A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty

Komninos, P.; Galanopoulos, G.; Kontogiannis, T.; Eleftheroglou, N.; Zarouchas, D.

DOI
10.1016/j.eswa.2025.127723

Publication date
2025

Document Version
Final published version

Published in
Expert Systems with Applications

Citation (APA)
Komninos, P., Galanopoulos, G., Kontogiannis, T., Eleftheroglou, N., & Zarouchas, D. (2025). A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty. *Expert Systems with Applications*, *288*, Article 127723.
https://doi.org/10.1016/j.eswa.2025.127723

Expert Systems With Applications 288 (2025) 127723

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A Bayesian inference-based framework for modeling imperfect post-repair behavior of remaining useful life under uncertainty

P. Komninos [a],[*], G. Galanopoulos [a], T. Kontogiannis [a,b], N. Eleftheroglou [a,b], D. Zarouchas [a]

[a] *Center of Excellence in Artificial Intelligence for Structures, Prognostics & Health Management, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands*

[b] *intelligent System Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands*

ARTICLE INFO

Dataset link: https://data.4tu.nl/datasets/b7d8031f-95a6-4ccc-8c08-802e694a0f40

Keywords:
Imperfect repair
Condition-based maintenance
Prognostics & health management
Remaining useful life
Structural damage analysis

ABSTRACT

Maintenance decisions often involve choosing between replacement and repair. The shortage of essential replacement parts has led to increased exploration of repair methodologies. However, repairs are often imperfect, leading to additional uncertainties in predicting the component's future condition. Existing approaches in the literature for modeling imperfect repairs struggle when repair dynamics are unknown requiring a large amount of data to be reliable. Furthermore, current methods are very task-specific, which limits the optimization of maintenance planning of varying components. This research addresses these challenges by conceptualizing imperfect repair effects as a stochastic increase in Remaining Useful Life (RUL). An existing deep learning model extracts prognostic-related features that can be utilized by any prognostic model to estimate RUL based on sensor data. Then, the proposed imperfect repair model predicts the RUL increase post-repair. This method offers three key benefits: (i) proactive post-repair assessment for improved maintenance, (ii) a data-driven repair model compatible with existing prognostic models, and (iii) flexibility in adapting to different repair techniques. Evaluation of the proposed model is conducted through tension-tension fatigue experiments on aerospace-grade aluminium specimens subject to imperfect repair. Results demonstrate the model's ability to accurately estimate the post-repair stochastic RUL increase.

1. Introduction

Maintaining primary load-bearing structures and components is vital to ensuring their safety, functionality, and durability. These structures, present in aircrafts, 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) and, currently, a further shift towards Condition-based maintenance (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' (Verhagen et al., 2023). 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 Carlo and Arleo (2017), 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

[*] Corresponding author.
E-mail addresses: P.Komninos@tudelft.nl (P. Komninos), G.Galanopoulos@tudelft.nl (G. Galanopoulos), A.Kontogiannis@tudelft.nl (T. Kontogiannis), n.eleftheroglou@tudelft.nl (N. Eleftheroglou), D.Zarouchas@tudelft.nl (D. Zarouchas).

https://doi.org/10.1016/j.eswa.2025.127723
Received 22 November 2024; Received in revised form 11 March 2025; Accepted 12 April 2025
Available online 21 May 2025

Nomenclature

α_1	Lower bound of Uniform distribution for μ_{mean}
α_2	Lower bound of Uniform distribution for σ^2_{mean}
α_{mean}	Lower bound of Truncated Normal distribution
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
μ	Mean
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
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
$ABAO$	As-Bad-As-Old
AE	Autoencoder
$AGAN$	As-Good-As-New
ANN	Artificial Neural Networks
CBM	Condition-based Maintenance
$CFRP$	Carbon Fiber Reinforced Polymer
D	Dataset
d	Number of parameters to train with MCMC
DIC	Digital Image Correlation
$DSMC$	Deep Soft Monotonic Clustering
EOL	End of Life
FC	Fully Connected
$GBDT$	Gradient Boosting Decision Tree
$HSMM$	Hidden-Semi Markov Models
KL	Kullback–Leibler
$LSTM$	Long Short-Term Memory
$MCMC$	Markov Chain Monte Carlo
MSE	Mean Squared Error
N	Number of samples
N_{traj}	Number of trajectories
$NUTS$	No U-Turn Sampler
$P(.)$	Probability
PDF	Probability Density Function
PHM	Prognostics and Health Management
$PPDM$	Post-Prognosis Decision-Making
R	Recovery
R_{mean}	Mean of recovery
R_{var}	Variance of recovery
RMS	Root Mean Square
RUL	Remaining Useful Life
SHM	Structural Health Monitoring
$TruncBNorm$	Truncated Bivariate Normal distribution
$TruncNorm$	Truncated Normal distribution
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}

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.

Fig. 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 (Pham & Wang, 1996). This results in an imperfect repair outcome and is the most realistic type of maintenance.

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's work (Nakagawa, 1979) 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. The authors in Block, Borges, and Savits (1985) 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 (Malik, 1979). Kijima, Morimura, and Suzuki (1988) 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 (Doyen, 2010; Kijima, 1989) collectively recognized as Kijima models. Several alternative treatment methodologies have been developed and documented within the literature, such as the shock model (Kijima & Nakagawa, 1992), (α, β) rule (Wang & Pham, 1996), and multiple (p, q) rule models (Shaked & Shanthikumar, 1986). Despite the multitude of approaches available, the widely adopted methods continue to be 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 modeling of imperfect repairs via the introduction of physics-based, data-driven, or hybrid approaches (incorporation of data-driven approaches into physical models). 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 (Do, Voisin, Levrat, & Iung, 2015; Hu, Pei, Wang, Si, & Zhang, 2018; Van & Bérenguer, 2012). This

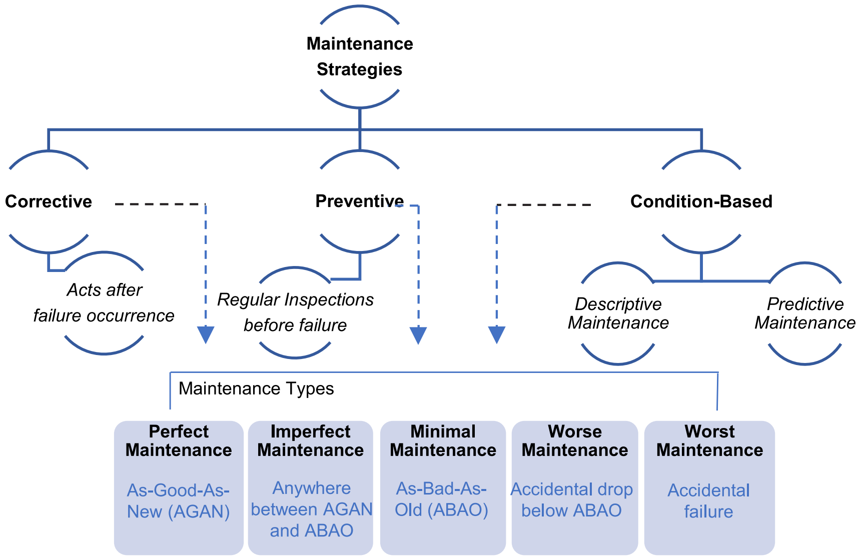


Fig. 1. Maintenance strategies and actions.
Source: Figure inspired by Carlo and Arleo (2017).

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 (Doyen & Gaudoin, 2004; Franck Corset & Gaudoin, 2012; Fuqing & Kumar, 2012; Pan & Rigdon, 2009) 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 (Cai et al., 2022; Liu, Chen, & Jiang, 2020; Ma et al., 2023; Nguyen, Do, Huynh, Bérenguer, & Grall, 2019; Skordilis & Moghaddass, 2020). 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 (Bousdekis & Mentzas, 2019; Do et al., 2015; Liu et al., 2020; Nguyen et al., 2019; Skordilis & Moghaddass, 2020), 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 (Song, Zhang, Shafieezadeh, & Xiao, 2022). 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.

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 modeling 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 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 modeling 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) (Kontogiannis, Salinas-Camus, & Eleftheroglou, 2025) are chosen due to their superior performance compared to other techniques (Eleftheroglou et al., 2019; Loutas et al., 2020; Loutas, Eleftheroglou, & Zarouchas, 2017).

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 (see Appendix A) 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 (van Ravenzwaaij, Cassey, & Brown, 2018), to effectively accommodate these diverse choices.

Performing an imperfect repair action leads to a RUL increase that requires modeling. Directly considering RUL as the sole information linked to imperfect repair offers four advantages: (i) modeling 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 modeling 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 modeling 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 article, 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.

Section 2 focuses on constructing the stochastic RUL model and building the imperfect repair model, while Section 3 describes the conducted experimental which serves as this case study. Finally, the evaluation of the methodology is presented in Section 4, and in Section 5, the main findings are discussed.

2. Methodology

This section focuses on the necessary steps for the development of the proposed imperfect repair model. 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] (Do et al., 2015; Hu et al., 2018; Wang, Hu, Si, & Zio, 2018). 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.

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 Fig. 2. The ultimate target is to model the stochastic level of repair in order to predict RUL post-repair before even maintenance occurs.

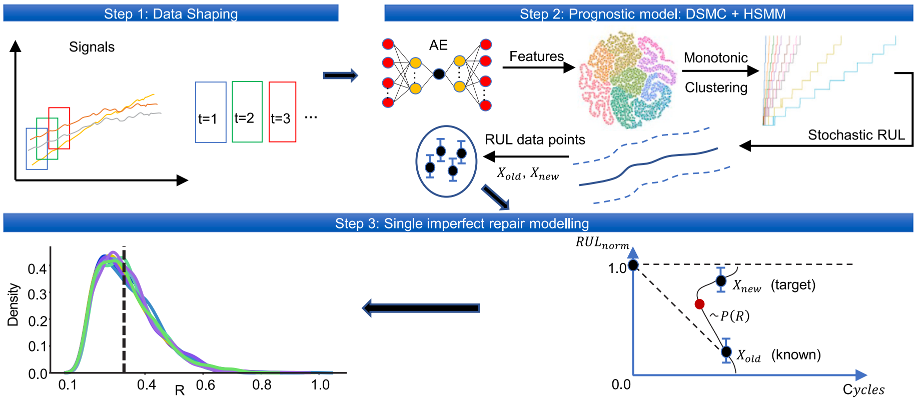


Fig. 2. Illustration of the entire process, from shaping the raw data to predicting RUL and using it to construct the imperfect repair model. Steps 1 and 2 could be replaced with any approach that predicts RUL under uncertainty.

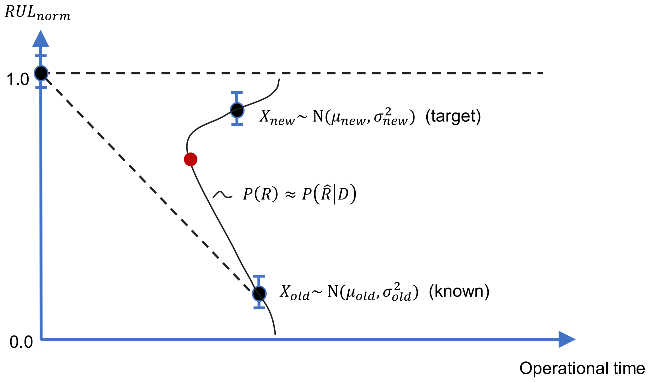

RUL_{norm}
1.0
$X_{new} \sim \mathrm{N}(\mu_{new}, \sigma^2_{new})$ (target)
$\sim P(R) \approx P(\hat{R}|D)$
$X_{old} \sim \mathrm{N}(\mu_{old}, \sigma^2_{old})$ (known)
0.0
Operational time

Fig. 3. 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 Eq. (8). 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.

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 Fig. 3). 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.

In the following subsections, each of the steps of the process will be extensively discussed.

2.1. 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.

2.2. 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 recent study that combines deep clustering and soft monotonic constraints for extracting monotonic features and clusters defined as the DSMC model (see Appendix A). The unique component of the DSMC model does not require any preprocessing step and can be applied directly to raw sequential data. It operates on the foundation of Artificial Neural Networks (ANN) as a core mechanism for extracting features through clustering analysis within the broad scope of deteriorating systems. Particularly noteworthy is the model's distinctive ability to derive pertinent prognostic-related features, i.e. features that exhibit an increasing trend over time, directly from raw data. This process occurs autonomously and comprehensively in a label-free format (unsupervised learning). The model's approach in selecting prognostic-related features is designed to detect a gradual progression (referred to as soft monotonicity) as opposed to an absolute and rigid progression (referred to as hard monotonicity). This choice is made to incorporate the potential occurrence of oscillations in the degradation trajectory of the analyzed system. Consequently, the DSMC model has the capability to consider noisy data while keeping the nature of the ever-increasing deterioration at the same time, thereby reflecting real-world systems and enabling a certain level of data comprehension.

Since 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 Long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997), that is, a recurrent layer, an appropriate candidate. Following this, the subsequent layers have the potential to comprise a series of Stacked Fully Connected (FC) layers. As an alternative, additional data, referred to as supplementary data, such as non-sequential data providing distinct extra information for each sample, can be incorporated into one of these FC layers. This integration facilitates their assimilation into the DSMC model, addressing the specific complexities associated with multi-modal data.

The DSMC model fundamentally operates as an adapted encoder that simultaneously extracts prognostic-related features and clusters those features accordingly. Essentially, the DSMC model adheres to a two-phase training procedure. Initially, the model engages in pre-training utilizing a deep autoencoder (AE), as shown in Fig. A.1. The

AE training is performed by the backpropagation algorithm considering the following loss:

$$\text{Loss}^{AE} = \text{MSE}\left(X, X'\right) + a * \left[\text{MSE}\left(t, t'\right) + \text{MSE}\left(O^{\text{enc}}_{\text{in}}, O^{\text{dec}}_{\text{out}}\right)\right]. \tag{1}$$

where O^{enc}_{in} and O^{dec}_{out} are the outcomes of the input layer of the encoder's monotonic module and the output layer of the decoder's monotonic module, respectively, 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 governs the significance attributed to the monotonicity within the clustering process.

Following this, the complete training dataset is passed through the encoder once, utilizing the resulting features to initialize the centroids generated by the k-means algorithm. Ultimately, the subsequent phase involves additional training of the DSMC model, specifically employing the encoder, to establish a connection between the features and the centroids, as depicted in Fig. A.2. The training of the second stage is performed via the following loss function:

$$\text{Loss} = \text{Loss}^{DSMC} + \beta * \text{Loss}^{AE}. \tag{2}$$

where Loss^{DSMC} is given by Eq. (A.1) and β is another tunable hyperparameter weighting the contribution of the $Loss^{AE}$. This entire procedure is performed in an unsupervised manner and the model's hyperparameters are automatically tuned by a customized Bayesian optimization algorithm (Victoria & Maragatham, 2021). Additional details about the two-stage training process and the role of each hyperparameter can be found in Appendix A.

The DSMC model is then paired with an HSMM for predicting the RUL under uncertainty as presented in Kontogiannis et al. (2025), which accepts the trajectories of clusters of the DSMC model as input and calculates the mean value of the RUL and the 95% CI. 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, as shown in Eq. (3). 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.

$$\begin{aligned} L(\lambda, O^{(1:K)}) &= \prod_{k=1}^{K} P(O^{(k)}|\lambda) \xrightarrow{L'=log(L)} \\ L'(\lambda, O^{(1:K)}) &= \sum_{k=1}^{K} log(P(O^{(k)}|\lambda)) \\ \lambda^* &= arg \max_{\lambda} \left(\sum_{k=1}^{K} log(P(O^{(k)}|\lambda)) \right) \end{aligned} \tag{3}$$

Once the model is trained, meaning that the degradation and observation processes are estimated, the Viterbi algorithm 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 Kontogiannis et al. (2025) and it is also presented Appendix B for completeness.

The prognostic measure can then be applied, considering the estimated state sequence. Kontogiannis et al. (2025) introduced a time-dependent prognostic measure which is presented Eq. (4). $D_i(d)$ 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. The variables $d_{i,i+1}$ and $d_{i,i}$ are defined as shown in Eqs. (5) and (6), respectively (Salinas-Camus & Eleftheroglou, 2024). 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%.

$$\begin{aligned} RUL_i^t = d_{i,i} \cdot &\left(D_i(d-\tau) + \sum_{k=i+1}^{N-1} D_k(d) + \mathcal{N}(1,\epsilon) \right) \\ &+ d_{i,i+1} \cdot \left(\sum_{k=i+1}^{N-1} D_k(d) + \mathcal{N}(1,\epsilon) \right) \end{aligned} \tag{4}$$

$$d_{i,i+1} = P(d \leq \tau | S_t = i) \tag{5}$$

$$d_{i,i} = 1 - d_{i,i+1} \tag{6}$$

Up until this point, the procedure could be replaced by any technique capable of predicting RUL under uncertainty, which is attributed to the proposed imperfect repair model's flexibility and applicability. The first 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 (X_{old}). 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.

2.3. Repair model and distribution of recovery

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. 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 modeling 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:

$$R = \frac{X_{new} - X_{old}}{1 - X_{old}}, \quad R \in [0,1] \tag{7}$$

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 Eq. (7) after estimating the distribution of R:

$$X_{new} = \left(1 - X_{old}\right) R + X_{old} \tag{8}$$

In this analysis, X_{old} and X_{new} are random variables that may follow any distribution according to the procedure of modeling RUL. Fig. 3 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 Eq. (8).

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 Eq. (7) 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 (Díaz-Francés & Rubio, 2013). An existing solution is to approximate R analytically by assuming a normal distribution with a mean $\mu = \mu_{new}/\mu_{old}$ and variance $\sigma^2 = \left(\mu_{new}/\mu_{old}\right)^2 [\left(\sigma_{new}/\mu_{new}\right)^2 + \left(\sigma_{old}/\mu_{old}\right)^2]$, if $\delta_{new} = \sigma_{new}/\mu_{new}$ is quite small ($\delta_{new} < 0.1$ is usually the condition for having a satisfying approximation Díaz-Francés & Rubio, 2013; Kuethe, Caprihan, Gach, Lowe, & Fukushima, 2000). 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, according to assumption 1, it should follow a Truncated Normal distribution representing the likelihood in Bayesian inference. This has also been mentioned in previous studies (Do et al., 2015; Van & Bérenguer, 2012; Wang et al., 2018). However, the specific parameters defining the distribution are still unknown and can be estimated differently by choosing 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 (van Ravenzwaaij et al., 2018) with No U-Turn Sampler (NUTS) (Hoffman, Gelman, et al., 2014) 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 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.

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.

Table 1
The necessary elements for the Bayesian modeling of a repair model.

Elements	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	$P\left(D\|\mu_{mean}, \sigma_{mean}, \mu_{var}, \sigma_{var}\right) \sim TruncBNorm$ $(\mu_{mean}, \sigma^2_{mean}, \mu_{var}, \sigma^2_{var}, a_{mean}, b_{mean}, a_{var}, b_{var})$
Priors	$\theta = \{\mu_{mean}, \sigma_{mean}, \mu_{var}, \sigma_{var}\}$, $\mu_{mean} \sim U\left(a_1, \ b_1\right)$, $\sigma_{mean} \sim \ U\left(a_2, \ b_2\right)$, $\mu_{var} \sim U\left(a_3, \ b_3\right)$
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 Eq. (8)

The Bayesian modeling consists of the components presented in Table 1. According to this table and based on assumption 1, the likelihood $P\left(D|\hat{R}\right)$ 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 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. Fig. 4 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 Eqs. (7) and (8), respectively as follows:

$$R_{mean} = \frac{\mu_{new} - \mu_{old}}{1 - \mu_{old}} \tag{9}$$

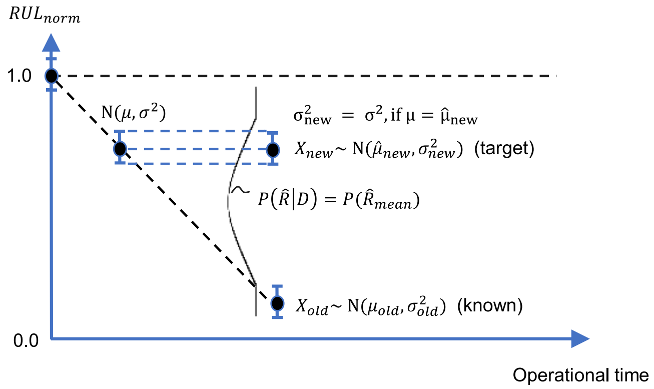

RUL_{norm}
1.0
0.0
$N(\mu,\sigma^2)$
$\sigma^2_{new} = \sigma^2$, if $\mu = \hat{\mu}_{new}$
$X_{new} \sim N(\hat{\mu}_{new}, \sigma^2_{new})$ (target)
$P(\hat{R}|D) = P(\hat{R}_{mean})$
$X_{old} \sim N(\mu_{old}, \sigma^2_{old})$ (known)
Operational time

Fig. 4. Recovery distribution under the assumption that the RUL model is independent of the repair process. Therefore, given all the stochastic RUL before repairing R_{mean} can be estimated by Bayesian inference, μ_{new} can be calculated, and the RUL before repair is identified that gives the same μ_{new} and simply the corresponding variance to construct X_{new} can be added.

$$\mu_{new} = \left(1 - \mu_{old}\right) R_{mean} + \mu_{old} \tag{10}$$

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 modeling consists of the elements stored in Table 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 (Srilakshmi, Ramji, & Chinthapenta, 2015). Additionally, this range allows a more flexible variance capable of shifting the distribution closer to 1.0, if necessary. Finally, the rejection sampling technique (Casella, Robert, & Wells, 2004; Robert & Casella, 2004) was applied to guarantee that the posterior predictive distribution will have only non-negative values at the required range.

3. Case study

In order 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 that given the periodic growth of an artificially induced crack – simulating a common damage case that can occur during operation due to fatigue loading – estimate, using the proposed model, the effects of a repair action on the lifetime. The test campaign involves tension-tension fatigue experiments on open-hole aluminium specimens. The specimens are cut from a 2 mm thick 7075-T6 aluminium sheet parallel to the rolling direction. The length of the specimens is 300 mm and the width is 45 mm. A 6 mm hole is drilled at the middle of the specimens, and 0.5–1 mm 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 100 MPa, a fatigue ratio of 0.1, and a frequency of 5 Hz.

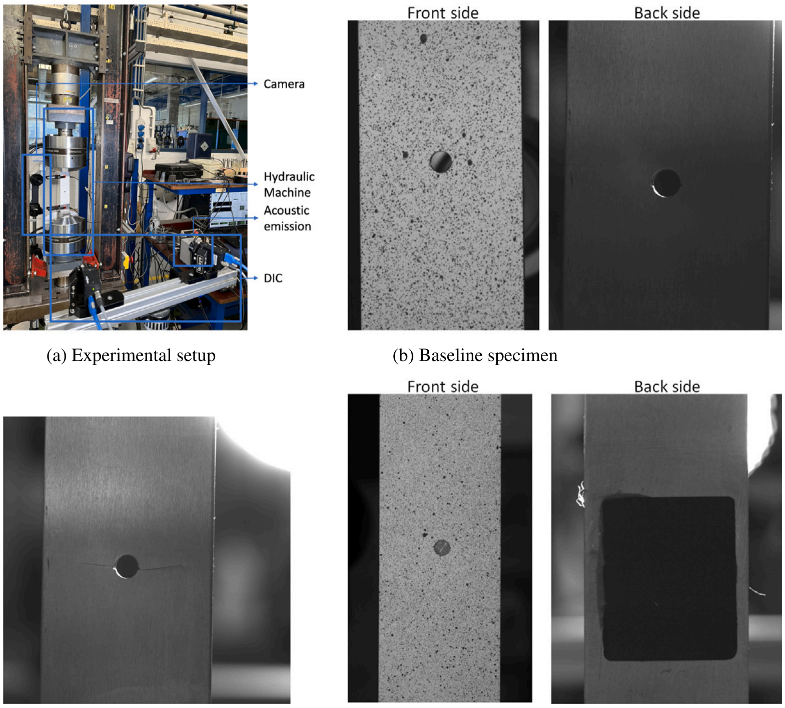
Table 2
The necessary elements for the Bayesian modeling of the proposed repair model.

Elements	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	$P\left(D\|\mu_{mean}, \sigma_{mean}\right) \sim TruncNorm\ (\mu_{mean}, \sigma^2_{mean}, a_{mean}, b_{mean})$
Priors	$\theta = \{\mu_{mean}, \sigma_{mean}\}$, $\mu_{mean} \sim U\left(a_1,\ b_1\right)$, $\sigma_{mean} \sim\ U\left(a_2,\ b_2\right)$
Posterior Predictive	One distribution for the mean of recovery: $P(R_{\hat{mean}}\|D)$
Process	Bayesian inference to estimate the posterior predictive $P(\hat{R}\|D) = P(R_{\hat{mean}}\|D)$, calculate $\mu_{\hat{new}}$ from Eq. (10), find the corresponding (μ, σ^2) before repairing such that $\left(\mu, \sigma^2\right) = (\mu_{\hat{new}}, \sigma^2_{new})$, estimate $X_{\hat{new}}$
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 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 void pressure for 24 h. 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 14 000 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 11 000 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 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.

Fig. 5(a) depicts the experimental setup and the different specimen conditions can be seen in Figs. 5(b)–5(d), for the baseline, damaged and repaired conditions respectively. To validate the methodology, an important aspect is the Structural Health Monitoring (SHM) data collected



(a) Experimental setup

(b) Baseline specimen

(c) Damaged specimen

(d) Repaired specimen

Fig. 5. Experimental setup (a) and specimen states (baseline (b), damaged (c), repaired (d)).

Table 3
Technical details. The specimens that were under repair contain two trajectories corresponding to one before and one after the repair.

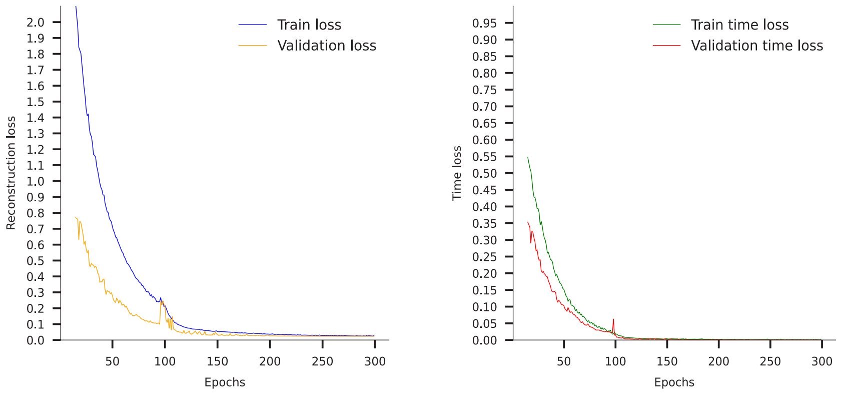
Specimen name(s)	Repair time (cycles) (T1)/observed crack size (mm)	Fatigue life after repair (cycles)	Fatigue life Total (cycles)
Baseline			
0_1	–	–	26 478
0_2	–	–	22 563
0_3	–	–	23 342
0_4	–	–	23 750
0_5	–	–	19 250
Average	–	–	23 076
1 Repair, reached EOL			
0_6, 1_6	14 000/1	24 565	38 565
0_7, 1_7	14 000/5	17 445	31 445
0_8, 1_8	14 000/6	17 250	31 250
Average	14 000/4	19 753	33 753
1 Repair, did not reach EOL, stopped after 11 000			
0_9, 1_9	14 000/1	–	–
0_10, 1_10	14 000/4	–	–
0_11, 1_11	14 000/0	–	–

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 were considered. To split the train and test data the dataset is formatted 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

Table 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^2 s [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} V s [nV s]	The integral of the rectified AE signal over time.



(a) (b)

Fig. 6. 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).

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 the results.

A Digital Image Correlation (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.

3.1. Values of hyperparameters highly dependent on the dataset

In this section, all the values of hyperparameters that are very task-specific are defined and, simultaneously, the reasons for choosing their values are explained.

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 A.1. Finally, Table 5 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 5
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. Results and discussion

In this section, the results of the clustering, prognostic, and repair models are presented and discussed.

4.1. 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 min. The training and validation losses concerning the reconstruction loss and the time domain loss, are shown in Figs. 6(a) and 6(b), respectively. These are the two losses based on which the DSMC model's autoencoder was trained. Following the entire procedure until RUL prediction, Fig. 7 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 (Eleftheroglou, Galanopoulos, & Loutas, 2024) could be a potential solution, it has not been chosen in this work. This decision was made to demonstrate that the proposed approach can be applied to any prognostic model, regardless of its

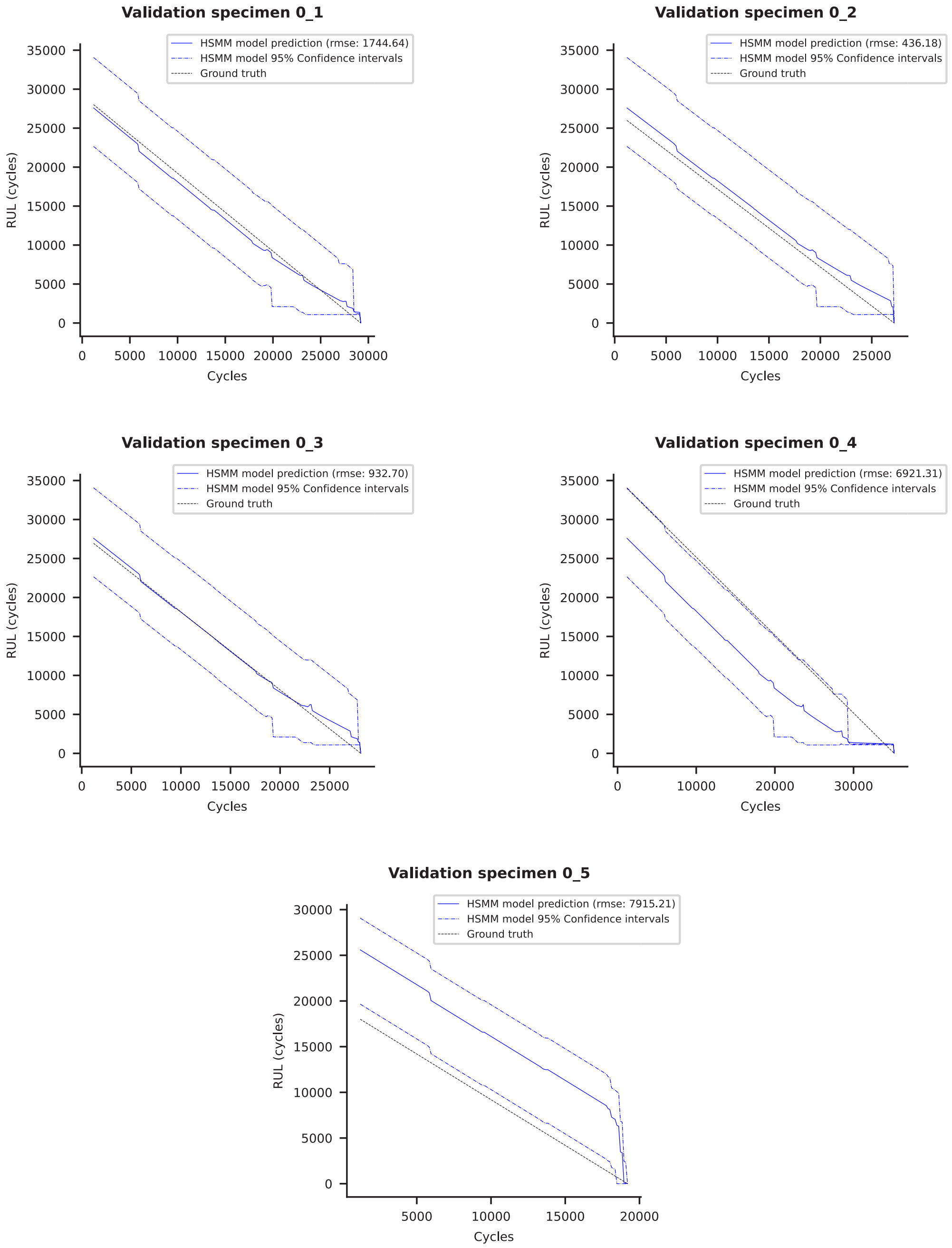


Fig. 7. Leave-one-out cross-validation for specimens '0_1' – '0_5' used for training the DSMC and HSMM models.

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.

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 Fig. 8. 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 behavior. 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 behavior can also be observed: all of the specimens' cluster assignments follow similar trends. Granted that the specimens fail at different times, it is expected that the damage accumulation (and thus the cluster assignment) would not follow the same path. This behavior is considered 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 input data for the DSMC and HSMM models. Such augmentation holds promise for introducing variability in clustering

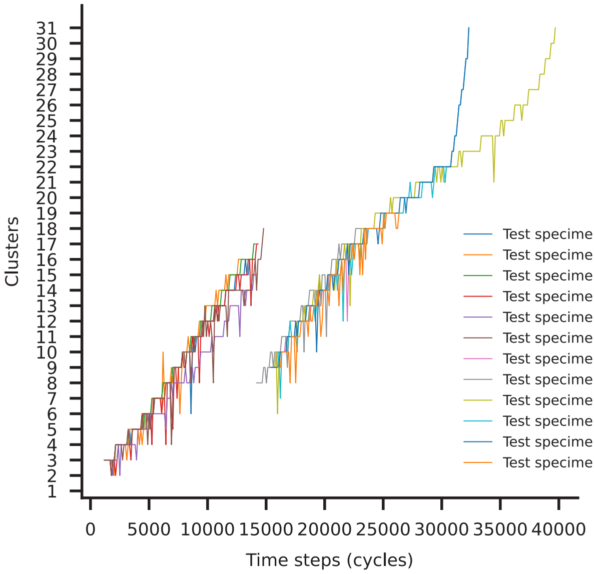


Fig. 8. The predicted by the DSMC model clustering assignments concerning the test specimens.

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. Fig. 9 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.

4.2. 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 Figs. 10(a) and 10(b) respectively. Based on these distributions, R_{mean} is estimated and illustrated in Fig. 11 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.

Additionally, by employing Eq. (10) 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 Fig. 12. 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 Srilakshmi et al. (2015), 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 2, σ_{new} can be calculated, and, simultaneously, the mean value and CI of the post-repair RUL can be approximated. As shown in the right column of Fig. 12 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. Fig. 12 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 a Gradient Boosting Decision Tree (GBDT) prognostic algorithm (Natekin & Knoll, 2013) 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 the 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 Fig. 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.

4.3. Total complexity of models

In this section, the total complexity of the models (DSMC, HSMM, and repairs models) is discussed. The total model complexity is equal to the sum of each model's complexity. However, the model complexity is different in the training and testing phases.

Starting from the training phase, the DSMC model's training is given by Eq. (A.6) as discussed in Appendix A.1. The model complexity of training the HSMM is $O^{train}_{HSMM}(IN_{traj}h_s^2 * t_{max})$ (Murphy, 2002),

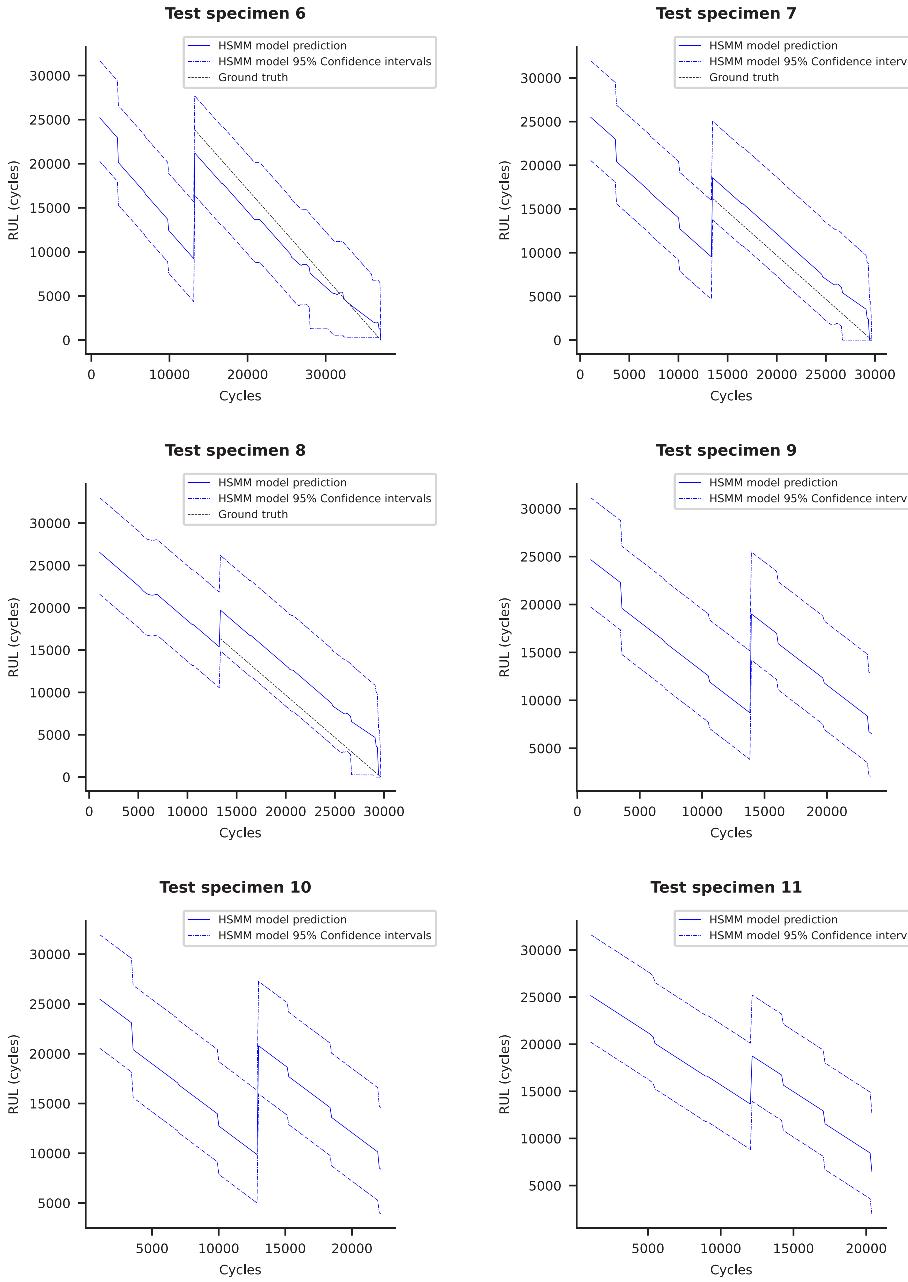


Fig. 9. Stochastic RUL predictions of testing specimens 6–11. Specimens 6–8 correspond to the ones with repair that reached the EOL. Specimens 9–11 represent the ones with repair that did not reach the EOL. The true RUL is known only for the specimens that reached the EOL; here, for the trajectory part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

where I is the number of iterations till convergence, N_{traj} is the number of trajectories, h_s is the number of hidden states, and t_{max} the length of the longest trajectory. In this case study, $I = 11, h_s = 8, N_{traj} = 5$, hence $O^{train}_{HSMM}(3520t_{max})$. The model complexity of MCMC considering the convergence time is $O^{train}_{MCMC}(Nd^3 \ln d)$ (including the burn-in time) (Belloni & Chernozhukov, 2009), where N is the number of samples and d is the number of parameters. Here, $d = 2$, reflects the mean and variance of the Truncated Normal distribution, hence $O^{train}_{MCMC}(8N \ln 2)$. Thus, the total model complexity of the training phase is:

$$O^{train} = O^{train}_{DSMC} + O^{train}_{HSMM}(3520t_{max}) + O^{train}_{MCMC}(8N \ln 2) \tag{11}$$

where O^{train}_{DSMC} is given by Eqs. (A.6), (A.4) and (A.5).

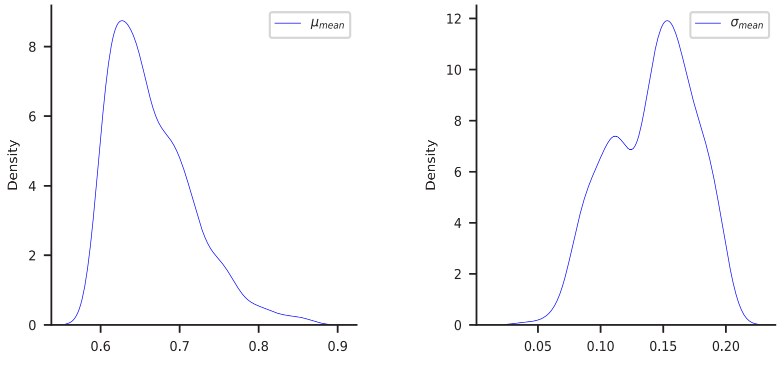
Concerning the testing phase, the DSMC model's testing is given by Eq. (A.7) as discussed in Appendix A.1. The Viterbi algorithm drives the HSMM model complexity and is $O^{test}_{HSMM}(N^2_{traj}N)$ (Quach & Farooq, 1994). Given that $N_{traj} = 12$ at the testing phase, this model complexity is $O^{test}_{HSMM}(144N)$. Regarding the MCMC algorithm, since there is a simple sampling over the posterior predictive distribution, the model complexity is $O^{test}_{MCMC}(dN)$. Given that $d = 2$, $O^{test}_{MCMC}(2N)$. Consequently, the total model complexity at the testing phase is:

$$O^{test} = O^{test}_{DSMC} + O^{test}_{HSMM}(144N) + O^{test}_{MCMC}(2N) \tag{12}$$

where O^{train}_{DSMC} is given by Eq. (A.7).

5. Conclusion and future work

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. These uncertainties often lead decision-makers to prefer replacements over repairs in maintenance planning, leaving



(a)

(b)

Fig. 10. 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}.

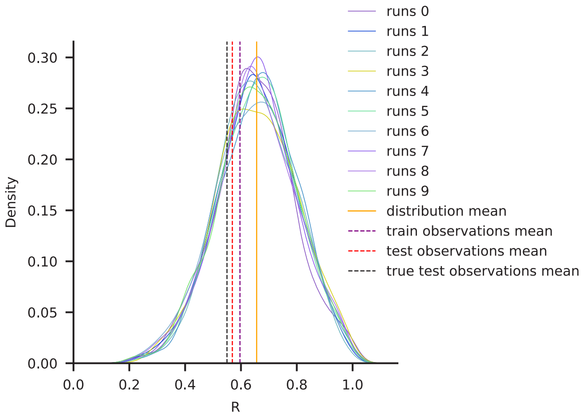


Fig. 11. Posterior predictive distribution of the mean of recovery (R_{mean}).

the economic and sustainability benefits of imperfect repairs underutilized. Existing literature lacks predictive models for estimating the post-repair health state of components before the actual maintenance action is performed. Thus, they cannot be integrated into a CBM framework where the maintenance planning is optimized beforehand and based on the current health state of the system. Additionally, the significant uncertainty introduced into the prediction of RUL due to an imperfect repair limits the applicability of existing approaches, since the acquired data before the repair no longer fully reflect the asset's health state post-repair. Moreover, current methodologies require task-specific degradation and repair modeling, making integration within a unified framework challenging and time-consuming. In this regard, 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 modeling, 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.

The final missing key aspect of making this approach applicable in an end-to-end CBM framework aiming at planning maintenance actions is its extension to post-repair RUL recovery modeling with multiple sequential repairs. It is believed that this extension is paramount to the model's real-world applicability, and it is the main focus for future work.

CRediT authorship contribution statement

P. Komninos: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Writing – review & editing. **G. Galanopoulos:** Methodology, Resources, Investigation, Writing – original draft. **T. Kontogiannis:** Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **N. Eleftheroglou:** Methodology, Validation, Writing – review & editing. **D. Zarouchas:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Generative AI in scientific writing

During the preparation of this work, the authors used ChatGPT based on GPT3.5 in order to improve the readability and language of some parts of the paper. The tool was in no way used to analyze and draw insights from the data, perform literature research, or extract any information other than feedback on the writing style based on the provided inputs. The tool was only used to perform minimal changes and provide feedback based on the provided input text, where the scientific content of the input sentences remains unchanged. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Fundings

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

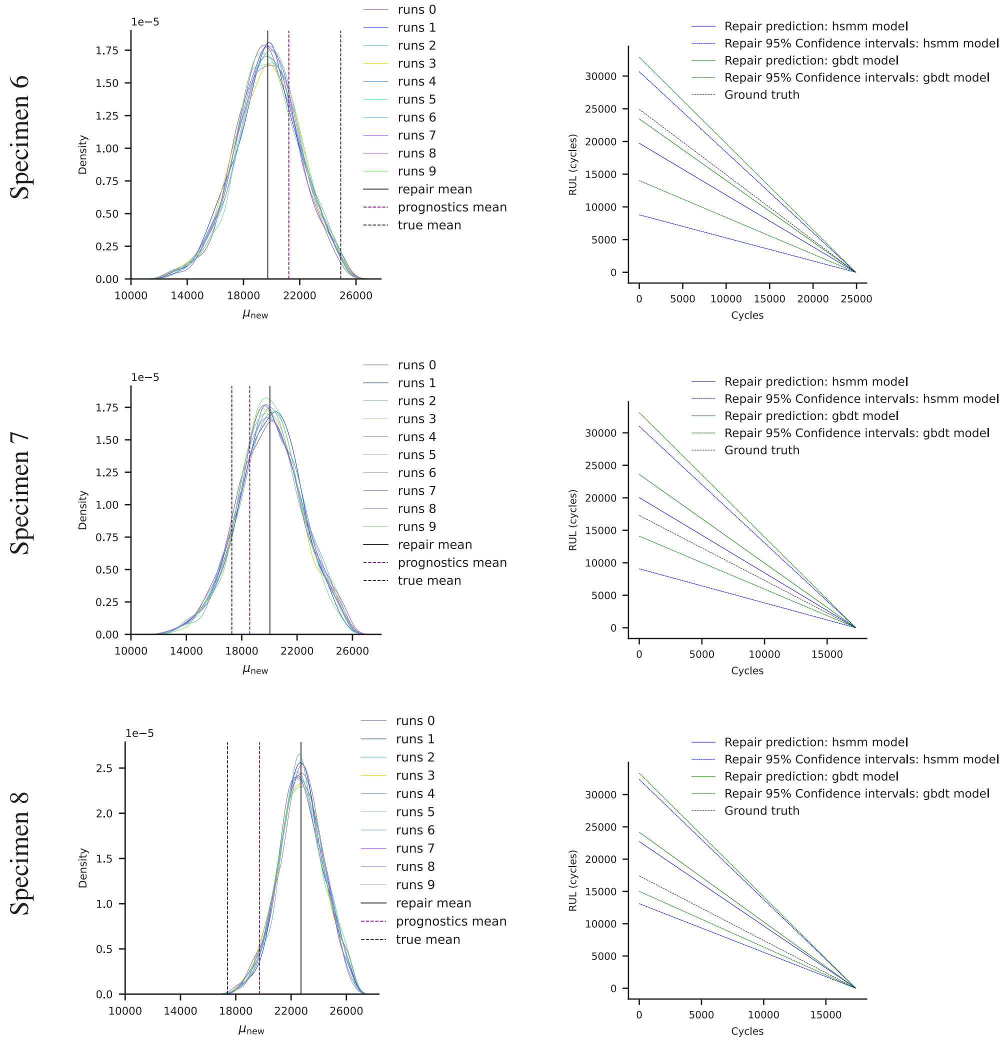


Fig. 12. Posterior predictive distributions of μ_{new} for test specimens 6, 7, and 8 (left column) and comparison of RUL predictions for the corresponding specimens after repair (X_{new}) between two prognostic models; HSMM and GBDT (left column).

Appendix A. Training the DSMC model

The training of the DSMC model involves a two-phase learning process: firstly, a pre-training phase utilizing an AE, which implements a strict constraint ensuring the non-negativity of the encoder's gradients concerning an embedded time feature. The second phase involves further training that exclusively employs the encoder to establish a connection between the features generated by the encoder and the centroids produced by the k-means algorithm. Here, the features denote the output of the encoder, while the centroids are the result of the k-means algorithm. As depicted in Fig. A.1, the AE consists of two modules stacked with multiple layers each. The first module, the feature extractor module, reduces the dimensionality of the input data to 1-dimensional arrays via a combination of LSTM and FC layers. The second module, referred to as the monotonic module, consists of a stacked arrangement of FC layers, integrated with a significant monotonic adjustment, playing a central role in developing an exceptionally robust model. The primary characteristic of this module revolves around its monotonous behavior, specifically in relation to time. To guarantee the intended incremental monotonic pattern in the output of the encoder, a vital additional time feature is introduced. Strict enforcement ensures that the gradients associated with the encoder's output concerning time remain non-negative.

There are two training losses that are utilized by those two learning stages. The first one is the AE loss, which is given by Eq. (1). The second loss function attributed to the DSMC is provided by:

$$Loss^{DSMC} = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{A.1}$$

where:

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

$$q_{ij} = \frac{\left(1 + \left\| z_i - \mu_j \right\|^2 / \alpha\right)^{-\frac{a+1}{2}}}{\sum_{j'} \left(1 + \left\| z_i - \mu_{j'} \right\|^2 / \alpha\right)^{-\frac{a+1}{2}}}. \tag{A.3}$$

The $Loss^{DSMC}$ is based on the Kullback–Leibler (KL) divergence loss between the distributions q and p, thus the goal is to make these distributions similar to each other. The probability distribution q typically achieves a soft assignment to the cluster labels that establishes the relationship between the extracted features of the AE and the centroids. This distribution corresponds to the Student's t-distribution, where α is the degrees of freedom of the distribution and in an unsupervised setting should be fixed to $\alpha = 1$. The distribution p corresponds to an auxiliary target distribution for performing the KL loss between q and p.

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

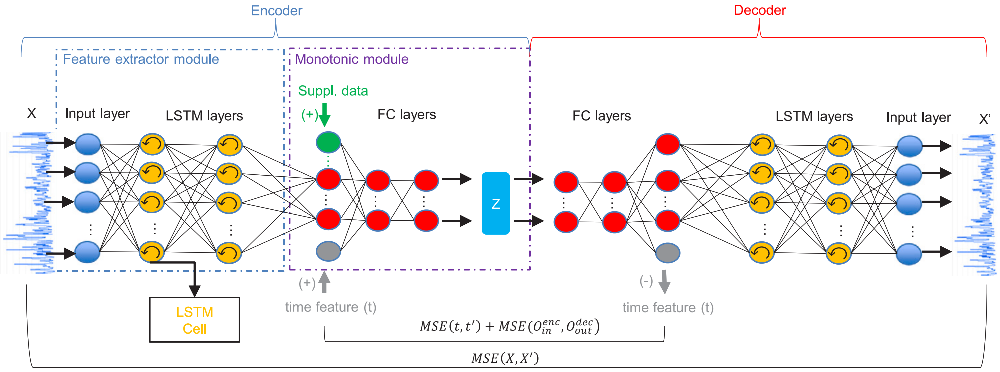


Fig. A.1. Detailed illustration of the deep AE architecture proposed in Komninos, Kontogiannis, Eleftheroglou, and Zarouchas (2024).

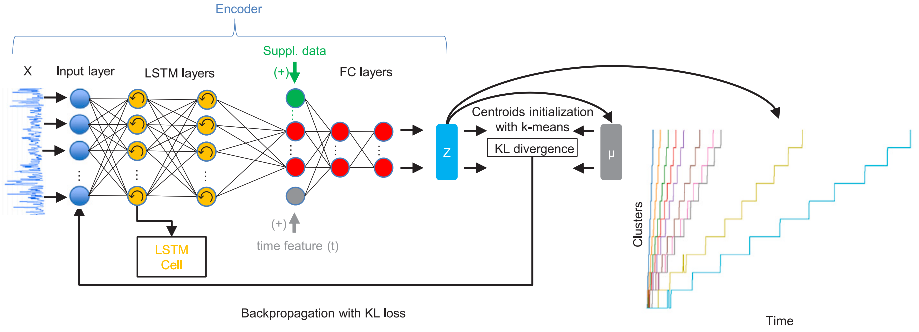


Fig. A.2. Detailed illustration of the second stage of the learning process as proposed in Komninos et al. (2024).

focuses on enhancing the assignments that exhibit low confidence. The optimization proceeds by jointly optimizing the ANN's parameters and the cluster centroids μ_j using the Stochastic Gradient Descent algorithm with momentum and applying a standard backpropagation with respect to the ANN's parameters. Consequently, the training loss of the second learning stage concerns both the AE and the deep clustering process and can be defined by Eq. (2). The rationale behind employing the $Loss^{AE}$ for the clustering process lies in the necessity to preserve the inherent soft monotonic characteristics within the embedding space, which would have gradually vanished otherwise. The second stage process is illustrated in Fig. A.2. Finally, during inference, the decoder is discarded whilst the trained encoder performs soft monotonic clustering. Interested readers may refer to Komninos et al. (2024) for additional details of the entire process. The hyperparameters that were considered for tuning via the Bayesian optimization algorithm are $h = \{Z, H_{in}^{enc}, lr^{AE}, lr^{DSMC}, epochs^{AE}, epochs^{DSMC}, \alpha, \beta, dr_{rate}\}$. Here, the hyperparameter H_{in}^{enc} corresponds to the number of neurons of the last hidden layer of the encoder's feature extractor module, dr_{rate} is the dropout rate, lr^{AE} and lr^{DSMC} are the learning rates of the AE and clustering model, respectively, and $epochs^{AE}$ and $epochs^{DSMC}$ are the corresponding number of epochs. Additional hyperparameters that were manually decided include L_{window}, S, $batch^{AE}$, and $batch^{DSMC}$, where L_{window} is the window length of each window with step size S, and $batch^{AE}$ and $batch^{DSMC}$ are the corresponding batch sizes for each training stage of the DSMC model.

A.1. DSMC model complexity

For each model, there are two model complexities: one for the training and another for the testing phase. Starting from the training phase, the DSMC model's training consists of two stages. In the first stage, the model is an autoencoder with a stack of LSTM and MLP layers. Therefore, the model complexity is the sum of LSTM and MLP model complexities of the encoder and decoder respectively, and equals to:

Table A.1
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

$$O_{AE} = O(2epochs^{AE} N L_{LSTM} L_{window}(n_{features} H_{LSTM} + H_{LSTM}^2)) + \\ +O(2epochs^{AE} N L_{FC} H_{in}^{FC} H_{out}^{FC}) \tag{A.4}$$

where L_{LSTM} and L_{FC} are the number of LSTM and FC layers respectively, N the number of samples, H_{LSTM} the number of units in each LSTM layer, H_{in}^{FC} and H_{out}^{FC} the number of input and output hidden units of each FC layer.

In the second stage, DSMC consists of the autoencoder's encoder part plus a k-means clustering. This is given by the following formula:

$$O_{DSMC} = O(30NZ) + O(epochs^{DSMC} N L_{LSTM} L_{window}(n_{features} H_{LSTM} \\ +H_{LSTM}^2)) + \\ +O(epochs^{DSMC} N L_{FC} H_{in}^{FC} H_{out}^{FC}) \tag{A.5}$$

where the term $O(30NZ)$ corresponds to k-means clustering with 30 clusters. The final model complexity of the entire training of the DSMC model is as follows:

$$O^{train}_{DSMC} = O_{AE} + O_{DSMC} \tag{A.6}$$

During the testing phase, only the encoder part is kept (without any k-means clustering), thus the DSMC model complexity for one sample is:

$$O^{test}_{DSMC} = O(L_{LSTM} L_{window}(n_{features} H_{LSTM} + H^2_{LSTM})) \\ + O(L_{FC} H^{FC}_{in} H^{FC}_{out}) \tag{A.7}$$

Appendix B. 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 $S = \{S_1, S_2, \dots, S_N\}$, 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 modeled with a continuous probability distribution (Gaussian), therefore the indicator space consists of all the real numbers, so $M \rightarrow \infty$ and $V = \{v \in \mathfrak{R}\}$.
- 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 $\{1, 2, \dots, D\}$ that the state duration random variable d obtains. Therefore, the degradation process is modeled 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:

$$\begin{gathered} \bar{\alpha}_{t+1}(i,d) = \sum_{j=1, j\neq i}^{N} \left(\alpha_{t+1-d}(j) * a_{ji}\right) * \prod_{\tau=t-d+2}^{t+1} \left(b_i(O_\tau)\right) * p_i(d) \\ \alpha_{t+1}(i) = \sum_{d=1}^{D} \bar{\alpha}_{t+1}(i,d), \\ \text{where } \ 1 \leq t \leq T-1 \ \text{ and } \ 1 \leq i \leq N \end{gathered} \tag{B.2}$$

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:

$$\begin{gathered} \delta_t(j,d) = \max_{\substack{1\leq i\leq N,\, i\neq j, \\ 1\leq d'\leq D}} \delta_{t-d}(i,d') * a_{ij} * \prod_{\tau=t-d+1}^{t+1} b_j(O_\tau), \\ \text{where } \ 1 \leq t \leq T, \ 1 \leq j \leq N, \ 1 \leq d \leq D \end{gathered} \tag{B.6}$$

3. Termination:

$$P(Q,O) = \max_{1\leq i\leq N} \delta_T(i,d) \tag{B.7}$$

$$\hat{q}_T = \arg\max_{\substack{1\leq i\leq N, \\ 1\leq d'\leq D}} \delta_T(i) \tag{B.8}$$

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad \text{where } \ t = T-1, T-2, \dots, 1 \tag{B.9}$$

In the termination step, Eq. (B.8) is the best last state, and Eq. (B.9) represents the backtracking used to get the best states in each time step.

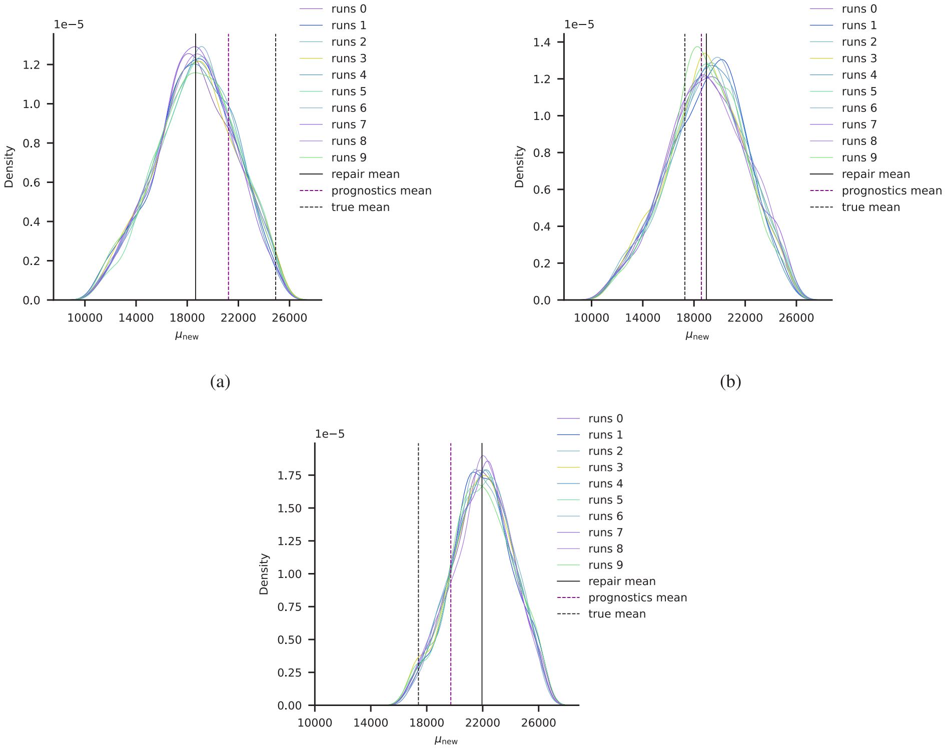
Finally, the last problem of the HMM, **the learning problem**, is solved by the Expectation-Maximization (E-M) algorithm, as mentioned in Section 2.2. The E-M algorithm utilizes both the forward and backward variables of the Forward-Backward algorithm and the parameters are updated to maximize the likelihood probability as defined in Eq. (3). The forward variable α is calculated with the iterative algorithm shown in Eqs. (B.1)–(B.3). The backward variable is defined as $\beta_t(i) = P(O_{t+1} O_{t+1} ... O_T | q_T = S_i, \lambda)$, 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:

$$\begin{gathered} \beta_t(i) = \sum_{\substack{j=1, \\ j\neq i}}^{N} \left(a_{ij} * \sum_{d=1}^{D} \left(p_i(d) * \beta_{t+d}(i) * \prod_{\tau=t+1}^{t+d} b_j(O_\tau) \right)\right), \\ \text{where } \ 1 \leq i \leq N \end{gathered} \tag{B.11}$$



(a)

(b)

(c)

Fig. 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).

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 Eqs. (4)–(6) are utilized to calculate the RUL.

Appendix C. Additional results of repair model

See Fig. C.1.

Data availability

The dataset used for this work is available at https://data.4tu.nl/datasets/b7d8031f-95a6-4ccc-8c08-802e694a0f40.

References

Belloni, Alexandre, & Chernozhukov, Victor (2009). On the computational complexity of MCMC-based estimators in large samples. *The Annals of Statistics*, 2011–2055.

Block, Henry W., Borges, Wagner S., & Savits, Thomas H. (1985). Age-dependent minimal repair. *Journal of Applied Probability*, *22*(2), 370–385. http://dx.doi.org/10.2307/3213780.

Bousdekis, Alexandros, & Mentzas, Gregoris (2019). A proactive model for joint maintenance and logistics optimization in the frame of industrial internet of things. In Angelo Sifaleras, & Konstantinos Petridis (Eds.), *Operational research in the digital era – ICT challenges* (pp. 23–45). Cham: Springer International Publishing, ISBN: 978-3-319-95666-4.

Cai, Baoping, Wang, Yuandong, Zhang, Yanping, Liu, Yiliu, Ge, Weifeng, Li, Rongkang, et al. (2022). Condition-based maintenance method for multi-component system based on RUL prediction: Subsea tree system as a case study. *Computers & Industrial Engineering*, [ISSN: 0360-8352] *173*, Article 108650. http://dx.doi.org/10.1016/j.cie.2022.108650, URL https://www.sciencedirect.com/science/article/pii/S0360835222006386.

Carlo, Filippo De, & Arleo, Maria Antonietta (2017). Imperfect maintenance models, from theory to practice. In Constantin Volosencu (Ed.), *System reliability*. Rijeka: IntechOpen, http://dx.doi.org/10.5772/intechopen.69286, chapter 18.

Casella, George, Robert, Christian P., & Wells, Martin T. (2004). Generalized accept-reject sampling schemes. *Lecture Notes-Monograph Series*, [ISSN: 07492170] *45*, 342–347, URL http://www.jstor.org/stable/4356322.

Díaz-Francés, Eloísa, & Rubio, Francisco J. (2013). On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables. *Statistical Papers*, [ISSN: 1613-9798] *54*(2), 309–323. http://dx.doi.org/10.1007/s00362-012-0429-2.

Do, Phuc, Voisin, Alexandre, Levrat, Eric, & Iung, Benoit (2015). A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *133*, 22–32. http://dx.doi.org/10.1016/j.ress.2014.08.011, URL https://www.sciencedirect.com/science/article/pii/S095183201400204X.

Doyen, Laurent (2010). Asymptotic properties of imperfect repair models and estimation of repair efficiency. *Naval Research Logistics*, *57*(3), 296–307. http://dx.doi.org/10.1002/nav.20406, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20406.

Doyen, Laurent, & Gaudoin, Olivier (2004). Classes of imperfect repair models based on reduction of failure intensity or virtual age. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *84*(1), 45–56. http://dx.doi.org/10.1016/S0951-8320(03)00173-X, URL https://www.sciencedirect.com/science/article/pii/S095183200300173X. Selected papers from ESREL 2002.

Eleftheroglou, Nick, Galanopoulos, Georgios, & Loutas, Theodoros (2024). Similarity learning hidden semi-Markov model for adaptive prognostics of composite structures. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *243*, Article 109808. http://dx.doi.org/10.1016/j.ress.2023.109808, URL https://www.sciencedirect.com/science/article/pii/S0951832023007226.

Eleftheroglou, Nick, Mansouri, Sina Sharif, Loutas, Theodoros, Karvelis, Petros, Georgoulas, George, Nikolakopoulos, George, et al. (2019). 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. *Applied Energy*, [ISSN: 0306-2619] *254*, Article 113677. http://dx.doi.org/10.1016/j.apenergy.2019.113677, URL https://www.sciencedirect.com/science/article/pii/S0306261919313649.

Franck Corset, Laurent Doyen, & Gaudoin, Olivier (2012). Bayesian analysis of ARA imperfect repair models. *Communications in Statistics. Theory and Methods*, *41*(21), 3915–3941. http://dx.doi.org/10.1080/03610926.2012.698688.

Fuqing, Yuan, & Kumar, Uday (2012). A general imperfect repair model considering time-dependent repair effectiveness. *IEEE Transactions on Reliability*, *61*(1), 95–100. http://dx.doi.org/10.1109/TR.2011.2182222.

Hochreiter, Sepp, & Schmidhuber, Jürgen (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hoffman, Matthew D., Gelman, Andrew, et al. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(1), 1593–1623.

Hu, Changhua, Pei, Hong, Wang, Zhaoqiang, Si, Xiaosheng, & Zhang, Zhengxin (2018). A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities. *Chinese Journal of Aeronautics*, [ISSN: 1000-9361] *31*(3), 514–528. http://dx.doi.org/10.1016/j.cja.2018.01.009, URL https://www.sciencedirect.com/science/article/pii/S1000936118300256.

Kijima, Masaaki (1989). Some results for repairable systems with general repair. *Journal of Applied Probability*, *26*(1), 89–102. http://dx.doi.org/10.2307/3214319.

Kijima, Masaaki, Morimura, Hidenori, & Suzuki, Yasusuke (1988). Periodical replacement problem without assuming minimal repair. *European Journal of Operational Research*, [ISSN: 0377-2217] *37*(2), 194–203. http://dx.doi.org/10.1016/0377-2217(88)90329-3, URL https://www.sciencedirect.com/science/article/pii/0377221788903293.

Kijima, Masaaki, & Nakagawa, Toshio (1992). Replacement policies of a shock model with imperfect preventive maintenance. *European Journal of Operational Research*, [ISSN: 0377-2217] *57*(1), 100–110. http://dx.doi.org/10.1016/0377-2217(92)90309-W, URL https://www.sciencedirect.com/science/article/pii/037722179290309W.

Komninos, Panagiotis, Kontogiannis, Thanos, Eleftheroglou, Nick, & Zarouchas, Dimitrios (2024). A robust generalized deep monotonic feature extraction model for label-free prediction of degenerative phenomena. https://pure.tudelft.nl/admin/files/222348032/DSMC_PK_TK_NE_DZ.pdf. [Accessed 10 October 2024].

Kontogiannis, Thanos, Salinas-Camus, Mariana, & Eleftheroglou, Nick (2025). Hidden Markov model applications: Aviation prognostics. In *Stochastic modeling and statistical methods: advances and applications* (pp. 191–213). Academic Press, ISBN: 9780443316951, http://dx.doi.org/10.1016/B978-0-44-331694-4.00015-3.

Kuethe, Dean, Caprihan, Arvind, Gach, H., Lowe, Irving, & Fukushima, Eiichi (2000). Imaging obstructed ventilation with NMR using inert fluorinated gases. *Journal of Applied Physiology (Bethesda, Md. : 1985)*, *88*, 2279–2286. http://dx.doi.org/10.1152/jappl.2000.88.6.2279.

Liu, Yu, Chen, Yiming, & Jiang, Tao (2020). Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, [ISSN: 0377-2217] *283*(1), 166–181. http://dx.doi.org/10.1016/j.ejor.2019.10.049, URL https://www.sciencedirect.com/science/article/pii/S0377221719309014.

Loutas, Theodoros, Eleftheroglou, Nick, Georgoulas, George, Loukopoulos, Panagiotis, Mba, David, & Bennett, Ian (2020). Valve failure prognostics in reciprocating compressors utilizing temperature measurements, PCA-based data fusion, and probabilistic algorithms. *IEEE Transactions on Industrial Electronics*, *67*(6), 5022–5029. http://dx.doi.org/10.1109/TIE.2019.2926048.

Loutas, Theodoros, Eleftheroglou, Nick, & Zarouchas, Dimitrios (2017). A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data. *Composite Structures*, [ISSN: 0263-8223] *161*, 522–529. http://dx.doi.org/10.1016/j.compstruct.2016.10.109, URL https://www.sciencedirect.com/science/article/pii/S0263822316316506.

Ma, Jie, Cai, Li, Liao, Guobo, Yin, Hongpeng, Si, Xiaosheng, & Zhang, Peng (2023). A multi-phase Wiener process-based degradation model with imperfect maintenance activities. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *232*, Article 109075. http://dx.doi.org/10.1016/j.ress.2022.109075, URL https://www.sciencedirect.com/science/article/pii/S0951832022006901.

Malik, Mazhar Ali Khan (1979). Reliable preventive maintenance scheduling. *IEEE Transactions*, *11*(3), 221–228. http://dx.doi.org/10.1080/05695557908974463.

Murphy, Kevin P. (2002). Hidden semi-markov models (hsmms). *Unpublished Notes*, *2*.

Nakagawa, Toshio (1979). Optimum policies when preventive maintenance is imperfect. *IEEE Transactions on Reliability*, *R-28*(4), 331–332. http://dx.doi.org/10.1109/TR.1979.5220624.

Natekin, Alexey, & Knoll, Alois (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, [ISSN: 1662-5218] *7*, http://dx.doi.org/10.3389/fnbot.2013.00021, URL https://www.frontiersin.org/articles/10.3389/fnbot.2013.00021 http://journal.frontiersin.org/article/10.3389/fnbot.2013.00021/abstract.

Nguyen, Khanh T. P., Do, Phuc, Huynh, Khac Tuan, Bérenguer, Christophe, & Grall, Antoine (2019). Joint optimization of monitoring quality and replacement decisions in condition-based maintenance. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *189*, 177–195. http://dx.doi.org/10.1016/j.ress.2019.04.034, URL https://www.sciencedirect.com/science/article/pii/S0951832018313097.

Pan, Rong, & Rigdon, Steven E. (2009). Bayes inference for general repairable systems. *Journal of Quality Technology*, *41*(1), 82–94. http://dx.doi.org/10.1080/00224065.2009.11917762.

Pham, Hoang, & Wang, Hongzhou (1996). Imperfect maintenance. *European Journal of Operational Research*, *94*(3), 425–438.

Quach, T., & Farooq, M. (1994). Maximum likelihood track formation with the Viterbi algorithm. *Vol. 1*, In *Proceedings of 1994 33rd IEEE conference on decision and control* (pp. 271–276). http://dx.doi.org/10.1109/CDC.1994.410918.

Robert, Christian P., & Casella, George (2004). Iterated and sequential importance sampling. In *Monte Carlo statistical methods* (pp. 545–580). New York, NY: Springer New York, ISBN: 978-1-4757-4145-2, http://dx.doi.org/10.1007/978-1-4757-4145-2_14.

Salinas-Camus, M., & Eleftheroglou, N. (2024). Uncertainty in aircraft turbofan engine prognostics on the C-MAPSS dataset. *Vol. 8*, In *Proceedings of the European conference of the PHM society 2024* (p. 10). http://dx.doi.org/10.36001/phme.2024.v8i1.4007.

Shaked, Moshe, & Shanthikumar, J. George (1986). Multivariate imperfect repair. *Operations Research*, *34*(3), 437–448. http://dx.doi.org/10.1287/opre.34.3.437.

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

Song, Chaolin, Zhang, Chi, Shafieezadeh, Abdollah, & Xiao, Rucheng (2022). Value of information analysis in non-stationary stochastic decision environments: A reliability-assisted POMDP approach. *Reliability Engineering & System Safety*, [ISSN: 0951-8320] *217*, Article 108034. http://dx.doi.org/10.1016/j.ress.2021.108034, URL https://www.sciencedirect.com/science/article/pii/S095183202100541X.

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

Van, Phuc Do, & Bérenguer, Christophe (2012). Condition-based maintenance with imperfect preventive repairs for a deteriorating production system. *Quality and Reliability Engineering International*, *28*(6), 624–633. http://dx.doi.org/10.1002/qre.1431, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.1431.

van Ravenzwaaij, Don, Cassey, Pete, & Brown, Scott D. (2018). A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin & Review*, [ISSN: 1531-5320] *25*(1), 143–154. http://dx.doi.org/10.3758/s13423-016-1015-8.

Verhagen, Wim J. C., Santos, Bruno F., Freeman, Floris, van Kessel, Paul, Zarouchas, Dimitrios, Loutas, Theodoros, et al. (2023). Condition-based maintenance in aviation: Challenges and opportunities. *Aerospace*, [ISSN: 2226-4310] *10*(9), http://dx.doi.org/10.3390/aerospace10090762, URL https://www.mdpi.com/2226-4310/10/9/762.

Victoria, A. Helen, & Maragatham, G. (2021). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*, [ISSN: 1868-6478] *12*(1), 217–223. http://dx.doi.org/10.1007/s12530-020-09345-2, URL https://link.springer.com/10.1007/s12530-020-09345-2.

Wang, Zhao-Qiang, Hu, Chang-Hua, Si, Xiao-Sheng, & Zio, Enrico (2018). Remaining useful life prediction of degrading systems subjected to imperfect maintenance: Application to draught fans. *Mechanical Systems and Signal Processing*, [ISSN: 0888-3270] *100*, 802–813. http://dx.doi.org/10.1016/j.ymssp.2017.08.016, URL https://www.sciencedirect.com/science/article/pii/S0888327017304429.

Wang, Hongzhou, & Pham, Hoang (1996). Optimal maintenance policies for several imperfect repair models. *International Journal of Systems Science*, *27*(6), 543–549. http://dx.doi.org/10.1080/00207729608929248.