A comprehensive review and evaluation framework for data-driven prognostics Uncertainty, robustness, interpretability, and feasibility

Salinas-Camus, Mariana; Goebel, Kai; Eleftheroglou, Nick

DOI
10.1016/j.ymssp.2025.113015

Publication date
2025

Document Version
Final published version

Published in
Mechanical Systems and Signal Processing

Citation (APA)
Salinas-Camus, M., Goebel, K., & Eleftheroglou, N. (2025). A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility. *Mechanical Systems and Signal Processing*, *237*, Article 113015. https://doi.org/10.1016/j.ymssp.2025.113015

Mechanical Systems and Signal Processing 237 (2025) 113015

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Invited review paper

A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility

Mariana Salinas-Camus [a,*], Kai Goebel [b,c], Nick Eleftheroglou [a]

[a] Intelligent System Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629HS, The Netherlands
[b] Fragum Global, Mountain View, CA 94040, USA
[c] Luleå University of Technology, Luleå, 971 87, Sweden

ARTICLE INFO

Communicated by X. Si

Keywords:
Prognostics
Remaining useful life
Data-driven
Robustness
Interpretability
Uncertainty
Feasibility

ABSTRACT

Prognostics and Health Management (PHM) is critical for predicting the Remaining Useful Life (RUL) of systems, a key enabler of Predictive Maintenance (PdM). This paper reviews state-of-the-art data-driven prognostic models, emphasizing four essential characteristics: uncertainty, robustness, interpretability, and feasibility. While traditional research has focused on enhancing RUL prediction accuracy, this review argues that these additional characteristics are equally vital for addressing the demands of PHM applications.

The review examines Machine Learning (ML) techniques, stochastic models, and Bayesian filters (BFs), analyzing their strengths, limitations, and trade-offs. ML models excel in accuracy but often lack robust uncertainty quantification and adaptability across varying operational conditions. Stochastic models demonstrate greater robustness and feasibility, performing reliably with limited or variable data. Bayesian filters provide high interpretability and do not require run-to-failure data but face challenges in adapting to diverse environments.

To bridge these gaps, this paper proposes a structured Model Evaluation Framework that integrates users' specific needs with key model characteristics identified in the review. By quantifying the importance of the four characteristics, the framework enables systematic evaluation and selection of prognostic models.

The findings underscore the need for advancements in uncertainty quantification, adaptive methods to improve robustness, and enhanced interpretability to meet practical and regulatory requirements. While current models offer valuable insights, further improvements are necessary to unlock their full potential for PHM and PdM applications, ensuring more reliable and actionable predictions.

Contents

* Corresponding author.
E-mail address: m.salinascamus@tudelft.nl (M. Salinas-Camus).

https://doi.org/10.1016/j.ymssp.2025.113015
Received 28 January 2025; Received in revised form 16 May 2025; Accepted 18 June 2025
Available online 7 July 2025

1. Introduction

Prognostics and Health Management (PHM) is a field that provides users with a thorough analysis of a system's current and future health conditions to maximize operational availability, reduce maintenance costs, improve reliability, and guarantee predefined safety standards. While PHM encompasses various modules, including feature extraction, diagnostics, prognostics, and decision-making, as shown in Fig. 1, this paper specifically focuses on the prognostics aspect of PHM. A general and brief overview of the complete PHM framework will be provided to give better context for understanding where prognostics fits within the framework.Feature extraction involves identifying and isolating relevant data from sensors and signals that indicate the health status of a system or component. Diagnostics is the process of analyzing these features to detect, isolate, localize, and identify faults or failures that have already occurred in the system. Prognostics extends this analysis to predict the future health state and Remaining Useful Life (RUL) of the system based on current and historical data. Finally, decision-making in PHM uses diagnostic and prognostic information to optimize maintenance strategies, operations, and logistics to reduce downtime, costs, and risks, ultimately improving system reliability and performance.

PHM is closely connected with Predictive Maintenance (PdM) since PdM uses PHM information to predict failures and plan maintenance tasks [1]. PdM changes the paradigm of other maintenance strategies, such as corrective and preventive. Corrective maintenance is performed once the fault has been detected. Preventive maintenance is scheduled at specific times, typically decided in the design phase.

Therefore, since PdM uses the prediction of failures to plan maintenance, prognostics becomes crucial for decision-making in PdM frameworks. As noted in [2], "prognostics is performed not as an objective in themselves, but for a decision-making process afterward". However, recent trends in prognostics have increasingly focused on performance metrics and minimizing prediction errors [3], [4]. While achieving high performance is important, focusing on the characteristics that make prognostic models effective for decision-making in PdM is equally vital.

In [5], it is highlighted that much of the literature emphasizes RUL prediction without addressing the subsequent health management tasks. Therefore, two characteristics of prognostic models are emphasized as necessary for decision-making in PdM. Prognostic models must provide uncertainty quantification because, as the authors describe, it is essential for effective maintenance planning since RUL prediction inherently has uncertainty. Additionally, the model must be feasible. A feasible model can perform well with the available conditioning monitor data, which is a fundamental characteristic for developing an end-to-end data-driven PdM framework.

Nomenclature

ANHHSMM	Adaptive Non-Homogeneous Hidden Semi Markov Model
BF	Bayesian Filter
Bi-GRU	Bidirectional Gated Recurrent Unit
BNN	Bayesian Neural Network
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DDM	Data-Driven Model
DL	Deep Learning
DNN	Deep Neural Network
EKF	Extended Kalman Filter
GAN	Generative Adversarial Network
GHMM	Generalized Hidden Markov Model
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
KF	Kalman Filter
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
MCMC	Monte Carlo Markov Chain
ML	Machine Learning
MLP	Multi-Layer Perceptron
NHHSMM	Non-Homogeneous Hidden Semi-Markov Model
NN	Neural Network
PbM	Physical-based Model
PdM	Predictive Maintenance
PF	Particle Filter
PHM	Prognostics and Health Management
RF	Random Forest
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
RVM	Relevance Vector Machine
SHAP	Shapley Additive Explanations
SLHSMM	Similarity Learning Hidden Semi Markov Model
SM	Stochastic Model
SPC	Statistical Process Control
SVM	Support Vector Machine
SVR	Support Vector Regressor
UQ	Uncertainty Quantification
WSU	Weighted Spread of Uncertainty

On a similar note, a review on PHM for predictive maintenance [6] highlights robustness as a characteristic that differentiates preventive maintenance from PdM. Preventive maintenance strategies may lose validity given that the maintenance schedule is decided in the design phase, but if the operation deviates from the design, it can lead to a maintenance schedule that is no longer applicable.

Finally, interpretability is considered a crucial characteristic for prognostic models given legal regulations for algorithms in decision-making. In Europe, the General Data Protection Regulation (GDPR) entails the right to an explanation of how a decision was made for systems and frameworks that can affect a human being's life [7].

Therefore, based on the available literature on PHM and PdM, it is proposed that uncertainty, robustness, interpretability, and feasibility be considered essential characteristics of prognostic models for effective decision-making, in addition to ensuring high accuracy.

Under the uncertainty characteristic, the identification and quantification of the impact of inherent uncertainty sources on RUL predictions, as well as the potential reduction of uncertainty in RUL predictions, are considered. Robustness is the capability of a

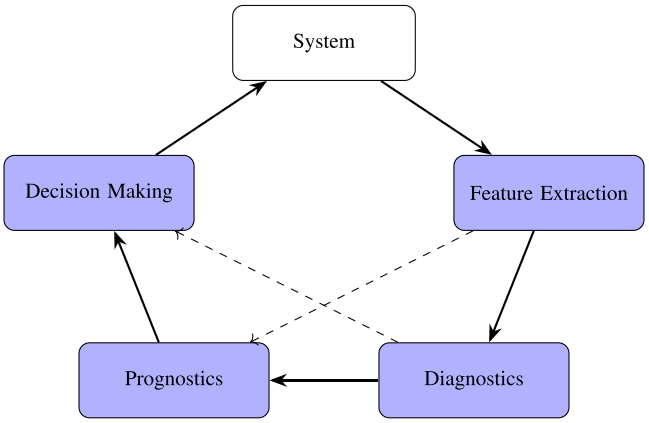


Fig. 1. PHM process flow diagram.

system to maintain high performance across diverse conditions. Interpretability means connecting the model's parameters with a physical aspect and explaining the predictions. Finally, feasibility is the ability of a model to be trained and perform well with a realistic amount of data.

It is important to note that among the four mentioned characteristics, uncertainty is crucial in prognostics. This is because predicting the RUL inherently involves uncertainties, and identifying and quantifying the sources of uncertainties provides critical information for the decision-making process. Therefore, a prognostic model should quantify uncertainty, ensure robustness, and provide interpretability and feasibility.

The goal of this literature review is to evaluate state-of-the-art prognostic models based on the aforementioned characteristics. Prior reviews in prognostics have been published, such as [8], which offers a general overview of prognostics within the PHM framework and is intended as a starting point for researchers in the field. Another review, [9], summarizes commonly used datasets in prognostics, discusses the construction and importance of health indicators, provides an overview of various prognostic approaches, and outlines the metrics commonly used to evaluate prognostic models. Although these papers contribute valuable insights to the field, the current paper provides a more focused and comprehensive examination of Data Driven Models (DDMs) in prognostics, while assuming familiarity with topics such as health indicators and evaluation metrics. In particular, we will analyze prognostic models through the lens of the four key characteristics: uncertainty, robustness, interpretability, and feasibility. In addition, this paper introduces a novel model evaluation framework designed to consider these characteristics and help in the selection of suitable DDMs for specific applications.

Before delving into the analysis of DDMs, it is essential first to understand the different prognostic approaches. For this purpose, the categorization from [10] is adopted, which includes three categories: Physics-based Models (PbMs), Data-Driven Models (DDMs), and Hybrid Models.

PbMs describe the physical process or evolution of the damage [11], which results in a complex model or in simple models that are unable to describe complex degradation processes. This approach has seen progress in recent years, with models developed for electrolytic capacitors [12,13], pneumatic valves [14], bearings [15], and metallic structures [16]. Despite good results, PbMs have limitations in validation, assumptions for operating conditions, and handling highly nonlinear equations, leading to increased computational costs. Additionally, PbMs are limited to components, constraining their application range as systems become more complex [17].

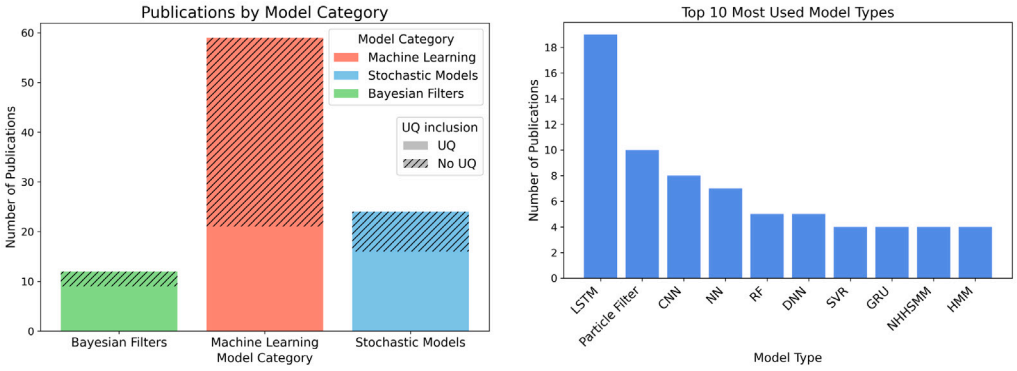
DDMs use historical data from sensors to identify characteristics of the damaged state of the engineering system. DDMs are widely used in prognostics because they can be applied without prior expert knowledge of the system and are flexible, allowing the same technique to be used for different systems or components [10].

Hybrid Models aim to bridge the gap between the incompleteness of PbMs and the need for data of DDMs, improving performance, reducing computational costs, and minimizing the data needed to train the model [18].

Most literature in the prognostic field focuses on DDMs, with a trend towards hybrid approaches, with DDMs at the core of the hybridization. DDMs are preferred for their flexibility and lack of need for prior system knowledge. However, when physical constraints are known, they can be integrated into DDMs.

Therefore, in this paper, prognostic models utilizing the data-driven approach are reviewed, with a focus on the four key requirements mentioned earlier. Given the extensive literature on DDMs, the techniques have been categorized into three groups: Machine Learning (ML), stochastic models, and Bayesian Filters (BFs).

The ML group includes models such as linear regression, decision trees, Support Vector Regressor (SVR), and Neural Network (NN), among others. Deep Learning (DL) is a subcategory of ML, encompassing models such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and many others. Under the category of stochastic models, Hidden Markov Models (HMMs) and Wiener processes, also referred to as Brownian motion, are included. Finally, BFs include Particle Filters (PFs) and Kalman Filters (KFs) and their variants. BFs can be used within a PbMs approach since they need a damage evolution function that can



(a) (b)

Fig. 2. (a) Publications by model category, including proportion of reported UQ. (b) Top 10 most used model types.

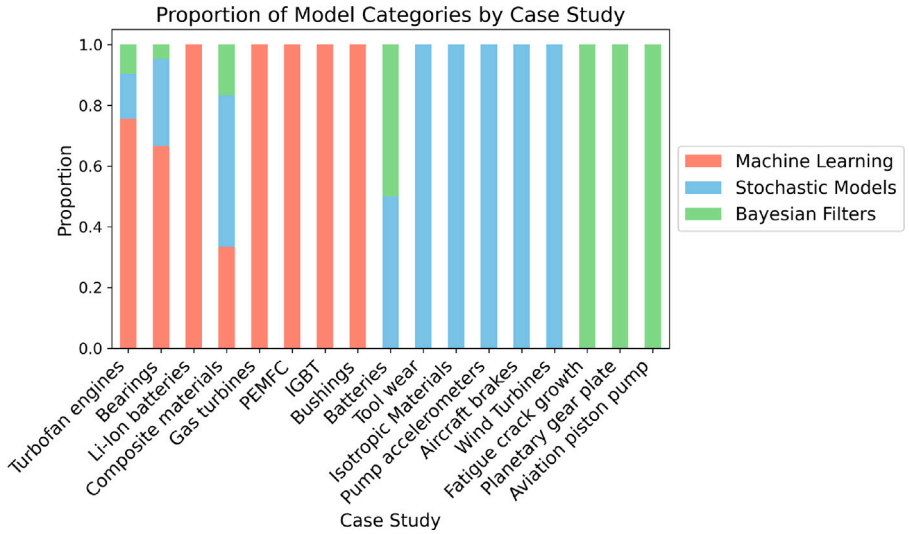


Fig. 3. Model category per case study.

be based on physical knowledge. However, they are also used in DDMs since this damage evolution function can be characterized with another data-driven model. Since this literature review focuses on DDMs, it will only cover BFs that rely solely on data-driven approaches within prognostic frameworks, excluding those that incorporate physical knowledge of the system.

Table 1 summarizes the selected publications that propose and evaluate data-driven prognostic models. The table is organized first by model category (ML, stochastic models, and BFs) and then by the engineering case study used for model validation. The model listed corresponds to the one performing the predictive step; for instance, in the case of BFs, while an ML model might be used for the damage evolution estimation, the BF is responsible for producing the final RUL prediction.

To give a clearer picture of the publications reviewed, Fig. 2 illustrates how the models are spread across different categories. In subplot (a), the number of publications by model category is shown, with hashed bars indicating those that do not report uncertainty quantification (UQ). This visual emphasizes that while ML dominates in terms of volume, a significant portion of ML-based works omit UQ, which is a crucial aspect of reliable prognostics, and it will be discussed later. Subplot (b) displays the top 10 most frequently used model types. LSTM models are the most widely adopted, followed by Particle Filters and CNNs. Variants of models are grouped under the same label (e.g., Bayesian LSTM and standard LSTM are both counted as LSTM).

Fig. 3 further explores the application of model categories across different engineering domains. It shows the proportion of each model category (ML, stochastic, BF) used in case studies, offering insight into the distribution of modeling strategies per context.

It is worth noting that the prognostic models listed in the table correspond to the model performing the predictive step. For example, in the case of BFs, an ML model may be used for the damage evolution equation; however, it is the BF model that performs the predictive step and provides the RUL as an output.

The publications were selected through a Google Scholar search using the keywords: “prognostics”, “robustness”, “interpretability”, “uncertainty”, and “feasibility”. Publications on ML models were chosen from 2018 onwards, as the number of such publications

Table 1
Summary of publications on the development of prognostic models included in the present literature review.

Group	Case study	Prognostic model	References
Machine Learning	Turbofan engines	SVR	[19]
		Bayesian RNN	[20]
		Natural and extreme gradient boosting, RF, MLP	[21]
		Bayesian Dual-Input-Channel LSTM	[22]
		Bayesian LSTM	[23,24]
		RF	[25]
		MLP	[26]
		LSTM	[27–32]
		DCNN	[33,34]
		NN	[35–37]
		Concise self-adapting deep learning network	[38]
		Multi-scale DCNN	[39]
		Multicellular-LSTM	[40]
		GRU	[41,42]
		Capsule neural network	[43]
		CNN	[44]
		Transformer	[45]
		Variational autoencoder	[46]
		DNN	[47,48]
		XGBoost, RF, logistic regression, feedforward NN	[49]
	Bearings	ϵ-SVR	[50]
		SVR	[51]
		Double-CNN	[52]
		Bi-GRU	[53]
		CNN	[54]
		Convolution-based LSTM	[55]
		GAN	[56]
		Bayesian DNN	[57,58]
		Graphical Convolutional Network	[59]
		Encoder–Decoder GRU	[60]
		MLP	[61]
		LSTM	[62]
		Self-surpervised LSTM-CNN	[63]
	Lithium-ion batteries	SVM	[64]
		LSTM	[65,66]
		Anti-noise adaptive LSTM	[67]
		Autoencoder CNN-LSTM	[68]
		Bayesian RNN	[69]
		Ensemble-Bayesian CNN	[70]
		Bayesian LSTM	[24]
	Composite materials	CNN, RF	[71]
		BNN	[72]
	Gas turbines	RVM	[73]
	Proton exchange membrane fuel cell	LSTM	[74]
	Insulated gate bipolar transistor	Self-attention NN	[75]
	Bushings	Naive Bayes, SVR, K-Neighbors regressor, linear discriminant analysis, XGBoost regressor, RF regressor	[76]

(*continued on next page*)

Table 1 (*continued*).

Stochastic Models	Bearings	SPC	[77]
		MCMC	[78]
		Adaptive skew-Wiener process	[1]
		Gaussian process	[79]
		HSMM	[80]
		Duration-dependent HSMM	[81]
	Composite materials	ANHHSMM	[82]
		NHHSMM	[83]
		SLHSMM	[84]
	Turbofan engines	HMM	[32,85]
		NHHSMM	[86]
		Wiener process	[87,88]
		Gaussian process	[89]
	Batteries	NHHSMM	[90]
		Gaussian process	[91]
		Nonparametric FPCA-Based Degradation Model	[92]
	Tool wear	HSMM	[93]
		GHMM	[94]
	Isotropic Materials	Multi-Branch HMM	[95]
	Pump accelerometers	HSMM	[96]
	Aircraft brakes	NHHSMM	[97]
	Wind Turbines	Self-data-driven statistical model	[98]
Bayesian Filters	Turbofan engines	Rao-Blackwellized PF	[99]
		Adaptive PF	[100]
		Discrete Bayesian filter	[101]
		PF	[102]
	Batteries	PF	[103–105]
	Composite materials	PF	[106]
	Fatigue crack growth	PF	[107]
	Bearings	EKF	[108]
	Planetary gear plate	High-order PF	[109]
	Aviation piston pump	Adaptive-order PF	[110]

is significantly higher and evolving more rapidly compared to other groups. Conversely, publications on stochastic models and BFs are more scattered across years. This variability is due to the challenge of finding relevant publications that address the specific topics covered in this literature review.

The paper is organized as follows: Section 2 explores the four key characteristics —uncertainty, robustness, interpretability, and feasibility —that are crucial for effective decision-making in prognostic models. Section 3 presents a case study that compares three prognostic models with different paradigms using the C-MAPSS dataset, evaluating them against these characteristics. Section 4 introduces a model evaluation framework designed to help users select the most appropriate prognostic model for their specific application. Section 5 provides an insight into the potential future research directions based on the challenges and research gaps found in the literature review. Finally, Section 6 discusses the findings and concludes the work.

2. Key characteristics in prognostics

Prognostic models must balance several critical characteristics to effectively support decision-making in PdM. While achieving high predictive accuracy is essential, it is equally important to address the inherent uncertainties in RUL predictions, ensure robustness under varying operational conditions, provide interpretability to comply with regulatory and practical requirements, and maintain feasibility for integration into real-world systems. These characteristics collectively ensure that prognostic models are both technically reliable and practically applicable for end-to-end PdM frameworks. The following subsections explore these key characteristics through a comprehensive literature review of prognostic models.

2.1. Uncertainty

Uncertainty Quantification (UQ) in prognostics refers to the process of quantifying the impact of various sources of uncertainty on the RUL prediction of a system. By definition, the prediction of RUL incorporates uncertainty because any prediction of the future is uncertain. Therefore, it is imperative to model RUL as a random variable rather than a deterministic one. The prediction of RUL is then used by a decision-making module, which will make health management decisions to fulfill PHM goals. Hence, to ensure the safety of an engineering system, it is important to have information on how uncertain the models are about the RUL predictions.

The "classical" categorization of sources of uncertainty divides them into aleatoric and epistemic. Aleatoric refers to the uncertainties that are intrinsic to the randomness of a phenomenon, hence, it is the uncertainty that the data holds. Epistemic refers to the uncertainty caused by a lack of knowledge, thus, the uncertainty comes from the model itself [111].

According to [112], uncertainty-related activities in prognostics are categorized into four processes: representation, quantification, propagation, and management. Representation involves choosing methods to model uncertain parameters, while quantification identifies and incorporates sources of uncertainty using statistical tools like Bayesian methods. Propagation predicts future states and RUL by accounting for quantified uncertainties through models and thresholds. Uncertainty management [112] reduces uncertainty by leveraging data to characterize the inherent prognostic uncertainties better. This process can minimize their impact on RUL predictions when possible, which is essential for effective decision-making.

Although UQ is essential for informed decision-making in prognostics, many publications omit it, providing only point estimates of RUL without confidence intervals. This trend is illustrated in Fig. 2(a), where the hashed portions of the bars represent studies that did not report UQ. The absence of UQ is particularly common in ML models, which are inherently deterministic. In contrast, BFs, show the highest rate of UQ reporting. This highlights a clear divide in UQ practices across model types in the prognostics field.

2.1.1. Machine learning models

UQ is a significant challenge within DDMs, and the methods used to quantify uncertainty often depend on the type of model employed. The low percentage of ML publications that address UQ may be attributed to the deterministic nature of ML models, which typically produce single-value outputs. To address this, various techniques have been developed to estimate epistemic uncertainty, though they often only approximate the posterior distribution. The posterior distribution cannot be directly calculated because it is intractable to calculate the marginal distribution; thus, there is no closed form for it. For instance, Variational Inference (VI) offers a good approximation of the posterior distribution, but it is still challenging to implement given its computational cost [113]. As a result, other techniques have arisen, such as Monte Carlo (MC) dropout, Deep Gaussian Processes, and Markov Chain Monte Carlo [114]. For a more detailed and in-depth explanation of these methods, readers can refer to the comprehensive review provided in [115], which also offers foundational guidance on their practical implementation in prognostics.

MC Dropout has been introduced as a technique to quantify epistemic uncertainty and is the most used one due to its simple implementation [116]. This technique approximates the posterior by randomly switching off neurons, given a dropout probability. The same architecture is run multiple times, and each dropout configuration corresponds to a different sample from the approximate posterior distribution.

However, MC Dropout struggles to approximate complex posterior distributions, which may lead to good approximations only in certain regions of the posterior distribution but poor approximations in others [117]. Even more, it has been questioned whether MC Dropout is truly Bayesian. MC Dropout fails sanity checks and is a design artifact since the posterior distribution converges to different values depending on the dropout probability assigned by a user [118]. Hence, these techniques, although easy to implement, do not always provide a good approximation of the desired distribution.

Building on these methods, several studies have applied these Bayesian techniques in prognostics and RUL prediction, each with varying levels of detail and success in capturing and interpreting uncertainty.

Some examples in prognostics are shown in [57] and in [22], where a point estimation of the failure time is made, but not the prediction of RUL through the operation time. Nevertheless, [22] includes an RUL prediction during the operation time by including a credible interval. To account for uncertainty, this study decomposes uncertainty into aleatoric and epistemic by using a Bayesian approach. However, when displaying the results, the distinction between the two sources of uncertainty is not made. The results are promising, yet the main drawback of this approach is the complexity of the model and its optimization, as the authors have claimed to be "extremely intractable and time-consuming".

In [70], an ensemble Bayesian CNN is proposed to predict the RUL of Li-ion batteries. Each model in the ensemble is trained on a separate run-to-failure trajectory to encourage diversity, and their posterior distributions are combined for prediction. The study compares this distribution stacking approach with traditional point prediction stacking, showing results through forecasted battery capacity and credible intervals. However, it does not clarify which uncertainty sources are captured or how uncertainty propagates across the ensemble, limiting the depth of its uncertainty analysis.

Other Bayesian approaches, such as [20], perform UQ including both aleatoric and epistemic uncertainty. However, similar to the previously cited work, it is not reported in the results how much each source contributes to the confidence intervals. Epistemic uncertainty is quantified with MC Dropout with a probability dropout value of 0.25, which is considered lower than the standard value of 0.5 [119].

In [24], a Bayesian DL model is proposed for RUL prediction with calibrated uncertainty using concrete dropout, where dropout probability is learned during training. This allows for a more flexible approximation of the RUL's probability distribution. The model separates aleatoric and epistemic uncertainties during calibration, leading to better performance than deterministic models

or untuned Bayesian ones. However, the model has limitations. It assumes a Gaussian RUL distribution, which may oversimplify uncertainty. Variance calibration assumes uniform miscalibration across the input space, ignoring potential regional variations. The use of isotonic regression for empirical CDF fitting risks overfitting on small datasets, although the original study used large datasets. Finally, the model assumes a linear, additive separation of epistemic and aleatoric uncertainties, which may not reflect their complex interdependence [115].

In [69], a Bayesian RNN is used with the dropout technique, however, they use a value of dropout between 0.05 to 0.2, which once again is considered low given that the standard dropout value. Low dropout values lead to narrow confidence intervals, meaning less estimated uncertainty in the RUL predictions, as shown in [32]. Thus, the choice of low dropout values can cause an underestimation of uncertainty that can be prejudicial for decision-making. Similarly, in [58], a CNN with a dropout value of 0.2 is proposed for epistemic UQ.

Beyond epistemic uncertainty, another critical aspect is aleatoric uncertainty, which many models overlook or only partially address. Aleatoric uncertainty in ML models is split into homoscedastic and heteroscedastic. Homoscedastic uncertainty corresponds to the noise in the data, and it remains constant through the whole dataset, while heteroscedastic uncertainty corresponds to the noise that varies with the input [115]. The few ML models that include aleatoric uncertainty include only one part of it. For example, in [120], a Bayesian DL framework is developed that takes into account heteroscedastic aleatoric uncertainty. In the already mentioned work of [20], only heteroscedastic aleatoric uncertainty is addressed. Similarly, in [24], aleatoric uncertainty is modeled as a simple linear decomposition of the total predictive uncertainty.

2.1.2. Stochastic models

Stochastic models inherently capture aleatoric uncertainty given that they model the randomness of the degradation process using a probabilistic approach [121]. Stochastic models use a prognostic measure to calculate the RUL; therefore, the RUL is represented as a probability density function (pdf) with a closed form for the posterior distribution. The confidence intervals of the RUL predictions are obtained by calculating the cumulative density function and, later, choosing the confidence level.

However, existing research on stochastic models often reveals wide confidence intervals. For instance, in [86], the NHHSMM model is employed, predicting a turbofan engine's RUL of 200 cycles. While the prediction is close to the actual RUL, the confidence intervals span from 300 cycles in the upper bound to 80 cycles in the lower bound. A similar situation occurs in [83] when using an NHHSMM applied to composite specimens. However, the same author later proposed an SLHSMM [84] that uses a similarity approach when training the model, which reduces the confidence intervals when calculating them, even when faced with unseen test data. The reduction of the confidence intervals is caused by the model leveraging data to manage and reduce RUL uncertainty, therefore, this model performs an uncertainty management step.

In a similar approach, [98] introduces a statistical/stochastic framework that addresses uncertainty arising from "unit-to-unit" variability. Wind turbines in operation often show different degradation patterns due to varying environmental and operational conditions. To manage this, the proposed framework selects the most appropriate model from a set of candidates, which helps reduce prediction uncertainty. By adapting to variability between units, the framework provides a more accurate representation of turbine behavior under diverse conditions.

It is important to remember that stochastic models include, by nature, only the aleatoric uncertainty. Epistemic uncertainty can be included through a time-consuming sensitivity analysis, Bayesian approaches, or imprecise probabilities. From the stochastic models reviewed, only [121] provides a model where epistemic uncertainty is quantified. The model called the Generalized Hidden Markov Model (GHMM) can identify both epistemic and aleatoric uncertainties by using imprecise probabilities. The results show that the GHMM can make more reliable decisions because the uncertainties can be differentiated. Yet, it is a computationally expensive model. Reliability refers to a model's or algorithm's ability to consistently deliver accurate and trustworthy outcomes. [122]. For instance, in the case of the GHMM, quantifying and differentiating the various sources of uncertainty enhances the reliability of the predictions, providing more detailed information.

2.1.3. Bayesian filter models

BFs can handle nonlinearity and non-Gaussian noise. A PF, which is a numerical approximation of a BF, estimates the system state by propagating a set of particles, each representing a possible condition. At each update step, particle weights are adjusted based on the similarity between predicted and observed sensor data. High-weight particles are more likely to be selected during resampling, focusing the estimate on the most probable states [123]. This iterative process is depicted in Fig. 4.

Additionally, as previously mentioned, PFs are combined with a model that characterizes the degradation evolution equation. In the case of a deterministic degradation model, such as an ML model, uncertainty is absent due to the deterministic nature of the model. However, part of the aleatoric uncertainty can be captured as new sensor data is used to update the weights of each particle. In the case of a stochastic degradation model, then aleatoric uncertainty is already present, and the output of a PF accounts for all the aleatoric uncertainty.

The fact that PFs, and BFs in general, by definition are updated based on new online data allows for the calculation of subjective uncertainty instead of population uncertainty, a characteristic that is preferred in prognostics [112]. Subjective uncertainty can be included in other frameworks, such as in the SLHSMM previously mentioned, however, these frameworks need modifications to achieve it, while BFs inherently include it. A key advantage of subjective uncertainty is that it accounts for the unique characteristics of each engineering system, given the specific conditions of its manufacture and operation. Therefore, the uncertainty should be calculated for the specific engineering system rather than relative to the population, as the goal in prognostics is to predict the RUL of an individual system based on the conditions under which it was operated [112].

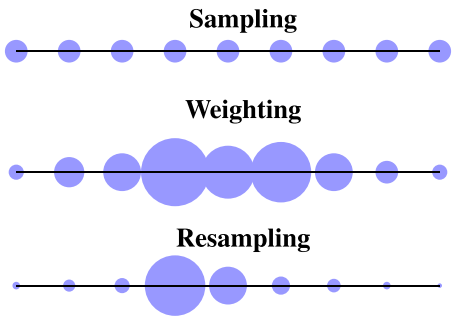


Fig. 4. Particle filter phases.

To improve efficiency in scenarios with slow or nonlinear fault progression, Lebesgue sampling has been proposed as an event-driven alternative to uniform time-based sampling [105]. Sampling is triggered when the fault indicator crosses predefined thresholds. An extension, Adaptive Lebesgue Sampling (ALS), dynamically adjusts these thresholds based on fault growth rate. When integrated with a PF under a Bayesian framework, ALS enables prediction along the fault state axis instead of time. This interaction reduces computational demands and long-horizon uncertainty, enabling efficient, real-time fault tracking and RUL estimation. In high-noise scenarios, however, ALS may revert to near-uniform sampling.

Furthermore, [102] proposed a hybrid approach combining PFs with a Wiener process to model unit-to-unit variability, referring to differences in degradation across systems due to varying conditions. A unit-specific parameter is inferred through the PF, enabling individualized RUL predictions while leveraging the stochastic properties of the Wiener process.

2.1.4. Challenges in uncertainty management

Given these considerations, it is clear that significant challenges persist in prognostics when it comes to quantifying uncertainty, especially in quantifying the contribution of each source of uncertainty in RUL predictions. Despite the challenges of DDMs, assume that all models can quantify the contribution of epistemic and aleatoric sources of uncertainty to RUL. Even in that ideal scenario, can uncertainty management be effectively performed? It is important to remember that epistemic uncertainty, stemming from a lack of knowledge, is reducible, but aleatoric uncertainty is irreducible. However, aleatoric and epistemic uncertainties often coexist, making it difficult to separate them [115]. In particular, for classification tasks, it was found that the rank correlations between both sources of uncertainty are within 0.8 to 0.99 [124]. Even more, when epistemic uncertainty is identified separately from aleatoric uncertainty, what kind of data is necessary to acquire to reduce the epistemic uncertainty? Is it just a matter of gathering more degradation histories? Some prognostic models were trained in big datasets, yet the confidence intervals of the RUL predictions remain wide. Consequently, managing uncertainty may not always be feasible under this categorization of uncertainty.

Therefore, a different categorization of uncertainties is needed to facilitate differentiation and enable uncertainty management. This categorization should be based on time, an inherent variable in prognostics. In addition, it should be subjective and focus on characterizing uncertainties specific to the system under study [84,98,102], rather than uncertainties in the population, as already mentioned. The need for a different categorization of uncertainty sources has been discussed in [112], where four sources of uncertainty were identified: present, future, model, and prognostic measure. This was further extended in [125] with the addition of a fifth source, past uncertainty.

The five proposed sources of uncertainty are as follows:

1. **Past Uncertainty**: Stemming from the manufacturing or assembly process and material quality.
2. **Present Uncertainty**: Arising from the lack of knowledge about the true state of health of an engineering system.
3. **Future Uncertainty**: The most difficult and important to address, as the future is unknown and it is impossible to foresee precisely environmental conditions, loading profiles, etc.
4. **Model Uncertainty**: Encompassing several aspects, such as model initial parameters and biases.
5. **Prediction Method Uncertainty**: Related to the uncertainty arising from the prognostic measure itself. In supervised techniques, such as ML models, model uncertainty and prediction method uncertainty become a single source.

It is important to note that past uncertainties are not uncertainties in the present once uncertainty management is performed. For example, if sufficient data is gathered about the manufacturing process, it is possible to manage past uncertainties and reduce their impact on RUL uncertainty.

2.2. Robustness

Robustness refers to the ability of a model to consistently provide accurate and consistent predictions, despite variations in operational conditions, environmental factors, and input data variability. This includes the model's ability to handle unexpected disturbances, sensor noise, incomplete data, and unseen data without a significant decline in performance.

In prognostics, the most concerning part comes with changes in the operational conditions or fault modes that cannot be taken into account in the training process. In addition, the operational conditions of the testing engineering system may differ from those observed in the training set. These issues can affect the performance of a prognostic model. Multiple publications on DL models [4,27,56] address robustness with "Domain Adaptation", referring to the techniques that reduce the negative effects in the performance of a prognostic model when dealing with domain shifts.

Nonetheless, despite the importance of robustness to ensure safety and reliability in the operation of a system, there are still big challenges to developing techniques that guarantee robustness to prognostic models.

2.2.1. Machine learning models

In [4], a review of ML models is provided, highlighting the challenges in these models. A consistent challenge with ML models is their tendency to exhibit higher errors under varying operational conditions. Nevertheless, several efforts have been made to address this problem. In [33], a Deep Convolutional Neural Network (DCNN) is used to estimate the RUL under different fault modes for turbofan engines. To do so, an adaptive batch normalization is integrated into the model. The results show an improvement when using adaptive normalization, however, the results still have high errors and are volatile. In [27,56], a domain adversarial neural network is validated under two case studies for turbofan engines and bearings. The same problems remain, with higher error rates and noisy results. One limitation that is highlighted is that better performance could be achieved with these models when training with more data; however, in industrial scenarios, data is scarce.

Transfer Learning has also been used to tackle robustness [47]. An important part of this framework is data alignment. However, it assumes a straight degradation trace pattern. This is a strong assumption, as the degradation process can be more complex and may not accurately capture the behavior of different fault modes. In addition, the authors mention that a main limitation of this framework is the availability of "no healthy" state data in industrial applications.

2.2.2. Stochastic models

Moving onto different prognostic models, stochastic models, in particular, HMMs, have been able to provide a methodology when testing the model with unseen data. In [82], an Adaptive Non-Homogeneous Hidden Semi-Markov Model (ANHHSMM) is used, which is an extension of the Non-Homogeneous Hidden Semi-Markov Model (NHHSMM). The model is trained with the degradation histories of 8 open-hole specimens that experienced fatigue loading and tested with 3 open-hole specimens that went under the same fatigue loading and in-situ impact loading. Impact loading can influence the lifetime of specimens, either reducing it or, counterintuitively, increasing it. To address this change in the lifetime of the specimens, the model adapts its parameters after the impact and, therefore, reports better results than non-adaptive models. The significance of this model lies in its ability to adapt without requiring a large amount of data, thus, a stochastic model can overcome the limitations of ML models. Nonetheless, the ANHHSM has its limitations, the most prominent being that it has not been tested under changes to the primary loading condition. Instead, it has only been applied when operating conditions change for a few seconds, such as during impact loading.

Another publication addresses this related problem by using an SLHSMM [84,126], which is another extension of the NHHSMM. The proposed framework characterizes the similarity between a testing degradation history and training data to perform a re-estimation process. The SLHSMM can provide good results when facing outliers (in this case, specimens that experienced impact loading) when compared to other frameworks, such as Gaussian process regression and NHHSMM. Even more, SLHSMM can reduce the confidence intervals and computational cost. The main limitation of this framework is the dependency that it has on the outliers present in the training set, meaning that for the SLHSMM to provide good results, it needs diverse data in the training set.

2.2.3. Bayesian filter models

In [104], a framework combining PFs and neural networks is developed to adapt to different scenarios by automatically detecting anomalies in a battery's expected behavior. The framework is not computationally expensive, and it can be run online. Another advantage of the framework is that it does not need physical knowledge or huge amounts of experimental data. The results show the capabilities of the framework to detect anomalies; however, RUL predictions tend to be volatile, and convergence to the true RUL is reached in the last cycles of the batteries.

2.3. Interpretability

Interpretability refers to the ability to understand and explain the predictions or results generated by prognostic models, meaning also that it is possible to relate the model parameters to a physical meaning [127]. Interpretability is an important aspect of prognostics because it helps users make an informed decision based on the predictions by knowing how the predictions were derived. Even more, based on European regulations, the General Data Protection Regulation (GDPR) [7] entails the right to explanation when it comes to algorithms in decision-making. Therefore, if a decision was made based on an algorithm and this decision can affect a person, then the person has the right to know how that decision was made. In terms of PHM, some components are critical and can compromise the safety of people, hence, PHM can also be affected by this regulation.

2.3.1. Machine learning models

The aspect of interpretability has gained attention over the years in the PHM community due to the increased use of ML models, which are mostly considered a "black box". However, within the ML community, the term "explainability" is frequently mentioned

and sometimes is used as a synonym for interpretability. Nonetheless, explainability refers to creating a model that can explain its actions, and interpretability refers to interpreting and comprehending the model's results [128]. In a review of DL applications for PHM [129], interpretability is considered a significant challenge, with current approaches often falling short in providing interpretability for physical systems. Consequently, most papers in prognostics that explicitly address interpretability are related to ML models.

Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) are widely used tools for interpretability. LIME focuses on local explanations by analyzing a single prediction through perturbations applied to the specific data point in question. Conversely, SHAP focuses on both local and global explanations. Global explanations in SHAP are used to understand the overall importance of each feature across the entire dataset, while local explanations are given by calculating how each feature contributes to predictions on a specific instance. Both of these tools have been applied to prognostic models based on ML in a few studies [42,48,76], primarily focusing on feature selection and feature importance.

Limitations of LIME and SHAP. Both LIME and SHAP exhibit notable limitations, particularly in their robustness and consistency. LIME's primary drawback lies in the unreliability of its generated explanations. Studies such as [130,131] highlight that LIME can produce significantly different explanations for inputs that are either identical or nearly identical. This inconsistency is a serious challenge, especially in prognostic models where stable and reliable explanations are essential for informed decision-making.

SHAP, similarly, has been shown to assign misleading relative feature importance in certain scenarios. To achieve computational efficiency, SHAP relies on approximations of Shapley values, but these approximations can diverge substantially from exact SHAP scores, particularly in models with a large number of features [132]. Exact SHAP scores aim to provide a precise measure of feature importance by evaluating all possible coalitions of features, yet their computation is impractical in most real-world applications due to exponential complexity. While approximations, as implemented in tools like SHAP, use heuristic or sampling-based methods, they can fail to preserve the true ranking of feature importance. Compounding this, exact SHAP scores themselves are not immune to issues; they may assign non-zero importance to irrelevant features or undervalue critical ones. This stems from the fact that not all feature subsets are equally relevant for interpretability in machine learning, contrasting with the assumptions of game theory [6]. Such limitations in both exact and approximate SHAP methods raise significant concerns about their reliability, particularly in high-stakes prognostic models where inaccuracies can compromise safety and outcomes.

In addition to the previously discussed issues, tools like LIME, SHAP, and similar methods often face a "disagreement problem", where they produce conflicting feature rankings. For example, if LIME and SHAP yield significantly different rankings for the same prediction, it can lead to confusion in interpreting the behavior of an engineering system. This ambiguity complicates decision-making, particularly in critical PHM applications, where understanding the root cause of a failure is essential for determining the appropriate corrective actions. Such inconsistencies result in unstable explanations that can have serious repercussions.

To mitigate this challenge, a metric called the "trust score" was proposed in [49] as a way to identify the most suitable explanation tool for a given scenario. While promising, the trust score has notable limitations. It has only been validated on simple DL architectures and does not address the root cause of biases in the generated explanations. Consequently, these tools and metrics still fall short of achieving the level of interpretability required for DL models, especially in high-stakes applications.

Alternative interpretability methods. Other approaches have been used to provide interpretability to ML models. In [35], variational Bayesian inferences were used to create a framework called a structured-effect neural network that quantifies how a change in a sensor measurement affects RUL predictions. In [46], a variational encoder is used to create a latent space that groups the degradation histories into different clusters. Once this latent space is created, it can be used to make a visual representation that is used for "interpretable diagnosis". When new unseen data is used as input, then the encoder will place this data in proximity to data points whose diagnosis is already known, in this way giving a visual interpretation of the data. Another approach for interpretability for data-driven prognostics is Relevant Vectors Machines [73], however, it only provides information about whether the degradation histories are linear, smooth, or stationary. Therefore, a more complete framework is needed to interpret the model and its predictions.

Many attempts have been made to give some explanation to RUL predictions about the features used by using methods such as LIME or SHAP to assess the importance of the features. However, DL and ML models are black boxes by nature, and even though there is a research area dedicated to providing interpretability to them, this area still has many challenges of its own.

2.3.2. Stochastic models

In [72], an NHHSMM, an extension of the HMM, is utilized to model the physical degradation process of composites. The model employs four hidden states intended to reflect the stages of composite degradation: matrix cracking, delamination, fiber breakage, and failure stage. While these hidden states have the potential to align with the aforementioned stages, this correlation has not been strictly defined.

In prognostics, HMMs are often constrained to a left-to-right transition structure, reflecting the irreversible nature of most degradation processes, except in cases of self-healing materials or maintenance. This structural restriction ensures that the model accurately represents systems that deteriorate progressively over time. Additionally, once state transition probabilities are estimated, metrics such as the average time spent in each damage state can be directly computed, providing intuitive insights into the physical degradation process. The observation distributions further clarify the behavior of sensor measurements or health indicators within each damage state.

A notable advantage of HMMs, derived from their unsupervised learning nature, is their ability to predict both the RUL and the current damage state, alongside estimating the time a system will spend in each damage state. Enhanced variants further improve

interpretability. For instance, the SLHSMM [84] provides an online estimation of similarity between the testing system and training systems, while the ANHHSMM [82] identifies when a testing system is an outlier. These advancements equip users with a deeper understanding of both the model's predictions and the underlying data, increasing trust and usability in critical applications.

2.3.3. Bayesian filter models

Finally, interpretability has not been addressed explicitly for BFs, similar to stochastic models. However, the nature of BFs provides interpretability, depending on the level of complexity. Consider the example of PFs to understand how these models can provide interpretability.

In this process, particles represent potential scenarios, making understanding and interpreting the model's predictions easier. When the particles are updated with new sensor data, it becomes clear which particles are assigned higher weights. This allows for the visualization of probability distribution functions, providing insights into the algorithm's behavior and the system's dynamic evolution. Additionally, a comparison can be made between the current estimated observation of the BF and the actual sensor data, enabling users to evaluate whether the BF accurately estimates the current observations.

2.4. Feasibility

Feasibility is defined as a model's ability to be trained and achieve accurate results with the available data, which varies by industry. Available data may be limited in number, contain missing or poor-quality entries, include noise, lack diversity, or present challenges in labeling.

For example, in the aviation industry, most of the data available corresponds to nominal operations and healthy states. The lack of diversity in the data leads to a challenge when predicting in different scenarios [133]. However, in general reviews of PHM [3,134], a challenge that is persistently found is the amount of data available to train the models, since in many industrial applications, there is no labeled data or missing measurements which makes it harder to create training datasets for prognostic models. To better understand the current state-of-the-art in prognostics on this topic, first examine the most commonly used datasets for validating prognostic models.

As shown in Table 1, a large number of publications validate their models using the case study of turbofan engines, coming from the C-MAPSS dataset [135]. The C-MAPSS dataset corresponds to simulated data on turbofan engines and is made of 4 sub-datasets, each one with more than 100 degradation histories for the training phase. These 4 sub-datasets have different conditions and fault modes, however, most papers only focus on the first sub-dataset, as it only has one condition and one fault mode.

This dataset has become a benchmark in the prognostic field, and as a consequence, it has helped to establish a comparison between models. However, it has had negative effects in the community since several publications only use C-MAPSS to test their models without taking into account the fact that in real-life scenarios there is not that amount of available degradation histories to train a model, and the data are artificially generated since it comes from a simulator that has its simplifications.

For the models that were validated with a case study on bearings, the majority use data from PRONOSTIA (also called FEMTO) [136]. PRONOSTIA is an experimental platform that can provide real data on the accelerated degradation of bearings for diagnostic and prognostic purposes. PRONOSTIA was used to generate the dataset "IEEE PHM 2012 Prognostic Challenge", which is composed of real data of bearings with accelerated degradation under three different conditions. The dataset has a training set of 6 degradation histories and a testing set of 11 degradation histories.

Another important case study that has a strong prevalence in prognostics is lithium-ion batteries, which have gained attention in PHM since they can play an important role in current technologies, from cell phones to electric cars and aircraft. Therefore, batteries will play a key part in the transition to sustainable transportation and renewable energy storage. But to do so, there is a need for reliability, which can be obtained via the accurate estimation of the state of health and the state of charge [137].

Given the importance of batteries in PHM, there are more than 20 datasets available for prognostics. However, the most popular ones are the Prognostic Center of Excellence (PCoE) Battery Dataset, Randomize Battery Usage Dataset, and CALCE Dataset. In PCoE, with 34 batteries that are cycled to 70% or 80% of the initial capacity, and are being charged and discharged at different temperatures [138]. In the Randomized Battery Usage Dataset, 28 cells are continuously cycled with current profiles generated by random walks, providing a benchmark for state of health estimation [139]. Finally, the CALCE Dataset has been used primarily for the state of health estimation. The dataset contains different discharge profiles and cutoff voltages to simulate real-life scenarios.

2.4.1. Machine learning models

ML models need large and labeled datasets for training and validation, which has been pointed out in reviews for prognostic [4] as a particular challenge since in industrial applications, datasets with those characteristics might not be available. In publications in the prognostic field, it is not addressed how the model's performance is affected by changes in the amount of data. However, there are attempts to address this issue, usually under the name "few-shot prognostics", which will be detailed below.

In [60], a Bayesian approximation enhanced probabilistic meta-learning (BA-PML) algorithm is used for prognostics with limited data. BA-PML integrates variational inference techniques within a meta-learning framework, enabling the estimation of uncertainty and the generation of probabilistic predictions. Unlike traditional ML approaches that rely on point estimates, BA-PML provides interval estimates, thereby offering a more comprehensive understanding of prediction uncertainty. By adopting an encoder–decoder architecture as its base predictor, BA-PML is capable of handling variable-length predictions, crucial for degradation prediction tasks common in machinery prognostics. Furthermore, the algorithm's episodic training stage facilitates cross-domain adaptation through fine-tuning to ensure robust performance in real-world applications where data distributions may vary.

The model was tested with bearings. The few-shot prognostics are referred not only to a limited amount of degradation histories but also to a limited number of samples within a degradation history. The results are promising, although for some cases the predictions are extremely noisy, and the RMSE of the predictions is higher when the amount of available samples increases. One of the limitations and future work for this study is the development of a more concise episodic training to reduce the computational cost of the framework.

A graph neural network is used in [59] in the context of few-shot prognostics. Instead of using traditional methods based on Euclidean measures, this approach creates dynamic graphs that show how different factors interact over time. By analyzing these graphs, the model can uncover hidden patterns and predict the remaining lifespan of the machines more accurately. First, it organizes the data into these graphs, considering various information channels. Then, it uses a smart method to decide how different parts of the machine relate to each other in these graphs. Finally, it trains a special kind of network called GSDeMLN, which learns from the graphs to make predictions about machine health, even when conditions change.

Another issue that ML models face when using supervised techniques is the need for labeled data. In [69], this challenge is addressed with a Bayesian DL-based approach. The proposed method preprocesses historical equipment monitoring data and field data to create samples labeled with degradation information. It then employs an RNN and incorporates a variational inference technique to quantify uncertainty in RUL prediction. By converting degradation uncertainty into RUL uncertainty from a reliability theory perspective, this approach helps make more informed maintenance decisions since it provides a probabilistic understanding of when a system or component is likely to fail.

In [63], the authors propose a self-supervised LSTM-CNN framework to tackle the challenges of sparse and unlabeled data in bearing prognostics. The model leverages contrastive learning during pretraining to extract discriminative temporal-frequency features from raw sensor signals, significantly reducing the dependence on labeled data. A subsequent supervised fine-tuning stage enables effective application to downstream tasks. This approach improves the practicality of predictive maintenance in label-scarce environments. However, the authors acknowledge that the method incurs high computational overhead, which may limit its deployment in resource-constrained settings.

2.4.2. Stochastic models

For stochastic models, feasibility is not a critical issue, as they can be trained with small datasets. For instance, in [83], an NHHSMM was trained using only 8 degradation histories. Stochastic models do not require large datasets, nor do they need labeled data. Moreover, models such as ANHHSMM or SLHSMM are particularly well-suited for few-shot prognostics, as they can adapt to or utilize just a single degradation history, respectively. Furthermore, [92] presents a nonparametric model based on functional principal component analysis and stochastic modeling to handle fragmented data. Fragmented data is defined as cases where sensor readings are missing during random periods of the degradation process due to monitoring interruptions or data loss. The model is tested under different levels of missing data and shows that RUL predictions remain accurate even with up to 25% of data missing.

2.4.3. Bayesian filter models

Finally, BFs have also proved to be feasible since they can have a good performance when trained with a small dataset. It is worth mentioning that BFs need an equation that describes the evolution of degradation of the engineering system. In the case of BFs with a DDM approach, it is important to take into account which models characterize the evolution of degradation and how much data is needed for this purpose. However, if only the prediction part is considered, BFs have the substantial advantage that it is not necessary to have a complete run-to-failure trajectory. Therefore, BFs are particularly feasible for applications where there is missing data.

3. Case study

In this case study, three models are compared based on the four key characteristics discussed throughout the paper. For the ML models, an LSTM is employed, as it is widely recognized in the prognostics field and typically delivers the best accuracy, inspired by the model developed in [28]. For stochastic models, the Adaptive Hidden Semi-Markov Model (AHSMM) is utilized, notable for its adaptability to unseen data, drawing inspiration from the framework presented in [82]. Finally, for BFs, a PF is applied, chosen for its popularity and proven performance across various engineering systems, based on the approach described in [99].

This comparison spans different paradigms of DDMs, with data sourced from the C-MAPSS dataset. The accuracy of the models is evaluated using the Root Mean Square Error (RMSE) metric complemented by the standard deviation (SD) of RMSE values across all test set predictions. Multiple experiments are conducted to evaluate each model's performance across the four key characteristics.

3.1. Data pre-processing

All three prognostic models are trained on data from a single sensor to ensure a fair comparison under identical input conditions. This deliberate simplification isolates each model's learning capability by minimizing confounding effects from feature engineering or data fusion. Although DL models typically perform better with rich, multivariate inputs, this study prioritizes a controlled evaluation of model behavior over maximizing predictive accuracy. As a result, the use of a single sensor intentionally limits absolute performance but enables a clearer analysis of RUL prediction characteristics.

The baseline case utilizes data from the first sub-dataset, FD001, with Sensor 11 chosen for its superior scores in prognosability, trendability, and monotonicity. These scores, along with the fitness value (calculated as the average of the three scores), are

Table 2
Prognosability, trendability, and monotonicity score per sensor data.

Sensor	Prognosability	Trendability	Monotonicity	Fitness
2	0.81	0.39	0.66	0.62
3	0.76	0.31	0.63	0.57
4	0.87	0.58	0.77	0.74
7	0.84	0.46	−0.75	0.68
8	0.66	0.01	0.67	0.45
9	0.33	0.01	0.42	0.25
11	**0.87**	**0.7**	**0.81**	**0.79**
12	0.85	0.55	−0.78	0.73
13	0.66	0.01	0.68	0.45
14	0.31	0.01	0.26	0.19
15	0.84	0.46	0.71	0.67
17	0.79	0.34	0.64	0.59
20	0.83	0.43	−0.7	0.65
21	0.81	0.47	−0.7	0.66

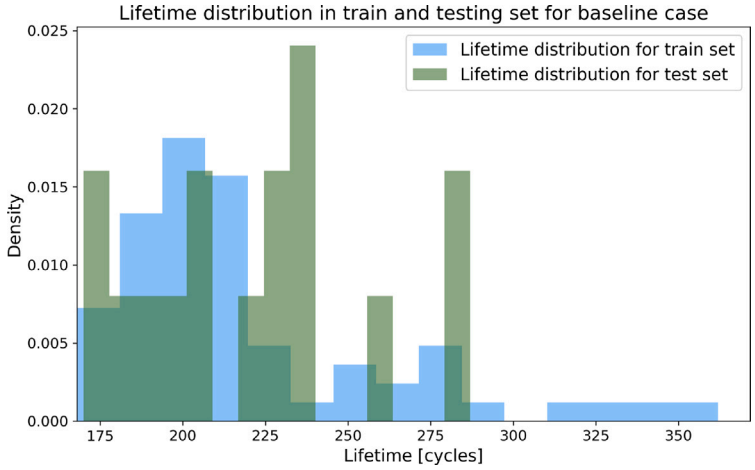


Fig. 5. Lifetime distribution for baseline case.

presented in Table 2. These characteristics are considered desirable for inputs in prognostic applications [140]. To process the sensor data, it is discretized into 20 clusters using the K-Means algorithm. The number of clusters is determined based on the Monotonicity Index (MI) [141], which helps identify the optimal number of clusters that can effectively represent the degradation process.

The C-MAPSS dataset provides 100 complete run-to-failure trajectories in the training file and 100 partial trajectories in the test file. For this study, we chose to work exclusively with the complete run-to-failure trajectories, splitting the original training set into 64 training and 16 testing instances. This approach ensures that true RUL labels are available for the full operational lifespan of each engine, which is essential for accurately computing evaluation metrics that require knowledge of the true EOL (e.g., coverage of uncertainty intervals across the entire life cycle). Using the provided test set would not allow for this level of evaluation consistency, as the trajectories are incomplete. Therefore, this split better supports the study's focus on model behavior and UQ. Fig. 5 presents a histogram showing the lifetime distributions of the training and testing sets.

Using the baseline training data, the hyperparameters of the three models were optimized. The architecture of each model and the corresponding training process are explained below.

The input for the LSTM consists of windows of three samples. This choice is driven by the need for the LSTM to receive inputs of consistent length and to be utilized in an online manner, necessitating the data to be windowed. The window length was selected based on its superior performance during validation.

The architecture of the LSTM is illustrated in Fig. 6. The model incorporates dropout to enhance accuracy and, subsequently, to account for uncertainty in the predictions through the use of the MC Dropout method. The number of neurons per LSTM layer and the dropout probability were determined via a random search.

The AHSMM used in this case study is based on the model proposed in [82]. It features an HSMM with 7 damage states, selected using the BIC criterion [86]. While adding more states could have slightly improved the BIC, the increased computational cost outweighed the benefit, leading to the choice of 7 states. In this HSMM the sojourn times are defined using a Weibull distribution. The Weibull distribution is determined by two parameters, shale and scale. An adaptive module modifies these sojourn times by adjusting the scale parameter after a transition between hidden states occurs. The scale parameter is adjusted based on the ratio of the expected sojourn time from the trained parameter to the actual time spent in a particular hidden state, considering the degradation history (see Fig. 7).

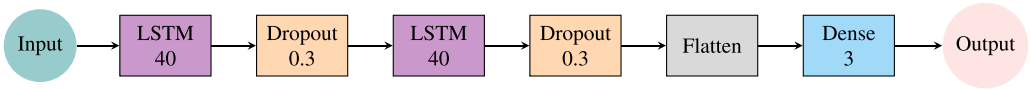


Fig. 6. LSTM architecture for prognostics.

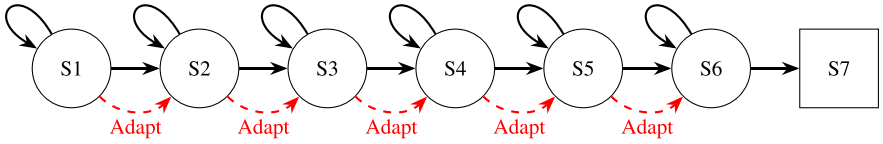


Fig. 7. HSMM with 7 states (last state observable) and Weibull-distributed sojourn time with adaptation mechanism.

Table 3
Results of prognostic models for baseline case.

Model	RMSE	SD
LSTM	38.05	8.91
AHSMM	34.48	9.34
PF	**24.05**	**15.69**

The PF used in this case study is inspired by the work presented in [99]. Specifically, a PF employing a similarity approach is utilized. The degradation model is represented by a degree 4 polynomial, chosen for its ability to capture nonlinear degradation trends while avoiding overfitting. This degree provides adequate flexibility without introducing unnecessary complexity. The similarity metric applied is the Maximum Mean Discrepancy (MMD) [142].

The framework operates as follows:

- Training Phase: Each training history is fitted to a polynomial, and the polynomial parameters are saved.
- State Transition Function: A particle, which is a set of parameters defining a curve, is perturbed with random noise.
- Observation Function: This function evaluates the value of the curve defined by a particle's parameters at a given time step.

For the testing phase, sensor data up to the current time step is collected. The MMD is used to identify which training histories are most similar to the current data. Training histories with an MMD value below a specified threshold (set to 0.01 in this paper) are considered. If no training histories meet this threshold, only the history with the best MMD value is used.

The PF is initialized using parameters derived from the fitted polynomials of the most similar training histories. Identifying these similar training histories requires access to some samples of the testing data. As a result, the PF can only begin providing predictions after a predefined number of samples, which in this study was selected to be 10 data points, have been observed during online operation. Consequently, in the upcoming prediction plots, the PF is the only model that does not generate predictions from the start. Using 150 particles, the PF achieves the desired particle count by randomly perturbing each set of parameters.

The PF operates until the mean value of all particles reaches the failure threshold, indicating that the predicted sensor values have reached a failure state. At this point, the PF is stopped. Fig. 8 summarizes the training and testing process for the PF.

3.2. Baseline

The results of the baseline case are summarized in Table 3. Among the three models, the PF outperforms the others with the lowest RMSE, although it has the highest standard deviation. The AHSMM model achieves an RMSE of 34.48, while the LSTM model shows a slightly higher RMSE of 38.05. Notably, the PF's superior performance is particularly impressive given its simplicity, relying on a fitted polynomial as the degradation model.

The results are visually represented by presenting the predictions for each model for three selected engines, as shown in Fig. 9. These engines were chosen based on their lifetimes: engine #37, which has the shortest lifetime in the test set; engine #81, which represents the median lifetime; and engine #2, which has the longest lifetime. For clarity, only the mean predicted RUL is shown, as including confidence intervals would complicate the visualization.

The visualization reveals that the predictions from both the PF and the LSTM models exhibit volatility, particularly in the case of LSTM. In contrast, AHSMM predictions are more stable, although they include noticeable jumps corresponding to transitions between hidden states. All three models perform better during the final cycles of the system's life. While this is beneficial for safety, as it indicates when the system needs to be decommissioned, it is less advantageous for predictive maintenance.

3.3. Uncertainty

As discussed in Section 2.1, dealing with uncertainty is a significant challenge in prognostics. Comparing models that capture different sources of uncertainty is even more difficult. In this section, an effort is made to draw such a comparison using two metrics: coverage and Weighted Spread of Uncertainty (WSU).

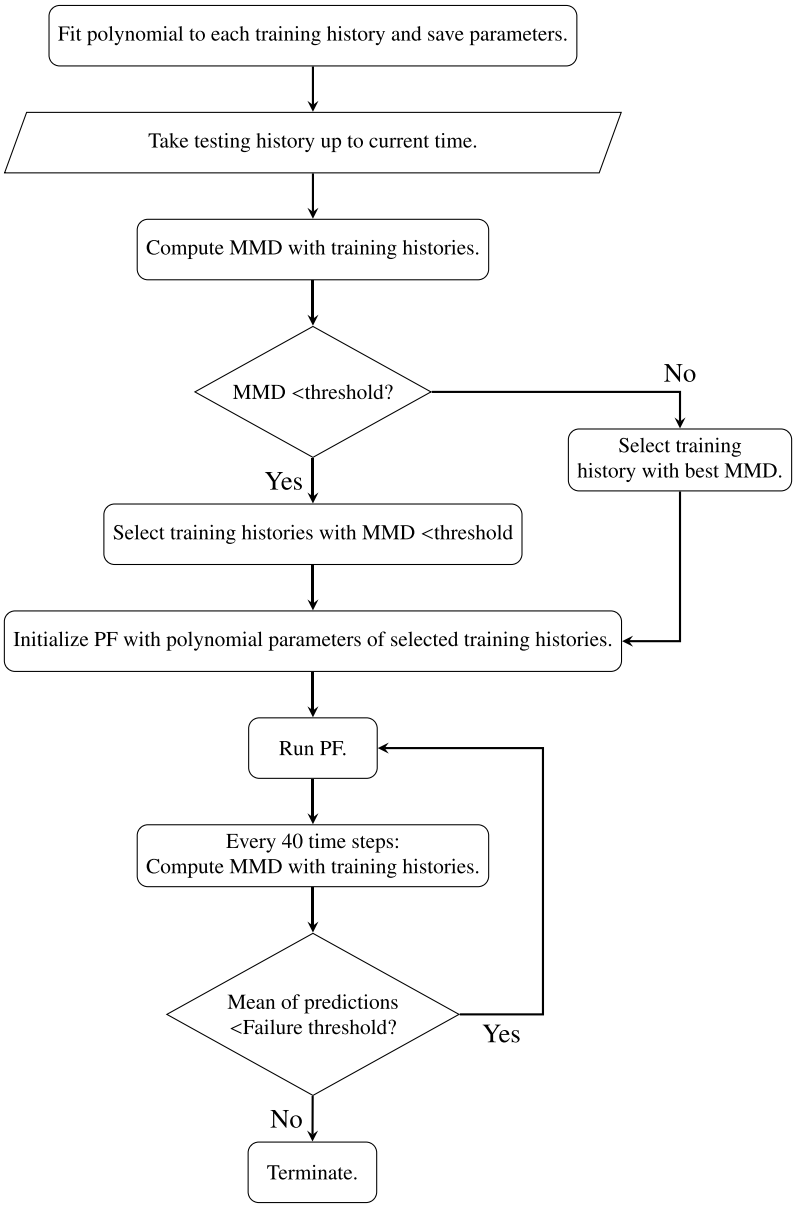


Fig. 8. Similarity PF algorithm.

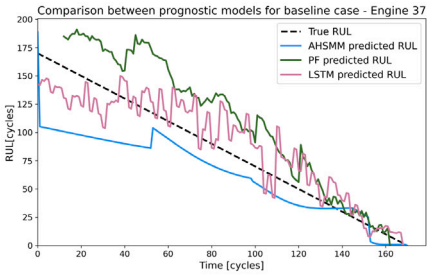
In this paper, the coverage metric is established to quantify how well the predicted intervals encapsulate the true outcomes over a series of observations or time steps. Therefore, it provides insight into whether the model's confidence intervals appropriately capture the variability in the data.

The coverage metric is defined as the proportion of true values that fall within the predicted confidence intervals. For a given predictive model, if y_t represents the true value at the time step t, and $[l_t, u_t]$ represents the predicted lower and upper bounds at the same time step, then the coverage for a single prediction can be expressed as:

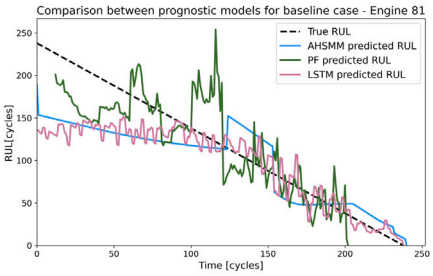
$$C_t = \begin{cases} 1, & \text{if } l_t \leq y_t \leq u_t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The coverage metric for predicting a complete degradation history of length T is calculated as the average of C_t. Hence, a coverage value of 1 indicates that all true values fall within their respective predicted intervals.

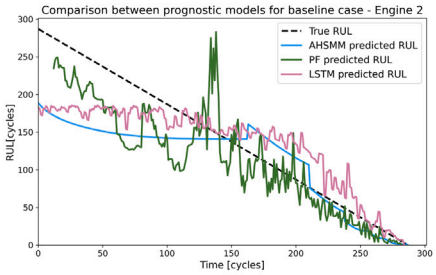
$$\text{Coverage} = \frac{1}{T}\sum_{t=1}^{T} C_t \tag{2}$$



(a) RUL predictions for engine #37.



(b) RUL predictions for engine #81.



(c) RUL predictions for engine #2.

Fig. 9. Results from prognostics model for three different engines in the baseline case.

Table 4
Results of prognostics models in terms of uncertainty for baseline case.

Model	Coverage	WSU
LSTM	0.33	680 357.55
AHSMM	0.97	3194276.47
PF	0.17	385 141.6

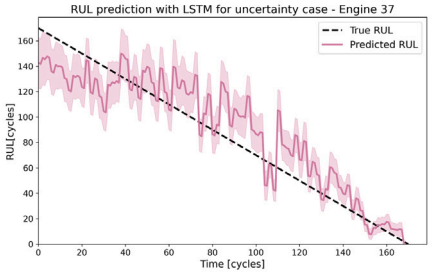
The WSU metric [32] is shown in Eq. (3). The expression calculates the area between the confidence intervals while penalizing wider confidence intervals at the end of the lifetime. The penalization is considered because the longer the time that has passed, the more information is available. Therefore, it is expected that the predictions hold less uncertainty as time passes.

$$WSU = \sum_{t=1}^{T-1}(t - t_0)\left(\frac{u_{t+1} + u_t}{2} - \frac{l_{t+1} + l_t}{2}\right) \tag{3}$$

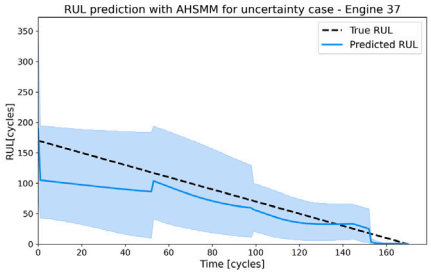
Table 4 presents the average coverage and WSU for the baseline test set. It is worth noting that the confidence intervals represent the 95% confidence level of the RUL probability density functions. The results indicate that AHSMM achieves the best coverage, with nearly perfect results. However, this high coverage comes at the cost of wide confidence intervals, as evidenced by AHSMM's highest WSU value. This trade-off becomes apparent when visualizing the confidence intervals in Fig. 10, where AHSMM's intervals are excessively wide, making them non-informative from a decision-making standpoint. This pattern is consistent with other stochastic models discussed earlier in Section 2.1.

In contrast, both the LSTM and the PF models exhibit low coverage percentages, making their predictions unreliable for decision-making since their confidence intervals fail to contain the true RUL values, leading to overconfident or inaccurate predictions that could lead to failures for PdM frameworks and a decrease in their trustworthiness.

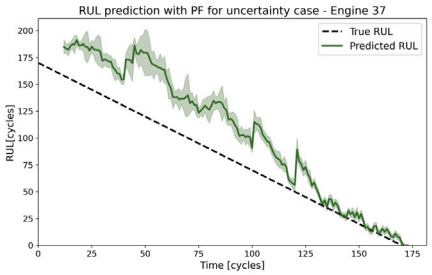
It is advisable to consider the coverage metric when calibrating uncertainty to develop more reliable models, ensuring it reaches a threshold that supports informed decision-making. In the case of LSTM and PF, this calibration is relatively straightforward since the depicted uncertainty depends on user-defined parameters: dropout probability for LSTM and process and observation noise for PF. However, adjusting these parameters can affect the accuracy and uncertainty coverage of the models. For example, using different dropout values can cause noticeable changes in coverage as shown in [32]. However, for this work, the parameters chosen were selected for their optimal accuracy. The critical question then becomes: How reliable do the models need to be for effective decision-making?



(a) LSTM RUL predictions.



(b) AHSMM RUL predictions.



(c) PF RUL predictions.

Fig. 10. Comparison between prognostics models for uncertainty case.

However, AHSMM faces the opposite challenge: reducing confidence intervals without significantly compromising coverage. For example, a similarity-based approach, as demonstrated in [84], has been shown to reduce confidence intervals. Furthermore, exploring alternative prognostic measures could further aid in narrowing the spread of these intervals.

This analysis does not explore the complexities of uncertainty categorization, as implementing such mechanisms in these models would require further development beyond what is covered in the reviewed literature. Instead, the focus is on understanding the capabilities of state-of-the-art models and assessing whether the calculated confidence intervals, along with the RUL predictions, are reliable for decision-making purposes. Moreover, the analysis may not provide a completely fair comparison, as the models capture different sources of uncertainty; for instance, LSTM captures epistemic uncertainty, while AHSMM addresses aleatoric uncertainty. Consequently, the results presented in Table 4 could vary significantly if all models accounted for both types of uncertainty.

3.4. Robustness

The robustness of the prognostic models is evaluated through two experiments. The first assesses performance when training and testing on data with different fault modes. The second introduces noise into the input data from the baseline case study to evaluate its impact on performance.

3.4.1. Adaptation across fault modes

This experiment evaluates the models' ability to generalize by comparing their performance on data with differing fault modes. Specifically, the models are trained on the FD001 dataset, which features a single fault mode (HPC degradation) under consistent operational conditions, and tested on the FD003 dataset, which introduces an additional fault mode (fan degradation) under the same conditions. As illustrated in Fig. 11, the testing dataset includes degradation histories that are outliers in terms of lifetime compared to the training dataset. This setup provides a rigorous test of the models' robustness in handling diverse fault modes and degradation patterns.

The results presented in Table 5 indicate a significant decrease in accuracy across all models when applied to the robustness test set. This drop is expected given the discrepancy between the fault modes present in the training and testing datasets. Among the models, AHSMM demonstrates the best performance with the lowest RMSE, suggesting that its adaptive mechanism allows it to handle variations in fault modes more effectively. In contrast, the PFs shows the worst performance, which aligns with expectations since PF heavily depends on the training data and struggles when exposed to unfamiliar degradation patterns.

Table 6 further examines the models' performance in terms of coverage and WSU. AHSMM, despite the significant drop in coverage, still outperforms the other models in this metric. However, its WSU increases substantially, indicating that while the model can maintain some predictive capability, the uncertainty in its predictions grows considerably under these challenging conditions.

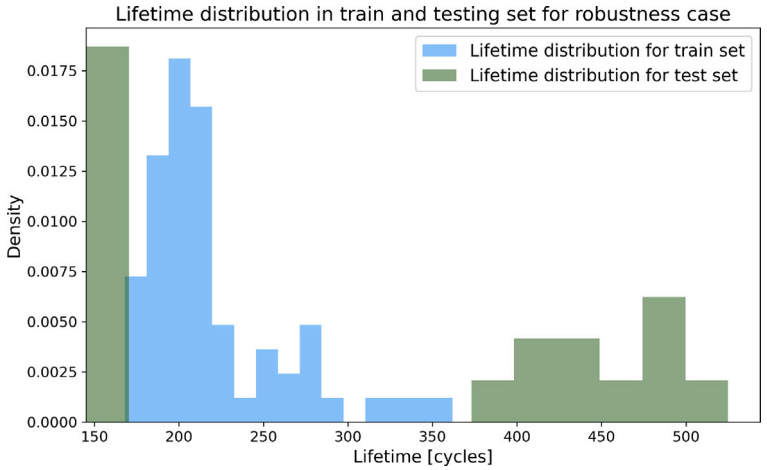


Fig. 11. Lifetime distribution for robustness case.

Table 5
Results of prognostic models for robustness case.

Model	RMSE	% change RMSE	SD
LSTM	94.32	+147.88%	63.42
AHSMM	**81.88**	**+137.47%**	**51.60**
PF	109.52	+355.38%	84.69

Table 6
Results of prognostics models in terms of uncertainty for baseline case.

Model	Cov.	% change coverage	WSU	% change WSU
LSTM	0.26	−21.21%	2047416.9	+200.93%
AHSMM	0.82	−15.46%	12219343.39	+282.53%
PF	0.08	−52.94%	829 690.81	+115.42%

The PF model suffers the most in terms of coverage, further underscoring its limitations in handling diverse fault modes, while LSTM maintains a middle ground between AHSMM and PF in both coverage and WSU.

To visualize some of the predictions on the testing set, Fig. 12 displays the predictions of the three models for three selected engines. These engines were chosen based on their lifetimes: engine #99, which has the shortest lifetime in the test set; engine #94, which represents the median lifetime; and engine #55, which has the longest lifetime.

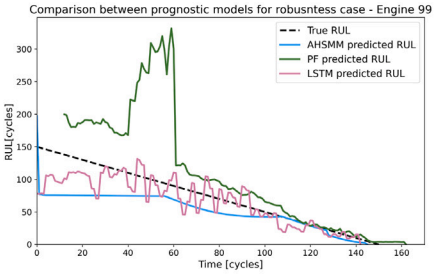
AHSMM's predictions improve when a transition occurs, thus the adaptation mechanism is triggered. This is due to the model's design, which accounts for discrepancies between the expected and actual sojourn times in a particular hidden state for a given testing engine.

However, for PF, the predictions for engines #94 and #55 are significantly inaccurate because the lifetimes of these engines differ greatly from those in the training set, leaving the model with little basis for comparison. One possible improvement is to implement a mechanism similar to AHSMM's, where the model adjusts by generating more particles that represent different degradation processes, or in other words, different polynomial curves, based on how far off the predictions are. This suggests that in stochastic models and BFs, adaptation based on current testing data is feasible since these models simulate the degradation process and can adjust individually to the online test degradation history.

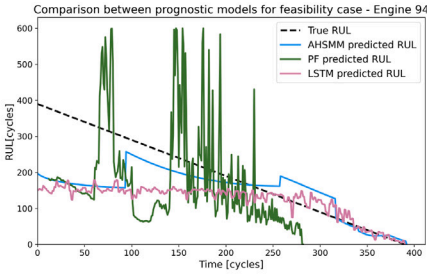
In contrast, LSTM directly maps sensor values to RUL labels, making it more challenging to adapt predictions when confronted with unseen data. This issue is common in supervised ML models, as mentioned in Section 2.2 since supervised ML models need labeled data across all the health conditions of the system to perform more accurately when dealing with data from different fault modes, which might not be available in industrial application. LSTM predictions tend to improve toward the end of the engine's life, where sensor values are more consistent across engines.

This experiment highlights the challenges faced by prognostic models when applied to datasets with different fault modes than those they were trained. The AHSMM model, due to its adaptive nature, performs better than the LSTM and PF models but still experiences a considerable decline in accuracy and increased uncertainty. These findings underscore the importance of developing models that can adapt to diverse operating conditions and fault modes, as real-world applications often involve varying and unpredictable degradation patterns. The significant decrease in model performance also suggests that further improvements in model generalization are necessary to ensure reliable prognostics across a wide range of scenarios.

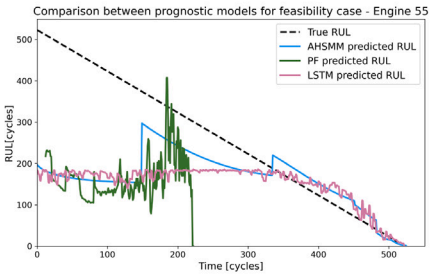
Models like LSTM could benefit significantly from incorporating techniques such as transfer learning or domain adaptation, which were discussed earlier in Section 2.2. These approaches allow the model to leverage knowledge from related tasks or datasets,



(a) RUL predictions for engine #99.



(b) RUL predictions for engine #94.



(c) RUL predictions for engine #55.

Fig. 12. Results from prognostics model for three different engines in the robustness case.

enabling it to perform better in situations with limited training data or when there are shifts in the operating conditions. Transfer learning could help the LSTM model adapt to new fault modes or degradation patterns by fine-tuning it with a smaller set of labeled data from the target domain. Similarly, domain adaptation could improve its robustness to variations in the testing data, ensuring more reliable performance across different operational environments. Implementing these techniques could enhance the LSTM's generalization capabilities, making it more effective in real-world applications where data may vary significantly from the training set.

3.4.2. Noisy input data

This experiment is designed to evaluate the models' ability to handle noisy input data. For this purpose, the data from the baseline case is used, where both the training and testing datasets contain the same fault mode. Gaussian noise, with a mean of 0 and a standard deviation of 0.25, is added to all degradation histories in both the training and testing sets. The addition of noise simulates real-world scenarios where sensor inaccuracies or data corruption may occur. A comparison between the original and noisy data is visually represented in Fig. 13, demonstrating the effect of the noise on the degradation histories. This setup tests the models' robustness and their ability to maintain accuracy despite the introduction of perturbations in the input data.

The results presented in Tables 7 and 8 highlight the varying performance of the prognostic models when faced with noisy input data. The PF shows a significant degradation in performance, with a dramatic increase in RMSE of +179.66%, resulting in an RMSE of 67.26. This indicates that the PF model is highly sensitive to the added Gaussian noise, leading to a notable loss in prediction accuracy. Similarly, the LSTM model experiences an increase in RMSE of +23.86%, resulting in a value of 47.13, showing that while the LSTM is impacted by noise, its performance degradation is less severe compared to the PF. The AHSMM model, on the other hand, exhibits a much smaller increase in RMSE of +7.83%, reaching 37.18, which suggests that it is more resilient to noise and better able to maintain prediction accuracy under challenging conditions.

In terms of coverage, the AHSMM model performs the best, with a coverage of 0.92, despite experiencing a small decrease of −5.15% compared to its performance with clean data. This indicates that the AHSMM model continues to capture the most true values within the predicted intervals, even in the presence of noise. The LSTM model, however, shows a significant drop in coverage, suggesting that the LSTM struggles to maintain accurate predictions when noise is added to the data, leading to a reduced ability to capture true values within the prediction intervals. The PF model shows the lowest coverage at 0.16, which aligns with its already limited performance observed in the baseline case.

Similar to the previous experiment with different fault modes, AHSMM demonstrates superior robustness, emerging as the most resilient model. It maintains relatively high prediction accuracy and coverage despite the presence of noise. In contrast, the PF and LSTM models are more sensitive to noise, leading to significant performance degradation and a reduced ability to handle uncertainty effectively. This highlights AHSMM's strength in managing challenging conditions, while the PF and LSTM models struggle to maintain reliability under noisy data inputs.

In Fig. 14, the predictions for the three selected engines are presented with noisy data. These results can be compared to those in Fig. 9, where the predictions are based on clean data. By examining both sets of results, the impact of noise on the RUL predictions

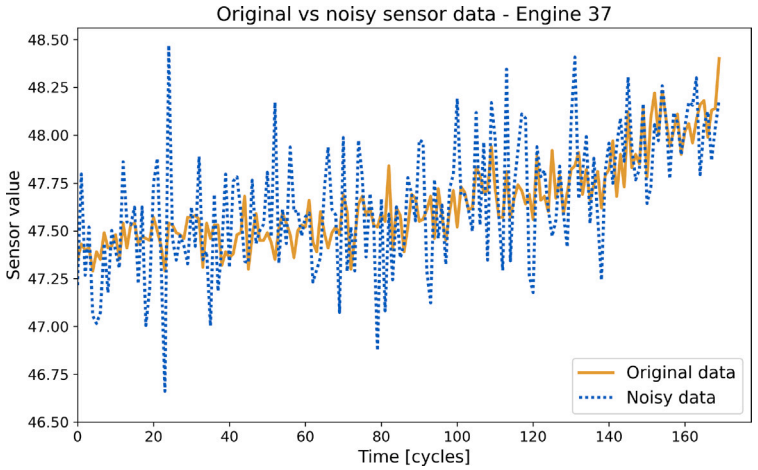


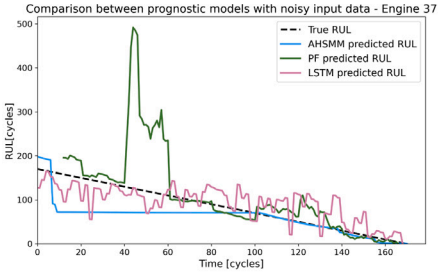
Fig. 13. Input noise data.

Table 7
Results of prognostic models with noisy data.

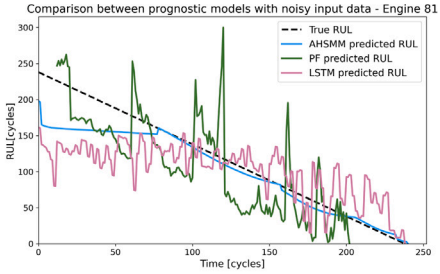
Model	RMSE	% change RMSE	SD
LSTM	47.13	+23.86%	9.09
AHSMM	**37.18**	**+7.83%**	**13.61**
PF	67.26	+179.66%	29.71

Table 8
Results of prognostics models regarding uncertainty with noisy data.

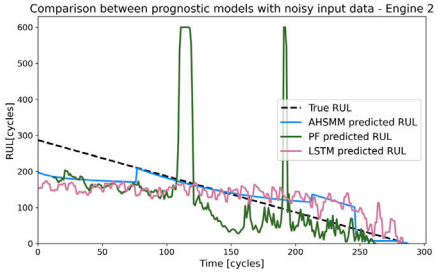
Model	Cov.	% change coverage	WSU	% change WSU
LSTM	0.24	−27.27%	747 704.15	+9.89%
AHSMM	0.92	−5.15%	3280421.27	+2.69%
PF	0.16	−5.88%	567 215.79	+47.27%



(a) RUL predictions for engine #37.



(b) RUL predictions for engine #81.



(c) RUL predictions for engine #2.

Fig. 14. Results from prognostics model for three different engines with noisy data.

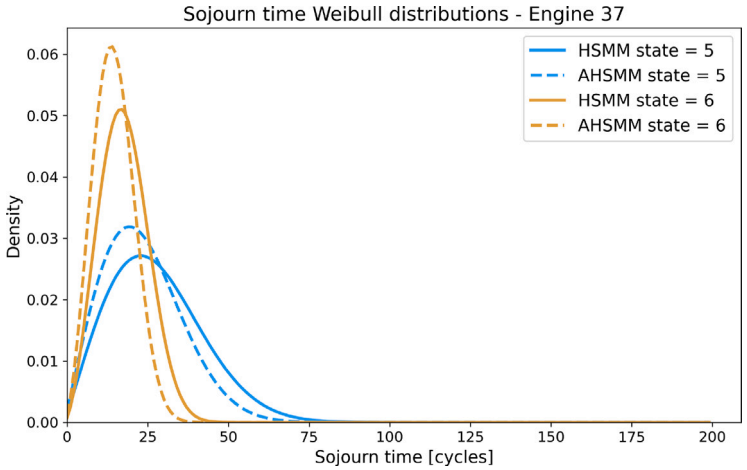


Fig. 15. Sojourn time Weibull distributions for hidden states 5 and 6 for engine #37 RUL prediction.

can be observed, highlighting how the accuracy and reliability of the predictions are affected when noise is introduced into the data.

3.5. Interpretability

Given the literature discussed in Section 2.3, most of the efforts done in prognostics in terms of interpretability are focused on feature importance. However, the three models under study were trained with only one feature, therefore, the relation between input and predictions is straightforward. This is why a qualitative analysis of the models for interpretability is conducted.

The AHSMM is a relatively interpretable model for RUL prediction, primarily due to its structured approach and the Weibull distribution, which is used extensively in reliability analysis. The parameters of the Weibull distribution, particularly the scale parameter, are dynamically adjusted based on real-time data, which allows the model to adapt to the actual degradation trajectory of the system. This adaptive mechanism provides a clear and understandable way to link the model's predictions to observed data, enhancing its interpretability. A visualization is offered in Fig. 15, the sojourn time Weibull distributions for hidden states 5 and 6 are shown for engine #37, which has a lifetime of 170 cycles. This short lifespan makes it a left outlier in the lifetime distribution previously depicted in Fig. 5. To account for the reduced time spent in each hidden state, the adaptive mechanism (illustrated by the dashed lines) shifts the Weibull distributions to the left.

Moreover, the AHSMM has clearly defined transitions and sojourn times, and offers a visual representation of the degradation process, making the model's operation more transparent. Each damage state corresponds to a different degradation level, and the transitions between states can be easily explained and tracked, providing an understanding of the model's predictions. While the use of hidden states and the complexity of the adaptive mechanism add some layers of abstraction, the overall structure of the model and the ability to visualize its predictions help maintain a high level of interpretability. Fig. 16, illustrates the estimated hidden states based on the sensor data from engine 37. These estimated hidden states can serve as a diagnostic tool to assess the system's level of damage, offering users a clearer understanding of the system's condition.

Understanding how these parameters influence predictions can be challenging for LSTM models, which have a total of 20,043 trainable parameters. With so many parameters and limited input data, the model may learn complex patterns that are difficult to interpret and trace back to specific factors in the data. Additionally, disentangling the temporal patterns or trends learned by the model can be problematic. One approach to address this black-box nature is to simplify the model, but this simplification often comes at the cost of reduced accuracy. In such cases, the performance of a simplified DL model may be comparable to or worse than other approaches that offer better interpretability, potentially negating one of the key advantages of DL models.

Fig. 17 illustrates a UMAP projection of the hidden states from the second LSTM layer. The plot shows a clear relationship between the projected hidden states and the RUL. Notably, for lower RUL values, the projections are more tightly clustered, suggesting that the model develops more consistent representations in the later stages of engine life. This is also reflected in improved prediction accuracy for low RUL values. In contrast, higher RUL values also form a relatively cohesive cluster, while mid-range RUL values (e.g., 100–200) appear more scattered. This dispersion may indicate less stability in the model's internal representation during this phase, leading to greater uncertainty and variance in its predictions.

For PF, interpretability will be analyzed step by step, beginning with the degradation model, which, in this case, is a polynomial. Polynomials are inherently interpretable as mathematical models with coefficients that indicate how they influence the shape of the degradation curve. Even more, since the state transition function involves perturbing the polynomial parameters with random noise, it is possible to interpret how changes in these parameters might affect the model's predictions. This gives insight into the model's sensitivity to variations in the degradation process.

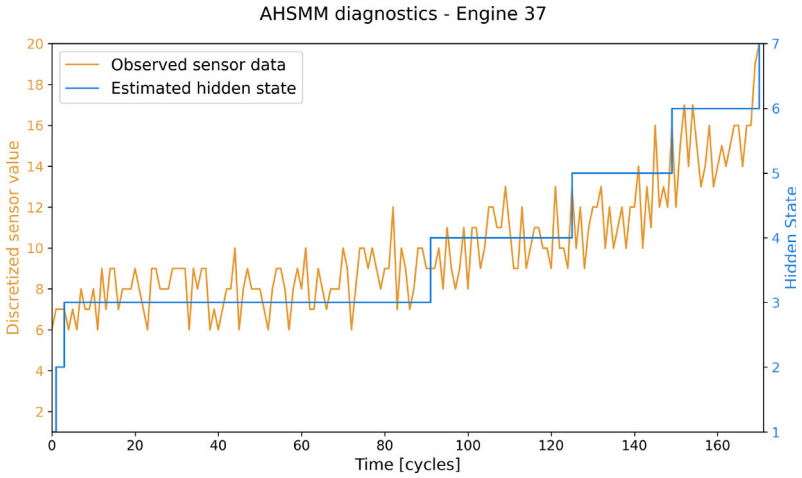


Fig. 16. Diagnostics of engine #37's degradation levels based on hidden state estimates.

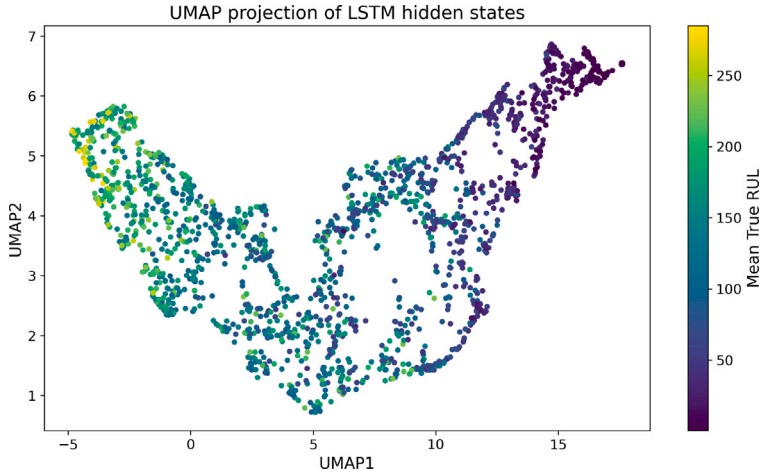


Fig. 17. UMAP projections for LSTM hidden states.

The similarity metric used is a transparent and interpretable method for identifying relevant training histories. MMD is a well-defined statistical measure that quantifies the difference between distributions, so its role in selecting the most similar training histories can be explained and understood in terms of matching historical degradation patterns to current observations. Finally, the PF nature mimics the possible variations in the degradation process over time.

Overall, the model's foundation on fitting and perturbing polynomial curves makes its predictions relatively easy to explain. Both the measured and estimated degradation curves can be visualized, aiding in the interpretation of how the system's current state aligns with the model's estimations. This visual interpretability is particularly valuable when communicating results to decision-makers. For instance, in Fig. 18, the observed sensor data is represented by the orange line, while the model's estimation is shown in green. The PF tracks the degradation fairly well, although, near the end of the system's lifetime, the estimated degradation progresses faster than the actual observations.

3.6. Feasibility

The analysis of the feasibility aspect is divided into the computational time of the training and testing for the baseline data and the effect that the number of available training histories has on the accuracy for the testing set.

3.6.1. Computational time

Table 9 shows the training time when the models are trained with the baseline data, as well as the testing time per sample. The testing time per sample is a critical factor to consider, particularly in real-time prognostic applications, where predictions must be made rapidly for each new sample during online operation.

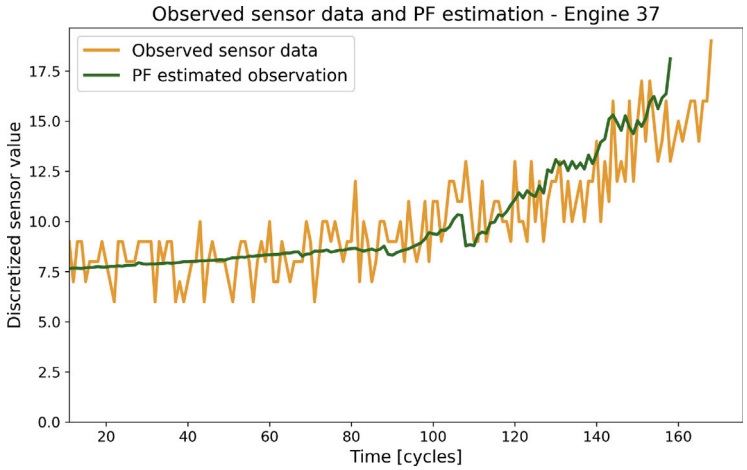


Fig. 18. Observed sensor data and PF estimated observation for engine #37.

Table 9
Computational time of training and testing per sample for each prognostic model.

Model	Training time [s]	Testing time (per sample) [s]
LSTM	8.57	0.39
AHSMM	1417.31	0.03
PF	0.02	0.41

Table 10
Results of prognostic models for feasibility case (training with two degradation histories).

Model	RMSE	% change RMSE	SD
LSTM	59.86	+57.31%	29.24
AHSMM	**38.99**	**+13.08%**	**7.05**
PF	132.21	+449.72%	104.21

The results in Table 9 show the computational times for training and testing each prognostic model. The training time for the LSTM model is 8.57 s, which is relatively fast considering the complexity of the deep learning architecture. However, it is worth noting that the LSTM model requires 0.39 s per sample for testing. In contrast, the AHSMM model has a significantly longer training time of 1417.31 s, which can be attributed to the complexity of training a semi-Markov model. However, it demonstrates an impressive 0.03 s per sample for testing, indicating its efficiency when making predictions after training is completed. The PF model, which is simpler in structure, achieves an exceptionally fast training time of 0.02 s, due to the computational simplicity of polynomial fitting. However, its testing time per sample is 0.41 s, slightly longer than that of LSTM.

It is important to highlight that these models are not fully optimized, as the codes were developed internally without a primary focus on computational efficiency. Consequently, the time results may reflect some inefficiencies in the implementation, especially for the PF and AHSMM models. With further optimization, such as employing more efficient coding practices or utilizing specialized hardware, the computational times could potentially be reduced. Despite this, the results provide a useful comparison of how the models perform in terms of computational demands. They offer valuable insights into the models' feasibility for real-time applications, where prediction speed is crucial, even though further optimizations could lead to improved performance.

3.6.2. Impact of available training histories

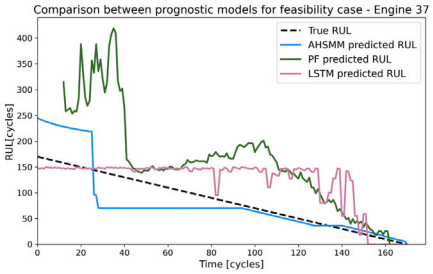
Each model was trained using only two degradation histories, as opposed to the full training set used in the baseline case. This setup allows for a comparison of how the models perform with limited training data, providing insight into their ability to generalize and maintain reliability under constrained conditions.

The results in Table 10 show the metrics for accuracy, with the mean RMSE and SD. Table 11 shows the results when evaluating confidence intervals by using the coverage and WSU metrics. In Fig. 19, a visualization of the predictions for three engines is presented when training the models with only two degradation histories of the baseline degradation histories.

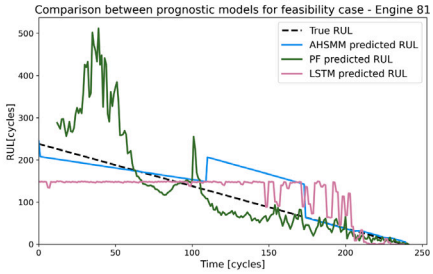
For accuracy, the LSTM model showed a 57.31% increase in RMSE, indicating a substantial drop in performance. This suggests that LSTM models, which depend on large datasets to capture temporal patterns effectively, struggled with the limited data, resulting in increased prediction instability, as reflected by their high standard deviation of 29.24. The AHSMM model demonstrated remarkable robustness, with only a 13.08% increase in RMSE and the lowest standard deviation. This minimal increase in error

Table 11
Results of prognostics models in terms of uncertainty for feasibility case (training with two degradation histories).

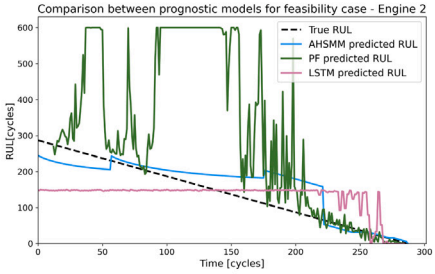
Model	Cov.	% change coverage	WSU	% change WSU
LSTM	0.2	−39.39%	766 143.51	+12.60%
AHSMM	0.62	−36.08%	1321686.13	−58.62%
PF	0.19	+11.76%	971 103.24	+150.14%



(a) RUL predictions for engine #37.



(b) RUL predictions for engine #81.



(c) RUL predictions for engine #2.

Fig. 19. Results from prognostics model for three different engines in the feasibility case.

indicates that AHSMM is better suited for situations with limited training data, likely due to its effective modeling of state transitions even with sparse input.

The PF model experienced the most severe performance decline, with a 449.72% increase in RMSE and an enormous standard deviation of 104.21. This sharp decline underscores the PF model's high sensitivity to reduced training data because the model heavily relies on the similarity approach. This result highlights the challenges PF faces in maintaining accuracy under data constraints.

In terms of uncertainty performance, the LSTM model's coverage decreased by 39.39%, suggesting that its confidence intervals cover the true RUL less frequently. The WSU increased by 12.60%, indicating wider confidence intervals that did not enhance coverage, reflecting inefficient uncertainty quantification. Conversely, AHSMM maintained a more balanced approach, with a 36.08% reduction in coverage and a 58.62% decrease in WSU. This shows that while the confidence intervals became narrower, AHSMM still provided reasonable coverage.

The PF model improved slightly in coverage by 11.76%, but this was offset by a drastic increase in WSU by 150.14%, indicating that while its confidence intervals became broader, they did not effectively enhance reliability. This inefficiency in handling uncertainty further illustrates the PF model's limitations when trained on minimal data. Overall, AHSMM stands out as the most resilient model for few-shot prognostics, maintaining more stable performance despite the reduction in available training data.

This observation is further supported by Fig. 20, which illustrates the relationship between RMSE and the number of degradation histories used for training. As shown in the figure, the PF model's performance significantly deteriorates when only two histories are used, confirming the earlier conclusion. However, when the PF is trained with four or more histories, its performance stabilizes. Notably, when the PF is trained with nine or more histories, its performance consistently outperforms that of the AHSMM model. This suggests that while AHSMM remains more robust in few-shot settings, the PF model can achieve competitive performance with small training sets.

3.7. Sensitivity analysis

To ensure fairness and reliability in the evaluation of the four key characteristics, a sensitivity analysis is conducted. This analysis verifies that the outcomes and conclusions are not overly dependent on the initialization parameters of the prognostic models, ensuring robustness and validity in the results.

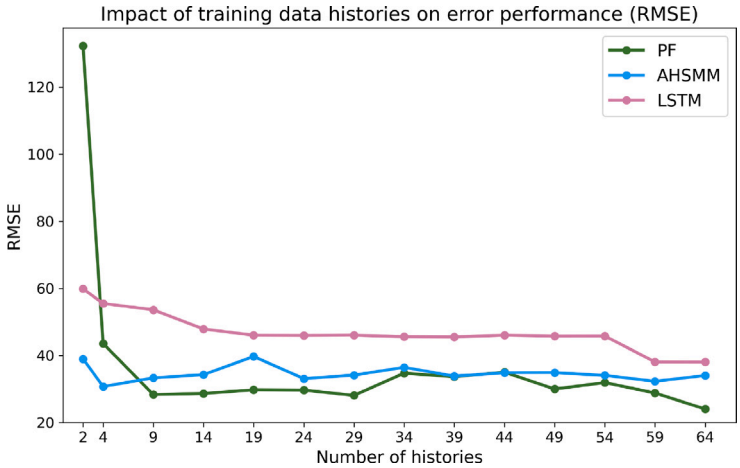


Fig. 20. Impact of number of training histories in accuracy.

Table 12
LSTM sensitivity analysis.

Dropout	RMSE	SD	Coverage	WSU
0.21	37.88	8.29	0.29	586 083.18
0.3	38.05	8.91	0.33	3194276.47
0.39	38.04	8.58	0.41	850 447.05

For the LSTM model, the dropout probability is selected as the parameter to be varied. This parameter is essential because it helps regularize the model, preventing overfitting by randomly dropping connections during training. The dropout probability directly influences both the accuracy of the predictions and the uncertainty of the model's estimates. The baseline dropout value is 0.3, and in this sensitivity analysis, it will be varied by ±30%, resulting in a lower limit of 0.21 and an upper limit of 0.39. The model will then be trained with three configurations: dropout = 0.21, dropout = 0.3 (baseline case), and dropout = 0.39.

For the AHSMM model, the number of hidden states is chosen as the parameter for sensitivity analysis. The number of states directly influences the model's ability to capture the underlying degradation process and its predictive accuracy. The baseline number of states is taken from the original model, and it is varied by ±30%, with the number of states rounded to integer values. This results in a lower limit of 5 and an upper limit of 9.

Finally, for the PF model, the number of particles is selected as the parameter to be altered. The number of particles determines the model's ability to approximate the system's state, and varying this number can influence both the accuracy and the spread of the predictions. In this case, the baseline number of particles is set to 150, and the number of particles is varied by ±30%, giving a lower limit of 105 and an upper limit of 195.

To evaluate the impact of these parameter changes, two key performance metrics — RMSE and coverage — are used. RMSE measures the accuracy of the predictions, while coverage assesses how well the prediction intervals capture the true values. By performing this sensitivity analysis, a deeper understanding of the models' behavior under different parameter settings is gained, which can aid in optimizing performance and ensuring robustness in real-world applications.

In Table 12, the results for the three configurations of the LSTM model are shown. It can be observed that the RMSE for all three dropout configurations is very similar, which suggests that varying the dropout probability (in this given range) has little effect on the prediction accuracy of the model. However, the standard deviation fluctuates slightly, with a decrease in SD when the dropout probability is reduced from 0.3 to 0.21, and an increase when it is raised to 0.39. This indicates some minor changes in the model's stability, but the RMSE remains largely unaffected.

Regarding UQ, the coverage increases as the dropout probability is raised, from 0.33 at a dropout of 0.3 to 0.41 at 0.39. This suggests that with higher dropout rates, the model's prediction intervals become wider, capturing more of the true values. However, this increase in coverage comes at a cost, as the WSU also increases substantially, from 3194276.47 at 0.3 to 850447.05 at 0.39.

To better understand the effects of varying the value of the dropout parameter, the pdf of the RUL obtained with the LSTM model, the mean predicted values and the true RUL are illustrated in Fig. 21. Due to challenges in visualizing all distributions, the RUL probability distributions are presented for two representative timesteps (cycles 100 and 200). For both timesteps, it is evident that the pdf corresponding to the model with the highest dropout value exhibits the largest standard deviation, resulting in a wider and flatter pdf. This highlights how increased dropout impacts uncertainty, aligning with the discussion in Section 2.1, which mentions that the uncertainty captured with MC Dropout is a design artifact and provides a partial representation of the epistemic uncertainty.

The sensitivity analysis for the AHSMM model, summarized in Table 13, demonstrates that increasing the number of hidden states significantly enhances the model's predictive accuracy. This improvement is evident in the sharp decrease in RMSE, from 39.05 at

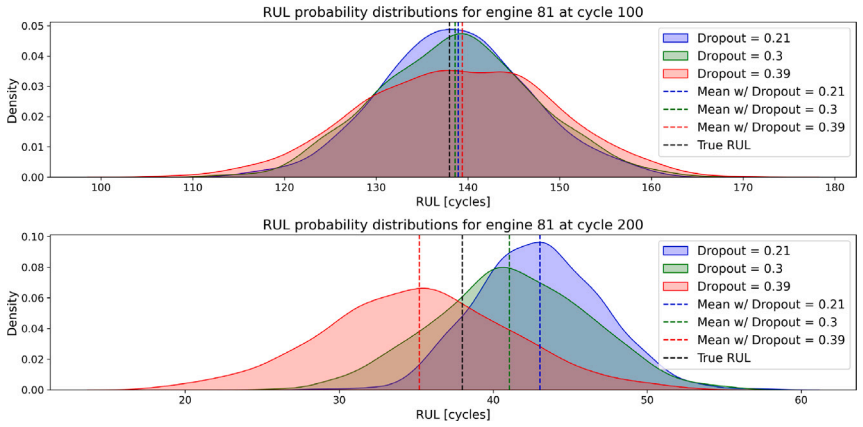


Fig. 21. RUL probability distribution obtained with LSTM for engine 81 at time steps 100 and 200 cycles.

Table 13
AHSMM sensitivity analysis.

N° of states	RMSE	SD	Coverage	WSU
5	39.05	8.91	0.97	3792994.09
7	34.48	9.34	0.97	3194276.47
9	22.68	7.41	0.98	3248370.73

Table 14
PF sensitivity analysis.

N° of particles	RMSE	SD	Coverage	WSU
105	30.08	8.85	0.18	379 971.91
150	24.05	9.34	0.17	385 141.6
195	31.96	12.67	0.17	450 366.9

5 states to 22.68 at 9 states, indicating that a higher number of states allows the model to capture the underlying degradation process better. Additionally, the SD of the predictions stabilizes as the number of states increases, reflecting greater consistency in performance. Notably, the coverage metric remains robust across all configurations, with values between 0.97 and 0.98, underscoring the model's reliable uncertainty quantification regardless of the parameter setting. Furthermore, the configuration with 7 states achieves the lowest WSU, indicating its superior ability to manage uncertainty while maintaining the same high coverage as the other configurations.

While a configuration with 9 hidden states achieves the best accuracy, the baseline configuration with 7 states offers a more practical balance between performance and computational efficiency. Training the model with 9 states is computationally intensive, making the 7-state configuration a better choice. This trade-off ensures an improvement in accuracy over the 5-state configuration while maintaining manageable computational demands. The RUL probability distributions at timesteps 100 and 200, displayed in Fig. 22, further demonstrate that the 7-state configuration exhibits less uncertainty, aligning with its lower WSU value from Table 13. In terms of accuracy, the mean predicted RUL values across configurations for these examples show minimal differences, highlighting that the 7-state configuration offers a reliable and efficient option for accurate predictions with reduced uncertainty.

At cycle 200, the RUL probability density function (pdf) for the AHSMM with five damage states shows a small initial peak, which is caused by an algorithm artifact. To improve efficiency, the way the model is programmed can occasionally result in pdfs with this shape. If the maximum time were set higher, requiring more computational time, the pdf would appear smoother like the others. However, this artifact is rare and does not affect the mean RUL prediction, and the changes to the confidence intervals are negligible.

The sensitivity analysis for the PF model, summarized in Table 14, reveals that varying the number of particles has a notable impact on the model's predictive performance and UQ. The configuration with 150 particles achieves the lowest RMSE of 24.05, indicating superior predictive accuracy compared to 105 and 195 particles, which have RMSE values of 30.08 and 31.96, respectively. Interestingly, the SD of the predictions increases with the number of particles, ranging from 8.85 (105 particles) to 12.67 (195 particles), suggesting that higher particle counts may lead to less consistent predictions. The coverage metric, which remains consistently low across all configurations, indicates that the model struggles to quantify uncertainty effectively, regardless of the number of particles. The WSU metric also highlights this trend, with higher values observed for configurations with more particles. (See Fig. 23)

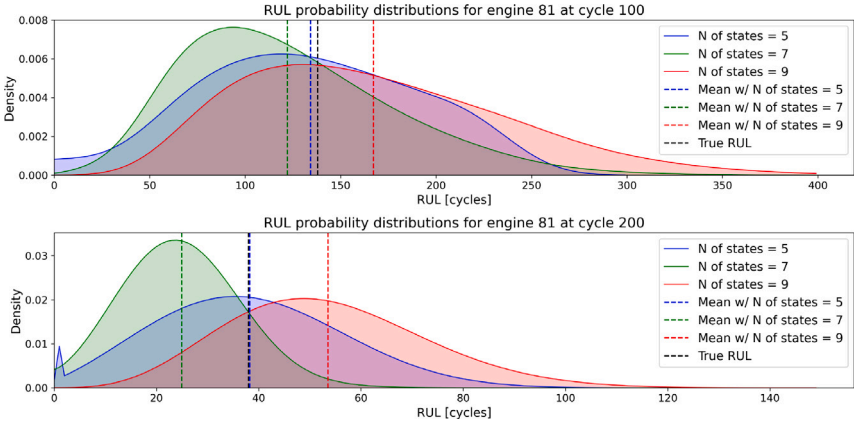


Fig. 22. RUL probability distribution obtained with AHSMM for engine 81 at time steps 100 and 200 cycles.

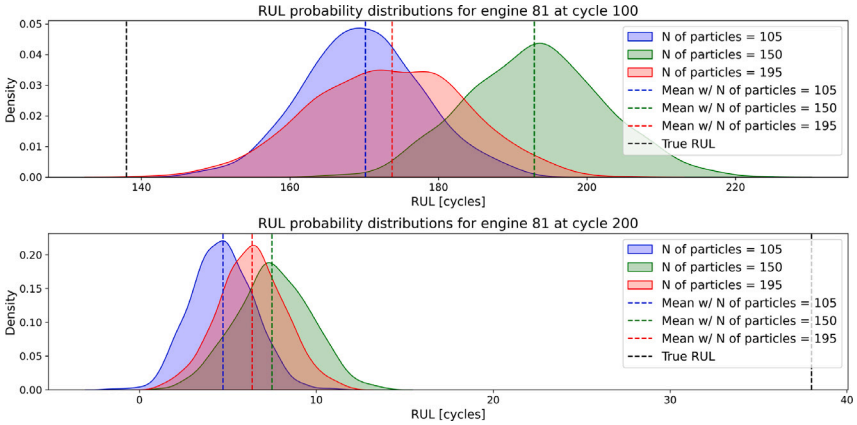


Fig. 23. RUL probability distribution obtained with PF for engine 81 at time steps 100 and 200 cycles.

4. Model evaluation framework

From the previous sections, it is evident that a wide variety of prognostic models are available, each with distinct strengths tailored to specific applications. Selecting the appropriate model is therefore crucial for optimizing performance and reliability. However, this selection process often involves navigating trade-offs between multiple characteristics. For instance, models with high interpretability may sacrifice computational efficiency, whereas models offering adequate UQ might be impractical when data or resources are constrained. To navigate these challenges, this section introduces a systematic methodology for prioritizing model characteristics based on organizational and operational needs.

Therefore, a decision-support tool is proposed to quantify the relative importance of uncertainty, robustness, interpretability, and feasibility in prognostic model selection. This approach incorporates stakeholder inputs and operational requirements into a weighted scoring system, enabling the systematic evaluation and ranking of candidate models.

First, it is necessary to understand the needs and priorities of key stakeholders. These priorities are evaluated across three dimensions:

- Strategic Priorities: Alignment with high-level organizational goals, such as enhancing reliability and availability, reducing operational costs, or meeting regulatory standards.
- Operational Constraints: Practical limitations, such as computational resources, time-to-deploy, or data availability.
- Technical Requirements: Specific performance criteria, such as accuracy, interpretability, or robustness, are needed to ensure reliable RUL predictions.

Second, each key characteristic — Uncertainty, Robustness, Interpretability, and Feasibility — is evaluated on a scale from 1 to 5, where 1 represents the least importance and 5 the greatest. These scores are assigned based on the following criteria:

Table 15
Decision matrix for selecting prognostic models.

Priority	Group of Models	SOTA	Metrics for Evaluation
Uncertainty	ML SMs BFs	[20,22] [84,121] [99]	Coverage, WSU
Robustness	ML SMs BFs	[27,47] [82,95] [104]	Accuracy on noisy data and different operational conditions
Interpretability	ML SMs BFs	[35,46,73] [72] [143]	Alignment with known failure modes, assessment on explainability
Feasibility	ML SMs BFs	[59,60,69] [82] *	Computational cost, accuracy on small datasets

Uncertainty.

- How crucial is the precise quantification of uncertainty in predicting RUL for safety-critical systems, where early failure could have catastrophic consequences?
- Are high-stakes decisions dependent on confidence levels in RUL estimates?

Robustness.

- Will the model operate under varying environmental or operational conditions?
- To what extent is the degradation process affected by environmental or operational conditions?

Interpretability.

- How essential is the ability to explain model predictions to stakeholders or regulators?
- Is model transparency required for decision-making or regulatory approval?

Feasibility.

- Are there sufficient computational resources and expertise to implement complex models?
- Is the available data adequate (e.g., in terms of volume, labeling, or consistency)?

Third, when scores are assigned by an individual stakeholder, raw scores are generally sufficient to determine the relative importance of each characteristic. However, when multiple stakeholders are involved, normalization becomes necessary. In such cases, the weight of each characteristic must be normalized using the expression shown in (4), where W_i represents the normalized weight of characteristic i, $S_{i,k}$ is the raw score assigned to characteristic i by stakeholder k:

$$W_i = \frac{\sum_k S_{i,k}}{\sum_i \sum_k S_{i,k}} \tag{4}$$

The decision-support tool provides a structured framework for selecting prognostic models that align with stakeholder priorities and operational requirements. For instance, a safety-critical system might prioritize robustness and interpretability, while a resource-constrained application may emphasize feasibility and UQ.

The decision matrix presented in Table 15 offers a practical framework for aligning model types with the key characteristics (uncertainty, robustness, interpretability, and feasibility), while suggesting appropriate metrics for their evaluation. It highlights the most suitable models within each group of prognostic approaches reviewed. By providing a structured and systematic methodology, the matrix empowers stakeholders to efficiently identify models that align with their operational and technical priorities.

Notably, Table 15 highlights the absence of state-of-the-art papers for BFs concerning feasibility. This omission is primarily due to the inherent capacity of BFs to predict RUL while using limited degradation histories or even incomplete ones, making them naturally suitable for scenarios where data availability is a constraint.

To evaluate and rank candidate models, a fitness function is proposed. This function quantifies the performance of candidate models by considering the key characteristics of uncertainty, robustness, interpretability, and feasibility, which are weighted according to stakeholder(s) priorities.

The fitness function is designed to maximize the overall score that reflects how well a model aligns with organizational and operational needs. For characteristics such as uncertainty, robustness, and interpretability, higher values of relevant metrics (e.g., coverage probability, model accuracy under varying conditions, or transparency in predictions) positively contribute to the fitness score. Conversely, for metrics that need to be minimized, such as error performance (e.g., RMSE), computational cost, or memory usage, the fitness function incorporates inverse scaling. This ensures that models with lower values for these metrics — indicating better performance or lower resource demands — are prioritized.

Table 16
Performance metric per model.

Characteristic	Metric	LSTM	AHSMM	BF
Uncertainty	Coverage	0.85	0.75	0.9
Robustness	RMSE (1/x)	0.75	0.92	0.8
Feasibility	Computational Cost (1/x)	0.60	0.80	0.70
Interpretability	Explainability assessment	0.60	0.75	0.85

Table 17
Metric normalization for each characteristic.

Characteristic	LSTM	AHSMM	BF
Uncertainty	0.94	0.83	1.0
Robustness	0.75	1.0	0.87
Feasibility	0.75	1.0	0.88
Interpretability	0.71	0.88	1.0

Table 18
Fitness calculation for each candidate model.

Model	Fitness calculation	Fitness score
LSTM	0.35*0.94 + 0.30*0.75 + 0.25* 0.75 + 0.10*0.71	0.852
AHSMM	0.35*0.83 + 0.30*1.00 + 0.25*1.00 + 0.10*0.88	0.915
BF	0.35*1.00 + 0.30*0.87 + 0.25*0.88 + 0.10*1.00	0.906

It is worth noting that interpretability often lacks a straightforward, quantifiable metric. Instead, it is assessed qualitatively or based on compliance with regulatory requirements. In this paper, stakeholders are encouraged to provide a percentage score (0 to 100%) reflecting how well interpretability needs are met. This score is normalized to a range of 0 to 1 for use in the fitness function.

The fitness function is then computed as follows:

$$F(M) = \sum_i W_i \cdot V_i(M) \tag{5}$$

where W_i represents the normalized weight for characteristic i, derived from stakeholder inputs, and $V_i(M)$ is the normalized value of the metric for characteristic i in model M.

By employing this method, models can be objectively compared based on their performance across all key characteristics. The model with the highest fitness score, considering both the prioritized characteristics and the operational constraints, is the one that best meets the needs of the organization.

The fitness function offers a transparent way to navigate trade-offs between conflicting priorities. For example, a model excelling in UQ but with high computational costs can be compared against another offering a better balance between performance and resource efficiency. This ensures that the final selection optimally aligns with organizational goals while adhering to operational constraints.

To illustrate the applicability of this framework, consider the following example. An airline operator wants to implement a prognostic model to predict the RUL of primary aviation structures. The primary organizational priorities are as follows: uncertainty quantification, which is crucial for safety-critical applications; robustness, accounting for variable operational conditions (e.g., environmental factors, pilot operational behavior); feasibility, given the limited availability of full-scale catastrophic data; and interpretability, as some level of model transparency is necessary for regulatory compliance.

Step 1: Prioritization. Stakeholders assign weights to each characteristic based on their importance. After input and normalization, the following normalized weights are assigned: Uncertainty = 0.35, Robustness = 0.30, Feasibility = 0.25, and Interpretability = 0.10.

Step 2: Evaluate candidate models. Three models are shortlisted: an LSTM with MC Dropout, an AHSMM, and a BF. Performance metrics are collected for each characteristic, as shown in Table 16:

The RMSE metric assesses how well the models predict the RUL under different environmental conditions. Both robustness and feasibility metrics are inversely scaled, meaning lower values are preferred. Interpretability is assessed through a qualitative score based on user feedback, with higher values indicating greater model explanability.

Step 3: Normalize metric values. Each metric is normalized to ensure comparability across models. The normalized values are shown in Table 17:

Step 4: Fitness calculation. The fitness score for each candidate model is calculated as a weighted sum of the normalized values. Table 18 shows the fitness calculation for each model:

Based on the fitness scores, the AHSMM model ranks highest, closely followed by the BF model. However, if model explainability becomes a more critical factor in the future, the BF model may be preferred due to its superior interpretability score.

After model selection, its implementation and deployment typically proceed in phases to minimize risks and ensure practical effectiveness. A recommended approach involves the following steps:

Pilot testing. Initially, the selected model is deployed in a controlled, small-scale environment. The pilot phase allows for testing its performance on real-world data while minimizing potential disruptions to operations. This step is critical for identifying gaps between theoretical expectations and practical outcomes. For instance, unexpected variability in data quality or computational demands can emerge during this phase.

Evaluation and adjustment. During the pilot test, the model's predictions are evaluated against benchmark datasets or historical ground truth. The pertinent metrics are monitored closely. Insights from this evaluation phase guide refinements in the model, such as parameter tuning, algorithmic adjustments, or preprocessing enhancements.

Scaling deployment. Once the pilot demonstrates satisfactory performance, the model can be scaled for full operational use. At this stage, additional considerations like integration with existing systems, user interfaces, and automation pipelines come into play. Robust error-handling mechanisms are also implemented to manage anomalies or system failures.

Continuous optimization. The deployment of a prognostic model is not a one-time process. Over time, the operational environment, system conditions, or data characteristics may change, potentially impacting the model's performance. Regular monitoring and periodic retraining using updated datasets are essential to maintaining the model's accuracy and reliability. Advanced methods such as online learning or dynamic domain adaptation can further enhance long-term performance.

Stakeholder feedback and adaptation. Engaging with end-users, engineers, or decision-makers provides critical insights into the model's usability and practical value. For example, stakeholders might request additional interpretability features or improved integration with diagnostic workflows. Incorporating this feedback ensures that the model remains aligned with organizational priorities.

By combining the decision matrix and the fitness function, organizations can systematically select, implement, and optimize prognostic models. This structured approach simplifies trade-offs, balances technical and operational demands, and ensures long-term success. The methodology enables actionable insights delivered with accuracy, reliability, and efficiency, supporting both immediate and strategic goals.

5. Potential research directions

Based on the findings of this review and the case study, future research directions are proposed to improve prognostic models for RUL prediction. These directions are grouped by the four key characteristics discussed: uncertainty, robustness, interpretability, and feasibility.

Uncertainty. For BFs and stochastic models, future research should aim to improve the reliability of confidence intervals, as existing methods often produce intervals that are overly wide and may result in uninformative predictions. Another area is the development of adaptive techniques that adjust uncertainty estimates based on system dynamics. For ML models, specifically DL models, future work should explore Bayesian DL approaches (e.g., variational inference, deep ensembles) to replace methods such as MC Dropout. Across all model types, a key direction is the management of uncertainty, not just its quantification. This includes using a more practical uncertainty categorization (past, present, future, model, and prediction method uncertainty), which enables uncertainty management by leveraging data to better characterize the sources of uncertainty, thereby reducing their impact on RUL prediction. Additionally, emphasis should shift toward subject-specific uncertainty rather than population-level uncertainty, a critical direction for all DDMs.

Robustness. Stochastic models could benefit from research into more flexible state-transition architectures that can capture non-stationary and multi-modal degradation behavior. BFs offer promise through online learning strategies that continuously adapt their degradation model for new conditions without requiring full retraining. For ML models, future work should prioritize self-supervised and unsupervised approaches to enhance generalization across domains. Across all model types, hybrid approaches that combine physics-based knowledge with data-driven learning warrant further exploration to improve adaptability and fault tolerance.

Interpretability. Stochastic models are generally interpretable, but their accessibility could be improved through visual tools that help practitioners understand state transitions and probabilistic outcomes. For BFs, interactive visualization techniques that show how belief states evolve could support explainability in practice. For DL models, interpretability remains a challenge. Future research should develop reliable explanation tools that avoid inconsistencies common in existing techniques like SHAP and LIME. Another important direction is quantifying the trade-off between interpretability and performance, particularly in contexts where regulatory or operational transparency is required.

Feasibility. For stochastic models and BFs, future research should investigate how to automatically configure models (e.g., selecting the number of states, particles, or distributions) when data is limited or noisy. In ML, promising directions include few-shot learning, self-supervised learning, and physics-informed learning to reduce data requirements, as mentioned in [144]. For all model types, better methods for handling missing data are essential. Lastly, developing lightweight implementations of these models for use in embedded or edge environments remains a research need.

6. Conclusions

This paper reviewed data-driven prognostic models for Remaining Useful Life (RUL) prediction, evaluating Machine Learning (ML) models, stochastic models, and Bayesian Filters (BFs) across four key characteristics: uncertainty, robustness, interpretability, and feasibility. A case study using the C-MAPSS dataset compared the performance of LSTM, AHSMM, and PF models, highlighting their strengths and limitations in addressing these characteristics.

Uncertainty quantification (UQ) remains a significant challenge in RUL prediction. Stochastic models and BFs are effective in reporting uncertainty but suffer from overly wide confidence intervals and computational complexity. ML models, while accurate, often overlook UQ, leading to potentially unreliable predictions. The findings underscore the need for better calibration and methods that balance accuracy with uncertainty management.

Robustness is crucial to ensure an adequate model performance under varying operational conditions. While ML models have shown promise through techniques like domain adaptation, challenges persist in industrial settings. Stochastic models and BFs, like AHSMM, demonstrated adaptability in unseen and noisy conditions, with AHSMM performing particularly well in the case study. However, PF models faced limitations due to degradation model constraints.

Interpretability is essential for safety-critical applications, with regulatory frameworks demanding transparency. AHSMM offers clear degradation representations, while LSTM models are more difficult to interpret. PFs provide some interpretability but depend heavily on the underlying degradation model. Balancing interpretability and performance remains a challenge across industries.

Feasibility is influenced by data availability, with scarcity and noise common in industries like aviation and machinery. Approaches like few-shot learning and Bayesian methods address these challenges. The case study showed that AHSMM performed well with limited data, while PF excelled with moderately sized datasets. This highlights the need for models that can adapt to varying data conditions.

The paper introduced a model evaluation framework, helping users select the most suitable model based on operational needs. It is emphasized that no one-size-fits-all solution exists; model choice depends on system-specific constraints and objectives. Implementing prognostic models requires ongoing retraining and stakeholder feedback.

In conclusion, this review highlights the progress and challenges in data-driven prognostics. Future research should focus on models that are not only accurate but also adaptable to real-world complexities, handling uncertainty, noisy data, and dynamic conditions. By aligning emerging techniques with industry-specific needs, the field can move towards creating reliable, transparent prognostic solutions for better decision-making and operational improvements.

CRediT authorship contribution statement

Mariana Salinas-Camus: Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Kai Goebel:** Writing – review & editing, Methodology, Conceptualization. **Nick Eleftheroglou:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

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.

Data availability

The data used in this study were sourced from a publicly available dataset. Details regarding the dataset, including its source and access information, are provided in the manuscript.

References

[1] Z. Huang, Z. Xu, X. Ke, W. Wang, Y. Sun, Remaining useful life prediction for an adaptive skew-Wiener process model, Mech. Syst. Signal Process. 87 (2017) 294–306.
[2] K. Goebel, M. Daigle, A. Saxena, S. Sankararaman, I. Roychoudhury, J. Celaya, Prognostics: The Science of Making Predictions, CreateSpace Independent Publishing Platform, 2017, URL: https://books.google.nl/books?id=M0AztAEACAAJ.
[3] E. Zio, Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice, Reliab. Eng. Syst. Saf. 218 (2022) http://dx.doi.org/10.1016/j.ress.2021.108119.
[4] S. Vollert, A. Theissler, Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, IEEE, 2021, pp. 1–8.
[5] A. Kamariotis, K. Tatsis, E. Chatzi, K. Goebel, D. Straub, A metric for assessing and optimizing data-driven prognostic algorithms for predictive maintenance, Reliab. Eng. Syst. Saf. 242 (2024) 109723.
[6] C. Huang, S. Bu, H.H. Lee, C.H. Chan, S.W. Kong, W.K. Yung, Prognostics and health management for predictive maintenance: A review, J. Manuf. Syst. 75 (2024) 78–101.
[7] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Mag. 38 (3) (2017) 50–57.
[8] H.M. Elattar, H.K. Elminir, A.M. Riad, Prognostics: a literature review, Complex Intell. Syst. 2 (2) (2016) 125–154.
[9] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to RUL prediction, Mech. Syst. Signal Process. 104 (2018) 799–834, http://dx.doi.org/10.1016/j.ymssp.2017.11.016, URL: https://www.sciencedirect.com/science/article/pii/S0888327017305988.

[10] J. Guo, Z. Li, M. Li, A review on prognostics methods for engineering systems, IEEE Trans. Reliab. 69 (3) (2019) 1110–1129.
[11] D. An, J. Choi, N. Kim, Options for prognostics methods: A review of data-driven and physics-based prognostics, in: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2013, p. 1940.
[12] J. Celaya, C. Kulkarni, K. Goebel, A model-based prognostics methodology for electrolytic CapacitorsBased on electrical overstress accelerated aging, in: Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011, PHM 2011, 2011.
[13] C. Kulkarni, J. Celaya, K. Goebel, G. Biswas, Physics based electrolytic capacitor degradation models for prognostic studies under thermal overstress, in: PHM Society European Conference, vol. 1, 2012.
[14] M. Daigle, C. Kulkarni, G. Gorospe, Application of model-based prognostics to a pneumatic valves testbed, in: 2014 IEEE Aerospace Conference, IEEE, 2014, pp. 1–8.
[15] J. Qiu, B. Seth, S. Liang, C. Zhang, Damage mechanics approach for bearing lifetime prognostics, Mech. Syst. Signal Process. 16 (5) (2002) 817–829.
[16] F. Zhao, Z. Tian, Y. Zeng, A stochastic collocation approach for efficient integrated gear health prognosis, Mech. Syst. Signal Process. 39 (1–2) (2013) 372–387.
[17] M. Chao, C. Kulkarni, K. Goebel, O. Fink, Fusing physics-based and deep learning models for prognostics, Reliab. Eng. Syst. Saf. 217 (2022) 107961.
[18] R. Nguyen, S.K. Singh, R. Rai, Physics-infused fuzzy generative adversarial network for robust failure prognosis, Mech. Syst. Signal Process. 184 (2023) 109611.
[19] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, N. Zerhouni, Direct remaining useful life estimation based on support vector regression, IEEE Trans. Ind. Electron. 64 (3) (2016) 2276–2285.
[20] J. Caceres, D. Gonzalez, T. Zhou, E.L. Droguett, A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties, Struct. Control. Heal. Monit. 28 (10) (2021) e2811.
[21] Y. Alomari, M. Andó, M.L. Baptista, Advancing aircraft engine RUL predictions: an interpretable integrated approach of feature engineering and aggregated feature importance, Sci. Rep. 13 (1) (2023) 13466.
[22] T. Xiahou, F. Wang, Y. Liu, Q. Zhang, Bayesian dual-input-channel LSTM-based prognostics: Toward uncertainty quantification under varying future operations, IEEE Trans. Reliab. (2023).
[23] L. Zhuang, A. Xu, X.-L. Wang, A prognostic driven predictive maintenance framework based on Bayesian deep learning, Reliab. Eng. Syst. Saf. 234 (2023) 109181.
[24] Y.-H. Lin, G.-H. Li, A Bayesian deep learning framework for RUL prediction incorporating uncertainty quantification and calibration, IEEE Trans. Ind. Inform. 18 (10) (2022) 7274–7284.
[25] X. Chen, G. Jin, S. Qiu, M. Lu, D. Yu, Direct remaining useful life estimation based on random forest regression, in: 2020 Global Reliability and Prognostics and Health Management, PHM-Shanghai, IEEE, 2020, pp. 1–7.
[26] D. Laredo, Z. Chen, O. Schütze, J.-Q. Sun, A neural network-evolutionary computational framework for remaining useful life estimation of mechanical systems, Neural Netw. 116 (2019) 178–187.
[27] P.R.d.O. da Costa, A. Akçay, Y. Zhang, U. Kaymak, Remaining useful lifetime prediction via deep domain adaptation, Reliab. Eng. Syst. Saf. 195 (2020) 106682.
[28] O. Asif, S. Haider, S. Naqvi, J. Zaki, K. Kwak, S. Islam, A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset, IEEE Access 10 (2022) 95425–95440.
[29] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, in: 2017 IEEE International Conference on Prognostics and Health Management, ICPHM, IEEE, 2017, pp. 88–95.
[30] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, X. Li, Machine remaining useful life prediction via an attention-based deep learning approach, IEEE Trans. Ind. Electron. 68 (3) (2020) 2521–2531.
[31] Y. Cheng, J. Wu, H. Zhu, S.W. Or, X. Shao, Remaining useful life prognosis based on ensemble long short-term memory neural network, IEEE Trans. Instrum. Meas. 70 (2020) 1–12.
[32] M. Salinas-Camus, N. Eleftheroglou, Uncertainty in aircraft turbofan engine prognostics on the C-MAPSS dataset, in: PHM Society European Conference, vol. 8, 2024, 10–10.
[33] J. Li, X. Li, D. He, Domain adaptation remaining useful life prediction method based on AdaBN-DCNN, in: 2019 Prognostics and System Health Management Conference, PHM-Qingdao, IEEE, 2019, pp. 1–6.
[34] X. Li, Q. Ding, J. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliab. Eng. Syst. Saf. 172 (2018) 1–11.
[35] M. Kraus, S. Feuerriegel, Forecasting remaining useful life: Interpretable deep learning approach via variational Bayesian inferences, Decis. Support Syst. 125 (2019) 113100.
[36] Y. Gao, Y. Wen, J. Wu, A neural network-based joint prognostic model for data fusion and remaining useful life prediction, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 117–127.
[37] A. Srinivasan, J.C. Andresen, A. Holst, Ensemble neural networks for remaining useful life (RUL) prediction, 2023, arXiv preprint arXiv:2309.12445.
[38] S. Xiang, Y. Qin, J. Luo, F. Wu, K. Gryllias, A concise self-adapting deep learning network for machine remaining useful life prediction, Mech. Syst. Signal Process. 191 (2023) 110187.
[39] H. Li, W. Zhao, Y. Zhang, E. Zio, Remaining useful life prediction using multi-scale deep convolutional neural network, Appl. Soft Comput. 89 (2020) 106113.
[40] S. Xiang, Y. Qin, J. Luo, H. Pu, B. Tang, Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction, Reliab. Eng. Syst. Saf. 216 (2021) 107927.
[41] J. Chen, H. Jing, Y. Chang, Q. Liu, Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process, Reliab. Eng. Syst. Saf. 185 (2019) 372–382.
[42] M. Baptista, M. Mishra, E. Henriques, H. Prendinger, Using explainable artificial intelligence to interpret remaining useful life estimation with gated recurrent unit, Annu. Conf. PHM Soc. 16 (2024) http://dx.doi.org/10.36001/phmconf.2024.v16i1.4124.
[43] A. Ruiz-Tagle Palazuelos, E.L. Droguett, R. Pascual, A novel deep capsule neural network for remaining useful life estimation, Proc. Inst. Mech. Eng. Part O: J. Risk Reliab. 234 (1) (2020) 151–167.
[44] Y. Keshun, Q. Guangqi, G. Yingkui, Optimizing prior distribution parameters for probabilistic prediction of remaining useful life using deep learning, Reliab. Eng. Syst. Saf. 242 (2024) 109793.
[45] X. Zhang, J. Sun, J. Wang, Y. Jin, L. Wang, Z. Liu, PAOLTransformer: Pruning-adaptive optimal lightweight transformer model for aero-engine remaining useful life prediction, Reliab. Eng. Syst. Saf. 240 (2023) 109605.
[46] N. Costa, L. Sánchez, Variational encoding approach for interpretable assessment of remaining useful life estimation, Reliab. Eng. Syst. Saf. 222 (2022) 108353.
[47] W. Zhang, X. Li, H. Ma, Z. Luo, X. Li, Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions, Reliab. Eng. Syst. Saf. 211 (2021) 107556.
[48] J. Figueroa Barraza, E. López Droguett, M.R. Martins, Towards interpretable deep learning: a feature selection framework for prognostics and health management using deep neural networks, Sensors 21 (17) (2021) 5888.
[49] R.K. Kundu, K.A. Hoque, Explainable predictive maintenance is not enough: quantifying trust in remaining useful life estimation, in: Annual Conference of the PHM Society, vol. 15, 2023.

[50] T.H. Loutas, D. Roulias, G. Georgoulas, Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic E-support vectors regression, IEEE Trans. Reliab. 62 (4) (2013) 821–832.
[51] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Remaining useful life estimation based on nonlinear feature reduction and support vector regression, Eng. Appl. Artif. Intell. 26 (7) (2013) 1751–1760.
[52] B. Yang, R. Liu, E. Zio, Remaining useful life prediction based on a double-convolutional neural network architecture, IEEE Trans. Ind. Electron. 66 (12) (2019) 9521–9530.
[53] D. She, M. Jia, A BiGRU method for remaining useful life prediction of machinery, Measurement 167 (2021) 108277.
[54] X. Li, W. Zhang, Q. Ding, Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction, Reliab. Eng. Syst. Saf. 182 (2019) 208–218.
[55] M. Ma, Z. Mao, Deep-convolution-based LSTM network for remaining useful life prediction, IEEE Trans. Ind. Inform. 17 (3) (2020) 1658–1667.
[56] X. Li, W. Zhang, H. Ma, Z. Luo, X. Li, Data alignments in machinery remaining useful life prediction using deep adversarial neural networks, Knowl.-Based Syst. 197 (2020) 105843.
[57] W. Peng, Z.-S. Ye, N. Chen, Bayesian deep-learning-based health prognostics toward prognostics uncertainty, IEEE Trans. Ind. Electron. 67 (3) (2019) 2283–2293.
[58] R. Zhu, Y. Chen, W. Peng, Z.-S. Ye, Bayesian deep-learning for RUL prediction: An active learning perspective, Reliab. Eng. Syst. Saf. 228 (2022) 108758.
[59] P. Ding, J. Xia, X. Zhao, M. Jia, Graph structure few-shot prognostics for machinery remaining useful life prediction under variable operating conditions, Adv. Eng. Inform. 60 (2024) 102360.
[60] P. Ding, M. Jia, Y. Ding, Y. Cao, J. Zhuang, X. Zhao, Machinery probabilistic few-shot prognostics considering prediction uncertainty, IEEE/ASME Trans. Mechatronics (2023).
[61] J. Zhu, N. Chen, C. Shen, A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions, Mech. Syst. Signal Process. 139 (2020) 106602.
[62] J. Yang, Y. Peng, J. Xie, P. Wang, Remaining useful life prediction method for bearings based on LSTM with uncertainty quantification, Sensors 22 (12) (2022) 4549.
[63] W. Deng, K.T. Nguyen, C. Gogu, K. Medjaher, J. Morio, Enhancing prognostics for sparse labeled data using advanced contrastive self-supervised learning with downstream integration, Eng. Appl. Artif. Intell. 138 (2024) 109268.
[64] V.M. Nagulapati, H. Lee, D. Jung, S.S. Paramanantham, B. Brigljevic, Y. Choi, H. Lim, A novel combined multi-battery dataset based approach for enhanced prediction accuracy of data driven prognostic models in capacity estimation of lithium ion batteries, Energy AI 5 (2021) 100089.
[65] D. Kong, S. Wang, P. Ping, State-of-health estimation and remaining useful life for lithium-ion battery based on deep learning with Bayesian hyperparameter optimization, Int. J. Energy Res. 46 (5) (2022) 6081–6098.
[66] K. Liu, Y. Shang, Q. Ouyang, W.D. Widanage, A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery, IEEE Trans. Ind. Electron. 68 (4) (2020) 3170–3180.
[67] S. Wang, Y. Fan, S. Jin, P. Takyi-Aninakwa, C. Fernandez, Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries, Reliab. Eng. Syst. Saf. 230 (2023) 108920.
[68] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, M.J. Deen, A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life, IEEE Trans. Ind. Inform. 17 (5) (2020) 3478–3487.
[69] H. Pei, X.-S. Si, C. Hu, T. Li, C. He, Z. Pang, Bayesian deep-learning-based prognostic model for equipment without label data related to lifetime, IEEE Trans. Syst. Man Cybern.: Syst. 53 (1) (2022) 504–517.
[70] J. Alcibar, J.I. Aizpurua, E. Zugasti, Towards a probabilistic fusion approach for robust battery prognostics, 2024, arXiv preprint arXiv:2405.15292.
[71] S.-Y.M. Louis, A. Nasiri, J. Bao, Y. Cui, Y. Zhao, J. Jin, X. Huang, J. Hu, Remaining useful strength (RUS) prediction of SiCf-SiCm composite materials using deep learning and acoustic emission, Appl. Sci. 10 (8) (2020) 2680.
[72] T. Loutas, N. Eleftheroglou, D. Zarouchas, A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data, Compos. Struct. 161 (2017) 522–529.
[73] M. Alamaniotis, Explainable prognostics method through differential evolved RVR ensemble of relevance vector machines, in: Annual Conference of the PHM Society, vol. 15, 2023.
[74] Z. Zhang, Y.-X. Wang, H. He, F. Sun, A short-and long-term prognostic associating with remaining useful life estimation for proton exchange membrane fuel cell, Appl. Energy 304 (2021) 117841.
[75] D. Xiao, C. Qin, J. Ge, P. Xia, Y. Huang, C. Liu, Self-attention-based adaptive remaining useful life prediction for IGBT with Monte Carlo dropout, Knowl.-Based Syst. 239 (2022) 107902.
[76] O. Serradilla, E. Zugasti, C. Cernuda, A. Aranburu, J.R. de Okariz, U. Zurutuza, Interpreting remaining useful life estimations combining explainable artificial intelligence and domain knowledge in industrial machinery, in: 2020 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2020, pp. 1–8.
[77] S. Liu, L. Fan, An adaptive prediction approach for rolling bearing remaining useful life based on multistage model with three-source variability, Reliab. Eng. Syst. Saf. 218 (2022) 108182.
[78] T. Gao, Y. Li, X. Huang, C. Wang, Data-driven method for predicting remaining useful life of bearing based on Bayesian theory, Sensors 21 (1) (2020) 182.
[79] S.A. Aye, P. Heyns, An integrated Gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission, Mech. Syst. Signal Process. 84 (2017) 485–498.
[80] Y.-F. Ma, X. Jia, Q. Hu, H. Bai, C. Guo, S. Wang, A new state recognition and prognosis method based on a sparse representation feature and the hidden semi-Markov model, IEEE Access 8 (2020) 119405–119420.
[81] N. Wang, S.-d. Sun, Z.-q. Cai, S. Zhang, C. Saygin, et al., A hidden semi-Markov model with duration-dependent state transition probabilities for prognostics, Math. Probl. Eng. 2014 (2014).
[82] N. Eleftheroglou, D. Zarouchas, R. Benedictus, An adaptive probabilistic data-driven methodology for prognosis of the fatigue life of composite structures, Compos. Struct. 245 (2020) 112386.
[83] N. Eleftheroglou, T. Loutas, Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes, Struct. Heal. Monit. 15 (4) (2016) 473–488.
[84] N. Eleftheroglou, G. Galanopoulos, T. Loutas, Similarity learning hidden semi-Markov model for adaptive prognostics of composite structures, Reliab. Eng. Syst. Saf. 243 (2024) 109808.
[85] A. Giantomassi, F. Ferracuti, A. Benini, G. Ippoliti, S. Longhi, A. Petrucci, Hidden Markov model for health estimation and prognosis of turbofan engines, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 54808, 2011, pp. 681–689.
[86] R. Moghaddass, M.J. Zuo, An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process, Reliab. Eng. Syst. Saf. 124 (2014) 92–104.
[87] Y. Deng, A. Di Bucchianico, M. Pechenizkiy, Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model, Reliab. Eng. Syst. Saf. 196 (2020) 106727.

[88] K. Le Son, M. Fouladirad, A. Barros, E. Levrat, B. Iung, Remaining useful life estimation based on stochastic deterioration models: A comparative study, Reliab. Eng. Syst. Saf. 112 (2013) 165–175.

[89] Y.-H. Lin, Z.-Q. Ding, Y.-F. Li, Similarity based remaining useful life prediction based on Gaussian process with active learning, Reliab. Eng. Syst. Saf. 238 (2023) 109461.

[90] N. Eleftheroglou, S.S. Mansouri, T. Loutas, P. Karvelis, G. Georgoulas, G. Nikolakopoulos, D. Zarouchas, Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the Lithium-Polymer batteries of unmanned aerial vehicles with uncertainty quantification, Appl. Energy 254 (2019) 113677.

[91] X. Hu, Y. Che, X. Lin, Z. Deng, Health prognosis for electric vehicle battery packs: A data-driven approach, IEEE/ASME Trans. Mechatronics 25 (6) (2020) 2622–2632.

[92] N. Li, M. Wang, Y. Lei, X. Si, B. Yang, X. Li, A nonparametric degradation modeling method for remaining useful life prediction with fragment data, Reliab. Eng. Syst. Saf. 249 (2024) 110224, http://dx.doi.org/10.1016/j.ress.2024.110224, URL: https://www.sciencedirect.com/science/article/pii/S0951832024002977.

[93] T. Liu, K. Zhu, L. Zeng, Diagnosis and prognosis of degradation process via hidden semi-Markov model, IEEE/ASME Trans. Mechatronics 23 (3) (2018) 1456–1466.

[94] F. Xie, B. Wu, Y. Hu, Y. Wang, G. Jia, Y. Cheng, A generalized interval probability-based optimization method for training generalized hidden Markov model, Signal Process. 94 (2014) 319–329.

[95] T. Le, F. Chatelain, C. Berenguer, Hidden Markov models for diagnostics and prognostics of systems under multiple deterioration modes, in: Proceedings of the in European Safety and Reliability Conference-ESREL, 2014, pp. 1197–1204, http://dx.doi.org/10.1201/b17399-166.

[96] Q. Liu, M. Dong, W. Lv, X. Geng, Y. Li, A novel method using adaptive hidden semi-Markov model for multi-sensor monitoring equipment health prognosis, Mech. Syst. Signal Process. 64 (2015) 217–232.

[97] A. Oikonomou, N. Eleftheroglou, F. Freeman, T. Loutas, D. Zarouchas, Remaining useful life prognosis of aircraft brakes, Int. J. Progn. Heal. Manag. 13 (1) (2022).

[98] N. Li, P. Xu, Y. Lei, X. Cai, D. Kong, A self-data-driven method for remaining useful life prediction of wind turbines considering continuously varying speeds, Mech. Syst. Signal Process. 165 (2022) 108315, http://dx.doi.org/10.1016/j.ymssp.2021.108315, URL: https://www.sciencedirect.com/science/article/pii/S0888327021006762.

[99] H. Cai, J. Feng, W. Li, Y. Hsu, J. Lee, Similarity-based particle filter for remaining useful life prediction with enhanced performance, Appl. Soft Comput. 94 (2020) 106474.

[100] F. Khan, O.F. Eker, A. Khan, W. Orfali, Adaptive degradation prognostic reasoning by particle filter with a neural network degradation model for turbofan jet engine, Data 3 (4) (2018) 49.

[101] A. Mosallam, K. Medjaher, N. Zerhouni, Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction, J. Intell. Manuf. 27 (2016) 1037–1048.

[102] N. Li, Y. Lei, T. Yan, N. Li, T. Han, A Wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability, IEEE Trans. Ind. Electron. 66 (3) (2018) 2092–2101.

[103] J. Wei, G. Dong, Z. Chen, Remaining useful life prediction and state of health diagnosis for lithium-ion batteries using particle filter and support vector regression, IEEE Trans. Ind. Electron. 65 (7) (2017) 5634–5643.

[104] F. Cadini, C. Sbarufatti, F. Cancelliere, M. Giglio, State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters, Appl. Energy 235 (2019) 661–672.

[105] W. Yan, B. Zhang, W. Dou, D. Liu, Y. Peng, Low-cost adaptive lebesgue sampling particle filtering approach for real-time li-ion battery diagnosis and prognosis, IEEE Trans. Autom. Sci. Eng. 14 (4) (2017) 1601–1611, http://dx.doi.org/10.1109/TASE.2017.2666202.

[106] P. Banerjee, O. Karpenko, L. Udpa, M. Haq, Y. Deng, Prediction of impact-damage growth in GFRP plates using particle filtering algorithm, Compos. Struct. 194 (2018) 527–536.

[107] T. Li, J. Chen, S. Yuan, F. Cadini, C. Sbarufatti, Particle filter-based damage prognosis using online feature fusion and selection, Mech. Syst. Signal Process. 203 (2023) 110713.

[108] R.K. Singleton, E.G. Strangas, S. Aviyente, Extended Kalman filtering for remaining-useful-life estimation of bearings, IEEE Trans. Ind. Electron. 62 (3) (2014) 1781–1790.

[109] C. Chen, G. Vachtsevanos, M.E. Orchard, Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach, Mech. Syst. Signal Process. 28 (2012) 597–607.

[110] L. Tongyang, W. Shaoping, S. Jian, M. Zhonghai, An adaptive-order particle filter for remaining useful life prediction of aviation piston pumps, Chin. J. Aeronaut. 31 (5) (2018) 941–948.

[111] A. Der Kiureghian, O. Ditlevsen, Aleatory or epistemic? Does it matter? Struct. Saf. 31 (2) (2009) 105–112.

[112] S. Sankararaman, Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction, Mech. Syst. Signal Process. 52 (2015) 228–247.

[113] C. Nastos, P. Komninos, D. Zarouchas, Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods, Compos. Struct. 311 (2023) 116815.

[114] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion 76 (2021) 243–297.

[115] V. Nemani, L. Biggio, X. Huan, Z. Hu, O. Fink, A. Tran, Y. Wang, X. Zhang, C. Hu, Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial, Mech. Syst. Signal Process. 205 (2023) 110796.

[116] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.

[117] S. Fort, H. Hu, B. Lakshminarayanan, Deep ensembles: A loss landscape perspective, 2019, arXiv 2019, arXiv preprint arXiv:1912.02757.

[118] L.L. Folgoc, V. Baltatzis, S. Desai, A. Devaraj, S. Ellis, O.E.M. Manzanera, A. Nair, H. Qiu, J. Schnabel, B. Glocker, Is MC dropout Bayesian? 2021, arXiv preprint arXiv:2110.04286.

[119] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[120] G. Li, L. Yang, C.-G. Lee, X. Wang, M. Rong, A Bayesian deep learning RUL framework integrating epistemic and aleatoric uncertainties, IEEE Trans. Ind. Electron. 68 (9) (2020) 8829–8841.

[121] F.-Y. Xie, Y.-M. Hu, B. Wu, Y. Wang, A generalized hidden Markov model and its applications in recognition of cutting states, Int. J. Precis. Eng. Manuf. 17 (2016) 1471–1482.

[122] E. Zio, Reliability engineering: Old problems and new challenges, Reliab. Eng. Syst. Saf. 94 (2) (2009) 125–141, http://dx.doi.org/10.1016/j.ress.2008.06.002, URL: https://www.sciencedirect.com/science/article/pii/S0951832008001749.

[123] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.

[124] B. Mucsányi, M. Kirchhof, S.J. Oh, Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks, 2024, arXiv preprint arXiv:2402.19460.
[125] N. Eleftheroglou, Adaptive Prognostics for Remaining Useful Life of Composite Structures (Ph.D. thesis), Delft University of Technology, 2020, http://dx.doi.org/10.4233/uuid:538558fb-ac9a-414d-8a59-4b523d8ff74c.
[126] N. Eleftheroglou, Adaptive prognostics: A reliable RUL approach, in: Annual Conference of the PHM Society, vol. 15, 2023.
[127] J. Sharma, M.L. Mittal, G. Soni, Condition-based maintenance using machine learning and role of interpretability: a review, Int. J. Syst. Assur. Eng. Manag. 15 (4) (2024) 1345–1360.
[128] K. Kobayashi, S.B. Alam, Explainable, interpretable, and trustworthy AI for an intelligent digital twin: A case study on remaining useful life, Eng. Appl. Artif. Intell. 129 (2024) 107620.
[129] O. Fink, Q. Wang, M. Svensen, P. Dersin, W.-J. Lee, M. Ducoffe, Potential, challenges and future directions for deep learning in prognostics and health management applications, Eng. Appl. Artif. Intell. 92 (2020) 103678.
[130] D. Alvarez-Melis, T.S. Jaakkola, On the robustness of interpretability methods, 2018, arXiv preprint arXiv:1806.08049.
[131] D. Garreau, Chapter 14 - theoretical analysis of LIME, in: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quénot (Eds.), Explainable Deep Learning AI, Academic Press, 2023, pp. 293–316, http://dx.doi.org/10.1016/B978-0-32-396098-4.00020-X, URL: https://www.sciencedirect.com/science/article/pii/B978032396098400020X.
[132] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J.M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, F. Herrera, Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence, Inf. Fusion 99 (2023) 101805, http://dx.doi.org/10.1016/j.inffus.2023.101805, URL: https://www.sciencedirect.com/science/article/pii/S1566253523001148.
[133] W.J. Verhagen, B.F. Santos, F. Freeman, P. van Kessel, D. Zarouchas, T. Loutas, R.C. Yeun, I. Heiets, Condition-based maintenance in aviation: Challenges and opportunities, Aerospace 10 (9) (2023) 762.
[134] N. Gebraeel, Y. Lei, N. Li, X. Si, E. Zio, et al., Prognostics and remaining useful life prediction of machinery: advances, opportunities and challenges, J. Dyn. Monit. Diagn. (2023) 1–12.
[135] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: 2008 International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–9.
[136] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, C. Varnier, PRONOSTIA: An experimental platform for bearings accelerated degradation tests, in: IEEE International Conference on Prognostics and Health Management, PHM'12, IEEE Catalog Number: CPF12PHM-CDR, 2012, pp. 1–8.
[137] H. Meng, Y.-F. Li, A review on prognostics and health management (PHM) methods of lithium-ion batteries, Renew. Sustain. Energy Rev. 116 (2019) 109405.
[138] B. Saha, K. Goebel, Battery data set, 2007, NASA AMES prognostics data repository.
[139] B. Bole, C. Kulkarni, M. Daigle, Randomized battery usage data set, 70, 2014, NASA AMES prognostics data repository.
[140] J. Coble, J.W. Hines, Identifying optimal prognostic parameters from data: a genetic algorithms approach, in: Annual Conference of the PHM Society, vol. 1, 2009.
[141] S. Yue, P. Pilon, A comparison of the power of the t test, Mann-Kendall and bootstrap tests for trend detection/une comparaison de la puissance des tests t de student, de Mann-Kendall et du bootstrap pour la détection de tendance, Hydrol. Sci. J. 49 (1) (2004) 21–37.
[142] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (1) (2012) 723–773.
[143] F. Cadini, C. Sbarufatti, M. Corbetta, F. Cancelliere, M. Giglio, Particle filtering-based adaptive training of neural networks for real-time structural damage diagnosis and prognosis, Struct. Control. Heal. Monit. 26 (12) (2019) e2451.
[144] W. Deng, K.T. Nguyen, K. Medjaher, Physics informed self supervised learning for fault diagnostics and prognostics in the context of sparse and noisy data, in: PHM Society European Conference, vol. 7, 2022, pp. 574–576.