.com/science/article/pii/S1568494619307008.

[54] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering". In: *ACM Computing Surveys* 31.3 (Sept. 1999), pp. 264–323. ISSN: 0360-0300. DOI: 10.1145/331499.331504. URL: https://dl.acm.org/doi/10.1145/331499.331504.

[55] T. Warren Liao. "Clustering of time series data—a survey". In: *Pattern Recognition* 38.11 (Nov. 2005), pp. 1857–1874. ISSN: 00313203. DOI: 10.1016/j.patcog.2005.01.025. URL: https://www.sciencedirect.com/science/article/pii/S0031320305001305%20https://linkinghub.elsevier.com/retrieve/pii/S0031320305001305.

[56] S. Al-Fahdawi, A. S. Al-Waisy, D. Q. Zeebaree, R. Qahwaji, H. Natiq, M. A. Mohammed, J. Nedoma, R. Martinek, and M. Deveci. "Fundus-DeepNet: Multi-label deep learning classification system for enhanced detection of multiple ocular diseases through data fusion of fundus images". In: *Information Fusion* 102 (2024),

p. 102059. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.102059. URL: https://www.sciencedirect.com/science/article/pii/S1566253523003755.

[57] J. Y. Jang, G. Yoo, T. Lee, Y. Uh, and J. Kim. “Identification of the robust predictor for sepsis based on clustering analysis”. In: *Scientific Reports* 12.1 (Feb. 2022), p. 2336. ISSN: 2045-2322. DOI: 10.1038/s41598-022-06310-8. URL: https://doi.org/10.1038/s41598-022-06310-8%20https://www.nature.com/articles/s41598-022-06310-8.

[58] R. Gutierrez-Osuna. “Pattern analysis for machine olfaction: a review”. In: *IEEE Sensors Journal* 2.3 (June 2002), pp. 189–202. ISSN: 1530-437X. DOI: 10.1109/JSEN.2002.800688. URL: http://ieeexplore.ieee.org/document/1021060/.

[59] H. Wang, H. Wang, G. Jiang, J. Li, and Y. Wang. “Early Fault Detection of Wind Turbines Based on Operational Condition Clustering and Optimized Deep Belief Network Modeling”. In: *Energies* 12.6 (Mar. 2019), p. 984. ISSN: 1996-1073. DOI: 10.3390/en12060984. URL: https://www.mdpi.com/1996-1073/12/6/984.

[60] Z. Xu, M. Bashir, W. Zhang, Y. Yang, X. Wang, and C. Li. “An intelligent fault diagnosis for machine maintenance using weighted soft-voting rule based multi-attention module with multi-scale information fusion”. In: *Information Fusion* 86-87 (2022), pp. 17–29. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2022.06.005. URL: https://www.sciencedirect.com/science/article/pii/S1566253522000562.

[61] J. Li, H. Izakian, W. Pedrycz, and I. Jamal. “Clustering-based anomaly detection in multivariate time series data”. In: *Applied Soft Computing* 100 (Mar. 2021), p. 106919. ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106919. URL: https://doi.org/10.1016/j.asoc.2020.106919%20https://linkinghub.elsevier.com/retrieve/pii/S1568494620308577.

[62] D. López, I. Aguilera-Martos, M. García-Barzana, F. Herrera, D. García-Gil, and J. Luengo. “Fusing anomaly detection with false positive mitigation methodology for predictive maintenance under multivariate time series”. In: *Information Fusion* 100 (2023), p. 101957. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101957. URL: https://www.sciencedirect.com/science/article/pii/S1566253523002737.

[63] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas. “A Review of Data-Driven Decision-Making Methods for Industry 4.0 Maintenance Applications”. In: *Electronics* 10.7 (Mar. 2021), p. 828. ISSN: 2079-9292. DOI: 10.3390/electronics10070828. URL: https://www.mdpi.com/2079-9292/10/7/828.

[64] M. A. G. Santos, R. Munoz, R. Olivares, P. P. R. Filho, J. D. Ser, and V. H. C. de Albuquerque. “Online heart monitoring systems on the internet of health things environments: A survey, a reference model and an outlook”. In: *Information Fusion* 53 (2020), pp. 222–239. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.06.004. URL: https://www.sciencedirect.com/science/article/pii/S1566253518309035.

[65] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra. “Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0”. In: *Information Fusion* 50 (2019), pp. 92–111. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2018.10.005. URL: https://www.sciencedirect.com/science/article/pii/S1566253518304706.

[66] J. Xu, Y. Ren, X. Shi, H. T. Shen, and X. Zhu. “UNTIE: Clustering analysis with disentanglement in multi-view information fusion”. In: *Information Fusion* 100 (2023), p. 101937. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101937. URL: https://www.sciencedirect.com/science/article/pii/S1566253523002531.

[67] G. M. Dimitri, S. Spasov, A. Duggento, L. Passamonti, P. Lió, and N. Toschi. “Multimodal and multicontrast image fusion via deep generative models”. In: *Information Fusion* 88 (2022), pp. 146–160. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2022.07.017. URL: https://www.sciencedirect.com/science/article/pii/S1566253522000720.

[68] J. Xie, R. Girshick, and A. Farhadi. “Unsupervised Deep Embedding for Clustering Analysis”. In: *33rd International Conference on Machine Learning, ICML 2016* 1 (Nov. 2015), pp. 740–749. URL: http://arxiv.org/abs/1511.06335.

[69] E. Mariotti, J. M. Alonso Moral, and A. Gatt. “Exploring the balance between interpretability and performance with carefully designed constrainable Neural Additive Models”. In: *Information Fusion* 99 (2023), p. 101882. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101882. URL: https://www.sciencedirect.com/science/article/pii/S1566253523001987.

[70] J. Sill. “Monotonic Networks”. In: *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*. NIPS ’97. Cambridge, MA, USA: MIT Press, 1998, pp. 661–667. ISBN: 0262100762.

[71] Hong Zhang and Zhen Zhang. “Feedforward networks with monotone constraints”. In: *IJCNN’99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)* 3 (1999), pp. 1820–1823. DOI: 10.1109/IJCNN.1999.832655. URL: http://ieeexplore.ieee.org/document/832655/.

[72] D. Runje and S. M. Shankaranarayana. “Constrained Monotonic Neural Networks”. In: *Proceedings of Machine Learning Research* 202 (May 2022), pp. 29338–29353. ISSN: 26403498. DOI: https://doi.org/10.48550/arXiv.2205.11775. URL: http://arxiv.org/abs/2205.11775.

[73] A. Wehenkel and G. Louppe. “Unconstrained Monotonic Neural Networks”. In: *Advances in Neural Information Processing Systems* 32 (Aug. 2019). Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett. DOI: https://doi.org/10.48550/arXiv.1908.05164. URL: https://proceedings.neurips.cc/paper˙files/paper/2019/file/2a084e55c87b1ebcdaad1f62fdbbac8e-Paper.pdf%20http://arxiv.org/abs/1908.05164.

[74] X. Liu, X. Han, N. Zhang, and Q. Liu. "Certified Monotonic Neural Networks". In: *Advances in Neural Information Processing Systems* 2020-Decem.NeurIPS (Nov. 2020). ISSN: 10495258. DOI: https://doi.org/10.48550/arXiv.2011.10219. URL: http://arxiv.org/abs/2011.10219.

[75] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus. "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". In: *JAMA* 315.8 (Feb. 2016), p. 801. ISSN: 0098-7484. DOI: 10.1001/jama.2016.0287. URL: http://jama.jamanetwork.com/article.aspx?doi=10.1001/jama.2016.0287.

[76] R. Liu, K. M. Hunold, J. M. Caterino, and P. Zhang. "Estimating treatment effects for time-to-treatment antibiotic stewardship in sepsis". In: *Nature Machine Intelligence* 5.4 (Apr. 2023), pp. 421–431. ISSN: 2522-5839. DOI: 10.1038/s42256-023-00638-0. URL: https://www.nature.com/articles/s42256-023-00638-0.

[77] K. Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 08936080. DOI: 10.1016/0893-6080(91)90009-T. URL: https://linkinghub.elsevier.com/retrieve/pii/089360809190009T.

[78] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

[79] G. Brugnara, M. Baumgartner, E. D. Scholze, K. Deike-Hofmann, K. Kades, J. Scherer, S. Denner, H. Meredig, A. Rastogi, M. A. Mahmutoglu, C. Ulfert, U. Neuberger, S. Schönenberger, K. Schlamp, Z. Bendella, T. Pinetz, C. Schmeel, W. Wick, P. A. Ringleb, R. Floca, M. Möhlenbruch, A. Radbruch, M. Bendszus, K. Maier-Hein, and P. Vollmuth. "Deep-learning based detection of vessel occlusions on CT-angiography in patients with suspected acute ischemic stroke". In: *Nature Communications* 14.1 (2023). ISSN: 20411723. DOI: 10.1038/s41467-023-40564-8.

[80] S. Müller, C. Sauter, R. Shunmugasundaram, N. Wenzler, V. De Andrade, F. De Carlo, E. Konukoglu, and V. Wood. "Deep learning-based segmentation of lithium-ion battery microstructures enhanced by artificially generated electrodes". In: *Nature Communications* 12.1 (2021), pp. 1–12. ISSN: 20411723. DOI: 10.1038/s41467-021-26480-9.

[81] D. Kahneman and A. Tversky. "Choices, values, and frames." In: *American Psychologist* 39.4 (Apr. 1984), pp. 341–350. ISSN: 1935-990X. DOI: 10.1037/0003-066X.39.4.341. URL: https://doi.org/10.1007/s12530-020-09345-2%20https://link.springer.com/10.1007/s12530-020-09345-2%20http://doi.apa.org/getdoi.cfm?doi=10.1037/0003-066X.39.4.341.

[82] B. Cao, S. Wang, R. Bai, B. Zhao, Q. Li, M. Lv, and G. Liu. “Boundary optimization of inclined coal seam open-pit mine based on the ISSA–LSSVR coal price prediction method”. In: *Scientific Reports* 13.1 (May 2023), p. 7527. ISSN: 2045-2322. DOI: 10.1038/s41598-023-34641-7. URL: https://doi.org/10.1038/s41598-023-34641-7%20https://www.nature.com/articles/s41598-023-34641-7.

[83] Z. Chao, F. Pu, Y. Yin, B. Han, and X. Chen. “Research on Real-Time Local Rainfall Prediction Based on MEMS Sensors”. In: *Journal of Sensors* 2018 (June 2018), pp. 1–9. ISSN: 1687-725X. DOI: 10.1155/2018/6184713. URL: https://www.hindawi.com/journals/js/2018/6184713/.

[84] T. Zebin, S. Rezvy, and T. J. Chaussalet. “A deep learning approach for length of stay prediction in clinical settings from medical records”. In: *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2019, pp. 1–5. DOI: 10.1109/CIBCB.2019.8791477.

[85] M. Scherpf, F. Gräßer, H. Malberg, and S. Zaunseder. “Predicting sepsis with a recurrent neural network using the MIMIC III database”. In: *Computers in Biology and Medicine* 113 (2019), p. 103395. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2019.103395. URL: https://www.sciencedirect.com/science/article/pii/S0010482519302720.

[86] S. Fu, S. Zhong, L. Lin, and M. Zhao. “A Novel Time-Series Memory Auto-Encoder With Sequentially Updated Reconstructions for Remaining Useful Life Prediction”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2022), pp. 7114–7125. DOI: 10.1109/TNNLS.2021.3084249.

[87] O. Asif, S. A. Haider, S. R. Naqvi, J. F. W. Zaki, K.-S. Kwak, and S. M. R. Islam. “A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset”. In: *IEEE Access* 10 (2022), pp. 95425–95440. DOI: 10.1109/ACCESS.2022.3203406.

[88] T. Kontogiannis, M. Salinas-Camus, and N. Eleftheroglou. “Hidden Markov model applications: Aviation prognostics”. In: *Stochastic Modeling and Statistical Methods: Advances and Applications*. Academic Press, 2025, pp. 191–213. ISBN: 9780443316951. DOI: https://doi.org/10.1016/B978-0-44-331694-4.00015-3. URL: https://doi.org/10.1016/B978-0-44-331694-4.00015-3.

[89] T. Quach and M. Farooq. “Maximum likelihood track formation with the Viterbi algorithm”. In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*. Vol. 1. 1994, 271–276 vol.1. DOI: 10.1109/CDC.1994.410918.

[90] M. Salinas-Camus and N. Eleftheroglou. “Uncertainty in Aircraft Turbofan Engine Prognostics on the C-MAPSS Dataset”. In: *Proceedings of the European Conference of the PHM Society 2024*. Vol. 8. 1. 2024, p. 10. DOI: https://doi.org/10.36001/phme.2024.v8i1.4007.

[91] S. R. Moss and H. C. Prescott. “Current Controversies in Sepsis Management”. eng. In: *Seminars in Respiratory and Critical Care Medicine* 40.05 (Oct. 2019), pp. 594–603. ISSN: 1069-3424. DOI: 10.1055/s-0039-1696981. URL: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0039-1696981.

[92] E. L. Kaplan and P. Meier. "Nonparametric Estimation from Incomplete Observations". In: *Journal of the American Statistical Association* 53.282 (June 1958), p. 457. ISSN: 01621459. DOI: 10.2307/2281868. URL: http://www.jstor.org/stable/2281868%20https://www.jstor.org/stable/2281868?origin=crossref.

[93] S. M. Lauritsen, M. Kristensen, M. V. Olsen, M. S. Larsen, K. M. Lauritsen, M. J. Jørgensen, J. Lange, and B. Thiesson. "Explainable artificial intelligence model to predict acute critical illness from electronic health records". In: *Nature Communications* 11.1 (2020), pp. 1–11. ISSN: 20411723. DOI: 10.1038/s41467-020-17431-x. URL: http://dx.doi.org/10.1038/s41467-020-17431-x.

[94] C. Wu, Y. Zhang, S. Nie, D. Hong, J. Zhu, Z. Chen, B. Liu, H. Liu, Q. Yang, H. Li, G. Xu, J. Weng, Y. Kong, Q. Wan, Y. Zha, C. Chen, H. Xu, Y. Hu, Y. Shi, Y. Zhou, G. Su, Y. Tang, M. Gong, L. Wang, F. Hou, Y. Liu, and G. Li. "Predicting in-hospital outcomes of patients with acute kidney injury". In: *Nature Communications* 14.1 (June 2023), p. 3739. ISSN: 2041-1723. DOI: 10.1038/s41467-023-39474-6. URL: https://www.nature.com/articles/s41467-023-39474-6.

[95] G. Kong, K. Lin, and Y. Hu. "Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU". In: *BMC Medical Informatics and Decision Making* 20.1 (Dec. 2020), p. 251. ISSN: 1472-6947. DOI: 10.1186/s12911-020-01271-2. URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-01271-2.

[96] M. M. Islam, T. Nasrin, B. A. Walther, C.-C. Wu, H.-C. Yang, and Y.-C. Li. "Prediction of sepsis patients using machine learning approach: A meta-analysis". In: *Computer Methods and Programs in Biomedicine* 170 (Mar. 2019), pp. 1–9. ISSN: 01692607. DOI: 10.1016/j.cmpb.2018.12.027. URL: https://doi.org/10.1016/j.cmpb.2018.12.027%20https://linkinghub.elsevier.com/retrieve/pii/S016926071831602X.

[97] M. Dong, D. He, P. Banerjee, and J. Keller. "Equipment health diagnosis and prognosis using hidden semi-Markov models". In: *The International Journal of Advanced Manufacturing Technology* 30.7-8 (Oct. 2006), pp. 738–749. ISSN: 0268-3768. DOI: 10.1007/s00170-005-0111-0. URL: https://doi.org/10.1007/s00170-005-0111-0%20http://link.springer.com/10.1007/s00170-005-0111-0.

[98] A. E. W. Johnson, T. J. Pollard, and R. G. Mark. "Reproducibility in critical care: a mortality prediction case study". In: *Proceedings of the 2nd Machine Learning for Healthcare Conference*. Ed. by F. Doshi-Velez, J. Fackler, D. Kale, R. Ranganath, B. Wallace, and J. Wiens. Vol. 68. Proceedings of Machine Learning Research. PMLR, 2017, pp. 361–376. URL: https://proceedings.mlr.press/v68/johnson17a.html.

[99] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan. "Multitask learning and benchmarking with clinical time series data". In: *Scientific data* 6.1 (2019), p. 96.

[100] S. Purushotham, C. Meng, Z. Che, and Y. Liu. “Benchmarking deep learning models on large healthcare datasets”. In: *Journal of Biomedical Informatics* 83 (2018), pp. 112–134. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2018.04.007. URL: https://www.sciencedirect.com/science/article/pii/S1532046418300716.

[101] H. L. Li-wei, R. P. Adams, L. Mayaud, G. B. Moody, A. Malhotra, R. G. Mark, and S. Nemati. “A physiological time series dynamics-based approach to patient monitoring and outcome prediction”. In: *IEEE journal of biomedical and health informatics* 19.3 (2014), pp. 1068–1076.

[102] J. Lee *et al.* “Patient-specific predictive modeling using random forests: an observational study for the critically ill”. In: *JMIR medical informatics* 5.1 (2017), e6690.

[103] M. C. Data and R. Pirracchio. “Mortality prediction in the icu based on mimic-ii results from the super icu learner algorithm (sicula) project”. In: *Secondary Analysis of Electronic Health Records* (2016), pp. 295–313.

[104] J. .-. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. K. Reinhart, P. M. Suter, and L. G. Thijs. “The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure”. In: *Intensive Care Medicine* 22.7 (July 1996), pp. 707–710. ISSN: 0342-4642. DOI: 10.1007/BF01709751. URL: https://doi.org/10.1007/BF01709751%20http://link.springer.com/10.1007/BF01709751.

[105] R. P. Moreno, P. G. H. Metnitz, E. Almeida, B. Jordan, P. Bauer, R. A. Campos, G. Iapichino, D. Edbrooke, M. Capuzzo, and J.-R. Le Gall. “SAPS 3—From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission”. eng. In: *Intensive Care Medicine* 31.10 (Oct. 2005), pp. 1345–1355. ISSN: 0342-4642. DOI: 10.1007/s00134-005-2763-5. URL: https://link.springer.com/10.1007/s00134-005-2763-5.

[106] W. A. KNAUS, E. A. DRAPER, D. P. WAGNER, and J. E. ZIMMERMAN. “APACHE II A severity of disease classification system”. In: *Critical Care Medicine* 13.10 (Oct. 1985), pp. 818–829. ISSN: 0090-3493. DOI: 10.1097/00003246-198510000-00009. URL: https://journals.lww.com/ccmjournal/fulltext/1985/10000/apache˙ii˙˙a˙severity˙of˙disease˙classification.9.aspx%20http://journals.lww.com/00003246-198510000-00009.

[107] A. Natekin and A. Knoll. “Gradient boosting machines, a tutorial”. In: *Frontiers in Neurorobotics* 7 (2013). ISSN: 1662-5218. DOI: 10.3389/fnbot.2013.00021. URL: https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021%20http://journal.frontiersin.org/article/10.3389/fnbot.2013.00021/abstract.

[108] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. “Support Vector Regression Machines”. In: *Advances in Neural Information Processing Systems*. Ed. by M. C. Mozer, M. Jordan, and T. Petsche. Vol. 9. MIT Press, 1996. URL: https://proceedings.neurips.cc/paper˙files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf.

[109] K. Deng, X. Zhang, Y. Cheng, Z. Zheng, F. Jiang, W. Liu, and J. Peng. "A remaining useful life prediction method with long-short term feature processing for aircraft engines". In: *Applied Soft Computing* 93 (Aug. 2020), p. 106344. ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106344. URL: https://www.sciencedirect.com/science/article/pii/S1568494620302842%20https://linkinghub.elsevier.com/retrieve/pii/S1568494620302842.

[110] H. Li, W. Zhao, Y. Zhang, and E. Zio. "Remaining useful life prediction using multi-scale deep convolutional neural network". In: *Applied Soft Computing* 89 (Apr. 2020), p. 106113. ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106113. URL: https://www.sciencedirect.com/science/article/pii/S1568494620300533%20https://linkinghub.elsevier.com/retrieve/pii/S1568494620300533.

[111] S. Yu, Z. Wu, X. Zhu, and M. Pecht. "A Domain Adaptive Convolutional LSTM Model for Prognostic Remaining Useful Life Estimation Under Variant Conditions". In: *2019 Prognostics and System Health Management Conference (PHM-Paris)* (May 2019), pp. 130–137. DOI: 10.1109/PHM-Paris.2019.00030. URL: https://ieeexplore.ieee.org/document/8756479/.

[112] C.-G. Huang, H.-Z. Huang, W. Peng, and T. Huang. "Improved trajectory similarity-based approach for turbofan engine prognostics". In: *Journal of Mechanical Science and Technology* 33.10 (Oct. 2019), pp. 4877–4890. ISSN: 1738-494X. DOI: 10.1007/s12206-019-0928-3. URL: https://doi.org/10.1007/s12206-019-0928-3%20http://link.springer.com/10.1007/s12206-019-0928-3.

[113] D. Laredo, Z. Chen, O. Schütze, and J.-Q. Sun. "A neural network-evolutionary computational framework for remaining useful life estimation of mechanical systems". In: *Neural Networks* 116 (Aug. 2019), pp. 178–187. ISSN: 08936080. DOI: 10.1016/j.neunet.2019.04.016. URL: https://doi.org/10.1016/j.neunet.2019.04.016%20https://linkinghub.elsevier.com/retrieve/pii/S0893608019301224.

[114] H. Cai, J. Feng, W. Li, Y.-M. Hsu, and J. Lee. "Similarity-based Particle Filter for Remaining Useful Life prediction with enhanced performance". In: *Applied Soft Computing* 94 (Sept. 2020), p. 106474. ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106474. URL: https://doi.org/10.1016/j.asoc.2020.106474%20https://linkinghub.elsevier.com/retrieve/pii/S1568494620304130.

[115] Z. Ye and J. Yu. "Health condition monitoring of machines based on long short-term memory convolutional autoencoder". In: *Applied Soft Computing* 107 (2021), p. 107379. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107379. URL: https://www.sciencedirect.com/science/article/pii/S1568494621003021.

[116] L. van der Maaten and G. Hinton. "Viualizing data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[117] G. Ciaburro and G. Iannace. "Machine-Learning-Based Methods for Acoustic Emission Testing: A Review". In: *Applied Sciences* 12.20 (Oct. 2022), p. 10476. ISSN: 2076-3417. DOI: 10.3390/app122010476. URL: https://www.mdpi.com/2076-3417/12/20/10476.

[118] N. Eleftheroglou and T. Loutas. “Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes”. In: *Structural Health Monitoring* 15.4 (May 2016), pp. 473–488. ISSN: 1475-9217. DOI: 10.1177/1475921716646579. URL: https://doi.org/10.1177/1475921716646579.

[119] D. Pathak, P. Krahenbuhl, and T. Darrell. “Constrained Convolutional Neural Networks for Weakly Supervised Segmentation”. In: *2015 IEEE International Conference on Computer Vision (ICCV)* 2015 Inter (Dec. 2015), pp. 1796–1804. ISSN: 15505499. DOI: 10.1109/ICCV.2015.209. URL: http://ieeexplore.ieee.org/document/7410566/.

[120] S. You, D. Ding, K. Canini, J. Pfeifer, and M. Gupta. “Deep Lattice Networks and Partial Monotonic Functions”. In: *Advances in Neural Information Processing Systems* 2017-Decem.Nips (Sept. 2017), pp. 2982–2990. ISSN: 10495258. URL: http://arxiv.org/abs/1709.06680.

4

A BAYESIAN INFERENCE-BASED FRAMEWORK FOR MODELING IMPERFECT POST-REPAIR BEHAVIOR OF RUL

This chapter proposes a Bayesian approach for modeling imperfect repairs and links them to the RUL of a component. Imperfect repairs, often a cost-effective alternative to full replacements, play a crucial role in maintenance strategies. By creating a model that accounts for both single and multiple imperfect repairs, decision-makers can gain valuable insights into the expected performance and reliability of a component after multiple successive repairs. This model can help optimize maintenance decisions by predicting future structural behavior even after sequential repairs, thus balancing cost and operational efficiency.

Part of this chapter has been published in Expert Systems with Applications **288**, 127723 (2025) [1].

4.1. INTRODUCTION

Maintaining primary load-bearing structures and components is vital to ensuring their safety, functionality, and durability. These structures, present in aircraft, bridges, buildings, industrial facilities, and infrastructure, experience deterioration due to diverse environmental and operational conditions, making regular maintenance actions necessary. These actions are typically tailored to the specific component, its operational environment, and the potential risks involved. In the past decades, the maintenance actions have evolved from corrective (a component is maintained after reaching failure) to preventive maintenance (a component is maintained at predefined time slots to prevent failure). Currently, a further shift towards CBM is being investigated, which is ‘a policy that uses information about the health condition of systems and structures to identify optimal maintenance interventions over time, increasing the efficiency of maintenance operations’ [2]. This offers earlier insights into the current and future condition of the component, which extends its useful life and minimizes maintenance costs.

Regardless of the maintenance strategy that has been implemented, a component can be either replaced or repaired. Nonetheless, the shortage of replacement parts and the increased costs and material waste associated with replacements have encouraged the exploration of effective repair methodologies. Repairing a component may lead to varying outcomes, which according to De Carlo and Arleo [3], are classified into five types:

- *Perfect maintenance.* Restores the system to its As-Good-As-New (AGAN) condition. This is the same as replacing the element or system with a brand-new one.

- *Imperfect maintenance.* Restores the system’s damage state somewhere between the AGAN and As-Bad-As-Old (ABAO) condition, i.e. the condition before maintenance.

- *Minimal maintenance.* Restores the system to the ABAO condition.

- *Worse maintenance.* This type of maintenance outcome accidentally causes a worsening operating condition of the system, i.e. the risk of failure and the deterioration level or speed, or both, increase.

- *Worst maintenance.* Additionally, to worse maintenance, failure occurs to the system after repair.

Figure 4.1 summarizes the above-mentioned maintenance actions and types. A common misassumption about a repair is the expectation of a perfect restoration of the component’s initial properties and functionality. However, repair actions rarely achieve the AGAN condition [4] (here, the term ‘condition’ is identical to the term ‘state’). This results in an imperfect repair outcome and is the most realistic type of maintenance.

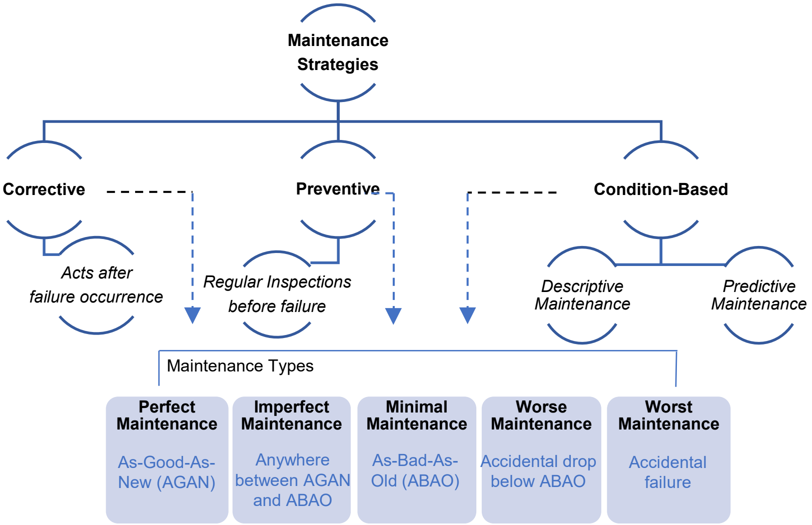


Figure 4.1.: Maintenance strategies and actions. Figure inspired by [3].

An imperfect repair, if not properly accounted for, may lead to an unexpected failure of the component with the implication of an unscheduled maintenance action. In the context of corrective and preventive maintenance – the very first applications of imperfect maintenance modeling– plenty of studies exist that address various methodologies concerning the estimation of the post-repair state. Nakagawa et al. [5] addresses the effects of imperfect repairs on the recovery of the system under consideration by utilizing the (p,q) rule, that is, the system returns to its AGAN condition with probability p and its ABAO condition with probability $q{=}1{-}p$. Therefore, minimal $(q{=}0)$ and perfect $(p{=}1)$ repair are special cases of imperfect repair. Block et al. [6] developed the $(p(t),q(t))$ rule, an extension of the above research where the imperfect repairs are age-dependent. A perfect repair is considered with probability $p(t)$ and a minimal repair with probability $q(t){=}1{-}p(t)$, where t is the age of the component at the time of repair. Another study proposed the improvement factor method, where the failure rate is reduced somewhere between its ABAO and AGAN values [7]. Kijima et al. [8] recommended a virtual age model where repairs are assumed to restore the system in the range of [ABAO, AGAN]. Subsequently, numerous virtual age models have been introduced [9, 10] collectively recognized as Kijima models. Several alternative treatment methodologies have been developed and documented within the literature, such as the shock model [11], (α,β) rule [12], and multiple (p,q) rule models [13]. Despite the multitude of approaches available, the widely adopted methods remain modifications of Kijima models. All these models assume that maintenance occurs either after a failure (corrective maintenance) or at predetermined intervals (preventive maintenance). Since these models define how a system is restored after maintenance, they are more suited for failure-response strategies rather than predictive ones, signifying the need for CBM.

The emerging trend to switch towards CBM demanded advanced modelling of imperfect repairs via the introduction of physics-based, data-driven, or hybrid approaches (incorporation of data-driven approaches into physical models). Here, physics-based approaches refer to closed-form mathematical equations that describe the process. Regarding CBM, where prognostics are required, the most common approach to model imperfect repairs is to incorporate the estimated state of the component after a repair into a predefined physical degradation model [14–16]. This can be done by assuming in prior a physical model which expresses the component's degradation behavior and then adding the estimated state to this model. However, this approach is case-specific and requires the full knowledge of the dynamics behind the application. Another family of imperfect repair approaches under the CBM umbrella considers parameterized physical degradation models [17–20] where the parameters, defined as random variables, are updated by Bayesian inference. Although the need for fully understanding the dynamics of the process is relaxed, choosing a suitable physical model is still a considerable challenge because it requires significant domain expertise. An extension of these approaches is the integration of imperfect repairs into a data-driven-based degradation model [21–25]. In such cases, the imperfect repair is parameterized inside the degradation model in a manner that affects the speed or level of degradation after the repair. Despite this solution offering a unified data-driven model that combines prognostics with imperfect repairs, it still requires a clear understanding of the data related to the component's degradation behavior and of the effects of repairing. For systems with complex non-linear degradation dynamics, there is a need for high-order models whose parameter estimation can be computationally expensive. Additionally, building both a prognostic and an imperfect repair model from scratch is time-consuming and difficult to optimize in an end-to-end manner.

Furthermore, these approaches consider a stationary transition function from one degradation state to another [16, 23–26], assuming a more deterministic behavior. However, it has been noted that since a time-dependent and non-stationary degradation process is usually observed, the above methodologies might not accurately capture the recovery of the component after an imperfect repair [27]. Temporal variability emerges when alterations in the degradation rate are observed, leading to potentially multiple transitions from the same health state to different recovery states. The abovementioned statement supports that proactively scheduling a maintenance action for a component subject to one or more imperfect repairs is challenging due to the inherent stochasticity. Finally, all data-driven or hybrid approaches require the component to reach the End of Life (EOL) condition after an imperfect repair. For real-case scenarios, this requires several components to reach failure in order to train an imperfect repair model, which can be cost-prohibitive. Additionally, such data may be simply not available for real applications [28].

Given the aforementioned limitations of existing works around imperfect repairs, the following research gaps can be identified:

- The literature lacks methods to proactively (before the imperfect repair takes place) predict the effects of imperfectly repairing a component. This gap limits the optimization of maintenance planning.

- An imperfect repair introduces significant uncertainty into the prediction of the Remaining Useful Life (RUL). This is because the data collected prior to the repair may no longer fully reflect the asset's health state after the repair, increasing post-repair uncertainty and making reliable predictions more challenging.

- All approaches are case-specific, requiring modelling the degradation and imperfect repair dynamics from scratch. Combining these models into a unified framework can be very challenging and time-consuming.

- To build an imperfect repair model, datasets that contain information regarding the component's EOL condition after an imperfect repair are required. Several components should reach failure after a repair to train the model, which is cost-prohibitive or absent in most real-world applications.

To address these challenges and the associated limitations of the existing frameworks, a new approach is proposed where the effect of imperfect repairs is converted into a stochastic RUL increase in the range of [ABAO, AGAN], which is defined as recovery. Instead of constructing a complex degradation model and incorporating the (often unknown) dynamics of imperfect repairs, all the information regarding the component's degradation behavior and health state improvement post-repair is translated into a recovery of the RUL. Consequently, the required data for modelling imperfect repairs is solely two RUL data points, one right before and another after the imperfect repair. These data points can be estimated by any prognostic model. Here, the Hidden Semi Markov Models (HSMM) [29] are chosen due to their superior performance compared to other techniques [30–32].

Since this work additionally aims at practical applications where only raw sensor signals are available that are coming from monitoring the health condition of the structure, it is necessary to extract prognostic-related features from the data that could be used as input to the prognostic model. Extracting such information from raw data alligns perfectly with Artificial Neural Networks (ANN) approaches. For this task, the Deep Soft Monotonic Clustering (DSMC) model [33] is considered that operates on the foundation of ANN as a core mechanism for extracting prognostic-related features directly from raw data. Any other deep learning model can be chosen capable of extracting prognostic-related features.

Because both RUL and the effects of imperfect repair actions are stochastic and limited trajectories are available, Bayesian inference is chosen as the most appropriate solution to develop the imperfect repair model. In Bayesian inference, the choice of specific prior and likelihood distributions is unavoidable. In practical approaches, conjugate priors are selected to receive analytical solutions (e.g. normal prior – normal likelihood distribution produces a normal posterior distribution). However, possessing a generalizable (not task-specific) model implies handling different combinations of prior and likelihood distributions that are not necessarily conjugate and, thus, cannot be calculated analytically. This necessitates selecting a more sophisticated Bayesian inference technique, the Markov Chain Monte Carlo (MCMC) approach [34], to effectively accommodate these diverse choices.

Performing an imperfect repair action leads to a RUL increase that requires modelling. Directly considering RUL as the sole information linked to imperfect repair offers four advantages: i) modelling the imperfect repair effects via RUL could provide experts with knowledge regarding the component's health state post-repair behavior before even implementing the maintenance action, thus assisting in optimizing maintenance plans, ii) RUL values before and after an imperfect repair suffices to build the imperfect repair model, i.e. to estimate the stochastic recovery, iii) the imperfect repair model only utilizes the output of the prognostic model (RUL prediction), therefore these models operate independently and they do not have to be combined, and iv) the recovery distribution can be estimated without necessitating the component to reach failure post-repair. Building on the first benefit, being able to predict the recovery of RUL before actually performing the maintenance action (by utilizing the estimated recovery distribution), enables the optimization of maintenance planning proactively, under a CBM framework (this looks over the first research gap). The second advantage simplifies the process of modelling the imperfect repair effects since one distribution regarding recovery suffices to include the unknown dynamics that are hidden in the raw signals (challenging the second research gap). This also means that this approach can be implemented directly with raw data. The third advantage supports that any prognostic model capable of predicting RUL under uncertainty can be incorporated, providing flexibility and allowing the focus to remain on imperfect repair modeling (thus facing the third research gap). The last benefit requires only the last and first RUL data points before and after the imperfect repair respectively to model the distribution of recovery. Consequently, in a real-case scenario, the examined component does not have to reach failure, i.e. EOL, to create a dataset for training (surpassing the fourth research gap).

Finally, the scientific contribution and novelty of this research can be summarized as follows:

- The imperfect repair model could offer valuable insights into the component's post-repair condition proactively, before maintenance occurs, thus assisting in improving maintenance planning policies.

- This work embarks on developing a data-driven imperfect repair model by looking at it from an integration point of view under the Prognostics and Health Management (PHM) umbrella that could work independently of the prognostic and decision-making phases. This offers flexibility and alleviates the need to modify or replace existing well-established degradation, prognostic, and decision-making models.

- This research marks the first application of Bayesian inference using MCMC algorithms for modelling imperfect repairs. This approach allows for a flexible selection of prior-likelihood distribution combinations based on domain expertise, enhancing generalizability and adaptability to different repair techniques (though not required).

- Acquiring data from repaired components that reached the EOL is not necessary to train the imperfect repair model, thus offering better data availability to train the models.

For the remainder of this chapter, the terms 'imperfect repair' and 'repair' are assumed to be interchangeable. The effectiveness of the repair model is assessed through evaluation in a real-case scenario involving an experimental campaign with repairing open-hole aluminium specimens subject to tension-tension fatigue experiments repaired with rectangular Carbon Fiber Reinforced Polymer (CFRP) patches. The decision of implementing such an experiment with aluminium specimens is to ease the process of repairing. Nevertheless, it should be noted that this methodology is independent of the component's material properties and the repair process, enhancing its applicability and generalizability.

4.2. THEORY

In this section, the methodologies for predicting the RUL and constructing the imperfect repair model are described. These methodologies represent the Bayesian inference and the MCMC algorithms responsible for constructing the imperfect repair model, whilst DSMC and HSMM constitute the fundamental chosen models for performing feature extraction, HI construction, and prognosis, where RUL prediction is a necessary input for the proposed model.

4.2.1. BAYESIAN INFERENCE

Building and training an imperfect repair model from data requires its parameterization. In general, two primary methodologies are employed for this purpose: the frequentist and Bayesian approaches. The frequentist statistics methodology does not involve calculating or utilizing the probability of a hypothesis. It does not necessitate establishing a prior probability distribution over parameters by relying solely on probabilities derived from observed data. Typically, the frequentist approach employs Maximum Likelihood Estimation as its primary objective function.

While this method is entirely data-centric, the Bayesian analysis integrates prior information and computes the probability that a specific hypothesis holds true by incorporating both prior probabilities and observed data. The Bayesian analysis enhances the accuracy of parameter estimations by combining expert knowledge with available data. Consequently, it can yield reliable parameter estimations even when dealing with limited datasets and expert knowledge is available. Moreover, it provides greater insights as it quantifies uncertainty regarding parameters, treating them as random variables. Therefore, given the scarcity of available data and the necessity to account for substantial uncertainty, the Bayesian approach is deemed preferable for constructing the models.

Within the Bayesian framework, the prior distributions representing the beliefs before encountering any data play a crucial role. Priors can either be non-informative (like a uniform distribution with broad boundary values) or informative (such as a normal distribution with a very small variance). When domain knowledge is limited, employing a non-informative prior is advisable due to the lack of information about the data. Conversely, if domain-specific information exists, an informative prior may lead to more accurate and less uncertain parameter estimations.

Subsequently, the observed data, constituting the likelihood distribution, is integrated with the prior through the Bayes theorem, resulting in the posterior distribution:

$$P(\theta|D) = \frac{P(D|\theta) * P(\theta)}{P(D)} \propto P(D|\theta) * P(\theta) \tag{4.1}$$

where θ is a set of parameters, D indicates the observed data, $P(\theta|D)$ is the posterior distribution of the parameters given the observed data, $P(D|\theta)$ is the likelihood distribution or the probability of the data given the parameters, $P(\theta)$ is the prior distribution over the parameters, and $P(D)$ is the distribution of the data that is treated as a normalization factor in Bayesian analysis and is called *evidence*, thus it is simply a constant which could be ignored. Consequently, removing the evidence leads to the right expression of the above equation, where the symbol $\propto$ means 'is proportional to'. Additional information on this practical simplification can be found in [35].

Applying Bayes' theorem to refine prior beliefs into posterior distributions based on acquired data forms the foundation of a statistical inference technique known as Bayesian inference. Subsequently, using the posterior distribution, it becomes possible to estimate the posterior predictive distribution of a random variable R, representing one or more unobserved samples from the observations D. This posterior predictive distribution accommodates uncertainty related to θ and is computed by integrating the distribution of R given θ across the posterior distribution:

$$P(\hat{R}\middle|D) = \int_{\theta} P(\hat{R}\middle|\theta,\ D)P(\theta|D)d\theta \tag{4.2}$$

It should be noted that the samples used to estimate R, i.e. $\hat{R}$, are independent of the observations D. Similarly, in a regression problem the above equations are reconstructed as follows:

$$P(\theta|X,\ Y) = \frac{P(Y|X,\ \theta) * P(\theta)}{P(Y|X)} \propto P(Y|X,\theta) * P(\theta) \tag{4.3}$$

$$p(\hat{y}|x,\ X,Y) = \int_{\theta} p(y|x,\ \theta)p(\theta|X,\ Y)d\theta \tag{4.4}$$

where $x \in X$ is a sample drawn from a dataset $D=\{X,Y\}$ with X and Y being random variables of the input and output respectively, and $\hat{y}$ is the predicted estimated random variable given a sample x. This process of estimating the posterior predictive distribution of y given a sample x from dataset D is called Bayesian regression.

Within Bayesian inference, correctly defining the nature of acquired data used for updating the prior beliefs -essentially, specifying the likelihood distribution- is crucial. Ideally, the likelihood should be formulated to encapsulate every conceivable combination of parameter values. When this is achievable, the equations mentioned earlier can be analytically resolved to save computational time and improve accuracy. This is facilitated by adopting a set of likelihood-prior distributions that ensure the resulting posterior distribution belongs to the same probability distribution family as the prior. In such instances, the prior and posterior are termed conjugate distributions, with the prior being referred to as a conjugate prior for the likelihood [36].

However, in practice, relying on conjugate priors presents considerable limitations regarding the variety of the distributions available. Addressing this challenge often necessitates numerical solutions employing techniques such as MCMC algorithms. These algorithms are utilized to manage the complexity associated with non-conjugate distributions, allowing for practical computation of posterior distributions.

MARKOV CHAIN MONTE CARLO

The fundamental concept behind MC involves drawing a sequence of samples from the posterior distribution to explore statistical measures such as mean, variance, median, and mode, among others. These drawn samples, often defined as simulation samples or iterations, allow for a comprehensive analysis of the posterior distribution. MCMC algorithms can be employed to estimate the posterior predictive distribution. This process involves generating random simulation samples derived from the posterior distribution through simulated Markov chains. A Markov chain constitutes a sequence of random samples, wherein the generation of the next sample relies on the current sample's position. It is mathematically proven that every Markov chain eventually converges to a stationary distribution when the number of simulation samples surpasses a certain threshold [37]. As the number of simulation samples increases, the Markov chain tends to converge towards regions of high density within the posterior distribution. Consequently, all the generated simulation samples can be regarded as random samples effectively representing the posterior distribution.

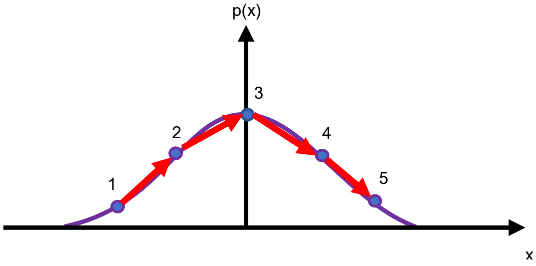


Figure 4.2.: Illustration of the MCMC process. The red arrows indicate the progression of MCMC iterations. Starting from an initial point (1), the MCMC algorithm proposes a new point (2) based on the current state. At each iteration, the algorithm decides whether to accept or reject the proposed point based on certain criteria (in this example, the Metropolis-Hastings acceptance criterion). If accepted, the chain moves to the new point; if rejected, it stays at the current point.

Figure 4.2 illustrates a simple MCMC implementation. Regardless of the specific MCMC algorithm employed, there exist practical recommendations and diagnostics aimed at enhancing performance and evaluating results, respectively:

- *Burn-in/Warmup*. In order to ensure a Markov chain reaches its stationary distribution, a considerable number of initial simulation samples need to be executed. These initial samples are commonly referred to as burn-in or warm-up samples within probabilistic programming languages. It is important to exclude these burn-in samples when approximating the posterior distribution, otherwise, the process of estimating the distribution will be unstable. Typical values for the number of burn-in simulation samples usually fall within a range of `[100,1000]` [37].

- *Acceptance rate*. The acceptance rate in MCMC represents the proportion of proposed samples that are accepted to be added to the chain. A value close to one indicates that nearly every proposed sample is accepted, slowing down convergence as it restricts exploration of the parameters' space. Conversely, a very low acceptance rate leads to a chain being restricted to local areas, inhibiting the generation of independent samples. It is crucial to define the acceptance rate before running the algorithm, often necessitating tuning. In certain algorithms such as the No U-Turn Sampler (NUTS) [38], initialization of the acceptance rate is essential, followed by automatic tuning during the process. This aids in achieving an optimal acceptance rate and balancing efficient exploration of the parameters' space.

- *Effective sample size (ESS)* [39, 40]. While the objective is to draw independent samples from a posterior distribution, achieving complete independence in practice is unfeasible due to the interdependence of samples generated by Markov chains based on their neighboring samples. ESS serves as a heuristic measure to estimate the degree of independence among the samples produced by the MCMC algorithm from the total number of simulation samples, excluding the warm-up samples. ESS is calculated as follows:

$$ESS = \frac{N}{1 + 2\sum_{t=1}^{\infty} \rho_t} \tag{4.5}$$

where N is the total number of samples used by MCMC and ρ_t is the correlation between the accepted samples at lag t. In the extreme case that all the samples are independent, ESS matches N. If the correlation at lag t decreases very slowly, ESS is zero and the produced samples do not estimate the posterior distribution satisfactorily. A commonly accepted criterion for the effectiveness of MCMC is when ESS is approximately greater than 40% of the total number of samples, i.e. when $ESS > 0.4N$ [40].

Different MCMC algorithms can be distinguished by the sampling methods used in simulating the chains. The simpler algorithm is the Metropolis-Hastings [41], which produces a chain of states called candidates from a chosen distribution, such as a normal distribution. In this case, the mean is equal to the current state of the chain and the variance is a measure of the noise that exists in that state. The candidate state is accepted and becomes the next stage of the chain with probability equal to the ratio of the likelihood of this candidate state and the likelihood of the current state. This implies that if the likelihood of the candidate state is higher, then it is always accepted otherwise, there is a chance of retaining the previous state.

Metropolis-Hastings algorithm converges slowly in general, mainly due to the tendency to explore the parameters' space by choosing a state randomly. When parameters are continuous rather than discrete, Hamiltonian Monte Carlo (HMC) cleverly converts the problem of sampling from a target distribution into a problem of simulating Hamiltonian dynamics. The mathematical formulation and the algorithm of HMC are given in [42, 43]. In HMC, the parameters that need to be estimated are vectorized as a particle in N-dimensional space corresponding to the number of parameters. This algorithm introduces an additional hyperparameter - the number of steps for running a simulated Hamiltonian system. This parameter dictates how long the algorithm follows a path before initiating a new random change in particle direction. If this step count is exceedingly low or high, the resulting samples tend to resemble each other too closely due to the paths making a complete U-turn (turnaround). Consequently, choosing an inappropriate value for this hyperparameter significantly impacts the efficiency of the HMC algorithm, rendering it less practical for real-world applications.

In response to these challenges, NUTS was developed. NUTS extends the capabilities of HMC by eliminating the need to specify a fixed number of steps for its simulated chains. The fundamental idea behind NUTS lies in identifying when the constructed path initiates a turnaround. This identification is accomplished by simulating the path in both forward and backward directions, and upon detecting a U-turn, NUTS ceases the simulation, stores a sample, and initiates a new simulation by randomly changing the particle's direction. This innovative approach enhances the algorithm's efficiency and mitigates issues associated with selecting an inappropriate step count in HMC simulations. Interested readers may refer to [38] for a detailed description of the algorithm and its mathematical formulation.

Employing MCMC with the NUTS algorithm enables the approximation of any posterior distribution without necessitating restrictions to conjugate priors. This freedom allows experts to select any combination of prior and likelihood distributions based on their domain-specific knowledge. Notably, the NUTS algorithm operates effectively without the need for fine-tuning its hyperparameters, rendering it highly suitable for practical problems. As a Bayesian inference-based approach, this algorithm demonstrates versatility by accommodating even a single observation, making it exceptionally convenient for modeling scenarios involving repairs constrained by limited datasets. Consequently, its flexibility and effectiveness make it a valuable tool for a wide range of practical applications, including ours.

4.2.2. PROGNOSTICS

The imperfect repair model requires RUL predictions to train with, thus a prognostic model is mandatory. In real-case scenarios, raw sensory data is the only information available. These data are usually time-series and multiple preprocessing steps are important before predicting RUL. These steps are quite different for each application, making each prognostic model task-specific which contradicts the generalizability of our proposed framework. In this regard, the chosen prognostic model is based on the recent study presented in chapter 3 that integrates the DSMC and HSMM models.

4.3. METHODOLODY

The state-of-the-art models considered in this work for feature extraction and RUL prediction consider the DSMC model and HSMM. Given the limited training dataset size (no more than 3 degradation histories after the repair) in this study, this stochastic model necessitates certain assumptions:

1. Imperfect repairs recover the component's condition somewhere between the ABAO and AGAN conditions by following a truncated normal distribution in the range [ABAO, AGAN] [15, 16, 44]. Hence, the RUL after a repair cannot be less than the ABAO condition and not higher than the AGAN condition.

2. The component must be repaired before reaching its predefined EOL.

3. A semi-dependency exists between the trajectories of an examined component both before and after the repair process. This dependency hiddenly exists in the monitoring data acquired before and after a repair. Since the recovery cannot be measured beforehand, the time index should not reset. Instead, time will always increase and the data should provide the information regarding the recovery.

4. Uncertainties associated with the prognostic and repair models are independent since both models operate autonomously. Uncertainties related to the manufacturing process and material quality are not considered. Thus, the proposed models consider two types of uncertainty: epistemic (uncertainty due to the model's parameters) and aleatoric (uncertainty due to noisy data).

5. For small datasets, there is a closed-form solution between RULs at ABAO and AGAN conditions. This is expressed via a random variable defined as recovery.

6. Multiple repairs result in a reduction of the maximum possible recovery of the component, i.e. the value of the AGAN condition decreases, while the ABAO condition still depends only on the corresponding RUL before repairing (see Figure 4.5b).

It should be highlighted that a larger training dataset size could assist in relaxing assumptions 1 and 4.

The entire process from acquiring raw sensory data, to predicting RUL and using it to construct the imperfect repair models is illustrated in Figure 4.3.The ultimate target is to model the stochastic level of repair in order to predict RUL post-repair before even maintenance occurs. First, the DSMC model as well as the HSMM model are trained with components that reach the EOL. Second, these models are applied during inference to components that did not reach the EOL, because a repair process has been performed. Given a sample of raw time-series data, the DSMC model assigns a cluster. This cluster is fed as input to HSMM to predict RUL under uncertainty. The last sample before a repair and the first post-repair form two RUL data points alongside confidence intervals (CI). The distance between those data points represents the percentage of recovery (see Figure 4.4). By providing multiple trajectories, several recovery values are estimated that are utilized to perform Bayesian inference and estimate the posterior predictive distribution of recovery. Having this distribution, the stochastic level of repair can be measured. In practice, when a RUL data point is predicted just before the repair, this distribution could be used to estimate the RUL post-repair before even maintenance is performed.

Up until this point, the procedure could be replaced by any technique capable of predicting RUL under uncertainty, which is attributed to our proposed imperfect repair model's versatility and generalizability. Upon obtaining the RUL predictions for each timestep of the examined structure, the model generates the necessary RUL data points immediately preceding a repair action (x_{old}). The objective here is to train the prognostic model on some baseline experiments that reached the EOL, hence ranging from the AGAN condition to failure. This training allows for estimating the RUL just before the repair. Subsequently, by evaluating the prognostic model to trajectories after a repair (these are not considered during training of the DSMC and HSMM models), RUL is predicted spanning from the time of the repair until failure (X_{new}). Thus, it becomes feasible to acquire the necessary samples just before and right after the repair. These samples are crucial for estimating the distribution that explains the dynamics of the recovery during repair.

In the following subsections, each of the steps of the process is discussed in detail.

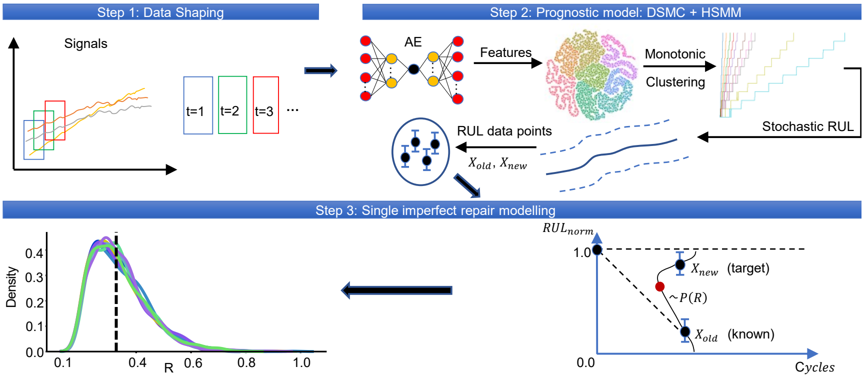


Figure 4.3.: Illustration of the entire process, from shaping the raw data to predicting RUL and using it to construct the imperfect repair model. Steps 1 and 2 could be replaced with any approach that predicts RUL under uncertainty.

4.3.1. DATA SHAPING

In the first step of the process, the raw time-series signals are acquired. These signals are normalized feature-wise to the range [0,1] with min-max normalization according to the training trajectories. Then, the same statistical values were applied to the testing ones to avoid data leakage. For each normalized trajectory, the time-series data are grouped into overlapping windows to create samples. It is noteworthy that no other preprocessing step is required.

4.3.2. FEATURE EXTRACTION AND PROGNOSTIC MODELS

The repair model requires RUL predictions to be trained with. Thus, a prognostic model is mandatory. In real-case scenarios, raw sensory data is the only information available. These data are usually time series and multiple preprocessing steps are important before predicting RUL. These steps are quite different for each application, making each prognostic model task-specific, which contradicts the generalizability of the proposed framework. In this regard, the chosen prognostic framework is based on the DSMC model [33]. The unique component of the DSMC model does not require any preprocessing step and can be applied directly to raw sequential data. The DSMC model is then paired with an HSMM for predicting the RUL under uncertainty as presented in [29], which accepts the trajectories of clusters of the DSMC model as input and calculates the mean value of the RUL and the 95% CI. These models have been extensively discussed in chapter 3.

4.3.3. THE REPAIR MODEL

Before constructing the repair model, a preprocessing step is necessary; rescaling the RUL so that its maximum value is one and the minimum is zero. This transformation allows for easy conversion into a percentage of recovery, which is the ultimate distribution to be modeled. However, the maximum RUL value is unknown before reaching the EOL. In this regard, the maximum value to be used for rescaling RUL is taken as the maximum value of the mean RUL prediction of each trajectory as estimated by the prognostic model (see Figure 4.4). Obviously, due to the induced uncertainties, there is a chance that the normalized RUL could become slightly higher than 1.0. In such a case, this value is manually converted to 1.0 as the upper limit to be compatible with assumption 1. Nevertheless, this modification can be translated as a risk-averse policy since the RUL predictions could indicate that the EOL will be reached faster than the real occurrence.

Subsequently, the modelling of the repairs' distributions will take place. Let X_{old} and X_{new} represent random variables denoting the normalized RUL before and after a repair, respectively. In other words, X_{old} denotes the ABAO condition, and X_{new} is a condition somewhere between the two extremes. The straightforward approach would be to model the stochastic relationship between X_{old} and X_{new} within a Bayesian regression task. However, due to the limited dataset size, there is a high risk of overfitting making it difficult for any machine learning technique to handle this formulation effectively. An alternative approach involves introducing an auxiliary random variable representing the percentage of improvement from X_{old} to X_{new}. This establishes a closed-form solution between the two random variables X_{old} and X_{new}, which is compatible with assumption 5. Subsequently, after applying Bayesian inference, sampling from this auxiliary random variable can help model the desired distribution of X_{new}, considering one or more new samples of $x_{old} \in X_{old}$. This auxiliary variable is defined as the recovery R, and its deterministic relation with X_{old} and X_{new} is expressed below:

$$R = \frac{X_{new} - X_{old}}{1 - X_{old}}, \quad R \in [0, 1] \tag{4.6}$$

The recovery R is a random variable with a probability distribution $P(R)$ that describes the percentage of improvement from X_{old} to X_{new} and could be applied to any imperfect repair model subject to stochastic RUL. In practical applications, the value of X_{old} is known and X_{new} is the target variable which should be predicted. This can be calculated using Equation 4.6 after estimating the distribution of R:

$$X_{new} = (1 - X_{old})R + X_{old} \tag{4.7}$$

In this analysis, X_{old} and X_{new} are random variables that may follow any distribution according to the procedure of modelling RUL. Figure 4.4 depicts the random variables and the distribution of recovery that needs to be addressed given a dataset D. An important remark here is that the left tail in the distribution of R should always be fixed to zero (no recovery) to be compatible with assumption 1 which supports that the RUL should always be equal or higher than the ABAO condition. The red dot represents a sample R. Then X_{new} is calculated by Equation 4.7.

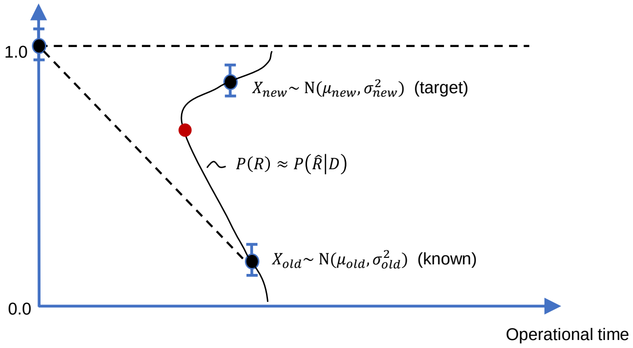


Figure 4.4.: Graphical representation of the random variables and the distribution of R that needs to be estimated. The red dot is a sample from R that recovers the component back to X_{new} after applying Equation 4.7. The minimum value of R is always fixed at zero (no recovery) indicating that after the repair the component's condition is always greater than its ABAO condition.

Data points $x_{old} \in X_{old}$ and $x_{new} \in X_{new}$ are estimated by the prognostic model representing RUL predictions for trajectories before and after a repair, respectively. From these data points, individual samples of R can be computed using Equation 4.6 to generate the desired dataset. Although the distributions of the random variables X_{old} and X_{new} are known, it is very challenging to express R analytically [45]. An existing solution is to approximate R analytically by assuming a normal distribution with a mean $\mu = \mu_{new}/\mu_{old}$ and variance $\sigma^2 = (\mu_{new}/\mu_{old})^2[(\sigma_{new}/\mu_{new})^2 + (\sigma_{old}/\mu_{old})^2]$, if $\delta_{new} = \sigma_{new}/\mu_{new}$ is quite small ($\delta_{new} < 0.1$ is usually the condition for having a satisfying approximation [45, 46]). Since μ_{new} can have values near 1.0 after a repair, having the above approximation requires $\sigma_{new} < 0.1$ which is not satisfied from the given stochastic RUL predictions as the uncertainty can easily surpass this limit.

Despite the difficulty in expressing R analytically, it can be assumed to follow a

Truncated Normal distribution for the likelihood in Bayesian inference, as mentioned in previous studies [14, 16, 44] and as mentioned in assumption 1. However, the specific parameters defining the distribution are still unknown and can be estimated differently by assuming varying prior distributions. This flexibility is left to the discretion of the expert, considering their domain expertise and knowledge. Notably, the use of conjugate priors could limit the model's capabilities and should be avoided, thus, Bayesian inference can only be approximated through sampling methods.

In this regard, the MCMC with NUTS algorithm has been selected to estimate the distribution of R from data. Employing MCMC with the NUTS algorithm enables the approximation of any posterior distribution without necessitating restrictions to conjugate priors. This freedom allows experts to select any combination of prior and likelihood distributions based on their domain-specific knowledge. Notably, the NUTS algorithm operates effectively by limiting the hyperparameters that require fine-tuning, rendering it highly suitable for practical problems. As a Bayesian inference-based approach, this algorithm demonstrates versatility by accommodating even a single observation, making it exceptionally convenient for modelling scenarios involving repairs constrained by limited datasets. Consequently, its flexibility and effectiveness make it a valuable tool for a wide range of practical applications, including ours.

The primary objective is to develop a model that effectively encapsulates the stochastic progression from the ABAO condition to an improved state, influenced by the uncertain dynamics of subsequent imperfect repairs. The parameters of the repair model need to undergo training within the concepts of Bayesian inference to effectively capture the underlying stochasticity. The target is to estimate the posterior predictive distribution that represents the random variable R, i.e. the distribution $P(\hat{R}|D)$. To achieve this, the likelihood distribution of the data $P(D|\theta)$ needs to be defined, as well as the prior distribution of the random variables $P(\theta)$. A general approach is to estimate the joint posterior predictive distribution utilizing two independent distributions for expressing $\hat{R}$; one for its mean and one for its variance, $P(\hat{R}_{mean}|D)$ and $P(\hat{R}_{var}|D)$, respectively.

The Bayesian modelling consists of the components presented in Table 4.1. According to this table, we assume that the likelihood $P\left(D\middle|\hat{R}\right)$ consists of two random variables that follow a bivariate truncated normal distribution with unknown mean μ_{mean} and variance σ^2_{mean} regarding R_{mean}, while μ_{var}, σ^2_{var} correspond to the unknown mean and variance of the second random variable R_{var}. The ranges (a_{mean},b_{mean}) and $(a_{var},\ b_{var})$ stand for these truncated normal distributions. Since R_{mean} and R_{var} are independent random variables $(R_{mean} \perp\!\!\!\perp R_{var})$, there is no correlation, thus the covariance of the aforementioned truncated bivariate normal distribution yields in

$$\Sigma=\begin{bmatrix} \sigma^2_{mean} & 0 \\ 0 & \sigma^2_{var} \end{bmatrix}.$$

This method allows for the approximation of any distribution R. However, practically implementing this model necessitates extensive knowledge about the range of prior distributions, essentially requiring an understanding of the data. Typically, in scenarios

Table 4.1.: The necessary components for the Bayesian modeling of a repair model. These are the basic components upon which our proposed repair models will be developed.

Components	Details
Idea	Decompose $P(\hat{R}\|D)$ into two independent distributions $P(\hat{R}_{mean}\|D)$, $P(\hat{R}_{var}\|D)$ s.t. $P(\hat{R}\|D){=}P(\hat{R}_{mean}\|D)P(\hat{R}_{var}\|D)$
Likelihood	$P(D\|\mu_{mean},\sigma_{mean},\mu_{var},\sigma_{var}){\sim}TruncBNorm$ $(\mu_{mean},\sigma^2_{mean},\mu_{var},\sigma^2_{var},a_{mean},b_{mean},a_{var},b_{var})$
Priors	$\theta{=}\{\mu_{mean},\sigma_{mean},\mu_{var},\sigma_{var}\}$, $\mu_{mean}{\sim}U(a_1,\ b_1)$, $\sigma_{mean}{\sim}\ U(a_2,\ b_2)$, $\mu_{var}{\sim}U(a_3,\ b_3)$
Posterior Predictive	One distribution for the mean and one for the variance of recovery: $P(\hat{R}_{mean}\|D)$ and $P(\hat{R}_{var}\|D)$, respectively
Process	Use the two independent distributions to estimate their joint distribution representing $\hat{R}$, then estimate $\hat{X}_{new}$ from Equation 4.7

with limited datasets, such comprehensive information about the data (informative priors) is unavailable. Therefore, the initialization of prior distributions often encompasses a broad range of values to consider potential variations in forthcoming samples. This broader initialization leads to larger variances and subsequently wider posterior predictive distributions. Sampling from such a wide distribution becomes impractical, especially when applying this model to a decision-making system that guides maintenance actions based on their effects on the component's recovery.

The approach employed in this work will solely focus on the posterior distribution of the mean of the recovery $P(\hat{R}_{mean}|D)$, whereas $P(\hat{R}_{var}|D)$ becomes calculable beforehand within this method. Particularly, since the prognostic model is independent of the repair process as well as their corresponding uncertainties (assumption 4), without loss of generality, by knowing the mean of the RUL after a repair, one can compute its corresponding variance by identifying the equivalent pair before the repair and matching their values individually. In essence, when the prognostic model operates independently of the repair model, the variance of RUL is influenced solely by external conditions and not inherent behaviors induced by the repairs. This is useful as the variance after repair will be known. Figure 4.5a illustrates this statement. Hence, only one distribution is needed to estimate R which is R_{mean}, thus $P(R|D){=}P(R_{mean}|D)$, where R_{mean} can be defined similarly to Equation 4.6 and Equation 4.7, respectively as follows:

$$R_{mean} = \frac{\mu_{new} - \mu_{old}}{1 - \mu_{old}} \tag{4.8}$$

$$\mu_{new} = (1 - \mu_{old})R_{mean} + \mu_{old} \tag{4.9}$$

Similarly, μ_{new}, μ_{old} are also random variables, thus an analytical solution for R_{mean} does not exist and it should be estimated via MCMC by defining a likelihood and some prior distributions. Then, after estimating the posterior predictive distribution of $\hat{R}_{mean}$, such that $P(\hat{R}|D)=P(\hat{R}_{mean}|D)$, a scalar from $\hat{\mu}_{new}$ can be calculated given a sample $x_{old} \in X_{old}$. With this scalar value and the stochastic RUL values stored prior to repair, it becomes straightforward to determine which specific stochastic RUL value corresponds to each scenario. After the identification, the variance σ^2 of the corresponding RUL can be received and used to estimate $\hat{X}_{new} \sim N(\hat{\mu}_{new}, \sigma^2_{new}=\sigma^2)$. It is noteworthy that the final RUL predictions after a repair should have a total variance that comes from the sum of the variances from the posterior predictive distribution of μ_{new} and the estimated by the prognostic model variance σ^2_{new}. This is the total uncertainty of the RUL after a repair that a decision-making system should consider.

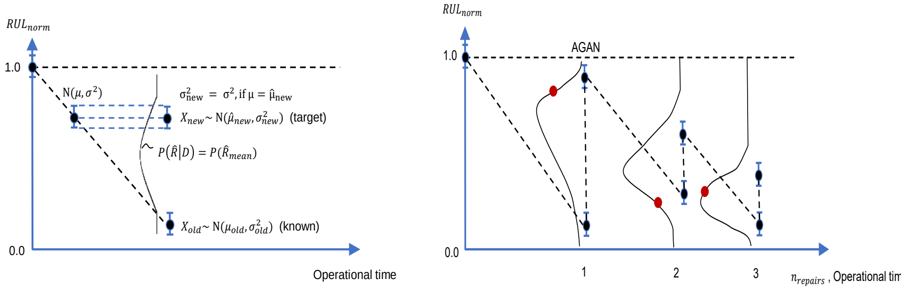
In this approach, the Bayesian modelling consists of the elements stored in Table 4.2. These hyperparameters were chosen in order to avoid an intersection between μ_{new} and μ_{old}, specifically to ensure the condition $\mu_{new} > \mu_{old}$ which is based on assumption 1. The standard deviation was assigned a relatively broad range of potential values owing to the absence of precise knowledge regarding the appropriate width of the posterior distributions. Ultimately, samples drawn from the likelihood distribution were constrained to have values no greater than 1.0 and no less than 0.4. Simultaneously, the prior distribution of μ_{mean} is a Uniform distribution with the lowest value of 0.6, whilst the highest value is set to 0.9. The selection of this range is based on a prior investigation that documented a recovery rate, subject to a comparable repair process, approaching more than 80% on average over three components [47]. Additionally, this range allows a more flexible variance capable of shifting the distribution closer to 1.0, if necessary. Finally, the rejection sampling technique [48, 49] was applied to guarantee that the posterior predictive distribution will have only non-negative values at the required range.

Table 4.2.: The necessary components for the Bayesian modeling of our proposed repair model.

Components	Details
Idea	$P(\hat{R}|D)=P(\hat{R}_{mean}|D)$, RUL model is independent of the repair process and variance is now known.
Likelihood	$P(D|\mu_{mean},\sigma_{mean}) \sim TruncNorm\ (\mu_{mean},\sigma^2_{mean},a_{mean},b_{mean})$
Priors	$\theta=\{\mu_{mean},\sigma_{mean}\}$, $\mu_{mean} \sim U(a_1,\ b_1)$, $\sigma_{mean} \sim\ U(a_2,\ b_2)$
Posterior Predictive	One distribution for the mean of recovery: $P(\hat{R}_{mean}|D)$
Process	Bayesian inference to estimate the posterior predictive $P(\hat{R}|D)=P(\hat{R}_{mean}|D)$, calculate $\hat{\mu}_{new}$ from Equation 4.9, find the corresponding (μ,σ^2) before repairing such that $(\mu,\sigma^2)=(\hat{\mu}_{new},\sigma^2_{new})$, estimate $\hat{X}_{new}$
Hyperparameters	$a_{mean}=0.4,\ b_{mean}=1.0,\ a_1=0.4,\ b_1=0.9,\ a_2=0.01,\ b_2=0.2$

The same process can be followed for multiple sequential repairs utilizing Equation 4.8

and Equation 4.9, where μ_{old} and μ_{new} are the normalized mean RULs exactly before and after the n-th imperfect repair action. Figure 4.5b illustrates the behavior of the distribution of R_{mean} for three sequential imperfect repairs. Notice how the distribution shifts towards zero with the increase of number of repairs. This is attributable to the assumption that each repair results in a reduced percentage of recovery (assumption 6).



(a) (b)

Figure 4.5.: Recovery distribution under the assumption that the RUL model is independent of the repair process (a). Recovery distribution subject to multiple repairs (b).

4.4. CASE STUDY

In to test the capabilities of the methodology, an experimental campaign is launched. Aluminium 7075-T6 is an aerospace-grade material that is commonly used in many structural aircraft elements like the wings and the fuselage. However, as with most metallic materials, its integrity is compromised by the initiation of cracks. These cracks develop either due to fatigue loading at critical points - like rivet holes - or due to some form of external damage-like impact. The motivation behind the current study is to estimate, using the proposed model, the effects of a repair action on the lifetime of a structure, given the periodic growth of an artificially induced crack simulating common fatigue damage that may occur during operation. The tests involve tension-tension fatigue experiments on open-hole aluminium specimens. Eleven specimens are cut from a $2mm$ thick 7075-T6 aluminium sheet parallel to the rolling direction. The length of the specimens is $300mm$ and the width is $45mm$. A $6mm$ hole is drilled at the middle of the specimens, and 0.5-$1mm$ notches are introduced at the sides of the hole to induce controlled crack growth perpendicular to the loading direction. The specimens are tested under tension-tension fatigue load, with a maximum stress of 100MPa, a fatigue ratio of 0.1, and a frequency of $5Hz$.

The purpose of the tests is to induce fatigue crack growth, repair the damage, restore some of the parent material's load-bearing capabilities, and investigate the extension of the fatigue life. The suggested repair methodology involves bonding a rectangular CFRP patch via secondary bonding using an Araldite 2015-1 two-part epoxy adhesive. This repair takes place at a predefined percentage of the average fatigue life of the tested specimens. To improve adhesion and remove air gaps between the bonded parts, the repaired specimens are cured at room temperature under a vacuum bagging process for 24 hours. This is kept similar for all the repaired specimens to remove any uncertainties associated with the repair procedure. The purpose of the repair is to halt the crack propagation, thus increasing the load-bearing capabilities of the parent material and allowing it to run for longer periods of time.

First, five specimens are tested until failure to get information regarding the fatigue life prior to repair and determine the constant lifetime percentage at which the repair is performed. The remaining six specimens are tested until 14000 fatigue cycles corresponding to 60% of the average fatigue life, when the developed crack was repaired with the aforementioned procedure. From those, half are tested until failure while the other half are stopped at 11000 cycles, approximately 60% of the averaged repaired specimen lifetime, to showcase that there is no need to reach failure to develop the repair model. More detailed information about the specimens is reported in Table 4.3. Since the trajectories of a specimen with and without a repair are separated, for the remainder of this work, the trajectories' names will be defined with two numbers; the first corresponds to the number of repairs being performed and the second to the corresponding specimen. For instance, the specimen no. '01' that has no repair is named specimen '0_1'. For example, specimen number '11' is split into two names, i.e. specimen '0_11' (trajectory with no repair) and '1_11' (trajectory with repair), and so on.

Table 4.3.: Test matrix. The specimens that were under repair contain two trajectories corresponding to one before and one after the repair.

Specimen name(s)	Repair time (cycles) (T1)/ observed crack size (mm)	Fatigue life after repair (cycles)	Fatigue life Total (cycles)
Baseline			
0_1	-	-	26478
0_2	-	-	22563
0_3	-	-	23342
0_4	-	-	23750
0_5	-	-	19250
Average	-	-	23076
1 Repair, reached EOL			
0_6, 1_6	14000/ 1	24565	38565
0_7, 1_7	14000/ 5	17445	31445
0_8, 1_8	14000/ 6	17250	31250
Average	14000/ 4	19753	33753
1 Repair, did not reach EOL, stopped after 11000			
0_9, 1_9	14000/ 1	-	-
0_10, 1_10	14000/ 4	-	-
0_11, 1_11	14000/ 0	-	-

Figure 4.6 depicts the experimental setup. To validate the methodology, an important aspect is the SHM data collected during the test. For that reason an acoustic emission

system, comprising of an AMSY-6 Vallen Systeme GmbH and two VS900-M wideband sensors - that was recording constantly during the experiment- was employed to monitor the specimens. This SHM technique was chosen due to its ability to capture and monitor damage evolution, both crack growth and the potential adhesive or cohesive damage of the bonded patch. The goal is to leverage this ability and extract features that can assist the model in finding and extracting the underlying degradation trend both before and after repair. From the acoustic emission low-level features that were recorded, the ones summarized in Table 4.4 were considered. To split the train and test data we formatted the dataset as follows:

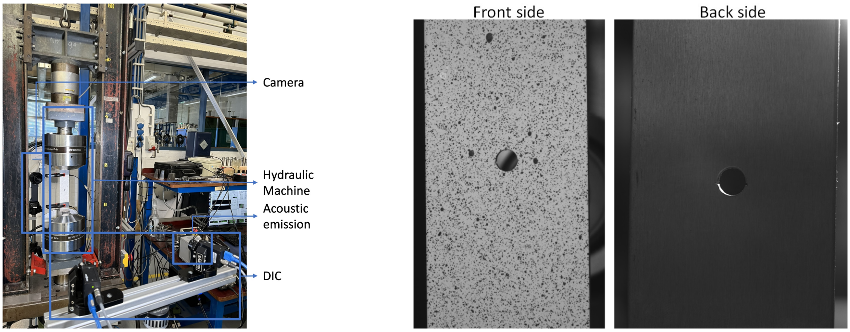
- To train the DSMC and the prognostic models only specimens ‘0_1’ - ‘0_5’ representing trajectories that reached the EOL without any repair were considered.
- From the above specimens, each time one was considered as a validation specimen of the other four, thus training the model 5 times (leave-one-out cross-validation). The final trained model considers all 5 specimens, meaning that the cross-validation technique was utilized only to show the models’ prediction capabilities.
- RUL estimations were made with the trained models to the remaining unseen specimens (specimens 6-11).
- From the above RUL estimations, trajectories ‘0_9’, ‘1_9’, ‘0_10’, ‘1_10’, ‘0_11’ and ‘1_11’ were considered as the training set for the repair model, whilst the rest of the specimens as the testing set. This decision was made, firstly, to show that the repair model can be trained to trajectories that did not reach the EOL and, secondly, to incorporate at the testing phase only the specimens that reached the EOL after the repair, thus having a ground truth for evaluating our results.

Table 4.4.: The low-level features that are considered and extracted by the AMSY-6 Vallen Systeme GmbH.

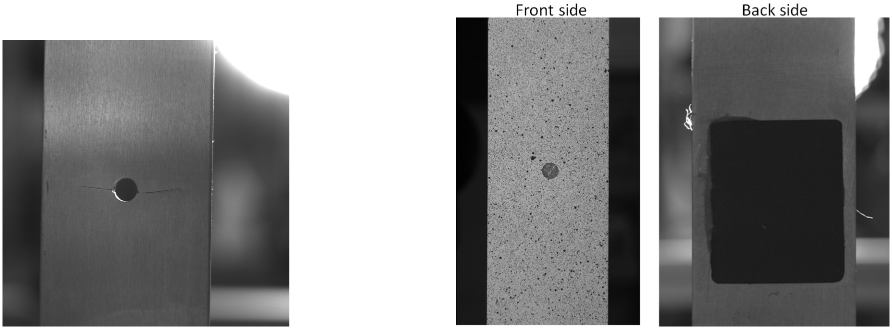
Feature name	Unit	Description
Threshold	Decibel $[dB]$	Values below this threshold are discarded.
Amplitude	Volts $[V]$	The amplitude of the corresponding signal.
Duration	Seconds $[s]$	The duration that a signal constantly remains above the threshold.
Energy	$10^{-14}V^2s$ $[eu]$.	Energy is the integral of the squared acoustic emission-signal over time
Counts	-	The number of positive threshold crossings of a hit.
Hit time	Seconds $[s]$	The absolute time when a hit is above the threshold.
Rise time	Seconds $[s]$	The time between the first threshold crossing and the maximum amplitude.
RMS	-	Root mean square (RMS) error.
Signal strength	$10^{-9}Vs[nVs]$	The integral of the rectified AE signal over time.

A DIC system was also present during the test to monitor the strain field around the hole and crack, however the collected data are not considered in the present research.

Since there was no practical experiments in the literature related to multiple imperfect repairs, additional trajectories have been generated based on the existing ones according



(a) Experimental setup (b) Baseline specimen



(c) Damaged specimen (d) Repaired specimen

Figure 4.6.: Experimental setup (a) and specimen states (baseline (b), damaged (c), repaired (d))

to the following two steps:

1. The first RUL data point after a repair corresponding to μ_{new} and, consequently, the mean of the distribution of recovery $\mu(R_{mean})$ is exponentially reduced with the number of repairs $n_{repairs}$ according to the following equation:

$$\mu\left(R_{mean}|n_{repairs}\right) = \kappa \cdot exp(-\lambda n_{repairs}),\ \ n_{repairs} \in 0,\ 1,\ 2,\ \ldots \tag{4.10}$$

   Where λ,κ are the parameters that need to be determined, with λ being the rate parameter, and κ a constant. By setting $n_{repairs}$ to zero, we have:

$$\kappa = \mu(\hat{R}_{mean}) \tag{4.11}$$

   The above equation can be used after training the imperfect repair model for one repair. The variable λ can be any float number in the range of $[0,1]$. To increase the complexity of the task, a sharp exponential decrease has been chosen with λ=0.7. Consequently, after the second repair, it is expected the distribution of recovery to be close to zero, thus the next repairs will not offer any significant recovery. This will increase the difficult of the PPDM task as will be shown in the next chapter.

2. Starting with the first RUL data point after the n-th repair, the corresponding RUL data point from the $(n-1)$-th repair with the same value is identified. The entire trajectory is then generated by using the mean and variance of the trajectory from the previous repair. Additional noise, following a Uniform distribution $U(-3000,3000)$ measured in cycles, is added to this generated trajectory.

Based on the above, three more trajectories have been generated representing the second, third, and fourth sequential repairs, respectively. These repairs concern one synthetic specimen. If more specimens were available regarding multiple repairs, our proposed repair model could be applied as is without generating data according to the aforementioned steps. However, with only one specimen available, a deterministic approach will be followed based on Equation 4.11 and Equation 4.10, which is discussed section 4.5.3.

4.4.1. VALUES OF HYPERPARAMETERS HIGHLY DEPENDENT ON THE DATASET

The input time-series data of each trajectory are grouped into overlapping windows. A high overlap percentage of 90% has been chosen because acoustic emission signals may contain important information at specific parts of the trajectory, which otherwise could not been observed accurately. In order to assist the LSTM layers in capturing the crucial information that may be observed at each window, a window length of 300 data points has been decided. This length is acceptable for LSTM to extract important temporal dependencies of the data while keeping memory capacity at low levels.

The tuned hyperparameters from the Bayesian Optimization algorithm that were used to train the DSMC are presented in Table 4.5. Finally, Table 4.6 summarizes the key hyperparameters that need to be tuned for applying the algorithm. It is worth mentioning that due to the limited number of samples, employing multiple Markov chains is inefficient. Therefore, the algorithm utilizes only one chain for estimation. The other hyperparameters of the table are the default values of such a dataset size.

Table 4.5.: Hyperparameter search ranges and final values optimized by the Bayesian Optimization algorithm. Both the automatically and manually tuned hyperparameters are included.

Bayesian Optimization	Hyperparameter	Search range	Optimized value
Yes	Z	[3, 32]	32
	H_{in}^{enc}	[32, 128]	128
	lr^{AE}	[10^{-4}, $5*10^{-3}$]	$8*10^{-4}$
	lr^{DSMC}	[$5*10^{-5}$, 10^{-3}]	$5*10^{-4}$
	$epochs^{AE}$	[50, 400]	299
	$epochs^{DSMC}$	[30, 60]	51
	α	[0.7, 2.0]	1.8
	β	[0.01, 5.0]	2.5
	dr_{rate}	[0.1, 0.3]	0.15
No	L_{window}	-	300
	S	-	30
	$batch^{AE}$	-	64
	$batch^{DSMC}$	-	64

Table 4.6.: Tuned hyperparameters for the initialization of the MCMC with the NUTS algorithm.

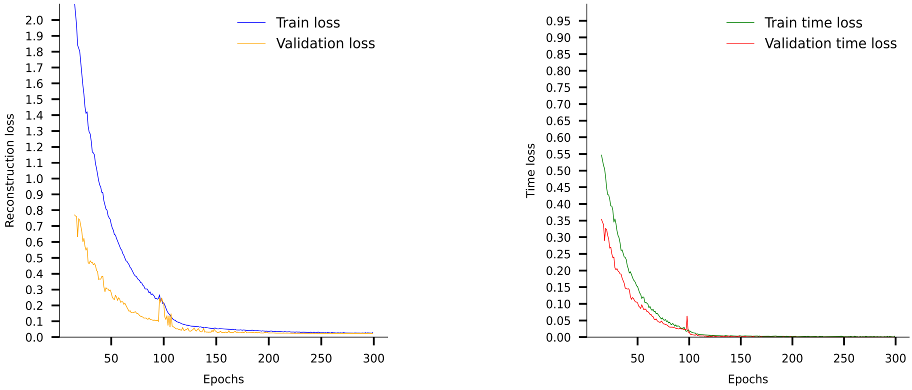
	Acceptance rate	Warmup samples	Iterations	No. chains
Values	0.8	200	1000	1

4.5. OUTCOMES OF IMPERFECT REPAIR MODEL

In this section, the results of the clustering, prognostic, and repair models related to the examined case study are presented and discussed.

4.5.1. CLUSTERING ASSIGNMENTS & PROGNOSTICS

Prior to training the repair model for the introduced case study, it is mandatory to train the DSMC model to produce the cluster assignments to be further utilized by a prognostic model. The DSMC model was trained on a single GPU (NVIDIA GeForce RTX 2080). The entire training process alongside the hyperparameter tuning via the Bayesian Optimization algorithm takes approximately 45 minutes. The training and validation losses concerning the reconstruction loss and the time domain loss, are shown in Figure 4.7a and Figure 4.7b, respectively. These are the two losses based on which the DSMC model's autoencoder was trained. Following the entire procedure until RUL prediction, Figure 4.8 shows the leave-one-out cross-validation results for the training set, i.e. each of the five specimens that were kept as a validation sample for each of the five training loops. It is clear that for the specimens '0_4' and '0_5' representing the left and right outliers of the dataset, respectively, the prognostic model struggles to capture the true RUL. Although an adaptive prognostic model [50] could be a potential solution, it was deliberately chosen not to use one. This decision was made to demonstrate that the proposed approach can be applied to any prognostic model, regardless of its adaptive capabilities. Nevertheless, the predicted uncertainty includes those true values showing a satisfying performance of the model based on which the samples for training and testing the repair model are produced.



(a) (b)

Figure 4.7.: The convergence of train and validation losses, including the reconstruction loss of the input (a) and reconstruction loss of the time feature (b) after the first stage of training of the DSMC model (AE training)

.

Upon evaluating the prognostic model's performance, the DSMC and HSMM models are re-trained from scratch with all five specimens. The evaluation of the DSMC model on the test data can be seen via the visualization of the clustering assignments. The assignments from the testing trajectories are illustrated in Figure 4.9. The specimens that did not reach the EOL have their last cluster assignment with a value less than the total number of clusters. As expected, the soft monotonicity observed in the cluster assignments reflects a realistic sensor measurement's noisy behaviour. The first cluster assignments for specimens after their repair are naturally predicted by a value larger than the first cluster. This roughly indicates the partial recovery of each specimen, which is better seen via the prognostic model. By examining the results, an unexpected behaviour can also be observed: all of the specimens' cluster assignments follow similar trends. Granted that the specimens fail at different times, we would expect that the damage accumulation (and thus the cluster assignment) would not follow the same path. We consider this behaviour to be attributed to the limited information that the raw acoustic signals provide at the beginning of each experiment, where no cracks are formed and thus no acoustic signals are emitted, and not to an issue of the DSMC model. A potential remedy to this issue involves integrating an additional SHM technique into the experimental protocol or incorporating crack growth measurements, thereby augmenting

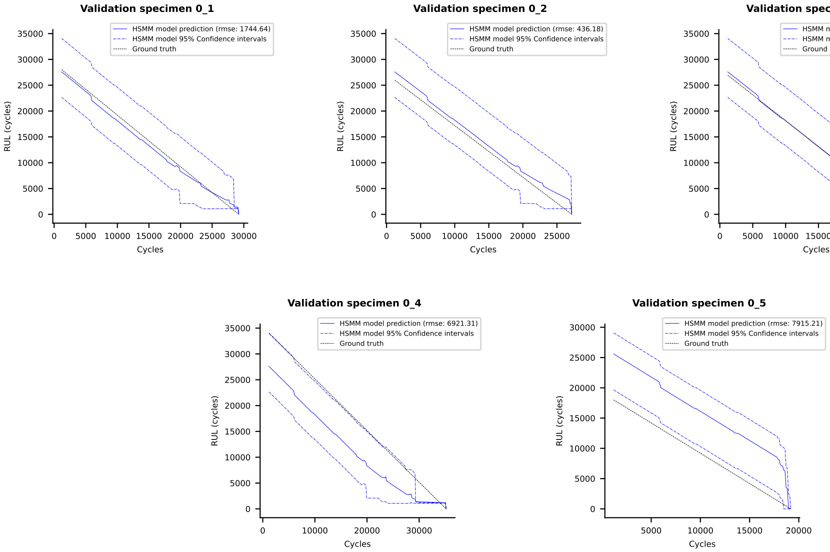


Figure 4.8.: Leave-one-out cross-validation for specimens ‘0_1’ - ‘0_5’ used for training the DSMC and HSMM models.

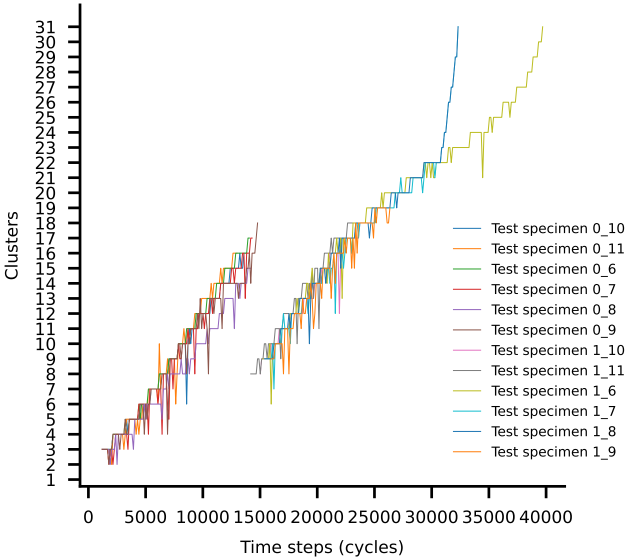


Figure 4.9.: The predicted by the DSMC model clustering assignments concerning the test specimens.

input data for the DSMC and HSMM models. Such augmentation holds promise for introducing variability in clustering assignments and stochastic RUL predictions, consequently enhancing the diversity of the recovery observations.

Passing all these trajectories through the trained prognostic model, one can receive the RUL predictions with uncertainty. Figure 4.10 visualizes the stochastic RUL for each specimen for both pre- and post-repair. The true RUL can only be measured in cases where the EOL is reached. Consequently, it is impossible to determine true RULs for testing specimens before and after repair that did not reach the EOL. Specimens that reached the EOL after repair were kept for evaluation to ensure a reliable ground truth for assessing the repair model. Another reason was to prove that the repair model could be trained even when no specimens reached failure, highlighting the reliability of this methodology.

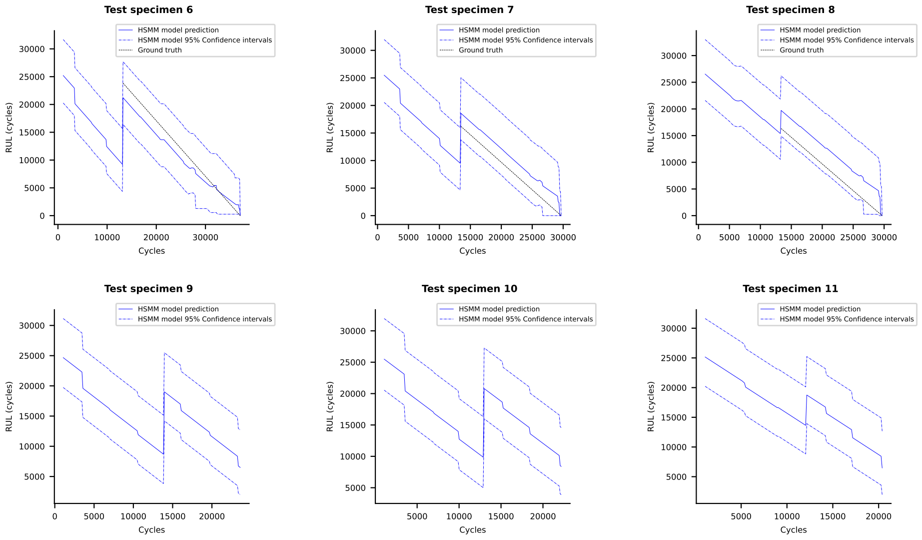
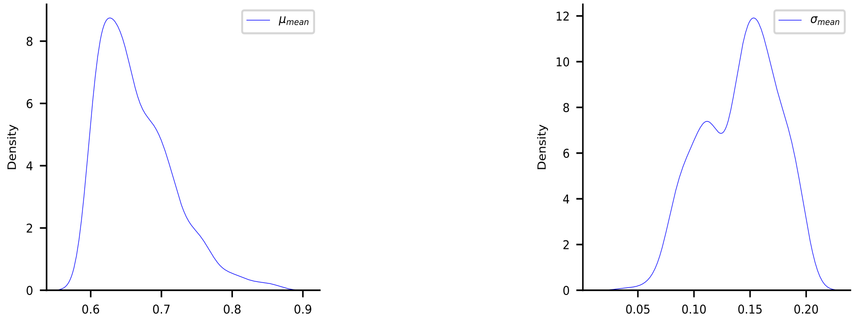


Figure 4.10.: Stochastic RUL predictions of testing specimens 6-11. Specimens 6-8 correspond to the ones with repair that reached the EOL. Specimens 9-11 represent the ones with repair that did not reach the EOL. The true RUL is known only for the specimens that reached the EOL; here, for the trajectory part that comes after the repair of specimens 6-8.

4.5.2. EVALUATION OF REPAIR MODEL

With access to trajectories both pre- and post-repair, it becomes feasible to retain the RUL prediction directly before and immediately after the repair event. These two data points are the only necessary inputs to be fed to the proposed repair model for training. In this regard, the estimated posterior distributions μ_{mean}, σ_{mean} that are considered to model the posterior predictive distribution of the mean of recovery R_{mean} are depicted in Figure 4.11a and Figure 4.11b respectively. Based on these distributions, R_{mean} is estimated and illustrated in Figure 4.12 after running the MCMC algorithm 10 times (runs). In this figure, the mean of the posterior predictive distribution is impressively close to the mean RUL prediction of the HSMM prognostic model and satisfactorily near the mean true RUL of the testing specimens after repair.



(a)

(b)

Figure 4.11.: Posterior distribution of the random variables μ_{mean} (left) and σ_{mean} (right) that are necessary to estimate the posterior predictive distribution of the mean of recovery R_{mean}.

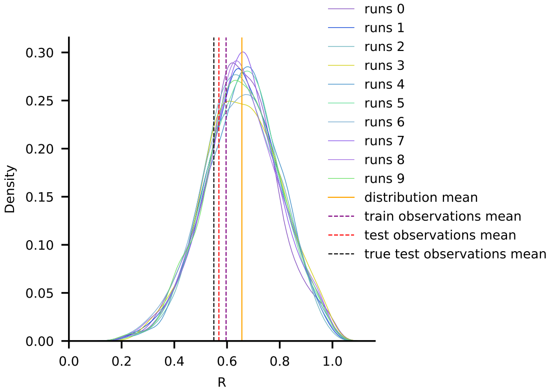


Figure 4.12.: Posterior predictive distribution of the mean of recovery (R_{mean}).

Additionally, by employing Equation 4.9 the posterior predictive distribution of μ_{new} can be estimated for each of the testing specimens accordingly. The posterior predictive distributions μ_{new} for testing specimens 6, 7, and 8 that reached the EOL after the repair are presented in the left column of Figure 4.13. Except for specimen 8, the repair model captures the true mean value in all other cases. Choosing different prior distributions could offer a posterior predictive distribution that better captures the true value of specimen 8 as well. However, since the prior distributions were chosen based on limited samples and according to [47], and specimen 8 had a much smaller recovery than the specimens used for training the repair model, it can be seen as a left outlier that makes it difficult to be accurately predicted. Nevertheless, after estimating μ_{new} and based on the process described in Table 4.2, σ_{new} can be calculated, and, simultaneously, the mean value and confidence intervals of the post-repair RUL can be approximated. As shown in the right column of Figure 4.13 the true RULs for each subfigure -even for specimen 8- belong inside the CI of the predicted by the repair model RUL based on the HSMM model.

Subsequently, a comparative study was conducted on various prognostic models and their impact on the repair model. Figure 4.13 additionally depicts predicted RUL under uncertainty by a repair model with an identical component. The RULs corresponding to observed recovery data points were predicted using the GBDT prognostic algorithm [51] utilizing Python's Scikit-learn package with default hyperparameters. Similar graphs for test specimens 6, 7, and 8 are presented. From these plots, it is clear that the HSMM model is a better candidate to employ for the repair model. Notably, the GBDT model's predictions based on which the observed recovery was calculated give almost identical predictions for every specimen. Consequently, this model struggles to capture the different patterns that each specimen may contain.

To sum up, it is noteworthy that even though our proposed repair model works for varying prognostic models, its accuracy depends on the feature extraction as well as the prognostic model. Moreover, it is affected by the chosen prior distributions of Bayesian inference. The latter can be better understood by constructing again the posterior predictive distributions of μ_{new} for each test specimen utilizing completely non-informative prior distributions. Such distribution corresponds to Uniform with wider ranges reflecting the minimum and maximum value of the mean and variance of the likelihood. Particularly, choosing prior distributions to be $\mu_{mean}{\sim}U(0.1, 0.9)$, $\sigma_{mean}{\sim}\ U(0.1, 0.3)$ the posterior predictive distributions are expected to be wider. Indeed, as shown in Figure C.1, the posterior predictive distributions of μ_{new} for test specimens 6, 7, and 8 managed to capture even the true recoveries. However, such distributions give predictions of RUL with increased uncertainty that are difficult to consider for a potential decision-making algorithm.

4.5.3. EXTENSION TO MULTIPLE REPAIRS

Although this methodology can be extended for multiple repairs, in this work where only one specimen subject to sequential repairs is available, a deterministic approach is followed by assuming an exponential decrease in the mean of recovery. Having the distribution of recovery after the first repair, κ can be calculated by Equation 4.11, hence $\kappa{=}0.67$. Determining the values of λ and κ, three additional distributions are estimated for representing two, three, and four sequential repairs utilizing Equation 4.10. These distributions alongside the distribution representing the first repair can be seen in

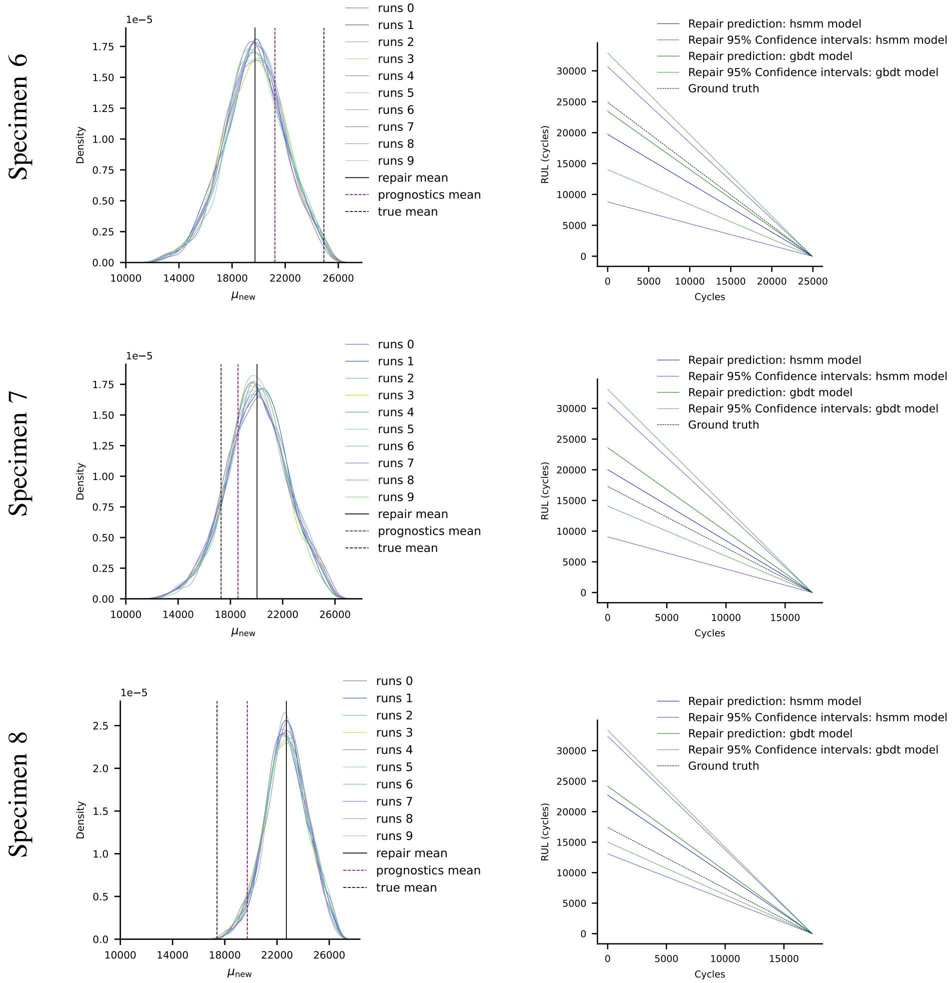
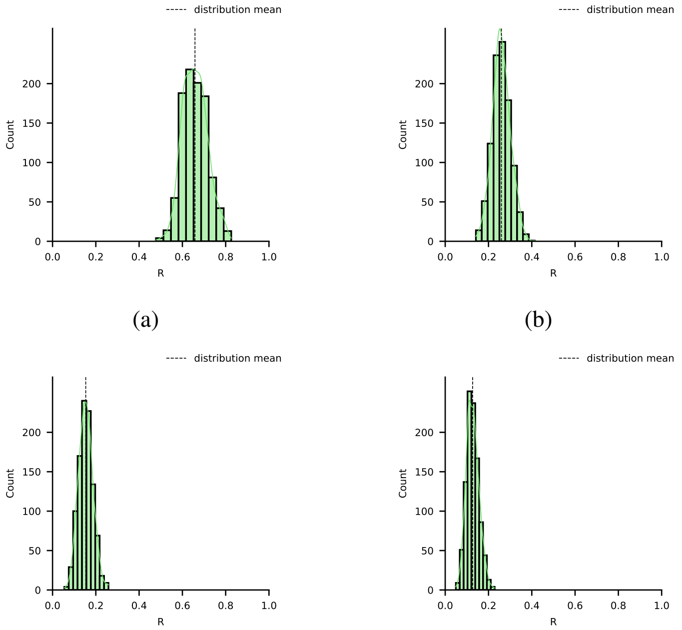


Figure 4.13.: Posterior predictive distributions of μ_{new} for test specimens 6, 7, and 8 (left column) and comparison of RUL predictions for the corresponding specimens after repair (X_{new}) between two prognostic models; HSMM and GBDT (left column). It is clear that the HSMM model is a better candidate to employ for the repair model. Notably, the GBDT model's predictions based on which the observed recovery was calculated give almost identical predictions for every specimen.

Figure 4.14.

As expected, because the exponential reduction was chosen on purpose to be sharp, from the second repair and thereafter, the distribution of recovery drastically tends to zero.

Consequently, in this case study, if PPDM was considered, the decision-maker should avoid performing more than one or two imperfect repairs sequentially as it will not assist in extending the useful life of the examined component.



(a) (b)

(c) (d)

Figure 4.14.: Posterior predictive distributions of the mean of recovery for one (**a**), two (**b**), three (**c**), and four (**d**) sequential repairs. It can be seen how these distributions shift towards zero by increasing the number of repairs.

4.6. CONCLUSION

There is a growing urgency to address imperfect repairs, driven by sustainability concerns and the shortage of replacement parts. Imperfect repairs inherently carry uncertainties regarding the extent of recovery they achieve. The proposed imperfect repair model conceptualizes the effect of imperfect repairs as a stochastic increase in RUL (recovery). Instead of constructing complex degradation models incorporating unknown repair dynamics, this approach leverages existing PHM techniques to estimate RUL based on sensory data.

By treating RUL as the primary contributor in linking degradation and imperfect repairs, this work offers some key advantages over existing methods. First, this work makes it possible to predict the health state of the component post-repair before taking any maintenance action. This serves as valuable information for optimizing maintenance plans. Second, the repair model can be developed independently of the rest of the PHM framework. Existing feature extraction and prognostic models could be considered, leveraging the existing state-of-the-art feature extraction and prognostic models introduced in the field of PHM. Third, this approach tackles the challenge of lacking run-to-failure trajectories of repaired specimens since the specimen does not have to reach the EOL condition after the repair to train the model. For imperfect repair modelling, the RUL value estimations directly before and after imperfect repair actions of a limited (here three) number of specimens are required (and three more were considered for testing). Finally, since the repair model is based on an MCMC-based Bayesian inference technique, it alleviates the need for predefining conjugate prior and likelihood distributions. This allows for the utilization of varying prior distributions depending on the domain expertise. Consequently, the repair model can be adapted to different material properties and repair techniques.

Concluding, the developed imperfect repair model can be integrated into the proposed PHM framework as a kind of maintenance decision. The model described in this chapter and the effects of a replacement action represent the maintenance action phase of the PHM strategy. This phase is necessary to proceed to the PPDM phase, which is discussed in the next chapter.

REFERENCES

[1] P. Komninos, G. Galanopoulos, T. Kontogiannis, N. Eleftheroglou, and D. Zarouchas. “A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty”. In: *Expert Systems with Applications* 288 (2025), p. 127723. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2025.127723.

[2] W. J. C. Verhagen, B. F. Santos, F. Freeman, P. van Kessel, D. Zarouchas, T. Loutas, R. C. K. Yeun, and I. Heiets. “Condition-Based Maintenance in Aviation: Challenges and Opportunities”. In: *Aerospace* 10.9 (2023). ISSN: 2226-4310. DOI: 10.3390/aerospace10090762. URL: https://www.mdpi.com/2226-4310/10/9/762.

[3] F. D. Carlo and M. A. Arleo. “Imperfect Maintenance Models, from Theory to Practice”. In: *System Reliability*. Ed. by C. Volosencu. Rijeka: IntechOpen, 2017. Chap. 18. DOI: 10.5772/intechopen.69286. URL: https://doi.org/10.5772/intechopen.69286.

[4] H. Pham and H. Wang. “Imperfect maintenance”. In: *European journal of operational research* 94.3 (1996), pp. 425–438.

[5] T. Nakagawa. “Optimum Policies When Preventive Maintenance is Imperfect”. In: *IEEE Transactions on Reliability* R-28.4 (1979), pp. 331–332. DOI: 10.1109/TR.1979.5220624.

[6] H. W. Block, W. S. Borges, and T. H. Savits. “Age-dependent minimal repair”. In: *Journal of Applied Probability* 22.2 (1985), pp. 370–385. DOI: 10.2307/3213780.

[7] M. A. K. Malik. “Reliable Preventive Maintenance Scheduling”. In: *IEEE Transactions* 11.3 (1979), pp. 221–228. DOI: 10.1080/05695557908974463. eprint: https://doi.org/10.1080/05695557908974463. URL: https://doi.org/10.1080/05695557908974463.

[8] M. Kijima, H. Morimura, and Y. Suzuki. “Periodical replacement problem without assuming minimal repair”. In: *European Journal of Operational Research* 37.2 (1988), pp. 194–203. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(88)90329-3. URL: https://www.sciencedirect.com/science/article/pii/0377221788903293.

[9] M. Kijima. “Some results for repairable systems with general repair”. In: *Journal of Applied Probability* 26.1 (1989), pp. 89–102. DOI: 10.2307/3214319.

[10] L. Doyen. “Asymptotic properties of imperfect repair models and estimation of repair efficiency”. In: *Naval Research Logistics (NRL)* 57.3 (2010), pp. 296–307. DOI: https://doi.org/10.1002/nav.20406. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20406. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20406.

[11] M. Kijima and T. Nakagawa. “Replacement policies of a shock model with imperfect preventive maintenance”. In: *European Journal of Operational Research* 57.1 (1992), pp. 100–110. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(92)90309-W. URL: https://www.sciencedirect.com/science/article/pii/037722179290309W.

[12] H. WANG and H. PHAM. “Optimal maintenance policies for several imperfect repair models”. In: *International Journal of Systems Science* 27.6 (1996), pp. 543–549. DOI: 10.1080/00207729608929248. eprint: https://doi.org/10.1080/00207729608929248. URL: https://doi.org/10.1080/00207729608929248.

[13] M. Shaked and J. G. Shanthikumar. “Multivariate Imperfect Repair”. In: *Operations Research* 34.3 (1986), pp. 437–448. DOI: 10.1287/opre.34.3.437. eprint: https://doi.org/10.1287/opre.34.3.437. URL: https://doi.org/10.1287/opre.34.3.437.

[14] P. D. Van and C. Bérenguer. “Condition-Based Maintenance with Imperfect Preventive Repairs for a Deteriorating Production System”. In: *Quality and Reliability Engineering International* 28.6 (2012), pp. 624–633. DOI: https://doi.org/10.1002/qre.1431. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.1431. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.1431.

[15] C. HU, H. PEI, Z. WANG, X. SI, and Z. ZHANG. “A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities”. In: *Chinese Journal of Aeronautics* 31.3 (2018), pp. 514–528. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2018.01.009. URL: https://www.sciencedirect.com/science/article/pii/S1000936118300256.

[16] P. Do, A. Voisin, E. Levrat, and B. Iung. “A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions”. In: *Reliability Engineering & System Safety* 133 (2015), pp. 22–32. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2014.08.011. URL: https://www.sciencedirect.com/science/article/pii/S095183201400204X.

[17] L. D. Franck Corset and O. Gaudoin. “Bayesian Analysis of ARA Imperfect Repair Models”. In: *Communications in Statistics - Theory and Methods* 41.21 (2012), pp. 3915–3941. DOI: 10.1080/03610926.2012.698688. eprint: https://doi.org/10.1080/03610926.2012.698688. URL: https://doi.org/10.1080/03610926.2012.698688.

[18] Y. Fuqing and U. Kumar. “A General Imperfect Repair Model Considering Time-Dependent Repair Effectiveness”. In: *IEEE Transactions on Reliability* 61.1 (2012), pp. 95–100. DOI: 10.1109/TR.2011.2182222.

[19] R. Pan and S. E. Rigdon. "Bayes Inference for General Repairable Systems". In: *Journal of Quality Technology* 41.1 (2009), pp. 82–94. DOI: 10.1080/00224065.2009.11917762. eprint: https://doi.org/10.1080/00224065.2009.11917762. URL: https://doi.org/10.1080/00224065.2009.11917762.

[20] L. Doyen and O. Gaudoin. "Classes of imperfect repair models based on reduction of failure intensity or virtual age". In: *Reliability Engineering & System Safety* 84.1 (2004). Selected papers from ESREL 2002, pp. 45–56. ISSN: 0951-8320. DOI: https://doi.org/10.1016/S0951-8320(03)00173-X. URL: https://www.sciencedirect.com/science/article/pii/S095183200300173X.

[21] J. Ma, L. Cai, G. Liao, H. Yin, X. Si, and P. Zhang. "A multi-phase Wiener process-based degradation model with imperfect maintenance activities". In: *Reliability Engineering & System Safety* 232 (2023), p. 109075. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2022.109075. URL: https://www.sciencedirect.com/science/article/pii/S0951832022006901.

[22] B. Cai, Y. Wang, Y. Zhang, Y. Liu, W. Ge, R. Li, Y. Liu, and G. Liu. "Condition-based maintenance method for multi-component system based on RUL prediction: Subsea tree system as a case study". In: *Computers & Industrial Engineering* 173 (2022), p. 108650. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2022.108650. URL: https://www.sciencedirect.com/science/article/pii/S0360835222006386.

[23] K. T. Nguyen, P. Do, K. T. Huynh, C. Bérenguer, and A. Grall. "Joint optimization of monitoring quality and replacement decisions in condition-based maintenance". In: *Reliability Engineering & System Safety* 189 (2019), pp. 177–195. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.04.034. URL: https://www.sciencedirect.com/science/article/pii/S0951832018313097.

[24] Y. Liu, Y. Chen, and T. Jiang. "Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach". In: *European Journal of Operational Research* 283.1 (2020), pp. 166–181. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2019.10.049. URL: https://www.sciencedirect.com/science/article/pii/S0377221719309014.

[25] E. Skordilis and R. Moghaddass. "A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics". In: *Computers & Industrial Engineering* 147 (2020), p. 106600. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106600. URL: https://www.sciencedirect.com/science/article/pii/S036083522030334X.

[26] A. Bousdekis and G. Mentzas. "A Proactive Model for Joint Maintenance and Logistics Optimization in the Frame of Industrial Internet of Things". In: *Operational Research in the Digital Era – ICT Challenges*. Ed. by A. Sifaleras and K. Petridis. Cham: Springer International Publishing, 2019, pp. 23–45. ISBN: 978-3-319-95666-4.

[27] C. Song, C. Zhang, A. Shafieezadeh, and R. Xiao. “Value of information analysis in non-stationary stochastic decision environments: A reliability-assisted POMDP approach”. In: *Reliability Engineering & System Safety* 217 (2022), p. 108034. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.108034. URL: https://www.sciencedirect.com/science/article/pii/S095183202100541X.

[28] L. Keizers. “Hybrid prognostics for predictive maintenance: Combining physics-based and data-driven methods to overcome prognostic challenges”. English. PhD Thesis - Research UT, graduation UT. Netherlands: University of Twente, May 2025. ISBN: 978-90-365-6573-8. DOI: 10.3990/1.9789036565745.

[29] T. Kontogiannis, M. Salinas-Camus, and N. Eleftheroglou. “Hidden Markov model applications: Aviation prognostics”. In: *Stochastic Modeling and Statistical Methods: Advances and Applications*. Academic Press, 2025, pp. 191–213. ISBN: 9780443316951. DOI: https://doi.org/10.1016/B978-0-44-331694-4.00015-3. URL: https://doi.org/10.1016/B978-0-44-331694-4.00015-3.

[30] T. Loutas, N. Eleftheroglou, and D. Zarouchas. “A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data”. In: *Composite Structures* 161 (2017), pp. 522–529. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2016.10.109. URL: https://www.sciencedirect.com/science/article/pii/S0263822316316506.

[31] N. Eleftheroglou, S. S. Mansouri, T. Loutas, P. Karvelis, G. Georgoulas, G. Nikolakopoulos, and D. Zarouchas. “Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the Lithium-Polymer batteries of unmanned aerial vehicles with uncertainty quantification”. In: *Applied Energy* 254 (2019), p. 113677. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.113677. URL: https://www.sciencedirect.com/science/article/pii/S0306261919313649.

[32] T. Loutas, N. Eleftheroglou, G. Georgoulas, P. Loukopoulos, D. Mba, and I. Bennett. “Valve Failure Prognostics in Reciprocating Compressors Utilizing Temperature Measurements, PCA-Based Data Fusion, and Probabilistic Algorithms”. In: *IEEE Transactions on Industrial Electronics* 67.6 (2020), pp. 5022–5029. DOI: 10.1109/TIE.2019.2926048.

[33] P. Komninos, T. Kontogiannis, N. Eleftheroglou, and D. Zarouchas. “A robust generalized deep monotonic feature extraction model for label-free prediction of degenerative phenomena”. In: *Data-Centric Engineering* (2025). Submitted for Publication (Second Revision).

[34] D. van Ravenzwaaij, P. Cassey, and S. D. Brown. “A simple introduction to Markov Chain Monte–Carlo sampling”. In: *Psychonomic Bulletin & Review* 25.1 (Feb. 2018), pp. 143–154. ISSN: 1531-5320. DOI: 10.3758/s13423-016-1015-8. URL: https://doi.org/10.3758/s13423-016-1015-8.

[35] M. D. Lee and E.-J. Wagenmakers. *Bayesian Cognitive Modeling: A Practical Course*. Cambridge University Press, 2014. DOI: 10.1017/CBO9781139087759.

[36] D. Fink. “A Compendium of Conjugate Priors”. In: *Technical Report.* (Jan. 1997).

[37] D. A. Freedman. "On the Convergence of Approximants to a Markov Chain, I". In: *Proceedings of the National Academy of Sciences of the United States of America* 60.1 (1968), pp. 66–72. ISSN: 00278424. URL: http://www.jstor.org/stable/58430 (visited on 11/16/2023).

[38] M. D. Hoffman, A. Gelman, *et al.* "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[39] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. "Markov Chain Monte Carlo in Practice: A Roundtable Discussion". In: *The American Statistician* 52.2 (1998), pp. 93–100. ISSN: 00031305. URL: http://www.jstor.org/stable/2685466 (visited on 11/16/2023).

[40] L. Martino, V. Elvira, and F. Louzada. "Effective sample size for importance sampling based on discrepancy measures". In: *Signal Processing* 131 (2017), pp. 386–401. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2016.08.025. URL: https://www.sciencedirect.com/science/article/pii/S0165168416302110.

[41] R. Patz and B. Junker. "A Straightforward Approach to Markov Chain Monte Carlo Methods for Item Response Models". In: *Journal of Educational and Behavioral Statistics* 24 (June 1999), pp. 146–178. DOI: 10.3102/10769986024002146.

[42] R. Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Tech. rep. University of Toronto Toronto, ON, Canada, 1993.

[43] R. Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* (June 2012). DOI: 10.1201/b10905-6.

[44] Z.-Q. Wang, C.-H. Hu, X.-S. Si, and E. Zio. "Remaining useful life prediction of degrading systems subjected to imperfect maintenance: Application to draught fans". In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 802–813. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.08.016. URL: https://www.sciencedirect.com/science/article/pii/S0888327017304429.

[45] E. Díaz-Francés and F. J. Rubio. "On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables". In: *Statistical Papers* 54.2 (May 2013), pp. 309–323. ISSN: 1613-9798. DOI: 10.1007/s00362-012-0429-2. URL: https://doi.org/10.1007/s00362-012-0429-2.

[46] D. Kuethe, A. Caprihan, H. Gach, I. Lowe, and E. Fukushima. "Imaging obstructed ventilation with NMR using inert fluorinated gases". In: *Journal of applied physiology (Bethesda, Md. : 1985)* 88 (July 2000), pp. 2279–86. DOI: 10.1152/jappl.2000.88.6.2279.

[47] R. Srilakshmi, M. Ramji, and V. Chinthapenta. "Fatigue crack growth study of CFRP patch repaired Al 2014-T6 panel having an inclined center crack using FEA and DIC". In: *Engineering Fracture Mechanics* 134 (2015), pp. 182–201. ISSN: 0013-7944. DOI: https://doi.org/10.1016/j.engfracmech.2014.12.012. URL: https://www.sciencedirect.com/science/article/pii/S0013794414004081.

[48] C. P. Robert and G. Casella. “Iterated and Sequential Importance Sampling”. In: *Monte Carlo Statistical Methods*. New York, NY: Springer New York, 2004, pp. 545–580. ISBN: 978-1-4757-4145-2. DOI: 10.1007/978-1-4757-4145-2˙14. URL: https://doi.org/10.1007/978-1-4757-4145-2˙14.

[49] G. Casella, C. P. Robert, and M. T. Wells. “Generalized Accept-Reject Sampling Schemes”. In: *Lecture Notes-Monograph Series* 45 (2004), pp. 342–347. ISSN: 07492170. URL: http://www.jstor.org/stable/4356322 (visited on 05/06/2024).

[50] N. Eleutheroglou, G. Galanopoulos, and T. Loutas. “Similarity learning hidden semi-Markov model for adaptive prognostics of composite structures”. In: *Reliability Engineering & System Safety* 243 (2024), p. 109808. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2023.109808. URL: https://www.sciencedirect.com/science/article/pii/S0951832023007226.

[51] A. Natekin and A. Knoll. “Gradient boosting machines, a tutorial”. In: *Frontiers in Neurorobotics* 7 (2013). ISSN: 1662-5218. DOI: 10.3389/fnbot.2013.00021. URL: https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021%20http://journal.frontiersin.org/article/10.3389/fnbot.2013.00021/abstract.

5

A FRAMEWORK FOR POST-PROGNOSIS DECISION-MAKING UNDER IMPERFECT MAINTENANCE DECISIONS

This final chapter combines all the PHM phases together for performing decision-making, i.e. PPDM. Here, the PPDM phase is first modeled as MDP and then optimized via DRL. The principles of Value of Information are integrated into this phase to choose the optimal time that new information should be acquired.

This chapter has been published in Array **27**, 100454 (2025) [1]

5.1. INTRODUCTION

Although substantial research has been conducted on PHM, relatively little attention has been devoted to the critical aspect of the PHM strategy: extending the useful life of components and structures through effective maintenance scheduling. This objective is the primary driver behind the development of prognostic models. The process of making decisions to prolong the operational life of structures and their components falls under the attention of PPDM. Despite PPDM being introduced differently, the most precise definition was proposed in [2], where PPDM is defined as *the set of actions that should be optimally taken at a given time by satisfying a set of constraints and optimizing a set of objectives formulated in an objective (cost) function, to overcome an undesirable upcoming predicted event*.

Although PPDM is still in its infancy, several published works under the CbM umbrella can be found in the literature [3–10]. Nevertheless, these works either assumed a numerical model for creating RUL trajectories or predicted deterministic RUL values which is not representative of a real-case scenario where structures are subject to stochasticity. Related works that considered RUL as the input to the PPDM can be divided into approaches that solve the decision-making problem either numerically or data-driven. The numerical approaches [7, 8, 11–14] form the cornerstones for constructing more complex PPDM strategies to tackle real-world problems. However, these models are usually tested in numerical examples that are much simpler and less dynamic than a real case. They often consider deterministic variables to describe the inputs, otherwise, it is very difficult to find an optimal solution. Additionally, Numerical approaches require domain knowledge and lots of adaptation and preprocessing to design an accurate model [15], thus lacking generalizability and robustness. For example, Pater et al. [7] followed a model-based PPDM strategy for multi-component systems related to the aircraft industry with a limited stock of spare components. Their novelty lies in maintenance planning for several repairable independent multi-component systems considering that spare parts are not always available. They added the cost of leasing parts from an external supplier when the aircraft shop ran out of spare parts. This strategy managed to schedule long-term horizons and incorporate many different variables. Despite their effectiveness in the aircraft industry, their approach assumes the available days for maintenance, leasing spare parts, and the repair process as deterministic values and not stochastic, which may give overconfidence to the decision-making model, hence increasing the gap between simulation and real-world applications. Similar works related to multi-component systems utilizing the k-out-of-n technique, which is a numerical approach, have been recently published [16–18] and evaluated on numerical examples.

As more data from varying sources are acquired and fused, PPDM solutions are rapidly evolving from numerical approaches to data-driven [5, 9, 10, 19–24], mainly through Reinforcement Learning (RL) after modeling the task as a Markov Decision Process (MDP) or Partially-Observable MDP (POMDP). Data-driven approaches may increase the accuracy and the future horizon on which the decisions take effect. In general, working with stochastic variables and noisy data leads to the need for ML models that are capable of capturing correlations between these variables and accurately approximating their probability distributions. When the demanding input and output data is large, deep learning models usually replace typical ML. In the context of decision-making, RL is replaced by DRL, i.e. an ANN is used as a function approximation of the policy (responsible for state-action mapping) that needs optimization. Very recently, DRL emerged on PPDM to offer solutions in tasks with multi-component systems [3, 5, 7, 10, 25] where state and action spaces are large, mainly via utilizing the deep Q-Network (DQN) algorithm.

When considering maintenance decisions for PPDM, mostly perfect and imperfect maintenance scenarios have been examined. On the one hand, performing perfect maintenance, such as a perfect repair, is usually identical to replacing the structure with a brand-new one. Yet, the significant expenses associated with replacements have prompted a thorough exploration of the viability of repair techniques. On the other hand, imperfect maintenance restores the structure somewhere between the current condition before the repair, i.e. the As-Bad-As-Old (ABAO) condition, and the brand-new condition, i.e. the As-Good-As-New (AGAN) condition. A limited number of studies has considered both perfect and imperfect repairs, with the majority being applied to numerical examples [12, 18, 26–28] rather than on real-world applications [29, 30]. Moreover, all these works have one common limitation; the lack of predicting a component's health state after an imperfect repair, before the repair is actually performed, thus limiting maintenance scheduling optimization. This work marks the first time maintenance actions are scheduled within a horizon even after deciding on an imperfect repair in advance by considering the accumulated uncertainty.

A typical question in PPDM and generally in sequential decision-making is related to how important the available information is to make a decision without requiring additional data. This is mathematically described by VoI [31]. In PPDM literature, VoI is viewed as a gauge of the significance of transitioning to inspection tactics in addition to primary maintenance activities. For instance, Song et al. [32] utilized the VoI analysis to facilitate the quantitative assessment of the expected net benefits of collecting new information for non-stationary stochastic or time-dependent decision environments modeled as POMDP with unknown uncertainties. They determined whether to use additional information from inspection actions before taking a maintenance action. Another work suggested improving the quality of maintenance decision-making with the help of three maintenance and three inspection strategies using VoI with an application to a safety-critical marine structure [33]. The objective was to increase the lifetime of the structure by considering both maintenance decisions and inspections, if necessary. Maintenance decisions were derived via integrated crack information through Bayesian updating, thus adding also a measure of the uncertainty. VoI has been additionally considered for making optimal maintenance decisions by taking into account imperfect maintenance scenarios [34–36]. However, these studies once more focused on numerical examples rather than realistic applications simplifying significantly the PPDM step.

To sum up, the existing literature on PHM reveals significant advancements in PPDM.

However, despite these strides, several research gaps persist, indicating that PPDM remains in its nascent stages. Firstly, due to the limited number of works related to modeling the RUL behavior after an imperfect repair, existing PPDM frameworks that consider imperfect repairs are scheduling maintenance actions within a horizon until the point of planning that repair. Furthermore, none of the extant works have proposed a comprehensive framework capable of providing actionable decisions and quantifying the associated confidence or probability pertaining to the decision-making process. In other words, interpreting the confidence of the decision-making model in making decisions, especially within the DRL context, is missing from the literature related to PPDM. Additionally, a lack of research exists concerning the correlation between VoI, uncertainty quantification, and decision probabilities, which are pivotal in determining the optimal timing for acquiring new information within the PPDM paradigm. Finally, a thorough understanding of how various sources of uncertainty -epistemic and aleatoric-influence decision-making can significantly enhance our comprehension of the DRL model. This, in turn, facilitates more informed judgments regarding when to trust the model and when to defer to the expertise of a human specialist.

Addressing the aforementioned identified research gaps is imperative for advancing the maturity and efficacy of PPDM methodologies within the broader domain of PHM. In this regard, this study proposes a novel framework related to PPDM based on DRL that works under the concepts of the PHM strategy. Particularly, the novelty of this research can be summarized as follows:

- This research marks the first PPDM framework that schedules maintenance actions even after an imperfect repair has been planned by estimating the component's health condition after the repair.

- The proposed framework deals with the uncertainty introduced by the stochastic RUL and imperfect repairs. The estimated uncertainty is initially decomposed into epistemic and aleatoric, then is passed through the framework and is converted to probabilistic decisions, thus offering interpretability and a better understanding of the maintenance actions being decided by the DRL model.

- The framework's generalizability based on the user's demands alongside the interpretation of the developed DRL model's confidence over its decisions provides a risk-averse policy. Based on the level of reliability the user demands, decisions taken with relatively low probabilities are transformed into an 'I don't know' output message by the model instead.

- VoI guides the framework in determining the optimal times for acquiring new sensory data to refine its decisions, resulting in computational efficiency and significant resource savings. Simultaneously, it filters out the corresponding unsuccessful runs of the RL agent, guaranteeing reliable scheduling recommendations by the DRL model.

The remainder of this chapter is organized as follows. section 5.2 is firstly devoted to the background related to the blocks necessary to build the PPDM framework. section 5.3 focuses on constructing the entire framework related to PPDM and the description of the experimental setup as a case study. The evaluation of the methodology is presented in section 5.5 and in section 5.6 the main findings and limitations are discussed.

5.2. THEORETICAL BACKGROUND FOR CONSTRUCTING THE PPDM FRAMEWORK

In this section, the theoritical background for constructing the PPDM framework will be described. Particularly, the corresponding theory related to DRL will be discussed in subsection 5.2.1 before describing the uncertainty management (subsection 5.2.3) and VoI (subsection 5.2.4). The necessary theory regarding feature extraction, HI construction, prognosis, and single and multiple imperfect repairs was discussed in the previous chapters.

5.2.1. MARKOV DECISION PROCESS & DEEP REINFORCEMENT LEARNING

MDP is a fundamental stochastic model for performing sequential decision-making. Originating from the field of stochastic processes and decision theory, MDP provides a mathematical formalism to represent the dynamics of systems where decisions must be made in a sequential manner, with outcomes influenced by both stochasticity and the decisions themselves.

Formally, an MDP is defined by a tuple (S,A,P_t,r,γ), where S, A, P_t, r represent states, actions, transition probabilities, and rewards respectively, and γ is the discount factor, representing the importance of future rewards relative to immediate rewards. The loop of an MDP system starts by observing the current state $s_t \in S$ at time t. Subsequently, the decision-maker (agent) takes an action $\alpha_t \in A$ and transits to the next state s_{t+1} with a transition probability $P_t(s_{t+1}|s_t,\alpha_t)$. The transition from t to $t+1$ is called a *step*. After the agent transits to the next state it receives a reward $r_t \in r$. All this information, including the system's dynamics, is collected in the *environment* with which the agent interacts. The entire process ends after the agent reaches the terminal state, i.e. a predefined state that satisfies some specific conditions defined based on the task. When the agent reaches the terminal state, an *episode* is finished and a new one begins from an initial (usually random) condition at $t=0$.

The goal of the agent is to follow a policy $\pi(\alpha_t|s_t)$ that maps the states to actions, in order to maximize the expected discounted future reward over a potentially infinite horizon:

$$\pi(a_t \mid s_t) \quad \text{s.t.} \quad \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, s_{t+1})\right] \tag{5.1}$$

The discount factor γ can be usually found in the range of $[0.95,1.0]$, which determines how far into the future the agent will accept rewards. Certainly, if $\gamma=1$, the agent will consider the expected undiscounted reward over a finite horizon. If the horizon is unknown in practice (infinite in theory), then $\gamma<1$.

When the states cannot be observed directly, but an approximation is possible or they are partially observed, then an MDP system becomes a Partially Observable Markov Decision Process (POMDP). Formally, a POMDP is a 8-tuple $(S,A,R,P_t,\gamma,\Omega,O)$, where Ω is a finite set of observations the agent experiences by interacting with his environment and $O{:}S\times A\rightarrow\Pi(\Omega)$ is the observation function that gives a probability distribution over the possible observations. The agent's goal remains to maximize the expected undiscounted reward, but this time following a policy $\pi(\alpha_t|b_t)$, where b is the belief state, i.e. a probability measure over past observations needed to determine the current state. The reason for using the belief state comes from the following statement: past observations depend on the current one, thus the Markov property [37] is violated; a belief state could summarize the past observations into one or more probability values representing the current state s_t, which is independent of the previous one. These distributions of the belief state encode the agent's observations into a representation of each state of the environment and provide a basis for acting under uncertainty [38].

The dynamics of an MDP or POMDP are often captured through the Bellman equation, which describes the optimal policy based on which the optimal value function is calculated:

$$V^*(s_t) = \max_{\alpha_t} \sum_{s_{t+1}} P_t(s_{t+1} \mid s_t, \alpha_t)[R(s_t, \alpha_t, s_{t+1}) + \gamma V^*(s_{t+1})] \tag{5.2}$$

where $V^*(s_t)$ is the optimal value function given the state s_t and $\max_{\alpha_t}$ denotes the maximization over all possible actions α_t in state s_t. This equation reflects the Bellman optimality equation, which reveals that the value of a state under the optimal policy is equal to the maximum expected return achievable by selecting the best action in that state and subsequently following the optimal policy. Based on the above equation, the RL field aims at approximating it when the dynamics of the environment are unknown or too complex to model explicitly. In other words, RL is crucial when the transition probability P_t is unknown. RL algorithms, being the best candidates for solving an MDP or POMDP system, aim to discover the optimal policy or value function by maximizing cumulative rewards obtained through sequences of actions.

There are two large categories of RL algorithms; model-based and model-free. On the first hand, model-based algorithms earn or exploit a model of the environment dynamics, which includes the transition probabilities and the immediate rewards. These methods typically involve building or approximating a model from data collected during interaction with the environment and then using this model to plan actions. On the other hand, model-free algorithms directly learn optimal policies or value functions from interactions with the environment, without explicitly building a model of the environment dynamics. These methods rely solely on experience obtained through trial and error. Model-free algorithms are preferred in cases with complex and unknown dynamics, making them more flexible and robust in real-world scenarios.

Under the concepts of model-free algorithms, there are value-based and policy-based methods. Q-learning is the most known value-based RL algorithm that learns directly the optimal action-value function $Q(s,\alpha)$ from experience without requiring a model of the environment. An extension of Q-learning that utilizes ANN to approximate the action-value function, enabling RL in high-dimensional state spaces is called DQN [39]. Policy-based methods directly learn the policy function without explicitly estimating value functions. They typically use gradient ascent to maximize expected cumulative rewards. A typical example of such an algorithm is the REINFORCE [40], i.e. a policy gradient algorithm that updates the policy parameters in the direction of the gradient of the expected return.

Policy-based and value-based methods in RL have notable drawbacks. Policy-based methods often struggle with high variance in gradient estimates, leading to slow convergence and instability, and require many samples to learn effectively, especially in complex environments [41]. Value-based methods, meanwhile, can suffer from overestimation bias and difficulties with the exploration-exploitation trade-off [41]. Actor-critic methods attempt to combine the strengths of both approaches. They use an actor to learn the policy and a critic to evaluate it. Proximal Policy Optimization (PPO) [42] is a leading actor-critic method that utilizes ANNs to approximate both the policy and value function. PPO addresses issues like high gradient variance and instability seen in traditional policy gradient methods and is an on-policy method, meaning it updates its policy based on data collected from the current policy. Additional details related to the formulation of PPO can be found in section D.1.

5.2.2. ACTION MASKING

In RL, action masking is a technique used to restrict the set of actions that an agent can take in a given state. This restriction is based on the environment's rules or constraints, preventing the agent from selecting actions that are not permitted or valid in that particular state. The theory behind action masking lies in the idea of creating a more realistic and efficient learning environment for the RL agent. By limiting the available actions, action masking reduces the complexity of the learning problem by focusing the agent's attention on only the relevant actions. This can be particularly useful in environments where certain actions are not feasible or allowed in specific states, or where the action space is large and needs to be pruned to improve learning efficiency.

Action masking is applied to the action space based on the current state of the environment. It involves determining which actions are permissible or valid in the current state and filtering out the rest. This is typically done by defining a mask vector that indicates the availability of each action. The agent's policy and value estimation are then computed based on the masked action space. By excluding invalid actions, the agent can focus its learning efforts on the subset of actions that are relevant to the current state. During the learning process, the agent explores the environment by selecting actions and observing the resulting rewards and next states. Action masking ensures that the agent only considers actions that are permissible in each state, thereby guiding exploration towards more promising areas of the state-action space.

In DRL, action masking is mainly applied in the last layer of the ANN. For the PPO algorithm in discrete action spaces, such as the one examined in this study, the ANN's last layer used to approximate the policy typically contains a Softmax activation function to produce a measure of probabilities for taking each action. Consider the outcome of the ANN's last hidden layer to be z_{α_t}. This represents the logits of each action α_t. Then the probability of each action α_t at step t is calculated by:

$$\pi_\theta(\alpha_t|s_t) = \frac{exp(z_{\alpha_t})}{\sum_{\alpha_t'} exp(z_{\alpha_t'})} \tag{5.3}$$

The invalid action masking technique [43–45] for discrete domains underlines that each invalid action can be masked (its probability is set to zero) by assigning a huge negative number to the corresponding logits z_{α_t}. Hence, after passing this logit through the Softmax activation function, the corresponding probability of taking this action will be very close to zero.

5.2.3. UNCERTAINTY QUANTIFICATION IN DISCRETE DOMAINS

Uncertainty is inherent in all machine learning problems and can sometimes be so pronounced that the model fails to provide accurate predictions. Understanding the fundamental sources of uncertainty is crucial in describing this issue. While various characterizations of uncertainty exist in the literature, the most prevalent involves its division into epistemic and aleatoric uncertainty [46]. Epistemic uncertainty pertains to model uncertainty and reflects the inability of a machine learning model to precisely comprehend the dataset's distribution. High epistemic uncertainty suggests that the model struggles to capture the entirety of the dataset's distribution or lacks accuracy. This form of uncertainty often manifests during the testing phase when encountering out-of-distribution data, which may substantially differ from the training set. Increasing the amount of data can mitigate epistemic uncertainty by providing a more comprehensive description of the problem's nature. In contrast, aleatoric uncertainty emerges from the complexity, noise, and multi-modality present in the data. Unlike epistemic uncertainty, this type of uncertainty cannot be diminished by augmenting the dataset; hence, it is considered irreducible. The total uncertainty comprises the sum of aleatoric and epistemic uncertainty.

When examining uncertainty in deep learning, epistemic uncertainty manifests in the weights of neurons, reflecting the challenge of attaining optimal weights that accurately characterize the data. However, this uncertainty can be mitigated by fine-tuning the hyperparameters of the ANN or by augmenting the dataset with more data. Conversely, detecting aleatoric uncertainty is more challenging and often involves expressing it through an additional output neuron that predicts the variance in regression tasks [46], or by measuring the entropy of the model's outputs in classification tasks [47].

MUTUAL INFORMATION

Entropy serves as a valuable metric for uncertainty in discrete domains, offering a concise indication of confidence in predictions for each input. It is particularly favored in DRL tasks involving discrete actions, where the outputs of ANNs are generated akin to classification, often utilizing a Softmax activation function before selecting the appropriate decision or class. Given that DRL entails a more intricate objective function compared to Supervised Learning, assessing variance across outputs could worsen the optimization process, making entropy the predominant choice within the RL domain.

For a dataset $D=\{X,Y\}$ within the context of a discriminative classification task, the aleatoric uncertainty can be determined using Shannon's entropy [48], computed as the expected conditional entropy:

$$\mathbb{E}_{p(x)}[H[(p(y \mid x)]] = \mathbb{E}_{p(x)}\left[-\sum_{c=1}^{K} p(y=c \mid x) \ln p(y=c \mid x)\right] \tag{5.4}$$

where c is the corresponding class (in a classification task) or action (in an RL task), K is the number of classes/actions, and $H[p(.)]$ is the entropy. The entropy of a discrete probability distribution is an information-theoretic measure of uncertainty and is calculated by the Mutual Information (MI) [49]:

$$I(y,x) = KL[p(x,y)||p(x)p(y)] = H[p(y)] - \mathbb{E}_{p(x)}[H[(p(y|x)]] \tag{5.5}$$

Where KL is the KL divergence [50]. It has been proven in [47] that the decomposition of the uncertainty into epistemic and aleatoric in deep learning via the MI theory can be performed utilizing M number of ANNs (deep ensembles) as follows:

$$I(y,\ M|,x,\ X,\ Y) = H\left[\mathbb{E}_{p(M|X,Y)}[p(y|x,M)]\right] - \mathbb{E}_{p(M|X,Y)}[H[p(y|x,M)]] \tag{5.6}$$

In this equation, the term to the left reflects the epistemic uncertainty, the first term on the right side is the total uncertainty, and the second on the right is the aleatoric uncertainty. From Shannon's entropy, the total uncertainty is easily calculated. To estimate the aleatoric uncertainty, it is crucial to approximate the epistemic first. However, estimating epistemic uncertainty is difficult since we need to calculate the posterior distribution over the parameters θ. To capture the epistemic uncertainty in an ANN, one approach is to use deep ensembles [51, 52], i.e. utilizing M models that converge differently due to varying random initializations. For DRL tasks this is computationally prohibitive. A better approach is to approximate these M models with the MC dropout technique. MC dropout is acknowledged as a Bayesian approximation, thereby enabling standard point predictions alongside meaningful uncertainty assessments [53].

Given a set of parameters Θ representing the weights of an ANN, the posterior predictive distribution of $y*$ given $x*$ and D equals:

$$p(y^* \mid x^*, D) = \frac{1}{T}\sum_{i=1}^{T} p\left(y_i^* \mid x_i^*, \Theta\right) \tag{5.7}$$

where T is the number of forward passes performed to predict a particular data point y given a single input sample x. In practice, T stochastic forward passes are conducted through the ANN with dropout applied at each layer for every sample, followed by extracting the mean and variance from the outcomes.

5.2.4. VALUE OF INFORMATION

The VoI theory is a concept used in decision theory to quantify the benefit or utility gained from acquiring new information. At its core, it explores how obtaining additional information can enhance decision-making processes, particularly in situations where uncertainty exists. This uncertainty may arise due to incomplete information, randomness, or the complexity of the decision environment. Decision-makers have the option to acquire additional information before making a decision. This information can range from market data and scientific research to expert opinions and personal observations. VoI represents the value gained from acquiring new information. This value can be measured in terms of improved decision outcomes, reduced uncertainty, increased expected utility, or monetary gains. Nevertheless, acquiring information typically involves costs, such as time, money, or effort. Therefore, decision-makers must weigh the potential benefits of acquiring information against the associated costs. This involves assessing the marginal benefit of additional information relative to its marginal cost.

In the RL field, VoI is often analyzed within the framework of expected value. The agent calculates the expected value of different actions based on their probabilities of occurrence and associated payoffs. By incorporating the potential value of acquiring additional information, the agent can adjust their actions to maximize expected utility. As introduced in varying prior works [32, 36, 54], VoI is defined by the following equation:

$$VoI = \mathbb{E}[C_{action}] - \mathbb{E}\left[C_{action} | I_{add}\right] \tag{5.8}$$

where C_{action} is the cost of taking a specific action by the agent and $C_{action|I_{add}}$ is the cost of taking an action given additional information. In this study, the expected action costs represent the total uncertainty of the agent's decision. When $VoI<0$, there is no need to acquire additional information and reconsider the chosen action.

5.3. METHODOLODY

The entire process of the PHM framework as described in Figure 1.1 (circled in the blue rectangle) is particularly depicted in Figure 1.2. After acquiring sensory data from each component of the structure, features are extracted and then clustered utilizing the proposed DSMC model (see chapter 3). Subsequently, the clusters are fed to the HSMM prognostic model that estimates RUL under uncertainty. The predicted RUL is afterward fed to the imperfect repair model to predict the stochastic recovery of each component after one or more sequential repairs (see chapter 4). Having information regarding the distribution of recovery and predicted RULs of each component, the final step is to implement the PPDM task modeled as an MDP (firstly, as a POMDP and then converted to MDP) as described in subsection 5.3.2, and solved via deep RL (subsection 5.3.3). Figure 5.1 depicts the necessary building blocks as a framework scheme to consider for implementing the PPDM framework. Except for the prognostic and imperfect repair model, putting constraints on the decisions via action masking (subsection 5.3.4) could improve the agent's performance. Additionally, managing the introduced uncertainty and mapping it with the decisions (subsection 5.3.5) is of paramount importance to attach interpretability to the framework. Finally, utilizing VoI assists in choosing the optimal time to acquire new information from sensors to make a new decision (subsection 5.3.6).

5.3.1. PROBLEM FORMULATION

Before defining the problem, the terminology of several keywords should be defined. 'Slots' represent predefined time intervals in a schedule where maintenance actions can be implemented. Each slot has a 'slot capacity' which defines the maximum number of components that can be put simultaneously for maintenance. A 'task' refers to the problem PPDM should solve. A 'horizon' concerns a time range in which meaningful decisions are made.

The target of PPDM is to make optimal decisions for extending the useful life of each structure's component, hence extending the lifetime of the structure. The current examined PPDM task considers a multi-component system with dependent components. The dependency comes from the slot availability and capacity, meaning that not all components can be maintained on the same day. Given the RUL predictions for each component of a multi-component system, decisions should be made inside a predefined horizon concerning when one or more maintenance actions should be taken. The

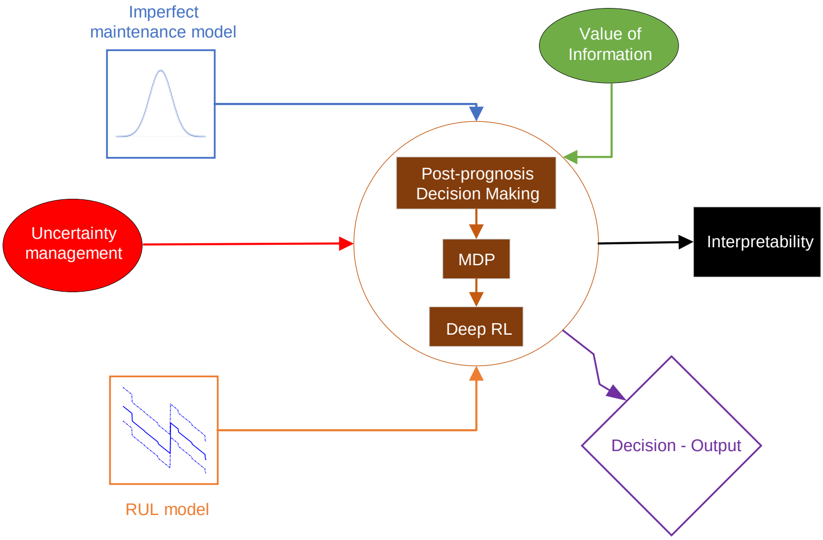
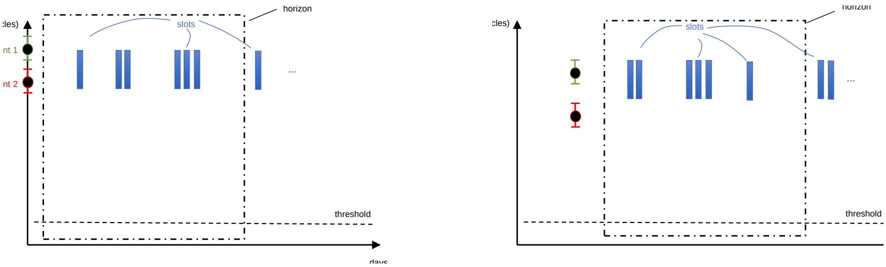


Figure 5.1.: Basic components for implementing the PPDM task. These blocks include the imperfect repair model, prognostic model, uncertainty management, VoI, and making PPDM interpretable.

decisions per component are ‘hold’, ‘imperfect repair’, or ‘replace’ corresponding to ‘do nothing’, perform an imperfect repair, and replace the component with a brand-new one, respectively. Within the horizon, there are available slots with different capacities per day. The existing slots are not guaranteed to be available every day. An example of the task given two components is illustrated in Figure 5.2. In Figure 5.2a, given the RULs of the two components and the corresponding available slots and their capacities (one, three, and two available slots respectively inside the horizon), the policy, i.e. the RL agent, should decide whether and when a maintenance action should be considered. When decisions have been scheduled, the horizon shifts to include the next available slot as shown in Figure 5.2b. Subsequently, the previous decisions are updated accordingly and a new decision is made for the newly available slot.

The difficulty of this task arises when an imperfect repair is decided as the estimated RUL after the repair will have a large uncertainty and the decisions thereafter might be untrustable. This highlights the importance of measuring this uncertainty and converting it to unbiased probabilities over the decisions. In this regard, a safety factor could be added to the framework, namely the probability threshold (p_{thresh}) based on which it is shown whether the RL agent is confident about its decision or not. If the decision has a lower probability than p_{thresh}, then all the decisions from this step till the end of the horizon are transformed into ‘N/A’, meaning that the agent does not know what decision to make. This increases the reliability of the framework, especially since p_{thresh} can be defined based on the user’s demands.

Ideally, an existing scheduling should be updated only when the uncertainty of the agent towards making decisions is high. Reducing the frequency of acquiring new data, extracting features, applying prognostics, and making decisions is of paramount importance for having a computationally efficient framework. Here comes the role of VoI in choosing which day should be the next to run the PPDM framework instead of running it daily.



(a) (b)

Figure 5.2.: Illustration of the PPDM task when the first (**a**) and second (**b**) data are acquired. The blue rectangles show the slot availability at each day. Different capacities of slots exist per day, making the task more complex to solve and converting the system to a multi-component one.

Before framing PPDM via implementing each building block, it is crucial to highlight all the necessary assumptions:

1. Imperfect repairs restore the state of the structure to a point between ABAO and AGAN conditions, adhering to a truncated normal distribution [12, 55, 56]. Consequently, RUL following a repair will fall within the range of ABAO and AGAN conditions, neither surpassing the latter nor falling below the former. Additional sequential repairs guarantee a smaller recovery rate of the component than the previous repairs.

2. The prognostic model operates independently of the repair model. The inherent uncertainties are independent.

3. Every specimen, either with or without repair, it exhibits similar sensory data values at the start of each trajectory, as crack growth is not yet detectable. This indicates that the clustering assignments and the RUL estimations are expected to be similar at the beginning of each trajectory.

4. Decisions should be made inside the horizon.

5. The task fails if any of the components' RUL drops below a predefined threshold (RUL_{thresh}).

6. The available slots and the corresponding capacities are static, i.e. if they are defined, they cannot change.

7. To be compatible with the available slots inside the horizon measured in days, the structure's operating cycles per day should be already determined. In this regard, for simplicity, it is assumed that the structure operates for a specific and constant amount of time per day.

8. The user of the PPDM framework defines the maximum number of repairs and replacements. Consequently, the costs are generally considered given a budget constraint.

9. The user additionally decides the horizon length ($L_{horizon}$), RUL_{thresh}, p_{thresh}, and the maximum number of steps (in days) to avoid data acquisition (this corresponds to VoI's contribution).

The choice of p_{thresh} and RUL_{thresh} makes it possible to have risk-averse or risk-prone policies. For example, a risk-averse policy will be considered if p_{thresh} is high or RUL_{thresh} low. Contrarily, if p_{thresh} is small or RUL_{thresh} large, the policy will be risk-prone.

5.3.2. PPDM TASK AND MDP FORMULATION

The PPDM task, as described in the previous page, naturally fits as a POMDP since the observations that come from the environment are noisy. Nevertheless, as will be further discussed in subsection 5.3.3, POMDP can be converted into a typical MDP by applying RNN to the ANN architecture to capture the unobserved states.

To establish a task as an MDP, it is crucial to define the environment in which the agent operates. The agent takes actions based on its observations within a finite horizon, $L_{horizon}$, with the goal of prolonging the lifecycle of a multi-component system. This involves extending the lifespan of each individual component or specimen by scheduling imperfect repairs and replacements. Figure 5.3 depicts the MDP related to the PPDM task. Particularly, the MDP consists of the following:

- Actions: The agent could take three discrete actions: 'hold', 'imperfect repair', and 'replace', encoded as $\{0, 1, 2\}$ respectively. The 'hold' action indicates that no action should be taken at the current step, 'imperfect repair' concerns the imperfect maintenance decision, and the 'replace' action is related to replacing the specimen with a brand-new one that starts from its AGAN condition. Depending on the number of specimens that are simultaneously examined, the total number of actions is $3N$, where N is the number of specimens.

- States: The observation (state) space consists of the stochastic RUL, the number of repairs ($n_{repairs}$) and replaces ($n_{replaces}$) that have already been done until step t, the horizon length (days) in which the agent can take actions, the 10 next available slot positions (days) inside the same horizon length, and the capacity of slots available at each slot position. The number of next available slot positions should be large enough to give the agent more exploration space available. All these variables are normalized in the `[0,1]` range according to their upper bounds. Particularly, the RUL is normalized based on the largest predicted RUL for each specimen separately, the repairs and replaces are normalized based on their maximum possible sequential repairs and replaces inside the predefined horizon, the normalized capacity is calculated by dividing with the maximum possible available slots given a slot position, and the next available slot positions are divided by the horizon length to calculate the normalized slot positions.

- Rewards: A reward function should be as simple as possible to help the agent reach the optimal policy. In this case, the agent must follow a policy that minimizes maintenance actions (and thus the costs) and extends the structure's health (by extending each specimen's EOL condition). An additional negative reward should be added to the agent when it reaches the terminal state. This state is reached if any specimen's RUL drops below a predefined threshold. After some trial and error, the simplest discovered reward function and its sub-components are given below:

$$r = r_{act} + r_{cost} + r_{maint} + r_{penalty} + r_{success} \tag{5.9}$$

$$r_{act} = \frac{5t}{L_{horizon}} \tag{5.10}$$

$$r_{cost} = -2, \text{ if any}(\alpha_t = 1 \text{ or } \alpha_t = 2) \tag{5.11}$$

$$r_{maint} = \frac{3.5 \cdot \sum_{i=1}^{N} 1 \cdot \left[\alpha_t^i \neq 0\right]}{N} \tag{5.12}$$

$$r_{penalty} = -50, \text{ if any}\left[RUL^i < threshold\right], \text{ i=1,2,..,}N \tag{5.13}$$

$$r_{success} = 100, \text{ if } t = L_{horizon} \tag{5.14}$$

In this study, $threshold$=0, but different values could be given depending on the desired system's safety. For instance, for an aircraft's engine, having $threshold$=0.8, i.e. the components' RUL should always be above 80% of the perfect healthy condition. The agent's ultimate target is to go as far as possible to the given horizon without letting any specimen drop its RUL below the threshold. Furthermore, the agent incurs a penalty for each scheduled maintenance action while also receiving a positive reward for consolidating as many maintenance actions as possible within the same time slot. This prompts the scheduling of maintenance actions at the same available slot, promoting a strategy with fewer schedules overall. Finally, the agent is rewarded with a huge value when the episode has been successfully finished.

- Environment: This MDP formulation is episodic, i.e. there is one or more conditions that terminate the episode before resetting the states. The episode terminates successfully if the agent accurately schedules maintenance actions within the horizon without expecting any failure from each of the specimens (at least one specimen's failure triggers the system's failure as well). Otherwise, the episode reaches an unsuccessful early termination. When an episode is finished, a new one starts with random initialization of the state space.

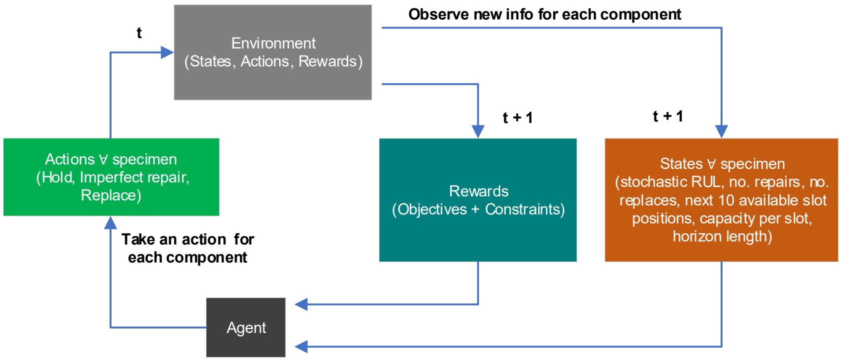


Figure 5.3.: Illustration of the MDP (states, actions, rewards). The illustration shows how an agent observes, takes actions, and receives rewards from the environment.

RANDOM INITIALIZATION OF ENVIRONMENT

The advantage of our approach is that the training is offline using synthetic data, thus having a theoretically infinite amount of training samples. The idea is to train the agent with high variance on data that always include the real data that will be observed at the evaluation step. Indeed, because sensor data is converted into RUL and the rest of the data in PPDM are related to logistics (slot position, capacity, etc.), achieving a generalized training subject to the user's demands is possible. This is accomplished by randomly distributing the initial RUL values of the environment and the other state variables per episode accordingly following a specified distribution, such as Gaussian, Uniform, and Poisson. Considering this, each state variable is distributed given a random distribution with the following characteristics[2]:

- Initial mean RUL [days]. Varying values of mean RUL are drawn from a Uniform distribution: $U(7,40)$.
- RUL noise [float]. RUL noise is drawn from a Uniform distribution: $U(1,5)$. This noise is added to create stochastic RUL trajectories.
- Maximum number of repairs [integer]. At each episode, a different number of maximum repairs is initialized, which follows a Uniform distribution: $U(0,4)$. The upper bound reflects our case study (four imperfect repairs are allowed before replacement).
- Maximum number of replacements. The maximum number of replacements follows a Uniform distribution: $U(0,2)$. The upper bound reflects our case study (two replacements are allowed within the horizon).
- Available slot position [days]. Only on specific days of the week, there are available slots. These days are determined by a Poisson distribution $Poisson(\lambda_{Poisson}=4)$. Thus, there will be available slots to schedule a maintenance action 4 days per week on average.
- Capacity per slot position [integer]. The capacity of each slot position is drawn from a Uniform distribution: $U(1,4)$. This implies that at each available slot position, 1-4 slots may be available to put specimens for imperfect repair or replacement. One slot corresponds to one component/specimen. The upper bound reflects our case study (four components).
- Distribution of recovery [float]. As estimated by the imperfect repair model. Since this random variable is very sensitive and significantly affects the entire process, it is decided to consider its real estimation based on the model's training utilizing the corresponding training specimens.

Since there is no prior information related to the PPDM task, Uniform distributions reflect the most uncertain condition.

5.3.3. THE DRL MODEL

The action space at each step in a system with N components should ideally be represented as an N-dimensional array, where each dimension corresponds to a specimen.

---

[2] It should be noted that each state variable is an array of N values representing each of the N specimens. Each value is randomly distributed, hence different values are drawn for each component which enlarges the observation space and increases the complexity of the environment.

However, due to the discrete nature of outputs from algorithms like DQN, which are commonly used in PPDM literature, representing such multidimensional arrays directly is challenging. To address this, a workaround involves mapping each combination of actions to an integer, typically using a ternary numerical system. For example, consider a multi-components system with four components and three possible actions each, and at step t the agent chooses the action array $[2,1,0,0]$ corresponding to [‘replace’, ‘imperfect repair’, ‘hold’, ‘hold’] for each specimen respectively. Since the DQN algorithm outputs a single integer, the agent should choose between a possible 3^4 discrete actions ranging between $[0,3^4)$. This integer should then be converted to the ternary system representing the desired array. Here, $\alpha_t=63$ since $63_3=2100\equiv[2,1,0,0]$. As such, a unique expression for each integer can be assigned. Note that if $\alpha_t<25$, say, $\alpha=4$, then $4_3=11$, hence $\alpha_t=[1,1]$ and left zero-padding should be applied until reaching the desired N-dimensional array; here, $\alpha_t=[0,0,1,1]$. Eventually, the action space for the DQN algorithm is $Discrete(3^N)$, but the desired actions should ideally form an N-dimensional array instead. If N is large, even DRL suffers from the exponentially increasing dimensionality of the action space. This can rapidly render the application of discrete-action RL algorithms intractable to domains with multi-dimensional action spaces [57]. This inspires motivation to choose an actor-critic method, such as PPO that considers N-independent Softmax functions to assign probabilities for each action.

Under the concepts of DRL, the PPO algorithm requires two ANN architectures; one that models the policy (policy network) and another that models the value function (value network). Figure 5.4 depicts this architecture. Observations are fed in the shared layers including two LSTM layers and an FC layer. Subsequently, the extracted hidden features are fed into the policy and value network. These networks have similar architectures, with the only difference on the final activation function, which is a Softmax for the policy network to produce probabilities[3] for each action, whilst the value network consists of a Rectified Linear Unit (ReLU) activation function that predicts a single value. Between each hidden layer, there are additional layers, namely a Dropout and a batch-normalization layer. Adjacent to each layer shown in Figure 5.4 there is a ReLU activation function, except between and after the LSTM layers where a hyperbolic tangent function (Tanh) is considered. In the same figure, the input dimensions of each layer are shown.

Although at each step inside an episode N_{obs} are stored, this architecture demands a length L of previous observations to be used, hence a two-dimensional array $[N_{obs},L]$. The hidden states extracted by the LSTM layers reflect the unobserved states of the environment, hence silently converting the POMDP into an MDP inside the ANN architecture [10, 21]. All hyperparameters related to DRL implemented with the PPO algorithm are stored in Table 5.1. A linear learning rate scheduler has been implemented to improve the model’s learning capabilities. Hence, the learning starts at a faster pace with $lr_{max}=10^{-2}$ and it linearly decreases until $lr_{min}=10^{-6}$. The number of observations

---

[3] The output of the Softmax activation function gives an overestimation of probabilities making the ANN overconfident about its predictions. By applying uncertainty quantification techniques to these predictions, it is possible to have an unbiased estimate of the model’s beliefs about its decisions [58, 59].

depends on the number of specimens N and is $N_{obs}=4N+21$, where 21 stands for the next 10 available slot positions and the corresponding capacity of each, plus the horizon length). Finally, to have a robust training process, 8 parallel environments are initialized with different randomness ($n_{workers}=8$). The PPO agent is trained in parallel in these environments.

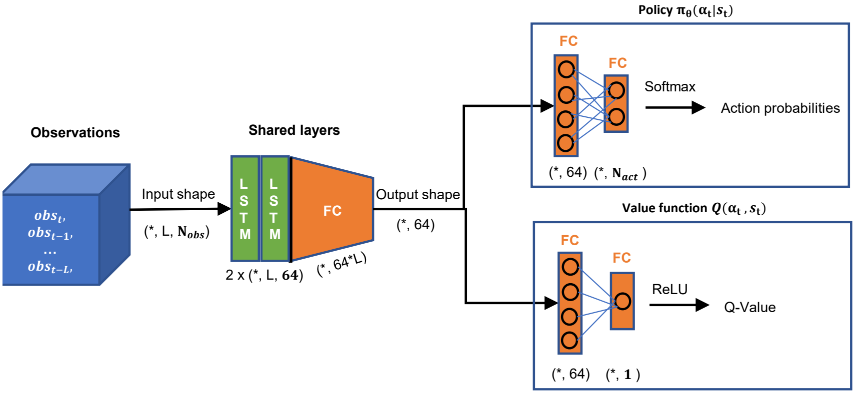


Figure 5.4.: ANN architecture for both policy network and value function. N_{obs} and N_{act} reflect the number of observations and actions respectively. The ‘*’ dimension reflects the batch dimension.

Table 5.1.: Hyperparameters related to DRL utilizing PPO algorithm.

Hyperparameter	Description	Value
lr_{min}	Minimum learning rate	10^{-6}
lr_{max}	Maximum learning rate	10^{-2}
n_{steps}	Number of steps per update of weights	64
$batch$	Batch size	128
$buffer$	Replay buffer limit to use for retraining the model with SL	$8.4 \cdot 10^6$
$epochs$	Number of epochs when optimizing the surrogate loss	5
n_{obs}	Number of observations	$4N+21$
L	Length of past observations to be used by the LSTM layers	8
d_{rate}	Dropout rate	0.5
γ	Discount factor	1.0
λ_{GAE}	Factor for trade-off of bias vs variance for GAE	0.95
ϵ	Clipping parameter	0.2
$C_{entropy}$	Entropy coefficient for the loss calculation (see [60])	0.01
$steps_{train}$	Total training steps	10^6
$n_{workers}$	Environments to run in parallel with different randomness	8

5.3.4. CONSTRAINTS & ACTION MASKING

The action masking technique is applied to guide the agent towards valid actions. Particularly, actions related to ‘imperfect repair’ or ‘replace’ are invalid when the agent has reached the maximum number of repairs or replacements, respectively, within an episode. Action masking is applied to the corresponding maintenance actions that ought to surpass those limits by assigning a large negative number before applying the Softmax activation function to assign a zero probability to the corresponding action. Additionally, the number of maintenance actions within an available slot should never exceed the capacity of that slot. A similar action masking approach is applied to the corresponding maintenance actions with the lowest probability scores until the capacity limit is satisfied.

5.3.5. MAPPING UNCERTAINTY WITH DECISION PROBABILITIES

As mentioned already, the policy network is responsible for assigning probabilities to each available action at each step. However, due to the tendency of ANNs to be overconfident, these probabilities can be easily either close to 1.0 or 0.0. Utilizing the MI theory, setting Equation 5.7 to Equation 5.6 gives the following:

$$\text{Model uncertainty} = \text{Total uncertainty - Aleatoric uncertainty} = \\ = \frac{1}{T} H\left[\sum_{i=1}^{T} p\left(y_i^* \mid x_i^*, \Theta\right)\right] - \frac{1}{T}\sum_{i=1}^{T} H\left[p\left(y^* \mid x^*, \Theta\right)\right] \tag{5.15}$$

In the case of three discrete actions that are passed through the Softmax activation function $p(y^*|x^*,\Theta)$ corresponds to the array $[p_{i1},p_{i2},p_{i3}]$ where p_{ij} is the probability of taking action $j=\{1,2,3\}$ at stochastic pass $i \in [1,T]$. In this study and according to a previous work [53], $T=30$ has been chosen. Taking the average over T stochastic passes makes an unbiased estimate of the probabilities of taking each action α_t at step t feasible. In this regard, the policy that the agent follows can give real probabilities of decisions. These decisions are interconnected with the measured uncertainty, which is crucial for bounding its upper values for normalization purposes. This is justified by Lemma 1 discussed in Appendix D.3. This means that the uncertainty can be normalized by dividing it by $log3$. This guarantees that the uncertainty will always be in the range of $[0,1]$. This is useful for training the VoI model and utilization of p_{thresh} concerning reliability.

Lemma 2 presented again in Appendix D.3 analyzes the relationship between p_{thresh} and the combined epistemic and aleatoric uncertainty. In this regard, we can ensure the physical meaning of the measured uncertainty, thus adding interpretability to our DRL model. Depending on the defined by the user p_{thresh}, if the maximum of the probabilities is less than this threshold, the agent’s decision is converted to the *‘N/A’* action. This is a hidden action that is not seen inside the MDP but is added after estimating the uncertainty during inference. In this context, when the agent is uncertain about its final decision, it is advisable to not rely solely on DRL and instead wait for human feedback. This approach renders the task risk-averse.

5.3.6. DECIDING OPTIMAL TIME TO ACQUIRE NEW INFORMATION

Acquiring new data in real time every day is cost-prohibitive and sometimes unnecessary. For instance, when all specimens are in their AGAN condition, it is already known that the agent should wait for the initial steps before taking any maintenance action. Consequently, there is no need to acquire new information in the subsequent steps, thus conserving resources. This can be achieved by utilizing VoI.

Figure 5.5 depicts the loop based on which the algorithm automatically decides whether to acquire new information. Recalling that each episode in the formulated MDP finishes successfully when the agent reaches the end of the horizon or fails when any of the specimens' RULs drops below the threshold. This means that given the current RUL information in real-time, operational conditions, and other logistic information, the agent creates a schedule for future maintenance actions within the horizon. During each episode, uncertainty can be estimated for each step. Recalling Equation 5.8, the expected costs in this study are related to the estimated uncertainty. The greater the degree of uncertainty, the higher the expected costs. VoI is applied throughout the entire episode as the average of the estimated uncertainty measured at each step. Therefore, the expected cost given a series of actions for each episode is as follows:

$$\mathbb{E}[C_{action}] = \frac{\sum_{i=0}^{steps_{episodic}} U^{(i)}}{steps_{episodic}} \tag{5.16}$$

where $steps_{episodic}$ represents the total steps until the episode has finished, either successfully or not. If $\mathbb{E}[C_{action}|_{I_{add}}]$ is estimated, VoI can be measured. If VoI is non-positive, the uncertainty estimated at the current episode is less than the expected uncertainty that will be observed in the next episode. In this case, acquiring information about the next episode is unnecessary. This can be extended to ignoring multi-steps ahead by counting the times that VoI is sequentially non-positive. Contrarily, when VoI becomes once positive, the expected uncertainty included in the next episode is less than the current one, hence the upcoming data are expected informative for the agent to update the decisions.

The formulation of VoI, though, demands the estimation of $\mathbb{E}[C_{action}|_{I_{add}}]$. Unfortunately, this information is not available beforehand. In this regard, the approach proposed in this work is to create a parameterized surrogate model f_θ based on ANN with parameters θ that outputs $\mathbb{E}[C_{action}|_{I_{add}}]$ s.t.:

$$\begin{aligned} \mathbb{E}\left[C_{action}|_{I_{add}}\right] &= f_\theta(\mathbb{E}\left[C_{action}^{tot}\right]_{n_{ep}}, \left[S_{0,n_{ep}}\right]) \\ \mathbb{E}\left[C_{action}^{tot}\right]_{n_{ep}} &= \{\mathbb{E}[C_{action}]_{n_{ep}-m}, \mathbb{E}[C_{action}]_{n_{ep}-m+1}, \ldots, \mathbb{E}[C_{action}]_{n_{ep}}\} \\ \left[S_{0,n_{ep}}\right] &= \{s_{0,n_{ep}-m}, s_{0,n_{ep}-m+1}, \ldots, s_{0,n_{ep}}\} \end{aligned} \tag{5.17}$$

Here, a state $s_{0,n_{ep}}$ corresponds to the first available observations of the episode n_{ep}, m is a hyperparameter representing the number of previous episodes to consider as input

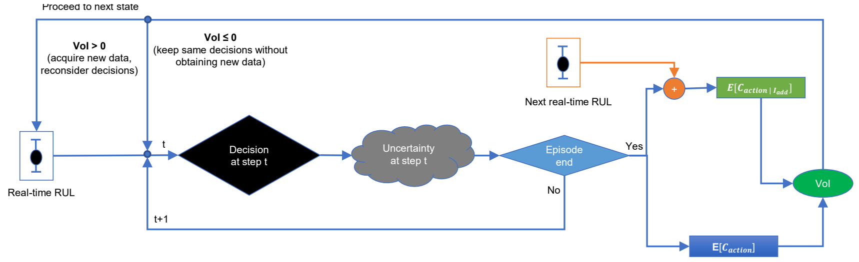


Figure 5.5.: Loop for deciding whether to acquire new information based on estimated VoI. MDP finishes successfully when the agent reaches the end of the horizon or fails when any of the specimens' RULs drops below the threshold. This means that given the current RUL information in real-time, operational conditions, and other logistic information, the agent creates a schedule for future maintenance actions within the horizon.

for the surrogate model to make n-episodic predictions ahead, and $\mathbb{E}[C_{action}]_{nep}$ is the expected cost at episode n_{ep} as calculated by Equation 5.16. Utilizing an ANN to make such predictions given sequential data leads to the choice of an LSTM-based architecture illustrated in Figure 5.6. Three non-bidirectional LSTM layers are structured. Between each layer, a Tanh activation function has been applied. Then, the dimensions are flattened into a one-dimensional array (the batch dimension remains untouched). This array is passed through three FC layers. Adjacent to the first two FC layers, a ReLU activation function is applied, whilst the last FC layer has a Tanh activation function. After each layer, batch normalization and Dropout are additionally considered. The hyperparameters of the surrogate model are stored in Table 5.2. A linear learning rate scheduler has been implemented to improve the model's learning capabilities. The optimization is performed with the Adam optimizer.

The key concept is to generate data for training the VoI model by evaluating the trained agent at each episode. For each episode n_{ep}, the corresponding $\mathbb{E}[C_{action}]_{nep}$ is calculated while simultaneously the initial observations $s_{0,nep}$ of that episode are being stored. Generating episodes sequentially based on a real case study means that each sequential episode contains the next available information regarding input data. For example, being at n_{ep}=2 means that the expected costs $\mathbb{E}[C_{action}]_{nep=1}$, $\mathbb{E}[C_{action}]_{nep=2}$ and $\mathbb{E}[C_{action}|I_{add}]_{1\rightarrow 2}$ are known. By specifically defining the case study, the RUL trajectories and the distribution of recovery for multiple repairs are available. Hence, it is possible to initially evaluate the RL agent for generating data related to VoI in an offline manner utilizing the training trajectories considered for estimating the distribution of recovery, before rendering the scheduling in real-time with the testing trajectories.

Generating data for training the VoI model requires an adaptation of the environment used for training the RL agent. By initializing the first episode to reflect a random day based on the RUL trajectories, each sequential episode should reflect the exact next day. Then, scheduling is applied for each episode and the uncertainty can be estimated. This time the episodes have a strong time dependency, which justifies the utilization of the LSTM module. After trial and error, generating 50 sequential episodes before randomly initializing the first episode for a new sequence was enough to train the VoI model. Having 50 generated sequential episodes corresponds to the hyperparameter m, hence m=`50`. Consequently, in this setup, the evaluation of the agent can be applied in an offline manner and is possible to generate as much episodic data as desired. Subsequently, labeled data can be generated to be fed into the surrogate modeling for training. In total, `10000` episodes were generated, hence `10000/50=200` random initializations were performed regarding the first episode of the sequence. From these samples, `8000` is considered to train and `2000` to validate the VoI model.

Finally, it is important to note that both successful and unsuccessful episodes are included as training data. An unsuccessful episode suggests that the (considerably well-trained) DRL model failed to make the correct decisions, resulting in higher uncertainty. In this context, the trained VoI model learns to filter out these unsuccessful episodes by suggesting the avoidance of data acquisition on these days. This indirectly enhances the DRL agent's performance, at the expense of fewer schedules within a given horizon. Although this approach recommends fewer schedule updates by the agent, it is more cost-efficient because it reduces the number of days considered to acquire new data and run the framework.

Table 5.2.: Hyperparameters related to surrogate model for the estimation of $C_{action}|_{I_{add}}$.

Hyperparameter	Description	Value
lr_{min}	Minimum learning rate	10^{-6}
lr_{max}	Maximum learning rate	10^{-2}
$batch$	Batch size	128
$epochs$	Number of epochs when optimizing the surrogate loss	400
m	Length of past data to be used by the LSTM layers	30
n	The number of multi-episode predictions	7
$samples_{train}$	The number of samples considered for training	6400
$samples_{val}$	The number of samples considered for validation	1600
$samples_{test}$	The number of samples considered for test	2000

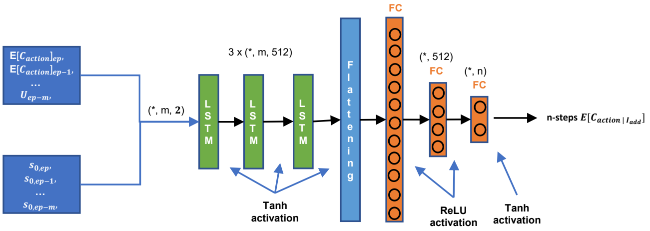


Figure 5.6.: ANN architecture of the surrogate model.

5.4. CASE STUDY

Consider a multi-component system consisting of a structure with four components/specimens. These components correspond to open-hole aluminum coupons. Each working day the structure functions for a specific and constant number of cycles. Maintenance actions should be made for the structure's components to extend the useful life of the entire structure. This extension concerns approximately doubling the average EOL of the components. It is assumed that this average EOL is 25-30 days, thus extending the structure's life up to 60 days ($L_{horizon}$=`60`) suffices. In order to increase the useful life by this amount within the horizon, four sequential repairs and two replacements are allowed. This means that the maximum possible number of maintenance actions is 10 if four repairs are planned between two replacements. Nevertheless, scheduling multiple maintenance actions is both cost-prohibitive and pointless, especially in case when additional imperfect repairs do not offer a significant recovery of the component.

To test the capabilities of the methodology, feature extraction, prognostics, and imperfect repair modeling have already been implemented in the experimental campaign described in chapter 4 and in Figure 4.6 involving tension-tension fatigue experiments on open-hole aluminum coupons subject to imperfect repairs. The raw signals corresponding to acoustic emission data are recorded by an AMSY-6 Vallen Systeme GmbH and two VS900-M wideband sensors. The initial five coupons are tested until failure to gather information about their fatigue life before repair and to determine the consistent lifetime percentage at which the repair is conducted. The remaining coupons undergo testing for 14,000 fatigue cycles, equivalent to 60% of the average fatigue life. Half of these are then tested until failure, while the other half are stopped at 11,000 cycles, which is about 60% of the average repaired coupon lifetime, demonstrating that it is not necessary to reach failure to develop the repair model. More detailed information about the specimens is given in Table 4.3. Since the trajectories of specimens with and without repairs are different, their names will be defined with two numbers for clarity; the first number indicates the number of repairs performed, and the second refers to the specific specimen. For example, the specimen labeled '02' without any repair is named specimen '`0_2`'. Similarly, specimen number '10' is split into two names: '`0_10`' for the trajectory without repair and ‘`1_10`’ for the trajectory with repair.

Since there have not been any practical experiments in the literature related to multiple imperfect repairs, additional trajectories have been generated based on the existing ones according to the following two steps:

i. The first RUL data point after a repair corresponding to μ_{new} and, consequently, the mean of the distribution of recovery $\mu(R_{mean})$ is exponentially reduced according to Equation 4.10. The constant variable κ is calculated by Equation 4.11. The variable λ can be any float number in the range of $[0,1]$. As already mentioned in chapter 4, to increase the complexity of the task, a sharp exponential decrease has been chosen with $\lambda=0.7$. Consequently, after the second repair, it is expected the distribution of recovery to be close to zero, thus the next repairs will not offer any significant recovery. It is expected that the RL agent should capture this phenomenon during training.

ii. Starting with the first RUL data point after the n-th repair, the corresponding RUL data point from the $(n-1)$-th repair with the same value is identified. The entire trajectory is then generated by using the mean and variance of the trajectory from the previous repair. Additional noise, modeled by a Uniform distribution $U(-3,3)$ measured in days, is added to this generated trajectory.

The DSMC model and the HSMM are trained on specimens ‘`01`’-‘`05`’, and, then, both models are utilized to predict RUL on the rest of the specimens. From the specimens subject to imperfect repairs, specimens ‘`09`’-‘`11`’ are taken into account for training the imperfect repair model based on which the RL agent is trained as well. Additionally, the RL agent was evaluated considering these specimens to generate the required data related to VoI. Specimens ‘`06`’-‘`08`’ correspond to ones used for evaluating the imperfect repair model and the outcome of the PPDM framework. During the evaluation phase of the RL agent, another RUL trajectory is generated, namely specimen ‘`12`’, by taking the average values of specimens ‘`06`’-‘`08`’, in order to include four components in the multi-components system. Trajectories for the second, third, and fourth sequential repairs are generated utilizing the aforementioned process.

Concerning the PPDM task, the hyperparameters for this case study are stored in Table 5.3. These hyperparameters are defined by the user of the framework each time before training the RL agent. In this setup, we seek to extend the EOL condition by doubling the average total number of working cycles. Particularly, since the average EOL before repair is approximately 25000 cycles, the structure should be extended up to 50000 cycles by performing maintenance actions to the components. This number of cycles is chosen to reflect a horizon of 60 working days. The structure containing these components is assumed to be working for a fixed and constant amount of time every day. In this regard, each day corresponds to $50000/60=833$ working cycles. In Table 5.4, the relation between working cycles and days for the four test specimens is shown.

Table 5.3.: Chosen hyperparameters related to PPDM for this case study.

Hyperparameter	Description	Value
$n_{components}$	No. components/specimens	4
$L_{horizon}$ [days]	Horizon length	60
$n^{max}_{repairs}$	Maximum no. sequential repairs	4
$n^{max}_{replaces}$	Maximum no. replaces	2
p_{thresh}	Probability threshold	0.6
$skip^{max}_{episodes}$ [days]	Maximum no. episodes to skip based on VoI	10
$slot_{pos}$ [days]	Weekly slot availability as a random variable	$\lambda_{poisson}$=4
$capacity$	Capacity of available slots per available day	$U(1,4)$

Table 5.4.: Characteristics of test specimens/components for this case study.

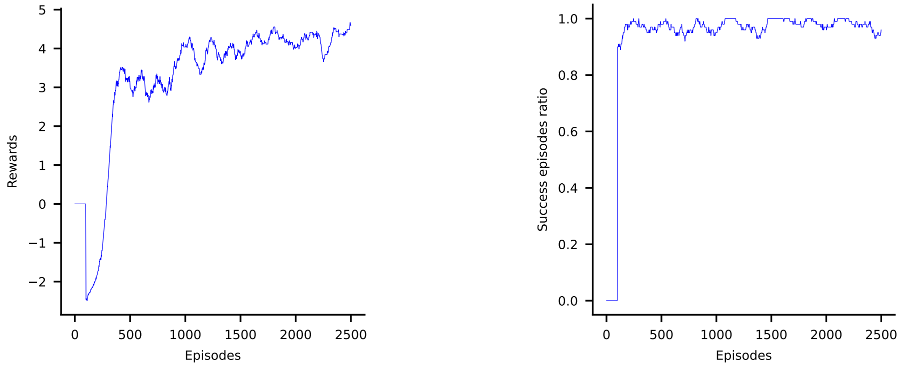
Test specimen No.	Component No.	EOL (cycles)	EOL (days)
06	0	24565	29.49
07	1	17445	20.94
08	2	17250	20.71
12	3	19753	23.71

5.5. PPDM OUTCOMES

In this section, the results of the entire framework are presented and discussed. Since the data of the chosen case study comes from the work presented in [61], the constructed HIs via the DSMC model have been already presented in Figure 4.9. The distribution of recovery for single and multiple repairs has been illustrated in Figure 4.14, for comprehensiveness, and will not be discussed again. Nevertheless, an important remark is that because the exponential reduction was chosen on purpose to be sharp, from the second repair and thereafter, the distribution of recovery drastically tends to zero. Consequently, in this case study, performing more than one or two repairs sequentially may not assist in extending the useful life of the examined component. Therefore, a well-trained agent is expected to make an 'imperfect maintenance' decision once or twice before considering a replacement of the component with a brand-new one.

5.5.1. PERFORMANCE OF THE RL AGENT

The RL agent was trained on a single GPU (NVIDIA GeForce RTX 2080). The entire training process for one million steps was approximately 40 minutes. The performance of the RL agent concerning the case study considering four components is depicted in Figure 5.7. Particularly, Figure 5.7a shows the agent's reward gains as training progresses and Figure 5.7b the ratio of successful episodes, i.e. the episodes where the agent has successfully reached the end of the horizon. Based on these graphs, training for 1000 episodes is enough to achieve high performance. These plots reflect the mean values after taking a rolling horizon with a rolling window of 100 for all plots.



(a)

(b)

Figure 5.7.: Training performance of the RL agent with four components. (**a**) rewards per episode and (**b**) ratio of successful episodes as the training progresses.

Decomposing the uncertainty into epistemic and aleatoric provides us with insightful information related to whether the RL agent struggles to take a decision due to difficulties in training (epistemic) or to noisy data (aleatoric). Recalling that the numbers 0, 1, and 2 represent the maintenance actions ‘hold’, ‘imperfect repair’, and ‘replace’ respectively, in Figure 5.8, the left part illustrates a part of an episode where the agent makes decisions with probabilities based on the red bars. Additionally, the corresponding uncertainties are depicted. During inference, each of these steps represents the average probability after running the step T times as discussed in subsection 5.3.5. The process of constructing two of these steps is illustrated in the right part of the figure where the probabilities are shown with blue bars. The upper graph shows the step with large aleatoric uncertainty since each of the MC samples gives almost equal probabilities in choosing either ‘hold’ or ‘imperfect maintenance’ decision. The lower graph has zero uncertainties since the agent chooses the ‘hold‘ decision with maximum confidence for all the MC samples. Although not shown here and because of the satisfying training of the RL agent, it is noteworthy that the epistemic uncertainty is always approximately or exactly zero. Nevertheless, if the agent is poorly trained, having large epistemic uncertainty would be easily observable since each MC sample representing a specific step would give quite different probabilities over the actions.

5.5.2. SCHEDULING MAINTENANCE ACTIONS

During the evaluation phase, the environment was run for 60 sequential episodes equal to the horizon length. In Figure 5.9, the scheduling representing the first episode is visualized. In this schedule, the actual EOL of each component is shown, only before

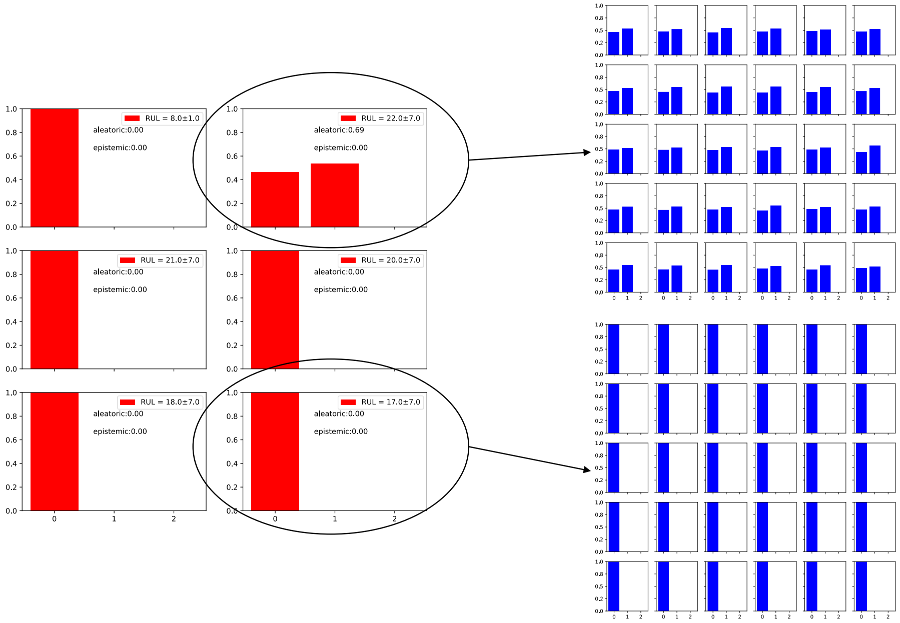


Figure 5.8.: Probabilities of taking each decision over a part of an episode and the corresponding uncertainty estimates are presented with red color in the graph. During inference, each of these steps is run T times. These samples and their corresponding probabilities are illustrated for two specific steps with blue color.

taking any maintenance action. Here, some key findings are observed that validate the satisfactory training of the RL agent based on the constructed reward function.

Firstly, the agent prioritizes scheduling maintenance actions for multiple components on the same day when the slots' capacity allows it. This leads to two consequences. On the first hand, there is a preference for maintaining the components earlier but on the same day rather than waiting for more days, even though RULs may allow it. On the other hand, the agent prioritizes scheduling earlier to avoid any RUL dropping below the threshold which signifies an unsuccessful episode.

Secondly, only one imperfect repair action is considered before replacement, which aligns with the sharp exponential decrease in recovery. The agent has learned that performing more than one sequential repair does not significantly increase the RUL and is cost-prohibitive. Simultaneously, a repair always precedes a replacement to extend the useful life as much as possible.

Thirdly, on day 57 and thereafter the agent considers deciding with a probability less than p_{thresh}. Thus, the scheduling from this step and so on becomes ‘N/A’, meaning that the agent does not know what decision to make. This points to the reliability of this framework in letting the user choose the amount of trust that should be given to the framework (based on the chosen by the user hyperparameter p_{thresh}) or when human feedback is preferred. As will be better shown in Figure 5.12, only a couple of days at the end of the horizon are becoming ‘N/A’. This occurs primarily because, after performing imperfect repairs, the RUL uncertainty increases significantly. Consequently, the agent is uncertain whether another repair close to the horizon’s end will be beneficial.

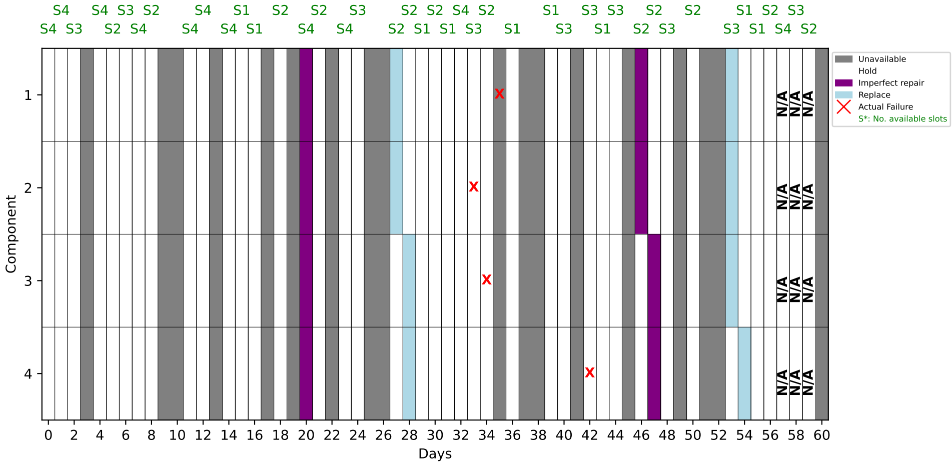


Figure 5.9.: Scheduling for the first episode. The decisions of the RL agent for the first episode are shown within the horizon of 60 days. In this schedule, the actual EOL of each component is shown only before taking any maintenance action.

To further show the behavior of each component in terms of RUL during the scheduling of the first episode, Figure 5.10 depicts how the predicted RUL is affected based on the agent’s maintenance actions. Since this refers to the first episode, i.e. when all components are in the brand-new condition, the corresponding shown RULs represent the first data point of each of the trajectories. This figure additionally presents the probabilities of taking each maintenance action. All other decisions made by the agent that are not shown correspond to the ‘hold’ decision and do not affect RUL behavior. As expected, after scheduling an ‘imperfect repair’, the variance of the RUL drastically increases until a ‘replace’ decision is made. Then, RUL contains only the variance related to its stochastic behavior as predicted by the prognostic model. From this figure, it is clear our previous statement related to scheduling components for maintenance on the same day rather than waiting for more days. In other words, although the agent could take additional ‘hold’ decisions before performing a maintenance action since RUL allows it, the overall cost is lower if more components are maintained on the same day than maintained later but on different dates.

Since those predicted RULs refer to the brand-new condition of each component, according to assumption 3, the plots have rationally similar trends, even after an imperfect repair maintenance action. In Figure D.1, an example of another RUL behavior is illustrated representing day 36, where the corresponding RUL data points have different values. For component 1, this day corresponds to the last row of the scheduling related to Figure 5.12.

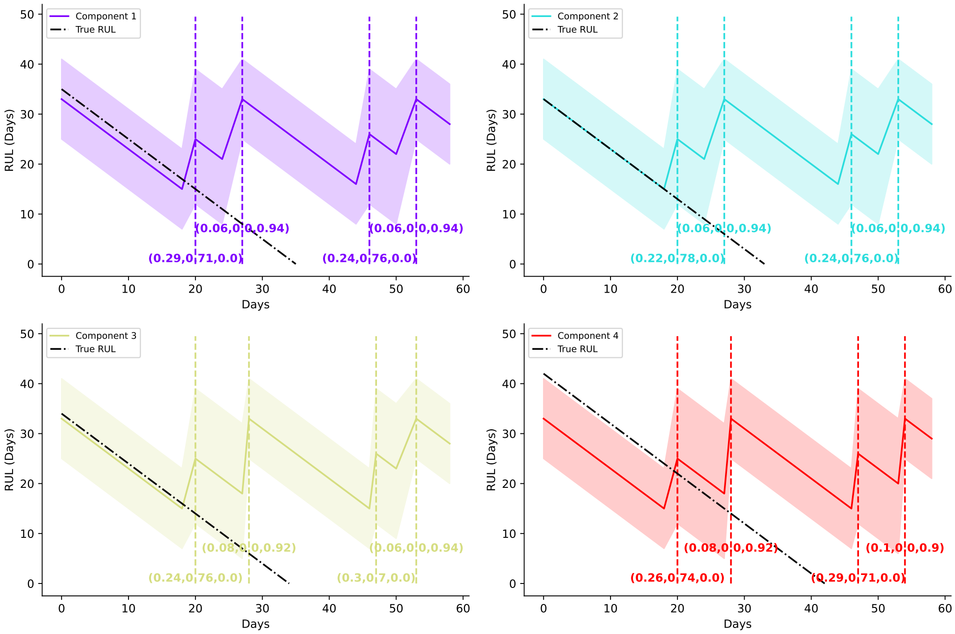
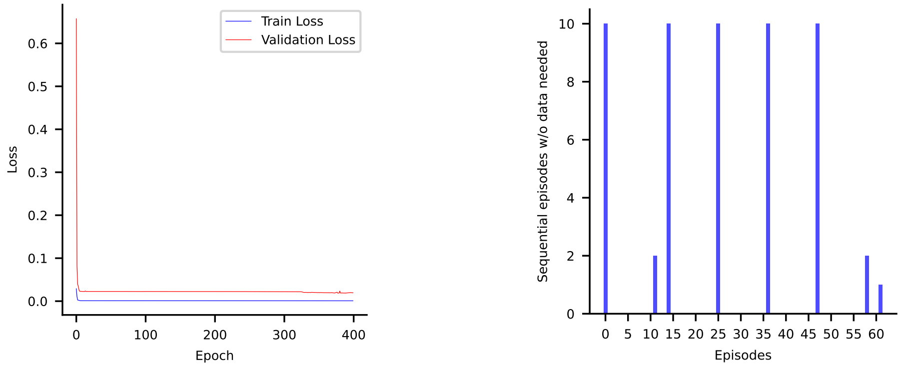


Figure 5.10.: RUL behavior of each component during the scheduling of the first episode within the horizon. For clarity, only the decisions related to maintenance actions are shown. The probabilities of the agent's decisions are illustrated as well.

5.5.3. THE ROLE OF VOI IN CHOOSING THE OPTIMAL TIME TO ACQUIRE NEW INFORMATION

Similarly to the RL agent, the VoI model was trained on a single GPU (NVIDIA GeForce RTX 2080). The entire training process for 400 epochs was approximately 40 minutes. However, generating data to train the VoI model required a huge amount of time up to 23 hours. This prolonged duration was primarily due to the uncertainty quantification and decomposition processes involved in generating data related to VoI. Each step within an episode was executed in a loop of 30 samples to compute MI, which contributed significantly to the overall time required.

The performance of the VoI model (training and validation losses) is illustrated in Figure 5.11a where both losses converge satisfactorily. Subsequently, after running the first episode related to the case study, the VoI model outputs the next 10 predicted uncertainties and the sequential number of future episodes that $VoI<0$ is measured by Equation 5.8. Following this process, the episodes where data are not required are depicted in Figure 5.11b. When the current episode in real-time is far from an 'imperfect repair' decision, the uncertainty remains mainly low, thus more episodes up to a maximum of 10 could be ignored. When such a maintenance action is approaching in real-time, the uncertainty increases, hence ignoring the next episodes should be avoided as shown in the last episodes of the same figure. Interestingly, for this case study consisting of a 60-day horizon, only 7 days were needed to acquire data, saving a vast amount of resources up to 88.3%.



(a)

(b)

Figure 5.11.: Results related to VoI model. (**a**) VoI model's training and validation loss, and (**b**) number of next sequential episodes to ignore after running the VoI model on the current episode.

Based on the VoI model and the episodes to ignore, it is important to visualize how the scheduling continues in real-time after having all the models trained. In Figure 5.12, the first five schedules of the first component are shown, considering VoI. In this regard, these schedules correspond to episodes 0, 11, 14, 25, and 36, as shown in Figure 5.11b. Importantly, it is observed that until the moment an 'imperfect repair' decision is made in real-time, the schedule stays almost the same. This shows the agent's confidence in its decisions, aligning with the low uncertainty and the multiple possible disregarded episodes. However, when an imperfect repair action is close while one or more of the components are close to the actual EOL condition, the uncertainty is increased and the schedule must be adjusted before proceeding with a new one. Thus, VoI suggests acquiring new information and updating the decisions. This is akin to how human experts update their schedules following a significant maintenance action.

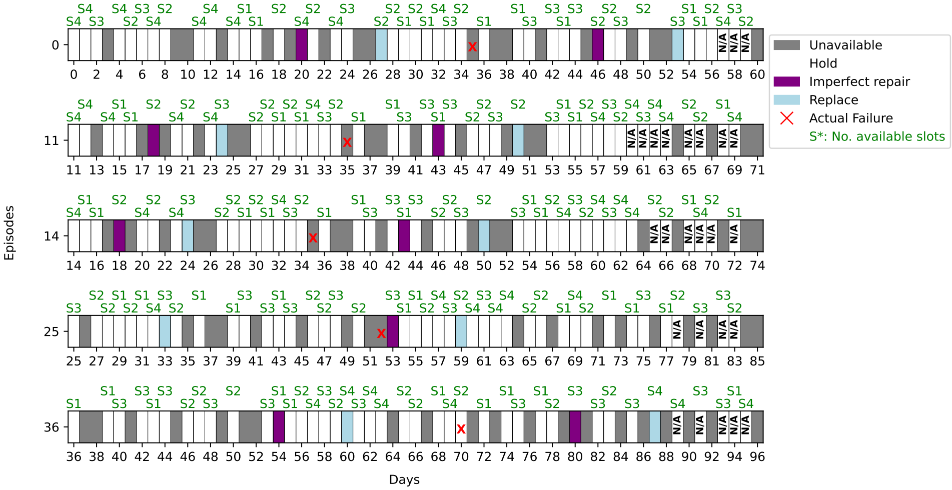
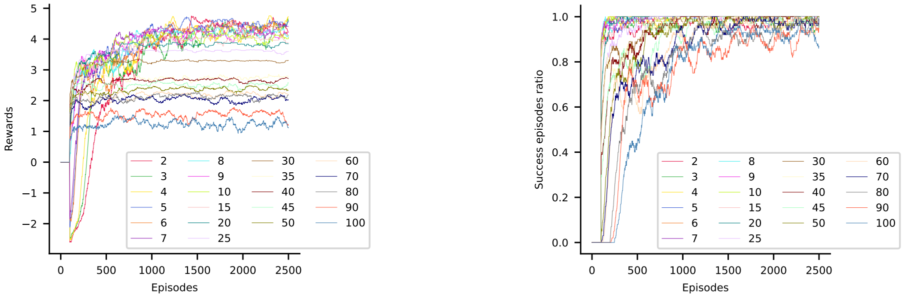


Figure 5.12.: Number of next sequential episodes to ignore after running the VoI model on the current episode. The first five schedules of the first component are shown, considering VoI. It is observed that until the moment an 'imperfect repair' decision is made in real-time, the schedule stays almost the same. This shows the agent's confidence in its decisions, aligning with the low uncertainty and the multiple possible disregarded episodes.

5.5.4. SENSITIVITY ANALYSIS BASED ON THE NUMBER OF SPECIMENS

To demonstrate the RL agent's performance limits, a sensitivity analysis was conducted for varying numbers of components. Similarly to Figure 5.7, the rewards and ratio of successful episodes for each number of components are depicted in Figure 5.13a and Figure 5.13b respectively, after taking the mean values of a rolling horizon with a rolling window of 100. From these plots, it is obvious that the performance drops as the number of components increases. Based on the plot of the ratio of successful episodes, one could agree that the performance of the agent is questionable for a multi-component system with more than 70 components. Further improvements to the framework should be made to tackle systems with a larger number of dependable components.



(a) (b)

Figure 5.13.: Training performance of the RL agent with different number of components ranging from 2 to 100. (**a**) rewards per episode and (**b**) ratio of successful episodes as the training progresses.

5.6. DISCUSSION AND CONCLUSION

The proposed PPDM framework integrates several sophisticated methodologies to address the challenges of maintenance scheduling in multi-component systems subject to imperfect repairs. This study introduces a novel approach that combines stochastic RUL estimations, imperfect repairs modeling, DRL, uncertainty quantification and decomposition, and VoI to decide the optimal time to acquire new information. The integration of stochastic RUL predictions and imperfect repairs allows for a more realistic and dynamic assessment of components' health over time. Improving the health state of each component can extend the useful life of the entire structure consisting of multiple dependent components. The primary advantage of this framework is its ability to integrate all steps of the PHM strategy while keeping each of them independent. This design provides flexibility, allowing for the use of various feature extraction, prognostic, and imperfect repair modeling techniques to input data into the PPDM framework. Consequently, one could choose to utilize all, a part, or none of the implemented in this work steps necessary to feed the PPDM framework with information.

By incorporating the uncertainty associated with RULs and repairs, the framework can more accurately predict the future condition of the components, which is crucial for making optimal maintenance decisions. The utilization of DRL enables the agent to learn optimal maintenance strategies through interaction with the environment, even with large observation and action space. A key feature of the framework is its ability to quantify and decompose uncertainty into epistemic and aleatoric through techniques such as MC dropout and MI. This allows the framework to transform the total uncertainty into decision probabilities, improving reliability and interpretability, something that is lacking particularly in DRL and scenarios where data is incomplete or noisy. Given measures of uncertainty, the VoI model is considered to determine the optimal timing for acquiring new information, thereby reducing unnecessary data collection and focusing resources on the most impactful days within the horizon.

The experimental evaluation demonstrated the framework's effectiveness in a real-world case study involving tension-tension fatigue experiments on open-hole aluminum coupons. The results showed that the framework could successfully schedule maintenance actions to extend the structure's lifecycle while satisfying operational and logistic constraints. The potential to perform scheduling given a horizon even under multiple repairs highlights the superiority of the proposed approach. Moreover, estimating the decision probabilities of the agent provides a framework that identifies its limits. Knowing when it is confident in its decisions and when it is uncertain facilitates a hybrid approach that combines the power of AI with human feedback, enhancing reliability and promoting feasibility and risk-averse policies. Finally, a sensitivity analysis has been conducted for varying numbers of components, demonstrating the performance limits of the designed framework.

Despite these key contributions, there exist two important limitations worth discussing to enhance the practical applicability of the framework. Firstly, due to the large number of hyperparameters, it is impossible with the current hardware to perform a parametric study (except for the necessary and already implemented sensitivity analysis) or to apply an automatic hyperparameter tuning, such as Bayesian Optimization. This limits the framework in a time-consuming trial-and-error approach. Secondly, the sensitivity analysis revealed that the performance of the DRL agent diminishes with an increasing number of components. This suggests that while the current model performs well for a moderate number of components, enhancements are needed to maintain efficacy in more complex scenarios. Further research should focus on developing more scalable algorithms to ensure that the agent remains effective as the number of components increases.

Overall, this study lays a strong foundation for advancing the field of PPDM. By addressing the identified limitations and building on the proposed framework, future research can further enhance its practical applicability. This framework could be a pillar for researchers to improve and promote PPDM into more advanced tasks that many industrial processes demand. Real-world validation of the model in various industrial settings will be crucial for establishing its effectiveness and ensuring its adoption in CbM practices across different sectors. Having a framework that integrates the steps of the PHM strategy independently could offer endless opportunities for constructing unique frameworks, applicable to wide or specific domains.

REFERENCES

[1] P. Komninos and D. Zarouchas. "A framework for post-prognosis decision-making utilizing deep reinforcement learning considering imperfect maintenance decisions and Value of Information". In: *Array* 27 (2025), p. 100454. ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2025.100454.

[2] K. Goebel, J. Celaya, S. Sankararaman, I. Roychoudhury, M. Daigle, and A. Saxena. *Prognostics: The Science of Making Predictions*. CreateSpace Independent Publishing Platform (April 3, 2017), Apr. 2017, p. 396. ISBN: ISBN-10: 1539074838 ISBN-13: 978-1539074830.

[3] K. Lepenioti, M. Pertselakis, A. Bousdekis, A. Louca, F. Lampathaki, D. Apostolou, G. Mentzas, and S. Anastasiou. "Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing". In: *Advanced Information Systems Engineering Workshops: CAiSE 2020 International Workshops, Grenoble, France, June 8–12, 2020, Proceedings 32*. Springer. 2020, pp. 5–16.

[4] Q. Liu, M. Dong, and F. Chen. "Single-machine-based joint optimization of predictive maintenance planning and production scheduling". In: *Robotics and Computer-Integrated Manufacturing* 51 (2018), pp. 238–247.

[5] K. S. H. Ong, D. Niyato, and C. Yuen. "Predictive maintenance for edge-based sensor networks: A deep reinforcement learning approach". In: *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE. 2020, pp. 1–6.

[6] H. Skima, C. Varnier, E. Dedu, K. Medjaher, and J. Bourgeois. "Post-prognostics decision making in distributed MEMS-based systems". In: *Journal of Intelligent Manufacturing* 30 (2019), pp. 1125–1136.

[7] I. de Pater and M. Mitici. "Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components". In: *Reliability Engineering & System Safety* 214 (2021), p. 107761. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107761. URL: https://www.sciencedirect.com/science/article/pii/S095183202100288X.

[8] A. Bousdekis and G. Mentzas. "A Proactive Model for Joint Maintenance and Logistics Optimization in the Frame of Industrial Internet of Things". In: *Operational Research in the Digital Era – ICT Challenges*. Ed. by A. Sifaleras and K. Petridis. Cham: Springer International Publishing, 2019, pp. 23–45. ISBN: 978-3-319-95666-4.

[9] Y. Hu, X. Miao, J. Zhang, J. Liu, and E. Pan. “Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization”. In: *Computers & industrial engineering* 153 (2021), p. 107056.

[10] C. Andriotis and K. Papakonstantinou. “Managing engineering systems with large state and action spaces through deep reinforcement learning”. In: *Reliability Engineering & System Safety* 191 (2019), p. 106483. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.04.036. URL: https://www.sciencedirect.com/science/article/pii/S0951832018313309.

[11] K. T. Nguyen, P. Do, K. T. Huynh, C. Bérenguer, and A. Grall. “Joint optimization of monitoring quality and replacement decisions in condition-based maintenance”. In: *Reliability Engineering & System Safety* 189 (2019), pp. 177–195. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.04.034. URL: https://www.sciencedirect.com/science/article/pii/S0951832018313097.

[12] P. Do, A. Voisin, E. Levrat, and B. Iung. “A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions”. In: *Reliability Engineering & System Safety* 133 (2015), pp. 22–32. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2014.08.011. URL: https://www.sciencedirect.com/science/article/pii/S095183201400204X.

[13] R. Meissner, A. Rahn, and K. Wicke. “Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making”. In: *Reliability Engineering & System Safety* 214 (2021), p. 107812. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107812. URL: https://www.sciencedirect.com/science/article/pii/S0951832021003331.

[14] Q. Deng and B. F. Santos. “Lookahead approximate dynamic programming for stochastic aircraft maintenance check scheduling optimization”. In: *European Journal of Operational Research* 299.3 (2022), pp. 814–833. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.09.019. URL: https://www.sciencedirect.com/science/article/pii/S0377221721007943.

[15] S.-J. Kim and C.-W. Lee. “Diagnosis of sensor faults in active magnetic bearing system equipped with built-in force transducers”. In: *IEEE/ASME Transactions on Mechatronics* 4.2 (1999), pp. 180–186. DOI: 10.1109/3516.769544.

[16] D. Lyu and S. Si. “Importance measure for K-out-of-n: G systems under dynamic random load considering strength degradation”. In: *Reliability Engineering & System Safety* 216 (2021), p. 107892. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107892. URL: https://www.sciencedirect.com/science/article/pii/S0951832021004105.

[17] J. Wang and X. Zhu. “Joint optimization of condition-based maintenance and inventory control for a k-out-of-n:F system of multi-state degrading components”. In: *European Journal of Operational Research* 290 (Aug. 2020). DOI: 10.1016/j.ejor.2020.08.016.

[18] M. C. Olde Keizer, R. H. Teunter, and J. Veldman. “Clustering condition-based maintenance for systems with redundancy and economic dependencies”. In: *European Journal of Operational Research* 251.2 (2016), pp. 531–540. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2015.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0377221715010218.

[19] P. Zhang, X. Zhu, and M. Xie. “A model-based reinforcement learning approach for maintenance optimization of degrading systems in a large state space”. In: *Computers & Industrial Engineering* 161 (2021), p. 107622. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2021.107622. URL: https://www.sciencedirect.com/science/article/pii/S036083522100526X.

[20] R. Rocchetta, L. Bellani, M. Compare, E. Zio, and E. Patelli. “A reinforcement learning framework for optimal operation and maintenance of power grids”. In: *Applied energy* 241 (2019), pp. 291–301.

[21] H. Wang, M. Gu, Q. Yu, Y. Tao, J. Li, H. Fei, J. Yan, W. Zhao, and T. Hong. “Adaptive and large-scale service composition based on deep reinforcement learning”. In: *Knowledge-Based Systems* 180 (2019), pp. 75–90. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2019.05.020. URL: https://www.sciencedirect.com/science/article/pii/S0950705119302266.

[22] Y. Liu, Y. Chen, and T. Jiang. “Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach”. In: *European Journal of Operational Research* 283.1 (2020), pp. 166–181. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2019.10.049. URL: https://www.sciencedirect.com/science/article/pii/S0377221719309014.

[23] E. Skordilis and R. Moghaddass. “A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics”. In: *Computers & Industrial Engineering* 147 (2020), p. 106600. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2020.106600. URL: https://www.sciencedirect.com/science/article/pii/S036083522030334X.

[24] P. Andrade, C. Silva, B. Ribeiro, and B. F. Santos. “Aircraft Maintenance Check Scheduling Using Reinforcement Learning”. In: *Aerospace* 8.4 (2021). ISSN: 2226-4310. DOI: 10.3390/aerospace8040113. URL: https://www.mdpi.com/2226-4310/8/4/113.

[25] Y. Zhou, B. Li, and T. R. Lin. “Maintenance optimisation of multicomponent systems using hierarchical coordinated reinforcement learning”. In: *Reliability Engineering & System Safety* 217 (2022), p. 108078. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.108078. URL: https://www.sciencedirect.com/science/article/pii/S0951832021005767.

[26] W. Cheng and X. Zhao. “Maintenance optimization for dependent two-component degrading systems subject to imperfect repair”. In: *Reliability Engineering & System Safety* 240 (2023), p. 109581. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2023.109581. URL: https://www.sciencedirect.com/science/article/pii/S0951832023004957.

[27] E. Mosayebi Omshi and A. Grall. “Replacement and imperfect repair of deteriorating system: Study of a CBM policy and impact of repair efficiency”. In: *Reliability Engineering & System Safety* 215 (2021), p. 107905. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107905. URL: https://www.sciencedirect.com/science/article/pii/S095183202100421X.

[28] F. Zhang, J. Shen, and Y. Ma. “Optimal maintenance policy considering imperfect repairs and non-constant probabilities of inspection errors”. In: *Reliability Engineering & System Safety* 193 (2020), p. 106615. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.106615. URL: https://www.sciencedirect.com/science/article/pii/S0951832019300274.

[29] X. Zhao, S. He, and M. Xie. “Utilizing experimental degradation data for warranty cost optimization under imperfect repair”. In: *Reliability Engineering & System Safety* 177 (2018), pp. 108–119. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2018.05.002. URL: https://www.sciencedirect.com/science/article/pii/S0951832017312206.

[30] T. I. Pedersen, X. Liu, and J. Vatn. “Maintenance optimization of a system subject to two-stage degradation, hard failure, and imperfect repair”. In: *Reliability Engineering & System Safety* 237 (2023), p. 109313. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2023.109313. URL: https://www.sciencedirect.com/science/article/pii/S0951832023002272.

[31] R. A. Howard. “Information Value Theory”. In: *IEEE Transactions on Systems Science and Cybernetics* 2.1 (1966), pp. 22–26. DOI: 10.1109/TSSC.1966.300074.

[32] C. Song, C. Zhang, A. Shafieezadeh, and R. Xiao. “Value of information analysis in non-stationary stochastic decision environments: A reliability-assisted POMDP approach”. In: *Reliability Engineering & System Safety* 217 (2022), p. 108034. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.108034. URL: https://www.sciencedirect.com/science/article/pii/S095183202100541X.

[33] G. Zou, A. González, and K. Banisoleiman. “A widely-applicable structural maintenance decision-analytic modelling approach assisted by information value computation”. In: *Ocean Engineering* 237 (2021), p. 109596. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2021.109596. URL: https://www.sciencedirect.com/science/article/pii/S002980182100980X.

[34] D. Straub. “Value of information analysis with structural reliability methods”. In: *Structural Safety* 49 (2014), pp. 75–85.

[35] W.-H. Zhang, J. Qin, D.-G. Lu, S. Thöns, and M. H. Faber. “VoI-informed decision-making for SHM system arrangement”. In: *Structural Health Monitoring* 21.1 (2022), pp. 37–58.

[36] A. Kamariotis, E. Chatzi, and D. Straub. “Value of information from vibration-based structural health monitoring extracted via Bayesian model updating”. In: *Mechanical Systems and Signal Processing* 166 (2022), p. 108465. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2021.108465. URL: https://www.sciencedirect.com/science/article/pii/S0888327021008104.

[37] Y.-F. Zhang, Q.-F. Zhang, and R.-H. Yu. "Markov property of Markov chains and its test". In: *2010 International Conference on Machine Learning and Cybernetics*. Vol. 4. 2010, pp. 1864–1867. DOI: 10.1109/ICMLC.2010.5580952.

[38] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. "Planning and acting in partially observable stochastic domains". In: *Artificial Intelligence* 101.1 (1998), pp. 99–134. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(98)00023-X. URL: https://www.sciencedirect.com/science/article/pii/S000437029800023X.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG].

[40] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper˙files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

[41] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans. "Bridging the Gap Between Value and Policy Based Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper˙files/paper/2017/file/facf9f743b083008a894eee7baa16469-Paper.pdf.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG].

[43] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, J. Quan, S. Gaffney, S. Petersen, K. Simonyan, T. Schaul, H. van Hasselt, D. Silver, T. Lillicrap, K. Calderone, P. Keet, A. Brunasso, D. Lawrence, A. Ekermo, J. Repp, and R. Tsing. *StarCraft II: A New Challenge for Reinforcement Learning*. 2017. arXiv: 1708.04782 [cs.LG].

[44] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, *et al.* "Dota 2 with large scale deep reinforcement learning". In: *arXiv preprint arXiv:1912.06680* (2019).

[45] S. Huang and S. Ontañón. "A closer look at invalid action masking in policy gradient algorithms". In: *arXiv preprint arXiv:2006.14171* (2020).

[46] A. Der Kiureghian and O. Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2 (2009), pp. 105–112.

[47] A. Malinin. "Uncertainty estimation in deep learning with application to spoken language assessment". PhD thesis. University of Cambridge, 2019.

[48] C. E. Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[49] T. M. Cover, J. A. Thomas, *et al.* "Entropy, relative entropy and mutual information". In: *Elements of information theory* 2.1 (1991), pp. 12–13.

[50] S. Kullback and R. A. Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[51] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in neural information processing systems* 30 (2017).

[52] S. Jain, G. Liu, J. Mueller, and D. Gifford. “Maximizing overall diversity for improved uncertainty estimates in deep ensembles”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 2020, pp. 4264–4271.

[53] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: https://proceedings.mlr.press/v48/gal16.html.

[54] D. Straub. “Value of information analysis with structural reliability methods”. In: *Structural Safety* 49 (2014). Special Issue In Honor of Professor Wilson H. Tang, pp. 75–85. ISSN: 0167-4730. DOI: https://doi.org/10.1016/j.strusafe.2013.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0167473013000611.

[55] C. HU, H. PEI, Z. WANG, X. SI, and Z. ZHANG. “A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities”. In: *Chinese Journal of Aeronautics* 31.3 (2018), pp. 514–528. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2018.01.009. URL: https://www.sciencedirect.com/science/article/pii/S1000936118300256.

[56] Z.-Q. Wang, C.-H. Hu, X.-S. Si, and E. Zio. “Remaining useful life prediction of degrading systems subjected to imperfect maintenance: Application to draught fans”. In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 802–813. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.08.016. URL: https://www.sciencedirect.com/science/article/pii/S0888327017304429.

[57] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[58] B. Zadrozny and C. Elkan. “Transforming classifier scores into accurate multiclass probability estimates”. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 694–699. ISBN: 158113567X. DOI: 10.1145/775047.775151. URL: https://doi.org/10.1145/775047.775151.

[59] M. Kull, T. M. S. Filho, and P. Flach. “Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration”. In: *Electronic Journal of Statistics* 11.2 (2017), pp. 5052–5080. DOI: 10.1214/17-EJS1338SI. URL: https://doi.org/10.1214/17-EJS1338SI.

[60] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans. “Understanding the impact of entropy on policy optimization”. In: *International conference on machine learning*. PMLR. 2019, pp. 151–160.

[61] P. Komninos, G. Galanopoulos, T. Kontogiannis, N. Eleftheroglou, and D. Zarouchas. “A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty”. In: *Expert Systems with Applications* 288 (2025), p. 127723. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2025.127723.

6
CONCLUSIONS AND RECOMMENDATIONS

6.1. CONCLUSIONS

In this thesis, the goal was to develop a holistic, end-to-end PHM framework that effectively implements the PHM strategy. However, successfully implementing such frameworks is challenging due to various complexities in their development. Therefore, the first step was to identify these challenges and associate them with the corresponding phases of the PHM strategy (refer to Table 1.1 for details). Once identified, each phase was implemented separately and seamlessly integrated with the next by addressing the associated challenges. By integrating advanced ML techniques, modular design principles, real-world applicability, and uncertainty quantification techniques, the proposed framework balances the need for precision, interpretability, and reliability while ensuring adaptability across various domains. This framework paves the way for advancements in CBM. Below are the key conclusions drawn from this research and according to the defined challenges:

- While it is feasible to use one NN-based model, here the ISTRUST model as proposed in chapter 2, that spans from data acquisition to the prognostics phase (phases 1-4), this approach is not recommended. A significant amount of information can be lost as data flows from input to output, reducing the accuracy of the predictions. Despite facing the challenge of interpretability, this loss makes it challenging to capture critical details necessary for precise prognostics, thereby diminishing the model's overall effectiveness.

- By incorporating monotonic behavior within the NN models, the generation of prognostic-related features and, hence, meaningful HIs are ensured. This demonstrates that NNs can effectively extract valuable prognostic-related insights from raw, multi-dimensional, and multi-modal data. The DSMC model was responsible to achieve this, as presented in chapter 3.

- Modeling the effects of repairs—especially when they are imperfect—is crucial for making proactive maintenance decisions. These effects can be effectively characterized by examining the Remaining Useful Life (RUL) behavior before and after a repair, as RUL captures all relevant information about a component's recovery. This approach naturally extends to scenarios involving multiple sequential repairs, as discussed in chapter 4.

- One of the key challenges in developing a holistic end-to-end PHM framework is the implementation of a decision-making algorithm that can proactively schedule practical maintenance actions, including imperfect repairs and replacements. This challenge becomes even more complex when dealing with multi-component systems. The PPDM phase presented in chapter 5 addresses this issue by offering a preliminary solution. However, even with a moderate number of components, this approach can be computationally intensive, and its performance degrades notably when scaling to larger systems (e.g., those with over 100 components). Nevertheless, it marks a significant step toward a more generalizable framework, capable of adapting to diverse problem settings with varying costs, logistical constraints, and component counts.

- The framework is highly generalizable across various domains. It can be applied to any context involving degenerative phenomena where time-series data is available and maintenance actions are defined. Importantly, these maintenance actions do not need to involve imperfect repairs. For instance, in healthcare, a "maintenance" action could be a treatment that improves a patient's health condition. As long as a model exists to capture the effects of such treatment, the framework can seamlessly transition into the PPDM phase. This versatility makes the framework adaptable to a wide range of applications.

- By accurately capturing the uncertainties introduced through imperfect repairs and RUL predictions, the DRL model was able to not only effectively schedule maintenance actions, but also to transform this uncertainty into probabilities across its decisions, increasing the levels of reliability.

- Although NNs are often considered unreliable due to their black-box nature, their interpretability across various architectures, including ISTRUST, DSMC, and DRL models, has been enhanced to address this limitation.

6.2. RECOMMENDATIONS

Despite the significant effectiveness of the PHM framework proposed in this thesis, several recommendations can enhance its capabilities further. Firstly, future research should prioritize improving performance through data scalability. This involves leveraging larger datasets, conducting more comprehensive experiments, and exploring the impact of multiple sequential imperfect repairs. In this work, we examined single repairs experimentally and simulated multiple ones. By expanding both the volume and variety of data, we can enhance the robustness and accuracy of PHM systems. Additionally, this approach will allow us to rigorously test the framework's limits, determining when performance begins to decline as sample sizes exceed certain thresholds.

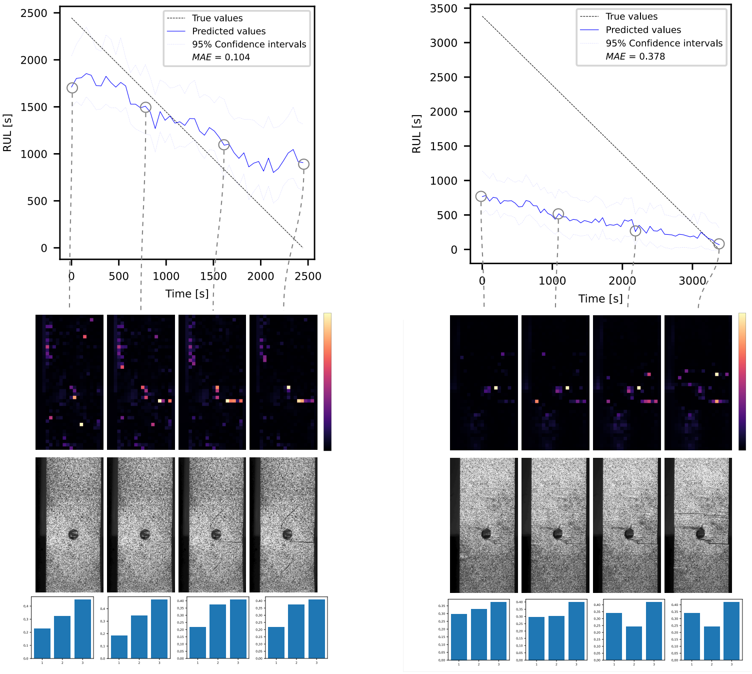
Secondly, given the variability in user demands and operational conditions, several formulations of PPDM warrant further exploration. This variability can be captured through different formulations of MDP. While this thesis addressed one general formulation based on specific assumptions, it is crucial to assess the proposed decision-making algorithm across various formulations tailored to distinct problems. This exploration will enhance the adaptability and effectiveness of the decision-making algorithm in the PPDM phase, ensuring that it can accommodate diverse operational scenarios.

Thirdly, conducting a sensitivity analysis for each phase of the PHM strategy is recommended to deepen our understanding of how various factors influence the outcomes of the PPDM, the final phase of the strategy. By identifying how outcomes are affected by different inputs and assumptions, we can gain insights into the robustness of the decision-making algorithms and enhance their reliability. Together, these recommendations will contribute to refining the PHM framework and ensuring its effectiveness in practical applications.

Finally, while we acknowledged the potential for integrating the PHM strategy with CBM, it is essential to demonstrate this integration in practical settings. Following the integration, a comprehensive performance evaluation should be conducted to assess its effectiveness in real-world applications. Although this thesis focused on a structure consisting of multiple components and aimed to extend its lifecycle by extending components' RUL via a set of maintenance actions, it is crucial to broaden our focus to entire systems composed of various structures and components. Understanding how the PHM framework interacts with CBM will yield valuable insights for enhancing maintenance practices and optimizing maintenance planning across a range of systems.

A

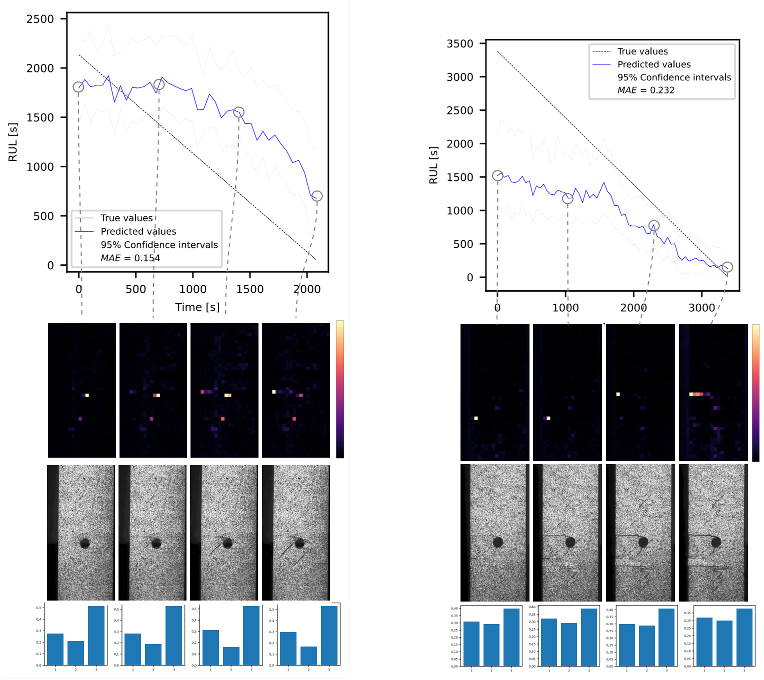
ADDITIONAL INFORMATION ISTRUST



(a) Specimen 1

(b) Specimen 2

Figure A.1.: Evolution of spatial and the corresponding temporal attention weights of testing specimens 1 and 2.



(a) Specimen 3

(b) Specimen 6

Figure A.2.: Evolution of spatial and the corresponding temporal attention weights of testing specimens 3 and 4.

B

ADDITIONAL INFORMATION REGARDING DSMC AND HSMM

B.1. EXTENDED DATA

Table B.1.: Reproducibility of the training process. Training and validation losses. The means and standard deviations were produced by training the DSMC model 10 times with random (uniform) weight initialization.

	MIMIC-III		C-MAPSS		F-MOC	
Loss type	Mean	Std.	Mean	Std.	Mean	Std.
Train reconstruction loss (first scale of Eq.3.11)	0.0461	0.0023	0.0123	0.0002	0.1315	0.0084
Validation reconstruction loss	0.0448	0.0094	0.0159	0.0083	0.1421	0.0093
Train time loss (second scale, first part of Eq.3.11)	0.0008	0.0	0.0032	0.0002	0.0421	0.0014
Validation time loss	0.0002	0.0002	0.0006	0.0002	0.0474	0.0023
Clustering loss (Eq.(3.12))	0.2485	0.0497	0.1917	0.0136	0.2923	0.0536
Validation clustering loss	0.2632	0.0502	0.2013	0.0507	0.3147	0.0619

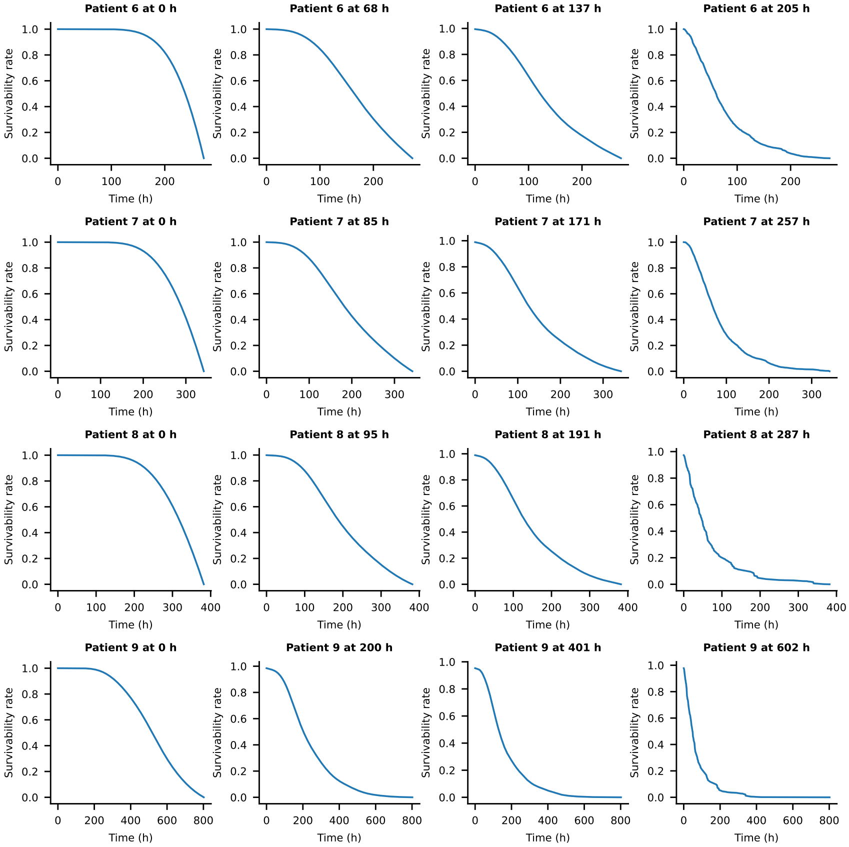


Figure B.1.: Survivability rates of a test subset of the patients at different timesteps. The timesteps are chosen at 0%, 25%, 50%, and 75% of the corresponding patient's true time of stay in the ICU until mortality.

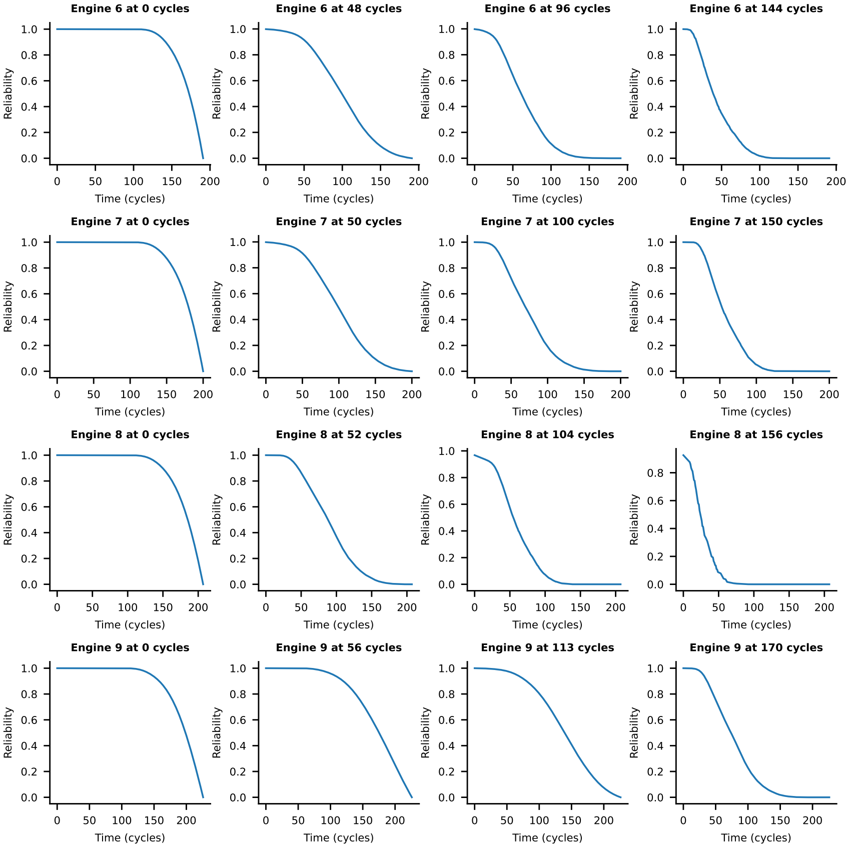
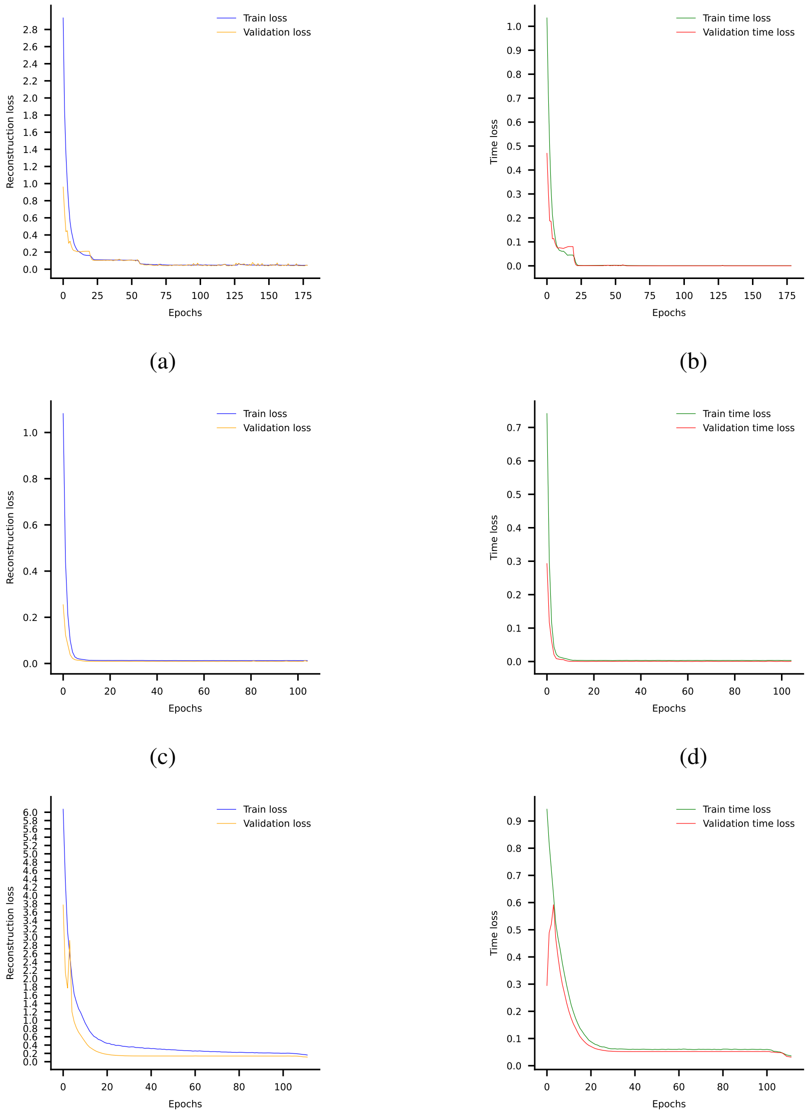


Figure B.2.: Reliability curves of a test subset of the engines at different timesteps. The timesteps are chosen at 0%, 25%, 50%, and 75% of the corresponding engine's true lifespan.



(a) (b)

(c) (d)

(e) (f)

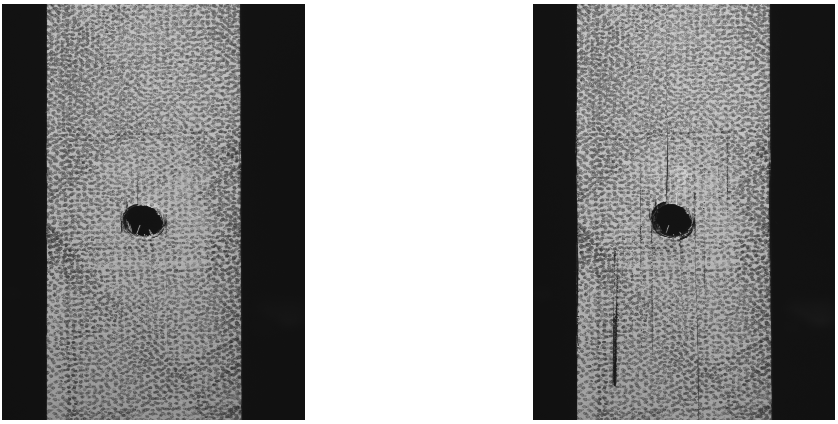
Figure B.3.: The convergence of train and validation losses, including the reconstruction loss of the input and reconstruction loss of the time feature, for the three datasets after the first stage of training of the DSMC model (AE training). (a) and (b) correspond to the MIMIC-III dataset. (c) and (d) correspond to the C-MAPSS dataset. (e) and (f) correspond to the F-MOC dataset.

B.2. IMPLEMENTATION DETAILS OF THE F-MOC DATASET

B.2.1. EXPERIMENTAL SETUP

The setup comprises a 100 kN MTS controller connected to a bench machine, an acoustic emission system, and two cameras used for DIC measurements. The cameras are capturing images of the specimen periodically every 50 seconds. To capture the acoustic emission signals generated during fatigue, an AMSY-6 Vallen Systeme GmbH, an 8-channel AE system with a sampling rate of 2 MHz, was utilized. A single broadband single-crystal piezoelectric transducer was affixed to the side of specimens using a clamping device situated between the lower grip of the fatigue machine and a safety aluminum cylinder. Ultrasound gel was applied for optimal acoustical coupling between the sensor and specimen surfaces. To ensure proper connectivity, a standard pencil lead break procedure verified the specimen-sensor connection before the fatigue test. Finally, the threshold was set at 50 dB.

The material at hand is a unidirectional prepreg tape Hexply® F6376CHTS(12K)-5-35. The laminate is manufactured using a hand lay-up of $[0/45/90/-45]_{2s}$ and is cured in an autoclave at a temperature of $180°C$ and pressure of $9\ bar$ for $120\ min$ as recommended by the manufacturer. The laminate is consequently cut to obtain specimens of $400\ mm\ x\ 45\ mm$ with an average thickness of $2.28\ mm$. Two examples of specimens can be found in Figure B.4 representing a healthy (Figure B.4a) and a damaged one (Figure B.4b). These examples correspond to the DIC part. In Figure B.5, an example of a specimen's low-level features from the recorded acoustic emissions across its lifetime is illustrated. Additional details about the used materials and the experimental setup can be found in [1].



(a) (b)

Figure B.4.: Two examples of image snapshots taken from a camera representing one specimen subject to fatigue loads. (**a**) Healthy. (**b**) Severely damaged.

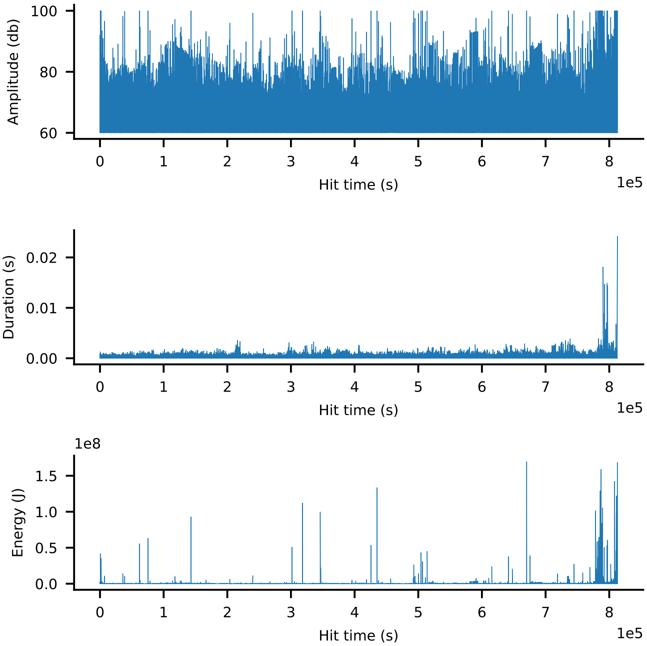


Figure B.5.: A specimen's low-level features representing amplitude, duration, and energy, as extracted from the AMSY-6 Vallen Systeme GmbH across its lifetime. These features are displayed along the y-axis, while the hit time feature is depicted along the x-axis.

B.2.2. SYNCHRONIZATION OF ACOUSTIC EMISSION AND DIC DATA

Considering a trade-off between memory efficiency and training accuracy, the decision was made to compose each time frame from 6 ($L_{window}=6$) sequential images captured at 50-second intervals by one camera. Considering that no damage is detected in any of the specimens during the initial approximately 8000 seconds, emphasis is directed toward images captured after 6000 seconds and beyond, with the preceding ones being disregarded, to face potential memory issues.

Opting for a 50% overlap, each successive frame incorporates the last 3 images from the preceding frame along with the subsequent 3 new ones, resulting in a step size of $S=3$. To synchronize these images with the acoustic emission signals, a window technique was employed based on the duration of each time frame. By determining the timestamps of the initial and final images within each frame, we matched these timestamps with the corresponding points in the acoustic emission signal to establish the signal's duration. Subsequently, a synthetic constant length for the signals was set as a baseline, which is the average of the corresponding signal lengths representing each frame. This baseline length is valued at 755 data points for each frame. Utilizing this baseline, a moving average filter was applied to adjust all signals to this standardized length. This filter was applied independently to each low-level feature extracted from each signal. If the length of one signal exceeds the baseline, the filter selectively eliminates data points; conversely, it interpolates data points when the signal length is shorter.

B.3. HSMM DEFINITION AND RE-ESTIMATION PROCESS

An HSMM is a stochastic model that describes a system evolving through time. The state of the system is hidden from the observer and can only be inferred from the observations emitted by the system in a probabilistic manner. HSMMs are extensions of HMM that introduce a variable duration for each state, thus allowing the underlying process to be semi-Markov. Therefore, the assumption of one emitted observation per state is relaxed since the number of observations emitted depends on the time spent in each state (duration d). An HSMM is then defined by the number of states (N), the number of distinct observations (M), transition probabilities between states (A), the probability distribution of observations in each state (B), and the initial state (π). The complete parameter set of the model is denoted as $\lambda=(A,B,\pi)$.

- N: number of states. Individual states are denoted as $S=\{S_1,S_2,...,S_N\}$, and the state at time t as q_t.
- State transition: the state transition probability distribution is denoted as $A=\{a_{ij}\}$, where $a_{ij}=P[q_{t+1}=S_j|q_t=S_i]$.
- M: The observation process is modelled with a continuous probability distribution (Gaussian), therefore the indicator space consists of all the real numbers, so $M\rightarrow\infty$ and $V=\{v\in\Re\}$.
- Observation distribution: The probability distribution for the observations is assumed to be a Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$. Therefore, the B parameters are the parameters of the μ and σ^2 observation vectors, with each row representing the hidden state and the values representing the mean and standard deviation of the $\mathcal{N}$ probability distribution that each state produces v observation.
- D: number of integer values in the space $\{1,2,...,D\}$ that the state duration random variable d obtains. Therefore, the degradation process is modelled in a non-parametric distribution.
- Initial state: the initial state distribution $\pi=\{\pi_i\}$ where $\pi_i=P[q_1=S_i]$ with $1\leq i\leq N$.

In order to characterize the degradation and observation processes within a system, three fundamental problems must be addressed. The three fundamental problems are inherent to

HSMMs and are key aspects of understanding utilizing them to model real-life situations effectively.

- **Likelihood**: Calculate the likelihood $P(O|\lambda)$ given the model λ and an observation sequence O.
- **Decoding**: Estimate the sequence of hidden states Q that best explains the observations, given the model λ and an observation sequence O.
- **Learning**: Estimate the transition and emission matrix that best describes the degradation process of the observation sequence O.

The Forward-Backward algorithm is used for the first problem, which is the **calculation of the likelihood**. However, to calculate the likelihood only the forward variable is needed. The forward variable is defined as $\alpha_t(i)=P(O_1 O_2 ... O_t, q_t=S_i|\lambda)$. This is the probability of a partial observation sequence (until time t) and state S_i at time t, given the model λ. The forward variable can be solved inductively as follows:

1. Initialization: Here the forward probabilities are initialized as the joint probability of state S_i and initial observation O_1.

$$\alpha_1 = \pi_i * b_i(O_1), \quad \text{where } 1 \leq i \leq N \tag{B.1}$$

2. Induction:

$$\begin{gathered}\bar{\alpha}_{t+1}(i,d) = \sum_{j=1, j\neq i}^{N} \left(\alpha_{t+1-d}(j) * a_{ji}\right) * \prod_{\tau=t-d+2}^{t+1} (b_i(O_\tau)) * p_i(d) \\ \alpha_{t+1}(i) = \sum_{d=1}^{D} \bar{\alpha}_{t+1}(i,d), \\ \text{where } 1 \leq t \leq T-1 \text{ and } 1 \leq i \leq N\end{gathered} \tag{B.2}$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{B.3}$$

The second problem, **decoding**, can be solved with the Viterbi algorithm. For this algorithm, it is necessary to define the variable $\delta_t(i)$, which is the highest probability for a path at time t ending at state q_i, and the variable $\psi_t(i)$, which is the path that has the highest probability until state q_i at time t.

1. Initialization:

$$\delta_1(i) = \pi_i * b_i(O_1), \quad \text{where } 1 \leq i \leq N \tag{B.4}$$

$$\psi_1(i) = 0 \tag{B.5}$$

2. Induction:

$$\delta_t(j,d) = \max_{\substack{1\leq i\leq N,\, i\neq j,\\ 1\leq d'\leq D}} \delta_{t-d}(i,d') * a_{ij} * \prod_{\tau=t-d+1}^{t+1} b_j(O_\tau), \quad \text{where } 1\leq t\leq T,\ 1\leq j\leq N,\ 1\leq d\leq D \tag{B.6}$$

3. Termination:

$$P(Q,O) = \max_{1\leq i\leq N} \delta_T(i,d) \tag{B.7}$$

$$\hat{q}_T = \arg\max_{\substack{1\leq i\leq N,\\ 1\leq d'\leq D}} \delta_T(i) \tag{B.8}$$

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad \text{where } t = T-1, T-2, \ldots, 1 \tag{B.9}$$

In the termination step, Equation B.8 is the best last state, and Equation B.9 represents the backtracking used to get the best states in each time step.

Finally, the last problem of the HMM, **the learning problem**, is solved by the Expectation-Maximization (E-M) algorithm. The E-M algorithm utilizes both the forward and backward variables of the Forward-Backward algorithm and the parameters are updated to maxize the likelihood propability as defined in Equation 3.15. The forward variable α is calculated with the iterative algorithm shown in Equation B.1, Equation B.2 and Equation B.3. The backward variable is defined as $\beta_t(i){=}P(O_{t+1}O_{t+1}\ldots O_T|q_T{=}S_i,\lambda)$, so $\beta_t(i)$ is the probability of the future observations from $t{+}1$ given the current state is S_i and the model λ. In the same manner as the forward variable, the backward variable can be solved inductively.

1. Initialization:

$$\beta_T(i) = 1, \quad \text{where } 1\leq i\leq N \tag{B.10}$$

2. Induction:

$$\beta_t(i) = \sum_{\substack{j=1,\\ j\neq i}}^{N} \left(a_{ij} * \sum_{d=1}^{D} \left(p_i(d) * \beta_{t+d}(i) * \prod_{\tau=t+1}^{t+d} b_j(O_\tau) \right) \right), \quad \text{where } 1\leq i\leq N \tag{B.11}$$

3. Termination:

$$P(O|\lambda) = \sum_{j=1}^{N} \pi_j * b_j(O_1) * \beta_1(j) \tag{B.12}$$

For the E-step, the variables $\gamma_t(j)$, which is the probability of being in a state j at time t given the observations and the model parameters, and $\xi_t(i,j)$, which is the probability of being in a state i at time t and state j at time $t{+}1$, given the observations and the model parameters and the auxiliary variable $\eta_t(j,d)$. Both $\gamma_t(j)$ and $\xi_t(i,j)$ can be written in terms of the forward α and backward β probabilities.

$$\gamma_t(j) = P(q_t = j|O,\lambda) = \frac{\alpha_t(j) * \beta_t(j)}{P(O|\lambda)} \tag{B.13}$$

$$\xi_t(i,j) = \frac{\sum_{d=1}^{D} \left(\alpha_t(i) * a_{ij} * \beta_{t+d}(j) * \prod_{\tau=t+1}^{t+d} b_j(O_\tau) \right)}{\sum_{j=1}^{N} \alpha_t(j) * \beta_t(j)} \tag{B.14}$$

$$\eta_t(j,d) = \bar{\alpha}_t(j,d) * \beta_t(j) \tag{B.15}$$

In the M-step, the variables $\gamma_t(j)$ and $\xi_t(i,j)$ are used to re-estimate the new probabilities for the transition A, the emission parameters (μ, σ^2) and the duration probability matrix $p_j(d)$, we define the following:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)} \tag{B.16}$$

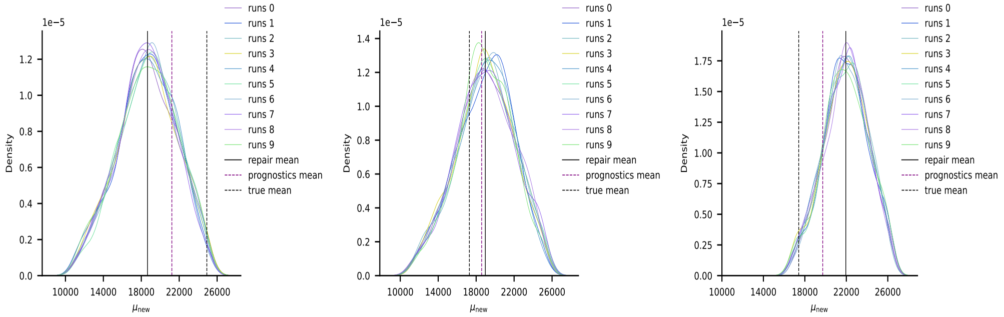
$$\hat{\sigma^2}(j) = \frac{\sum_{t=1}^{T} \gamma_t(j) * (O_t - \mu(j))^2}{\sum_{t=1}^{T} \gamma_t(j)} \tag{B.17}$$

$$\hat{p}(j,d) = \frac{\sum_{t=1}^{T} \eta_t(j,d)}{\sum_{d=1}^{D} \sum_{t=1}^{T} \eta_t(j,d)} \tag{B.18}$$

In the particular case of prognostics, some assumptions are made. First, the last state is not hidden but observable and represents failure. Second, in the failure state, only one observation value is emitted. Third, only left-to-right transitions are allowed and the transition can occur only to a neighbor's hidden state. Fourth, the initial state is always the first state. After the parameters are re-estimated Equation 3.16, Equation 3.17 and Equation 3.18 are utilized to calculate the RUL.

C

ADDITIONAL RESULTS OF IMPERFECT REPAIR MODEL



(a) (b) (c)

Figure C.1.: Posterior predictive distributions of the random variable μ_{new} using non-informative (and wider) prior distributions for specimens 6 (left), 7 (middle), and 8 (right).

D

ADDITIONAL INFORMATION PPDM

D.1. PROXIMAL POLICY OPTIMIZATION

Consider a stochastic policy $\pi_\theta{:}S{\times}A{\rightarrow}[0{,}1]$, parameterized by a parameter vector θ, which assigns probabilities to each of the available actions given a state. For a finite horizon, the goal is to maximize the expected discounted return of the policy:

$$J = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \gamma^t r_t \left(s_t, s_{t+1} \right) \right] \tag{D.1}$$

Where $\tau{=}(s_0{,}\alpha_0{,}r_0{,}\ldots{,}s_{T-1}{,}\alpha_{T-1}{,}r_{T-1})$ is a trajectory related to the environment's variables. Policy gradient algorithms aim to derive the gradient of the expected discounted return $\nabla_\theta J$ concerning the policy parameter θ as their fundamental concept. According to Sutton & Barto [2] and the introduced advantage function $A(s_t{,}\alpha_t)$ [3], the proposed policy gradient estimate to the objective J is as follows:

$$\nabla_\theta J = \mathbb{E}_{\tau\sim\pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\alpha_t|s_t) A(s_t, \alpha_t) \right] \tag{D.2}$$

Where:

$$A(s_t, \alpha_t) = Q(s_t, \alpha_t) - V(s_t, \alpha_t) \tag{D.3}$$

This is the advantage function, representing the advantage of taking action α_t in state s_t compared to the average action value. Then the parameters θ are updated:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J \left(\pi_{\theta_k} \right) \tag{D.4}$$

In PPO, the advantage function is often estimated using the generalized advantage estimation (GAE) technique [4], which combines rewards and value estimates:

$$A^{GAE}(s_t, a_t) = \sum_{\tau=0}^{\infty} (\gamma \lambda_{GAE})^{\tau} \left(r_{t+\tau} + \gamma V(s_{t+\tau+1}) - V(s_{t+\tau})\right) \tag{D.5}$$

PPO parameterizes the policy using an ANN with parameters θ. This network outputs probabilities for selecting each action given a state, denoted as $\pi_\theta(a|s)$. Moreover, PPO introduces a surrogate objective function that constrains the policy update to be close to the current policy. This helps prevent large policy updates that can lead to instability. The surrogate objective is defined as:

$$L(\theta, s_t, a_t) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{old}}(a_t \mid s_t)} A^{GAE}(s_t, a_t) \tag{D.6}$$

Where π_{old} is the probability of selecting action a_t given state s_t under the old policy at step t. To further stabilize training and prevent large policy updates, PPO introduces a clipped surrogate objective. Instead of directly maximizing the surrogate objective, PPO maximizes a clipped version of the surrogate objective, which is bounded by a specified clipping parameter ϵ :

$$L_{clip}(\theta,s_t,a_t)=\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}A^{GAE}(s_t,a_t),\ \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)},1-\epsilon,1+\epsilon\right)A^{GAE}(s_t,a_t)\right) \tag{D.7}$$

PPO updates the policy parameters θ by maximizing the clipped surrogate objective typically using stochastic gradient ascent with Adam optimizer. This involves computing the gradient of the clipped surrogate objective with respect to θ and updating θ in the direction that increases the objective as follows:

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{s_t, a_t \sim \pi_{\text{old}}} \left[L_{clip}(\theta, s_t, a_t)\right] \tag{D.8}$$

To further stabilize training, PPO typically performs multiple optimization epochs on the collected data before updating the policy. This helps to reduce the variance in gradient estimates and improve sample efficiency. Finally, PPO can also update the parameters of the value function network to improve value estimation by regression on MSE given the equation below:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - R_t\right)^2 \tag{D.9}$$

Where $\mathcal{D}_k$ is a dataset containing older steps stored in the buffer, $V_\phi(s_t)$ is the value function approximated by an ANN with parameters ϕ. The reward R_t at step t comes from these collections stored in the buffer.

PPO is designed to provide stable and reliable training by constraining policy updates. It typically requires fewer samples to achieve good performance compared to some other policy gradient methods and performs well across a wide range of tasks and environments, making it a popular choice in the RL community.

D.2. MUTUAL INFORMATION THEORY

For a dataset $D{=}\{X{,}Y\}$, given the Shannon's entropy as presented in Equation 5.4 the MI equation can be derived (Equation 5.5). The marginal distribution $p(y)$ of Equation 5.5 is given by:

$$p(y) = \int_D p(x,y)dx \tag{D.10}$$

MI quantifies how much information x provides about y. When MI is substantial, it indicates that x carries significant information about y, consequently leading to high aleatoric uncertainty.

Applying the formulations inside an ANN in discrete domains where a Softmax activation function usually prevails, necessitates the minimization of the expected negative log-likelihood to update the ANN's parameters θ:

$$L^{NLL}(x,y,\theta) = -\sum_{c=1}^{K} p(y=c|x,\theta) ln\, p(y=c|x,\theta) \tag{D.11}$$

where c is the corresponding class (in a classification task) or action (in an RL task). It has been shown in [5] that the minimization of the expected negative log-likelihood is equivalent to minimizing the expected KL-divergence between the ANN's outcome $p(y|x{,}\theta)$ and the true conditional distribution $p(y|x)$. Therefore, the expected negative log-likelihood is lower-bounded by the expected entropy $H[p(y|x)]$ that is a measure of the aleatoric uncertainty as follows:

$$L^{NLL}(x{,}y{,}\theta){=}\mathbb{E}_{p(x)}[KL[p(y|x)||p(y|x{,}\theta)]{+}H[p(y|x)]]{\geq}\mathbb{E}_{p(x)}[H[p(y|x)]] \tag{D.12}$$

Finally, as proven in [5], the total uncertainty can be decomposed into epistemic and aleatoric in deep learning via the MI theory utilizing M number of ANNs (deep ensembles), which leads to Equation 5.6.

D.3. PROOFS

Lemma 1. *Given an agent that chooses over three possible decisions, the measured total uncertainty produced by Shannon's entropy and MC dropout is upper bounded to* $\log 3$, *s.t.:*

$$U = \frac{1}{T} H\left[\sum_{i=1}^{T} p(y_i)\right] \leq log3 \tag{D.13}$$

Where U *is the total uncertainty.*

Proof. According to Shannon's entropy and Jensen's inequality [6], for k decisions we have:

$$-\sum_{y=1}^{k}[p_y log(p_y)] = \sum_{y=1}^{k}\left[log(\frac{1}{p_y})\right] \leq logk \tag{D.14}$$

The given total uncertainty from Equation 5.15 for 3 decisions is:

$$U = \frac{1}{T} H\left[\sum_{i=1}^{T} p(y_i)\right] = -\sum_{y=1}^{3}\left[\frac{1}{T}\sum_{i=1}^{T} p(y_i) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y_i)\right)\right] \tag{D.15}$$

Setting $P_y = \frac{1}{T}\sum_{i=1}^{T} p(y_i)$ then Equation D.15 becomes:

$$U = \frac{1}{T} H\left[\sum_{i=1}^{T} p(y_i)\right] = -\sum_{y=1}^{3}\left[P_y log(P_y)\right] \leq log3 \tag{D.16}$$

□

Lemma 2. *Given an agent that chooses over three possible decisions, the probability threshold* p_{thresh} *based on which a decision is considered acceptable has a connection with the total uncertainty, s.t.:*

$$U \leq -p_{thresh} log(p_{thresh}) - (1-p_{thresh}) log\left(\frac{1-p_{thresh}}{2}\right) \tag{D.17}$$

Proof. Since only one decision should be made for each component, we care only about the maximum probability $p(y_i^{max}) = max(p(y_i), i = \{1,2,3\} \geq p_{thresh}$. The worst case scenario is $p(y_i^{max}) = p_{thresh}$. Let us assume that the first decision is the maximum, thus $p(y_1) = p_{thresh}$. Then, based on Equation D.16, the maximum corresponding acceptable uncertainty is:

$$\begin{aligned} U = -\sum_{y=1}^{3}\left[\frac{1}{T}\sum_{i=1}^{T} p(y_i) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y_i)\right)\right] = \\ = -\frac{1}{T}\sum_{i=1}^{T} p(y^i_{thresh}) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y^i_{thresh})\right) - \sum_{y=2}^{3}\left[\frac{1}{T}\sum_{i=1}^{T} p(y_i) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y_i)\right)\right] = \\ = -\frac{1}{T} T p_{thresh} log\left(\frac{1}{T} T\right) p_{thresh} - \sum_{y=2}^{3}\left[\frac{1}{T}\sum_{i=1}^{T} p(y_i) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y_i)\right)\right] = \\ = -p_{thresh} log(p_{thresh}) - \sum_{y=2}^{3}\left[\frac{1}{T}\sum_{i=1}^{T} p(y_i) log\left(\frac{1}{T}\sum_{i=1}^{T} p(y_i)\right)\right] \end{aligned} \tag{D.18}$$

Setting again $P_y = \frac{1}{T}\sum_{i=1}^{T} p(y_i)$ we have:

$$U = -p_{thresh} log(p_{thresh}) - \sum_{y=2}^{3} P_y log(P_y) \tag{D.19}$$

If the first decision corresponds to p_{thresh} then $P_1 = p_{thresh}$ and $\sum_{y=2}^{3} P_y = 1 - p_{thresh}$. According to Jensen's inequality the entropy of the terms P_2 and P_3 is maximized when $P_2 = P_3$, hence:

$$P_2 + P_3 = 1 - p_{thresh} \Leftrightarrow P_2 = P_3 = \frac{1-p_{thresh}}{2} \tag{D.20}$$

Setting Equation D.20 to the second term ofEquation D.19 we have:

$$\begin{aligned} U = -p_{thresh} log(p_{thresh}) - -\sum_{y=2}^{3} \frac{1-p_{thresh}}{2} log(\frac{1-p_{thresh}}{2}) = \\ = -p_{thresh} log(p_{thresh}) - (1-p_{thresh}) log\left(\frac{1-p_{thresh}}{2}\right) \end{aligned} \tag{D.21}$$

The same equation will be given assuming any of the probabilities P_1,P_2,P_3 equals to p_{thresh}. If any of the maximum P_i is less than p_{thresh}, then the total uncertainty in Equation D.21 is always smaller than the above expression. ☐

D.4. SUPPLEMENTARY RESULTS

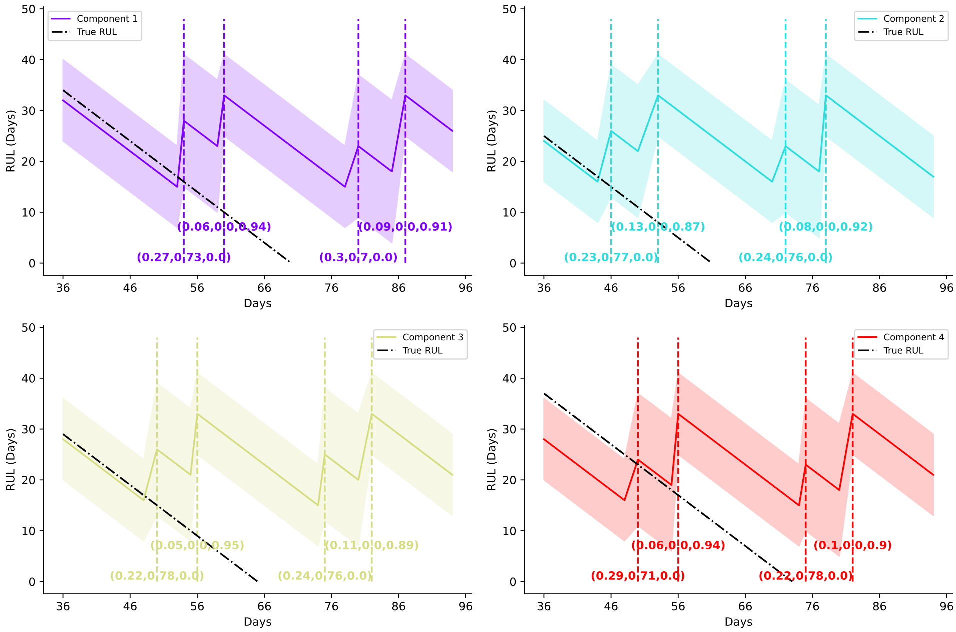


Figure D.1.: RUL behavior of each component during the scheduling of day 36 within the horizon. For clarity, only the decisions related to maintenance actions. Additionally, the probabilities of making a decision by the agent are shown, corresponding to ‘hold’, ‘imperfect repair’, and ‘replace’ respectively.

ACKNOWLEDGEMENTS

Completing my PhD has been a challenging yet deeply fulfilling journey, and I owe a great deal of gratitude to the many people who have supported and encouraged me along the way.

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Dimitrios Zarouchas. His continuing guidance, insightful feedback, and constant encouragement have been invaluable in shaping both this research and my development as a scholar. I am particularly grateful for his supportive, easy-going approach, especially during the most challenging phases of this work. Dimitri, your unique humor can bring lightness even to the toughest moments. Do not ever change that for anyone.

I would also like to thank the members of my dissertation committee for their feedback and time. Each of you has provided unique perspectives that have enriched this research, and I am deeply appreciative of your input and encouragement.

To my research team in the department, thank you for creating an inspiring and collaborative environment. I am especially grateful to Morteza, Thanos, and Nick for the countless discussions and brainstorming sessions that made this journey not just bearable but truly interesting. My heartfelt appreciation goes to George, whose invaluable help with the experimental setup brought my research ideas to life. George, you were also the first to offer support during a serious injury, and your kindness was really helpful during that challenging time. Special thanks to my teammate, Stratos, whose assistance with the annoying Greek bureaucracy saved me time and energy, allowing me to focus on my work. Strato, thank you also for your delightfully terrible humor, which provided much-needed cheerful breaks. To my friend and former teammate, Christos, thank you for the endless conversations and motivation, especially during those chaotic early days of the PhD. You taught me a lesson that reaches far beyond academia, which is how to find joy in every moment of my life. Finally, to the rest of my research team, Iordanis, Mariana, and Ferda, I am truly grateful to have shared this journey with such talented and compassionate people. Each person on this team has contributed to this experience in ways I will always treasure.

My sincere appreciation also goes to the administrative and technical staff at the Faculty of Aerospace Engineering for their assistance and support throughout my time here. Gemma, dank u wel for your invaluable help in handling endless Dutch bureaucratic tasks.

On a more personal note, I am truly thankful to my friends. Your constant encouragement and gentle reminders to balance work with life have meant so much to me.

Last but not least, to my family and my closest people, the most important people in my life, thank you for encouraging me to follow my dreams and inspiring me to keep learning. Your love and support have been the foundation of everything I do. Σας ευχαριστώ από τα βάθη της καρδιάς μου!

To everyone who has been part of this journey, whether directly or indirectly, thank you for helping me reach this milestone. This thesis would not have been possible without each of you.

Thank you all.

Panagiotis Komninos, November 2025

CURRICULUM VITæ

Panagiotis Komninos

22-02-1997	Born in Patras, Greece.

EDUCATION

2015–2020	Dipl. Ing (Integrated Master) in Mechanical Engineering & Aeronautics University of Patras
2021-2025	PhD. Aerospace Engineering Delft University of Technology *Thesis:* An end-to-end framework for Prognostics and Health Management *Promotor:* Dr. D. Zarouchas

AWARDS

2012	Gold Medal, Team Championship of Greece, Table Tennis
2017	Second place in “People of the Year of Western Greece” with ROBOTICS GROUP
2024	Best Work Award issued by Second International Conference for CBM in Aerospace

LIST OF PUBLICATIONS

1. P. Komninos, T. Kontogiannis, N. Eleftheroglou, and D. Zarouchas. “A robust generalized deep monotonic feature extraction model for label-free prediction of degenerative phenomena”. In: *Data-Centric Engineering* (2025). Submitted for Publication (Second Revision).

2. P. Komninos and D. Zarouchas. “A framework for post-prognosis decision-making utilizing deep reinforcement learning considering imperfect maintenance decisions and Value of Information”. In: *Array* 27 (2025), p. 100454. ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2025.100454

3. P. Komninos, G. Galanopoulos, T. Kontogiannis, N. Eleftheroglou, and D. Zarouchas. “A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty”. In: *Expert Systems with Applications* 288 (2025), p. 127723. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2025.127723

4. P. Komninos, A. Verraest, N. Eleftheroglou, and D. Zarouchas. “Intelligent fatigue damage tracking and prognostics of composite structures utilizing raw images via interpretable deep learning”. In: *Composites Part B: Engineering* 287 (2024), p. 111863. ISSN: 1359-8368. DOI: https://doi.org/10.1016/j.compositesb.2024.111863

5. X. Yang, S. Cantero-Chinchilla, M. Moradi, P. Komninos, C. Fang, Y. Liao, P. Kundu, D. Zarouchas, and D. Chronopoulos. “Damage imaging in structural health monitoring with fine-tuned conditional diffusion model”. In: *Mechanical Systems and Signal Processing* 236 (2025), p. 112996. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2025.112996

6. D. V. Srinivasan, M. Moradi, P. Komninos, D. Zarouchas, and A. P. Vassilopoulos. “A generalized machine learning framework to estimate fatigue life across materials with minimal data”. In: *Materials & Design* 246 (2024), p. 113355. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2024.113355

7. M. Moradi, P. Komninos, and D. Zarouchas. “Constructing explainable health indicators for aircraft engines by developing an interpretable neural network with discretized weights”. In: *Applied Intelligence* 55 (2024), p. 143. ISSN: 1573-7497. DOI: 10.1007/s10489-024-05981-2

8. C. Nastos, P. Komninos, and D. Zarouchas. “Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods”. In: *Composite Structures* 311 (2023), p. 116815. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2023.116815

9. M. Moradi, P. Komninos, and D. Zarouchas. “Interpretable Neural Network with Limited Weights for Constructing Simple and Explainable HI using SHM Data”. In: *Proceedings of the Annual Conference of the PHM Society 2022*. Vol. 14. Annual Conference of the PHM Society, 2022. ISBN: 978-1-936263-37-0. DOI: https://doi.org/10.36001/phmconf.2022.v14i1.3185

TU Delft
Delft
University of
Technology