Counterfactual explanations for remaining useful life estimation within a Bayesian framework

Andringa, Jilles; Baptista, Marcia L.; Santos, Bruno F.

DOI
10.1016/j.inffus.2025.102972

Publication date
2025

Document Version
Final published version

Published in
Information Fusion

Citation (APA)
Andringa, J., Baptista, M. L., & Santos, B. F. (2025). Counterfactual explanations for remaining useful life estimation within a Bayesian framework. *Information Fusion*, *118*, Article 102972. https://doi.org/10.1016/j.inffus.2025.102972

Contents lists available at ScienceDirect

Information Fusion

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

Counterfactual explanations for remaining useful life estimation within a Bayesian framework

Jilles Andringa [a], Marcia L. Baptista [b],*, Bruno F. Santos [c]

[a] Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands
[b] NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Lisboa, 1070-312, Portugal
[c] KLM Royal Dutch Airlines, P.O. Box 7700, ZL Schiphol., Amsterdam, 1117, The Netherlands

ARTICLE INFO

Keywords:
Explainable Artificial Intelligence
Prognostics and Health Management
C-MAPSS dataset
Bayesian uncertainty
Counterfactual explanations
Interpretability and explainability
Data augmentation

ABSTRACT

Machine learning has contributed to the advancement of maintenance in many industries, including aviation. In recent years, many neural network models have been proposed to address the problems of failure identification and estimating the remaining useful life (RUL). Nevertheless, the black-box nature of neural networks often limits their transparency and interpretability. Interpretability (or explainability) in maintenance refers to the ability of a predictive model to provide insights into its decision-making process for predicting failures or estimating metrics like RUL. Counterfactual Explanations (CFEs) from Explainable AI (XAI) addresses this problem by explaining model decisions through hypothetical scenarios leading to alternative outcomes. A kind of neural network that could benefit from increased interpretability is Bayesian networks. In general, Bayesian models improve interpretability by quantifying uncertainty. However, incorporating Bayesian uncertainty into neural networks adds complexity because we often need a statistical distribution for each network parameter. This study investigates the use of CFEs within a Bayesian framework to achieve two key objectives simultaneously: (1) enhance the interpretability of RUL estimations and (2) improve model accuracy. We generate two types of CFEs: (1) RUL CFEs that increase/decrease the RUL estimation and (2) uncertainty CFEs with reduced estimation uncertainty, which we use to augment the dataset and increase model accuracy. We apply this method to a classical case study, the C-MAPSS dataset, using a Bayesian Long Short-Term Memory (B-LSTM) model. We demonstrate that CFEs can help identify critical features and fine-tune corrective actions to achieve specific outcomes. For example, following a maintenance action that increased the temperature by 1° F, CFEs can reveal that this adjustment extended the equipment's useful life by 30 cycles. This ability to correlate specific actions with effects enhances both decision-making and maintenance efficiency. Additionally, our data augmentation approach results in a 5% improvement in $\alpha - \lambda$ accuracy for a strict α of 20%. The root mean square error (RMSE) of the B-LSTM model decreases from 9.56 to 8.47 cycles, demonstrating the potential of Uncertainty CFEs to improve accuracy in aircraft maintenance. The code is publicly available at Github.

1. Introduction

In a typical scenario of preventive maintenance (PrvM) without artificial intelligence (AI), decisions are made based on scheduled maintenance or reactive approaches when equipment fails. Aircraft maintenance has been undergoing a transformative evolution, aligned with advances seen in other fields [1]. This evolution is driven by the integration of more in-depth information into maintenance systems. The research area of **predictive maintenance** (PrdM) [2] is already based on the idea of making maintenance decisions based on data-driven models. With predictive maintenance, we rely on AI to analyze the data and predict when maintenance should occur. **Prescriptive maintenance** (PrcM) [3–5] takes maintenance one step further by **explaining** decision-making processes, vital to guarantee high standards of safety, efficiency and regulatory compliance. This is of particular importance for the aviation sector. Much of this transformation is based on the new technologies of explainable AI (XAI) [6]. XAI is a set of techniques to explain the internal workings of AI models, including the purpose of their components, logical reasoning, and decision-making processes [7]. Fig. 1 illustrates the described evolution of maintenance.

Currently, predictive maintenance (PrdM) relies (mostly) on neural networks [8]. Although these predictive models have been shown to be satisfactorily accurate in their predictions, they pose challenging issues

* Corresponding author.
E-mail addresses: j.y.andringa@student.tudelft.nl (J. Andringa), m.baptista@novaims.unl.pt (M.L. Baptista), bruno.f.santos@klm.nl (B.F. Santos).

https://doi.org/10.1016/j.inffus.2025.102972
Received 8 July 2024; Received in revised form 26 October 2024; Accepted 18 January 2025
Available online 28 January 2025

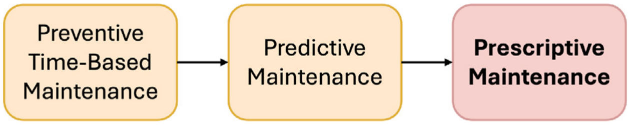


Fig. 1. Evolution of Maintenance. Preventive maintenance involves scheduled inspections and routine tasks. Predictive maintenance utilizes sensor data and AI to perform maintenance just-in-time. Prescriptive maintenance goes beyond prediction, offering actionable insights based on explainable artificial intelligence (XAI).

regarding their interpretability [9]. For example, neural networks suffer from a **"black box"** nature, which limits the transparency of their predictions. The field of XAI [1] attempts to address this problem by developing methods to make machine learning more **explainable** and **interpretable**. Here, we treat interpretability and explainability interchangeably, following the work and review of Kumar et al. [10]. Both concepts aim to address the challenge of making machine learning models more understandable and trustworthy for the different stakeholders.

Quantifying interpretability and explainability in the context of machine learning and artificial intelligence can be challenging, as these concepts are often subjective and context-dependent [11]. However, interpretability in maintenance can be generally defined as the ease by which the maintenance stakeholders can understand the system's behavior, potential failure modes, and the impact of various maintenance actions [6].

A method from XAI is **Counterfactual Explanations** (CFEs), which aim to provide more information on a model and its predictions by suggesting alternative scenarios that would have led to a different outcome [12]. Apart from CFEs, there are other explanation methodologies in XAI, such as decision-theory-based explanations, contrastive explanations, example-based explanations, or attributional explanations [13]. CFEs are of special interest, as counterfactual analysis enables researchers to make inferences by comparing observed outcomes with hypothetical outcomes under different conditions [14].

In the context of maintenance, a CFE scenario will involve asking "What changes in input would impact the prediction of remaining useful life (RUL)?" These input changes can result from two main factors: maintenance actions or improvements in data quality. Corrective maintenance actions, like adjusting operational parameters or replacing components, can be impacted by CFE analysis. On the other hand, the CFEs might signal the need for improving data quality to enhance the accuracy of predictions. In our specific work, we are interested in two subquestions (see Fig. 2):

- **RUL CFEs**: What changes in input (alternative scenarios) would increase/decrease the Remaining Useful Life (RUL) prediction?
- **Uncertainty CFEs**: What changes in input would yield a more precise RUL prediction?

The first question is particularly important as it enables us to assess the effects of maintenance interventions. For instance, if we reason (based on CFEs) that by reducing a specific temperature we could extend the life of the equipment, we could tailor the effectiveness of our maintenance efforts to target those particular values. This type of CFEs we designate as Increasing/Decreasing **RUL CFEs**.

We were also interested in **Uncertainty CFEs** (adapted from Ley et al. [15]), which address a distinct question: "What alterations in the input variables would lead to a more precise RUL prediction?" Here, our focus lies in narrowing the uncertainty bounds surrounding the current RUL prediction. This approach holds potential for data augmentation. Our rationale is centered on ensuring that the augmented data maintains quality standards with respect to RUL prediction.

In this paper, we study the viability of CFEs to attain two important goals: (1) interpretability and (2) accuracy. In AI, there is much discussion about these dimensions. There is a well-known paradigm stating that interpretability and accuracy (performance) are two perspectives that are difficult to balance in machine learning. For example, Espinosa et al. [16] advocate that the more accurate a model is, the less explicable it becomes. A goal of this project is to show empirically that **interpretability** and **accuracy** can be improved using an XAI technique: the CFEs in this case.

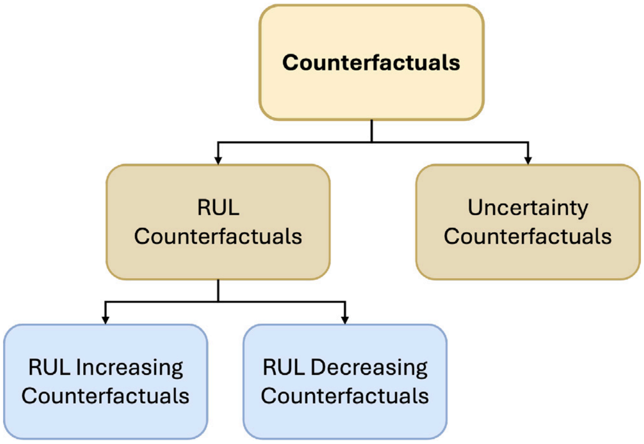

Counterfactuals
RUL Counterfactuals
Uncertainty Counterfactuals
RUL Increasing Counterfactuals
RUL Decreasing Counterfactuals

Fig. 2. Counterfactual Explanations (CFEs). In the context of maintenance, we distinguish between RUL CFEs and Uncertainty CFEs. The first designates a change in the model inputs (sensor data) to produce a RUL change. The second decreases the uncertainty of the RUL prediction.

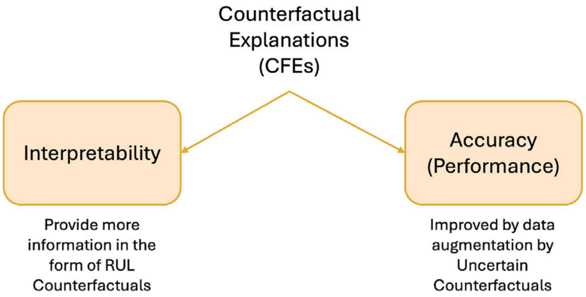

Counterfactual Explanations (CFEs)
Interpretability
Accuracy (Performance)
Provide more information in the form of RUL Counterfactuals
Improved by data augmentation by Uncertain Counterfactuals

Fig. 3. Goals of this work.

As illustrated in Fig. 3, this work explores the utility of different types of CFEs to enhance the (a) interpretability and (b) performance of predictive models in aircraft maintenance. Our main research question is as follows.

RQ: How can Counterfactual Explanations (CFEs) be used in predictive maintenance to improve the interpretability and performance of a Bayesian Long Short Term Memory (B-LSTM)?

We are going to analyze the interpretability of CFEs based on various visual tools. Our contribution is to show that RUL CFEs can be utilized to analyze the impact of maintenance events (maintainability) and to understand the different degradation stages of an equipment.

In this study, we argue that CFEs (RUL and Uncertainty) contain extra information that can enhance model performance when fed back into the system. Achieving accurate RUL models typically requires a substantial volume of data, which can be both expensive and time-intensive to acquire. This is particularly true for critical aircraft systems, where airlines and manufacturers have only a limited number of failure signatures. Here, **data augmentation** is a possible strategy. We combine these ideas by generating new training data based on CFEs (on a holdout set).

We have opted for a Bayesian Long Short Term Memory (B-LSTM) as our primary RUL model since Bayesian models naturally capture the uncertainty associated with RUL estimation [17]. Instead of providing a single-point estimate for each prediction, Bayesian models offer a distribution defining the range of possible RUL values. The LSTM component captures the temporal dependencies within the time series data. In general, we can summarize the contributions of our work in two dimensions:

- **Interpretability:** Discussion and proposal of visual tools to analyze the maintainability and reliability of engineering systems with **RUL CFEs**
- **Accuracy:** Comparison of the performance of various RUL models augmented with different types of **CFEs** (RUL and Uncertainty).

The remainder of this paper is structured as follows. Section 2 will cover the theoretical background and previous literature on related work. Section 3 describes the methodology used in this study followed by a description of the C-MAPSS case study in Section 4. Finally, we will discuss the results in Section 5. We conclude in Section 6.

2. Related work

This section reviews related work in the field of Predictive Maintenance (PrdM) (Section 2.1), and in explainable AI (XAI) (Section 2.2). We conclude with a revision of the latest advancements in counterfactual analysis (Section 2.3).

2.1. Predictive maintenance

Preventive maintenance (PrvM), also known as calendar-based maintenance, is the traditional approach to maintenance that involves performing routine inspections, servicing, and repairs on equipment or machinery according to a predetermined schedule [18]. Unlike Predictive Maintenance (PrdM), which uses advanced algorithms to predict equipment failures based on real-time data, preventive maintenance does not use AI. This type of maintenance may be effective for routine tasks and preventive care, but it may not be as efficient as AI-driven predictive maintenance, since it can lead to overmaintenance, where equipment is serviced more frequently than necessary [19,20].

Predictive maintenance is different from preventive maintenance in that it utilizes historical and real-time data to perform failure prognostics [21], which entails predicting a system's future behavior and how it will fail. This includes predicting a system's Remaining Useful Life (RUL) [22].

In general, there are three main approaches to estimating the RUL: model-based, data-driven, and hybrid, as described by Chao et al. [23]. Model-based approaches (also known as physics-based approaches) are a popular PHM approach due to their accuracy, precision, and real-time performance. However, these methods require a deep understanding of the physics of the system [24]. Data-driven methods offer an alternative to model-based approaches [25]. Their optimal performance depends on the availability of substantial historical and current data. The hybrid approach aims to combine the strengths of both model-based and data-driven approaches [26].

The selection of a data-driven approach in this study was motivated by its practicality and effectiveness in handling extensive historical data. These methods, particularly when built upon machine learning pipelines, tend to be more challenging to interpret [27], which aligns well with the analytical goals of our study. We aim at maximizing the benefits of proposed interpretability techniques.

An overview of the data-driven methods applied in RUL estimation is provided by Ansari et al. [28]. The authors classify the methods into two main categories: statistical and machine learning. Statistical methods use empirical knowledge and data to build statistical models for the estimation of RUL. Statistical models used in previous work are, for instance, the Auto-Regressive Integrated Moving Average (ARIMA) technique [29–31], the Gray Model (GM) [32–34], the Wiener Process (WP) [35–38], and entropy analysis [39].

In previous work on RUL estimation, various machine learning models have been applied, such as Naive Bayes [40–42], Support Vector Regression [43,44], Relevance Vector Machines [45,46], Gaussian Process Regression [47–49] and Deep Neural Networks (DNNs) [49–60].

In this research, we use a Bayesian Long Short-Term Memory (B-LSTM). The LSTM is a specialized type of neural network, a recurrent neural network (RNN) designed to learn long-term dependencies. Although evaluation of different neural networks has shown varying performance results, the LSTM appears to be a favored method for estimating RUL [28]. It has been applied to RUL estimation in previous work [49–52,56]. We selected this network because of its ability to discern complex, nonlinear temporal relationships [61].

It is possible to apply the Bayes rule to different models to quantify the uncertainty of the models. Examples of such models are Naive Bayes [62], Bayesian linear regression [63], Bayesian Networks [64], Gaussian Process Regression [65] and Relevance Vector Machines [66]. Our research focuses on the Bayesian LSTM. B-LSTMs have been applied to RUL estimation in previous work. For example, Caceres et al. [67] compared multiple Bayesian RNN networks (including LSTMs).

2.2. Explainable Artificial Intelligence (XAI)

Prescriptive maintenance (PrcM) [5,68,69] is an advanced maintenance strategy that goes beyond predictive maintenance (PrdM) by not only predicting equipment failures but also providing actionable recommendations to optimize maintenance activities. Explainable Artificial Intelligence (XAI) has emerged as a promising technology in this domain, providing the possibility to develop interactions between AI systems and various stakeholders while deciphering the decision-making processes of complex "black box" models to enhance understandability.

Concerns about the lack of interpretability in data-driven methods were raised early when neural networks were first introduced in predictive maintenance [70]. However, this problem has been aggravated with the widespread adoption of deep learning models. Some reviews and papers [5,68,69] notice this research gap. However, in the predictive maintenance community, the lack of interpretability is often regarded as a consequence of data-driven methods rather than as an issue that can be effectively addressed.

In prescriptive maintenance, we can distinguish between two approaches to XAI: intrinsically interpretable models and post-hoc explanations. Authors sometimes use other taxonomies with more dimensions to categorize models in XAI. For example, Speith [71] discusses four different approaches to constructing taxonomies, such as the functioning-based approach, the result-based approach, the conceptual approach, and the mixed approach. For simplicity we rely solely on the distinction between interpretable models and post-hoc explanations. This coincides with the conceptual approach of Speith [71].

In intrinsically interpretable approaches, models are designed to provide transparency and understandability by their inherent structure, making them readily interpretable without the need for additional post-hoc explanations. In predictive maintenance, a variety of intrinsically interpretable models have been employed, ranging from methods such as ontologies [72–74], decision trees [75,76] and filtering approaches [77] to stochastic models like the Wiener process [78] and hidden Markov models [79]. While interpretable models for diagnosis and detection often rely on knowledge, rules, and decision trees, intrinsically interpretable models for RUL estimation predominantly consist of stochastic models.

The surge of models such as deep neural networks prompted a shift away from intrinsically interpretable ML models. Consequently, there has been further developments in post-hoc methods aimed at explaining the decisions of complex black-box models. Post-hoc explanations are generated after the model has made predictions and aim to elucidate its decision-making process, often through techniques like feature importance. Following Arrieta et al. [80], post-hoc explanations can be classified into visual, feature relevance-based explanations, knowledge-extraction as well as example-based.

- **Visual explanations** utilize graphical plots and summaries, such as heatmap overlays, decision boundary plots, and feature importances, to provide insights into model behavior.
- **Feature-relevance explanations** quantify feature contributions, often combined with visual explanations for clarity.
- **Knowledge-extraction explanations** transfer hidden knowledge within the model to more transparent representations, such as symbolic rules or surrogate models approximating predictions.
- **Example-based explanations** select specific data instances, like prototypes or underrepresented examples, to elucidate model behavior, while counterfactual examples describe changes needed to alter predictions.

Examples of works with visual explanations in predictive maintenance include the work of Kozielski [81]. The authors presented SHAP-based explanations using local context heatmaps. The use of correlation maps to express feature importance is also a common methodology [82,83]. Another work using visual explanations to explore feature importance is by Alomari et al. [84]. These explanations are often combined with feature-relevance explanations to improve comprehensibility.

Knowledge-extraction explanations [85] may involve algorithms for rule extraction [86–88] or the use of surrogate models such as SHAP and LIME. One example of XAI applied to RUL estimation comes from Hong et al. [89], who applied it to C-MAPSS using the SHAP explanation model. Other previous work regarding XAI in maintenance is for instance from Sundar et al. [90] using the LIME model, and Onchis and Gillich [91] who used LIME and SHAP. Baptista et al. [6] showed that SHAP explanations formed meaningful trajectories.

Work on example-based explanations covers counterfactuals and causal inference [92]. Kozielski [81] studied contextual explanations for decision support in maintenance. By contextual explanations, the authors meant local (single prediction) explanations providing an understanding of what influenced a model decision for a particular data instance. Counterfactual explanations (CFEs) is a method introduced by Wachter et al. [93] that is extensively studied in other fields [94,95] but not extensively in maintenance. Exceptions are the works of Pileggi et al. [96],Jakubowski et al. [97] and Barraza et al. [98].

The field of prescriptive maintenance is currently characterized by a significant lack of research and exploration. Despite the growing recognition of the importance of prescriptive maintenance in optimizing asset performance and minimizing downtime, there remains a notable gap in the literature concerning the development and application of advanced technologies.

2.3. *Counterfactual explanations (CFEs)*

Nor et al. [99] performed a comprehensive literature review of XAI in maintenance. They concluded that XAI is mainly applied in the form of inherently interpretable models, rule-based and knowledge-based models and attention mechanisms. Counterfactual explanations (CFEs) is a post-hoc XAI method introduced by Wachter et al. [93]. CFEs work by asking the question What if? such as for instance:

> What if the output was Y instead of X what would have been the input?

In general, a counterfactual explanation can be defined as a perturbation of the input $\mathbf{x}$ to generate a different output y. This perturbed input can be seen as a counterfactual example $\mathbf{c}$. In mathematical form (see Eq. (1)), our objective is to minimize the yloss such that a different prediction is generated, while also minimizing the distance between the original input $\mathbf{x}$ and the counterfactual input $\mathbf{c}$ (referred to as proximity).

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} \left[\text{yloss}(f(\boldsymbol{c}), y) + |\boldsymbol{x} - \boldsymbol{c}| \right] \tag{1}$$

In this research, we are interested in questions such as:

- **RUL CFE:** What change in input could result in a specific increase/decrease in predicted RUL?
- **Uncertainty CFE:** What change in input could result in a more precise (less uncertain) prediction?

There are many platforms and strategies to generate CFEs. We review some important contributions. For example, Wiratunga et al. [100] proposed DisCERN, a nearest unlike neighbor (NUN) approach combined with model-agnostic feature relevance algorithms to generate counterfactuals with minimal feature change. Chen et al. [101] proposed RELAX, a model-agnostic platform to generate CFEs. The platform generates optimal CFEs via deep reinforcement learning (DRL). Another work of note is SAC-FACT which also used DRL for counterfactual generation [102]. Hamman et al. [103] focused on making counterfactual generation robust to small changes or updates to the model. Poyiadzi et al. [104] proposed FACE which is an algorithm that enforces that the generated counterfactuals meet the underlying data distribution and follow one of the "feasible paths" of change. AlJalaud and Hosny [105] proposed a genetic approach to generate counterfactuals. Genetic approaches to counterfactuals are typically model-agnostic, which makes them versatile across different model types. Other examples of evolutionary models for CFEs are in [106–108]. Other authors have used techniques such as autoenconders to generate counterfactuals. For example, Guyomard et al. [109] combined a predictor and a counterfactual generator, jointly trained, to produce counterfactuals. Other authors such as Sarathi et al. [110] focused on monotonic constraints while generating counterfactuals. Kuratomi et al. [111] proposed a counterfactual generation algorithm that provides justification based on features that rank high on plausibility, mutability, and directionality. Fernández et al. [112] proposed to generate instead of a single counterfactual a set of counterfactuals based on random forests. Mothilal et al. [113] proposed DiCE (Diverse Counterfactual Explanations), a counterfactual model-agnostic platform that focuses on preserving the diversity of the explanations while approximating local decision boundaries. Importantly, DiCE fulfills several critical properties necessary for generating effective counterfactuals namely the plausibility, proximity, diversity, and sparsity. [114]. In terms of diversity, this platform has been shown to outperform other approaches and is considered a reference.

3. Methodology

This section describes the methodology used in this research. We describe the general architecture in Section 3.1. The constituents of the architecture are explained in Section 3.2 (B-LSTM) and Section 3.3 (DiCE). We present the evaluation metrics in Section 3.4.

3.1. *General architecture*

The proposed architecture is a Bayesian long short-term memory (B-LSTM), from which we obtain probabilistic remaining useful life (RUL) predictions. These predictions are subject to an interpretability analysis using counterfactuals (RUL CFEs) generated from the DiCE platform [115]. We also use counterfactuals (RUL CFEs and Uncertainty CFEs) to feed back information to the network. In sum, this architecture involves the key components:

- **Bayesian Long Short-Term Memory (B-LSTM) Network**: The B-LSTM (see Fig. 4) is central to the architecture, capable of capturing temporal dependencies in the sensor data. The Bayesian approach is integrated into the LSTM architecture to account

Table 1
Hyperparameters for Bayesian LSTM (B-LSTM) model.

Hyperparameter	Value	Hyperparameter	Value
LSTM Neurons	32	LR Decay [epochs]	60
Dense Layers	2	Final LR	70%
Dense Neurons	32, 16	Validation split	20%
Epochs	100	Minimum delta	0.25
Learning Rate (LR)	1e-3	Patience [epochs]	5

for uncertainty in the model's predictions. This involves representing the network parameters (weights, bias) as probability distributions rather than fixed values.

- **Counterfactual Generation**: We generate counterfactuals from the B-LSTM with DiCE (Diverse Counterfactual Explanations) (see Fig. 5) by perturbing the input data to explore alternative scenarios. We use different plots to interpret the model and highlight the factors contributing to RUL estimation. These mechanisms help to make the model's behavior more transparent and understandable to end-users. We also augment the dataset with different types of counterfactuals (RUL CFEs and Uncertainty CFEs).

We focus on Bayesian Neural Networks (BNNs), which extend traditional neural network architectures by incorporating probabilistic reasoning into their framework. Unlike conventional neural networks that utilize fixed weights and biases, BNNs model these parameters as probability distributions. This approach is to some extent more complex than a traditional neural network. However, the quantification of uncertainty in predictions makes BNNs particularly suitable for applications where interpretability is important, such as this work.

It is important to explain that the outputs of a B-LSTM and a regular LSTM differ, even when optimized with the same parameters [116]. In a B-LSTM, the predicted output is generated by averaging over multiple probabilistic samples from the model's posterior distribution. Instead, a traditional LSTM provides deterministic predictions, without accounting for uncertainty. As a result, the mean prediction of a B-LSTM can vary from the corresponding output of a conventional LSTM. This justifies our choice of the B-LSTM to generate counterfactuals.

In general, the proposed architecture integrates probabilistic modeling, interpretability mechanisms, and data augmentation strategies to provide predictions and explanations while accounting for the uncertainty in the RUL outcomes.

3.2. Bayesian LSTM (B-LSTM)

The architecture of the B-LSTM model was inspired by the work of Caceres et al. [67]. Caceres et al. compared the performance of different Bayesian recurrent neural network (RNN) models in the C-MAPSS data set. The selected architecture, shown in Fig. 4, performed the best in our dataset.

To find the optimal distributions when training the Bayesian models, we used variational inference (VI), which aims to find the best-fitting distribution by minimizing the Kullback–Leibler (KL) divergence between distributions. When inference is performed on the model, the weights and biases need to be randomly sampled from their distributions in each prediction. In order to generate the RUL statistical distributions, we performed Monte Carlo simulations for each input.

Regarding hyperparameters, we relied on the optimization results of Caceres et al. [67] that are shown in Table 1. During training, a decaying learning rate (LR) was used to aid learning in later epochs. In addition, an early stop method was applied to stop the training process and prevent overfitting.

3.3. DiCE

The framework used to generate the counterfactuals used in this work is the DiCE model, developed by Mothilal et al. [113]. We have selected this framework for its ease of implementation, extensive documentation, compatibility with ML models, customizability, and overall performance.

DiCE is based on the counterfactual concept as described in Eq. (1). It adapts the concept as shown in Eq. (2). In this equation, the first term encourages the counterfactual input to produce a different output ($f(\mathbf{c}_i) = y$). The second term aims to keep the counterfactual input as close to the original input as possible, the third term seeks diversity among the k counterfactuals, and λ_1 and λ_2 are hyperparameters. DiCE iterates over the loss function until it converges and meets the desired condition (achieving a different output). It is important to note that all $\mathbf{c}_i$ values are initialized randomly.

$$C(\boldsymbol{x}) = \underset{c_1,\ldots,c_k}{\arg\min} \left[\frac{1}{k} \sum_{i=1}^{k} \text{yloss}\left(f(\boldsymbol{c}_i), y\right) + \frac{\lambda_1}{k} \sum_{i=1}^{k} \text{dist}(\boldsymbol{c}_i, \boldsymbol{x}) - \lambda_2 \text{dpp\_diversity}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k) \right] \tag{2}$$

The DiCE model was applied to this research as shown in Fig. 5. Firstly, to accommodate DiCE's limitation with temporal inputs, the (30 steps x 14 sensors) time series data was reshaped into a (1x420) vector. DiCE then slightly altered this reshaped input and evaluated whether the modified input yielded the desired output using the Bayesian Neural Network (BNN) model. If the desired RUL outcome was achieved, the altered input was accepted as a valid counterfactual (CF) and reshaped back to its original (30x14) format. If not, DiCE continued iteratively adjusting the input.

We also ensured that altering a feature value on a data point had a subsequent impact on later data points. figure 6 depicts a process of generating temporal counterfactual explanations for time-series data. The diagram represents 3 trajectories (the x-axis represents time and the y-axis represents temperature, a sensor feature). Generating temporal counterfactual trajectories (orange and green) involves tweaking the temperature at a specific time step and propagating through the time sequence from that time step onward.

DiCE is designed for models that produce fixed outputs, typically suitable for deterministic frameworks. In our approach, we used the mean value of a B-LSTM model instead of the deterministic output of a traditional LSTM to generate CFEs for RUL predictions. This distinction is important because B-LSTM outputs are derived from probabilistic sampling, reflecting uncertainty, unlike single-point predictions of a standard LSTM. These two networks, B-LSTM and LSTM, are different and produce different results.

Using DiCE, we were able to generate two kind of CFEs. The first CFEs in this research (**RUL CFES**) had the goal of finding explanations for how to increase or decrease the predicted RUL. To generate this type of counterfactuals, we calculated the mean of the prediction distribution as the output to adjust. By generating counterfactual inputs leading to a higher/lower RUL, we aimed to find explanations on how the life of the engine could have been changed.

The second application of CFEs (**Uncertainty CFEs**) had the goal of improving predictive performance. Generally, the more data you have available for a model to train on, the better its performance. As acquiring more data can be a long and costly effort, we propose a method to synthetically create more training data based on counterfactual inputs. In this case, we generate five different "augmented" models:

- **Baseline (without CFEs)**: This is the baseline model that uses all available training data
- **RUL CFEs (Increased RUL)**: This model is trained also on counterfactuals that increase the RUL

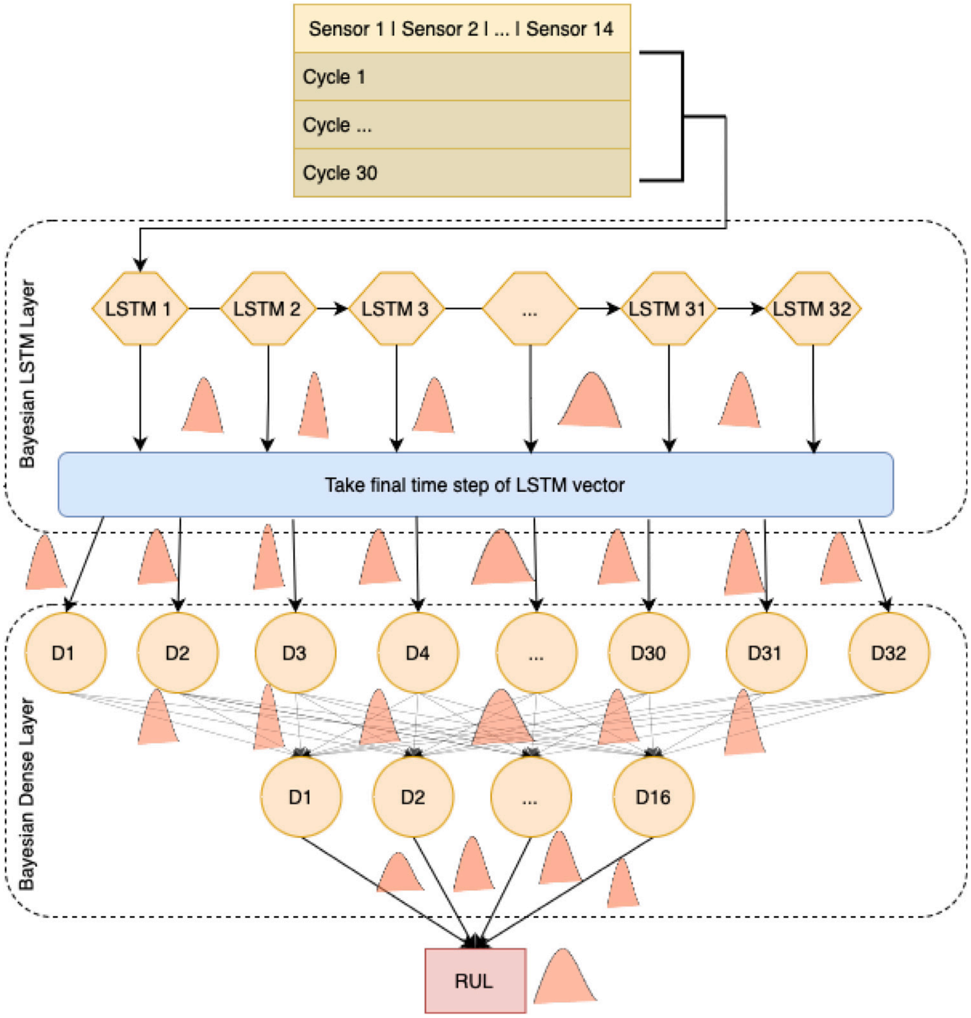


Fig. 4. Bayesian LSTM (B-LSTM) architecture. The model architecture consists of 1 LSTM layer, followed by 2 dense layers of 32 and 16 neurons respectively. Each input window of size (30 x 14) is fed into the LSTM layer sequentially. The architecture leads to a single output representing the RUL. Each weight and bias is represented by a distribution rather than a deterministic value, giving the model its probabilistic features.

- **RUL CFEs (Decreased RUL):** This model is trained also on counterfactuals that decrease the RUL
- **RUL CFEs (Combined):** This model is trained also on counterfactuals that decrease and increase the RUL
- **Uncertainty CFEs**: This model is trained also on CFEs generated from a hold-out set

3.4. Evaluation

In order to analyze the performance of the B-LSTM with each of these augmented datasets, we applied the Root Mean Squared Error (RMSE), the $\alpha - \lambda$ accuracy [117], and the asymmetric scoring function introduced by Saxena et al. [118].The normalized diversity score (NDS) was the metric used to evaluate the diversity of the generated counterfactuals. It is calculated as the ratio of the average pairwise distance to the maximum distance between all CFEs in the set.

$$D_{\text{norm}} = \frac{\frac{1}{\binom{N}{2}} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sqrt{\sum_{k=1}^{M} (x_{ik} - x_{jk})^2}}{\max_{i,j} \sqrt{\sum_{k=1}^{M} (x_{ik} - x_{jk})^2}}$$

Where:

- D_{norm}: Normalized diversity score.
- N: Total number of counterfactual vectors.
- M: Dimensionality of each vector.
- $\mathbf{x}_i$: The ith counterfactual vector.
- $d(\mathbf{x}_i, \mathbf{x}_j)$: Euclidean distance between vectors.
- $\binom{N}{2}$: Number of unique pairs of vectors.
- D_{avg}: Average pairwise distance between vectors.
- D_{max}: Maximum pairwise distance between vectors.

4. Case study: C-MAPSS

For this research, we used the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset as a case study to apply our proposed methods. C-MAPSS is a tool created by NASA [119] which can simulate a large commercial turbofan engine. Using this tool, a data set was created for the 2008 international conference on Prognostics and Health Mangement (PHM08) where attendees were challenged to create their best RUL prediction methods. Currently, it is a widely used data set in RUL estimation research [118].

The data set includes four separate sub data sets (FD001, FD002, FD003, FD004), which vary in number of engines, operating conditions and failure modes. For the goal of this research, the FD001 set was selected. It has 100 train/test trajectories, one operating condition, and one failure mode. Each engine within the dataset can be regarded as a member of an identical fleet of engines. Each engine contains a

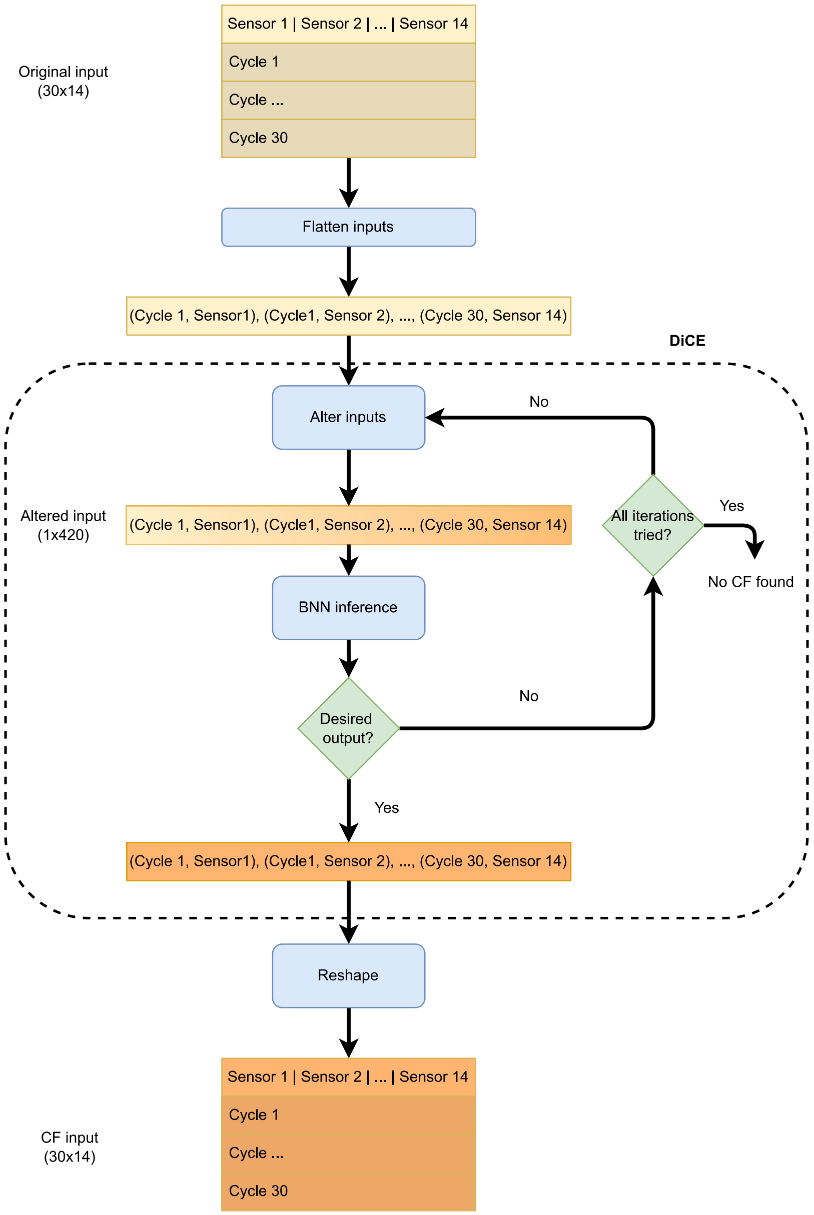

Original input
(30x14)
Sensor 1 | Sensor 2 | ... | Sensor 14
Cycle 1
Cycle ...
Cycle 30
Flatten inputs
(Cycle 1, Sensor1), (Cycle1, Sensor 2), ..., (Cycle 30, Sensor 14)
DiCE
Alter inputs
No
Altered input
(1x420)
(Cycle 1, Sensor1), (Cycle1, Sensor 2), ..., (Cycle 30, Sensor 14)
All iterations tried?
Yes
No CF found
BNN inference
Desired output?
No
Yes
(Cycle 1, Sensor1), (Cycle1, Sensor 2), ..., (Cycle 30, Sensor 14)
Reshape
CF input
(30x14)
Sensor 1 | Sensor 2 | ... | Sensor 14
Cycle 1
Cycle ...
Cycle 30

Fig. 5. DiCE model for counterfactual explanations (CFEs) generation. The (30 timesteps x 14 features) input is converted to a (1 timestep x 420 features) input, as DiCE is not able to handle 2D inputs such as our time series data. After this, DiCE alters the inputs slightly and checks if the desired output is achieved with this new input using the B-LSTM model. If so, the altered input is accepted as a valid counterfactual explanation and is then converted back to the original shape of (30x14).

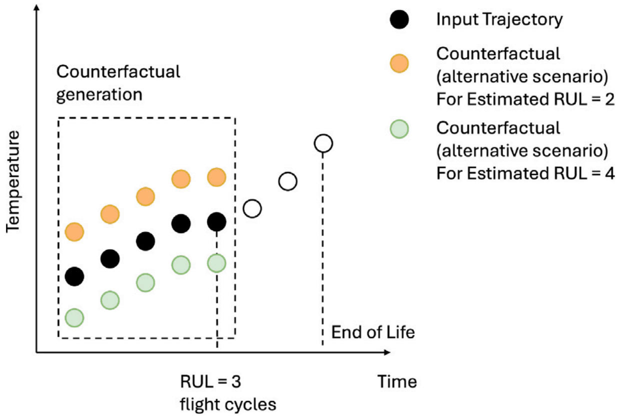


Fig. 6. Example of counterfactual explanations generation.

time-series set of data, where the amount of time steps represents an operating cycle of the engine. We assume that every engine operates at its standard capacity and begins to deteriorate at some point in the time series. However, the initial wear state of each engine remains unknown. Once a specific degradation threshold is reached, the engine is considered non-operational and has effectively reached the end of life (EOL). Furthermore, the dataset is affected by a certain level of noise.

For this research, we performed some preprocessing steps on the dataset before applying it to our models. For each cycle per engine, the data set contains the following [Engine number, cycle number, operational setting 1–3, sensor measurement 1-21]. As FD001 only has one operating condition, we removed the operational settings from the data set. In addition, we removed the cycle number to prevent subsequent overfitting. This leaves us with the raw sensor data found in Fig. 7.

The next steps in the pre-processing process consist of firstly, removing the unnecessary sensors, where it is clear that sensors 1,5,6,10,16,18 and 19 do not provide useful input, leaving 14 useful

Table 2
Sensor descriptions.

Sensor index	Description	Units
1	Total temperature at fan inlet	° R
2	Total temperature at low-pressure compressor (LPC) outlet	° R
3	Total temperature at high-pressure compressor (HPC) outlet	° R
4	Total temperature at LPC outlet	° R
5	Pressure at fan inlet	psia
6	Total pressure in bypass duct	psia
7	Total pressure at HPC outlet	psia
8	Physical fan speed	rpm
9	Physical core speed	rpm
10	Engine pressure ratio	–
11	Static pressure at HPC outlet	psia
12	Ratio fuel flow to Ps30	pps/ps
13	Corrected fan speed	rpm
14	Corrected core speed	rpm
15	Bypass ratio	–
16	Burner fuel-air ratio	–
17	Bleed enthalpy	–
18	Demanded fan speed	rpm
19	Demanded corrected fan speed	rpm
20	High-pressure turbine (HPT) coolant bleed	lbm/s
21	Low-pressure turbine (LPT) coolant bleed	lbm/s

sensors. All sensor names can be found in Table 2. Secondly, de-noising the sensor trajectories, where each sensor was subjected to a Savitzky-Golay smoothing and differentiation filter [120] using a 3rd degree polynomial. Third, all sensor inputs were normalized to a scale of [−1,1], to ensure that each feature is interpreted the same by the model.

The final step of the pre-processing process was inspired by Caceres et al. [67], who also used the C-MAPSS data set for RUL prediction. They propose a sliding window approach as shown in Fig. 8, where the ground truth RUL is calculated by counting the amount of cycles remaining until the end of the data set per engine. A window size of 30 cycles was used, ensuring each RUL prediction was based on not only the current cycle, but the 30 cycles before. By using a sliding window, we increased the amount of training data and ensured that all inputs were of equal size. For the ground truth RUL, we also incorporated a piece-wise linear correction used by Benker et al. [121]. This limits the maximum ground truth RUL to 120 cycles, which attempts to prevent the model from trying to find fault modes in the healthy regime of the engine lifetime, but rather focused on finding degradation patterns more towards the EOL region. In previous work, Libera [122] applied a similar approach to these pre-processing steps which we followed.

Performing all steps for 100 engines with varying lifetimes results in 17731 samples of size (30, 14), one of which can be seen in Fig. 9.

5. Results & discussion

We present and discuss the results in this section. We first analyze the performance of the B-LSTM developed for RUL prediction (Section 5.1). Secondly, in Section 5.2 we discuss the findings and explanations obtained from the B-LSTM model with DiCE (RUL CFEs). Finally, in Section 5.3, we analyze the results of our data augmentation methods (described in Section 3.3).

5.1. Predictive performance of B-LSTM

In this subsection, we evaluate the performance of the B-LSTM model compared to the standard LSTM across multiple dimensions of accuracy. Specifically, we evaluate how well each model predicts Remaining Useful Life (RUL) and examine several key performance metrics, including RMSE, the $\alpha - \lambda$ accuracy, and the scoring function. We also evaluate the counterfactuals.

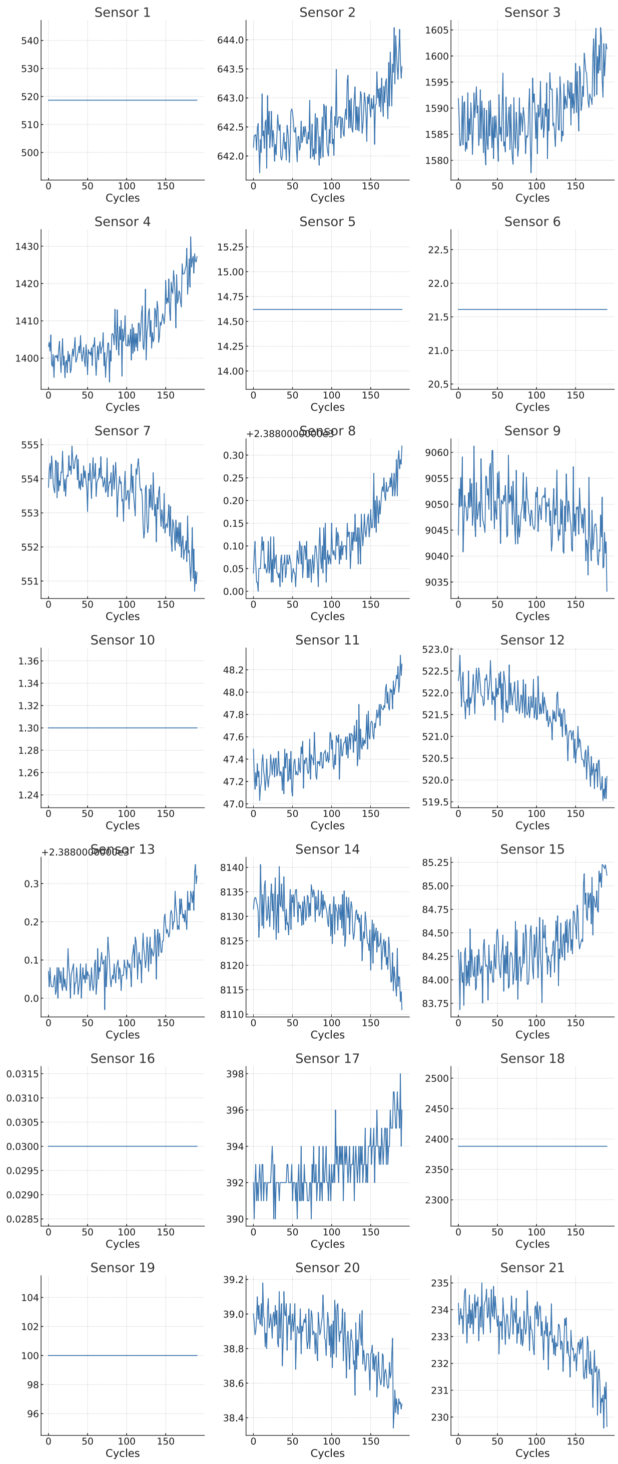


Fig. 7. Raw sensor data of engine 1.

From our comparison between LSTM and B-LSTM, we observed that the predictions were both accurate even though the B-LSTM had more favorable results (see Fig. 10). In the figure, we can see the distinct output of the two LSTM variants. The predicted RUL values closely track the true RUL values, remaining largely within the $\alpha - \lambda$ bounds, as indicated by the blue and red lines and shaded gray area.

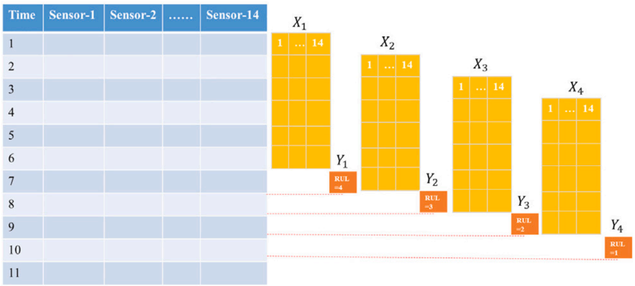


Fig. 8. Sliding window representation of data set [67].

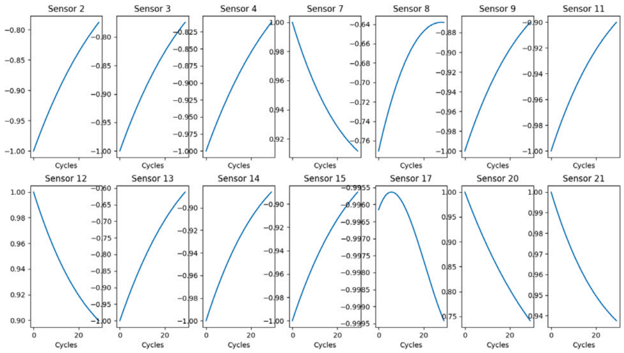


Fig. 9. Processed sensor data for a certain engine (1).

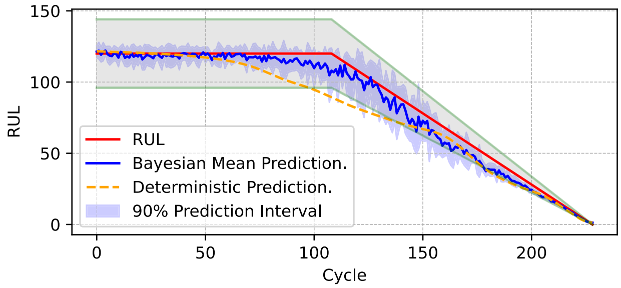


Fig. 10. RUL Predictions. B-LSTM RMSE: 5.81 cycles, LSTM RMSE: 13.51 cycles.

Table 3
Performance of B-LSTM.

Metric	Denoised	Noisy
RMSE	10.95	13.67
STD	7.40	4.70
Total score	3662.34	6919.16
Predictions in $\alpha = 0.2$	66%	63%

The overall performance of the B-LSTM with noisy and with denoised data can be seen in Table 3. In general, the model trained on the denoised data performs better than the one trained on the noisy data.

5.2. Interpretability analysis (RUL CFEs)

As described in Section 3.3, we run the DiCE model to create the RUL CFEs. We were interested in what inputs could generate an increased RUL, and also which inputs could generate a decreased RUL. By increased/decreased RUL, we mean an increase or decrease of 10 flight cycles. Ten cycles represent approximately 10% of the total lifecycle in C-MAPSS, which we consider an acceptable value for our experiments.

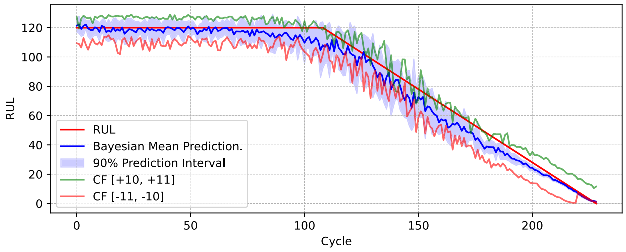


Fig. 11. RUL prediction of engine 4 including CF output for −10 RUL cycles (red) and for +10 RUL cycles (green).

One of the visualizations that we propose to examine interpretability in maintenance is the plot presented in Fig. 11. In the picture, we show the increasing RUL CFEs in green and the decreasing RUL CFEs in red. As expected, the increasing CFEs result (with some oscillations) in higher RUL predictions and the decreasing CFEs in lower RUL predictions. This visualization demonstrates how different input scenarios – represented by the counterfactual explanations – can push the model's predictions in either direction. This interpretive output is particularly useful for tuning maintenance actions based on anticipated RUL outcomes.

In the visualization, we can observe that as the system approaches the end of its life, the uncertainty associated with the CFEs decreases significantly. This reduction in uncertainty suggests that the model becomes more confident in its explanations during this critical phase. Consequently, it becomes easier to trust that any corrective maintenance action proposed based on these CFEs will yield the expected results. This behavior is intuitive since, near the end of life, the degradation patterns are more apparent and predictable, allowing the model to generate more reliable predictions and reduce variability in the outcomes.

Another visualization tool that we propose is shown in Fig. 12, where we present the counterfactuals from the perspective of input features. In this plot, we illustrate how varying specific input parameters impacts the model's predictions, offering a clear view of which features lead to an increase or decrease in predicted RUL. The central lines depict the denoised sensor inputs throughout the life cycle of an engine. Although the example engine is used for illustration purposes, the overall trends remain consistent across the entire testing dataset. The green lines represent modifications to the original inputs aimed at increasing the RUL at that specific time point, while the red lines indicate alterations intended to decrease the RUL.

The overlap of CFE points in Fig. 12 is because each sliding window input is adjusted to its respective counterfactual input, and neighboring windows exhibit significant overlap. The shown patterns enable us to assess whether the different CFE explanations for the same time point align. The concentration of green points provides strong evidence that the system is in a condition where corrective actions—such as maintenance or operational adjustments – would have a positive effect on extending the RUL. This type of insight is of value for maintenance planning. Additionally, this approach facilitates the identification of outliers and trends, as subsets of extreme or conflicting counterfactual explanations become more discernible.

To produce “green counterfactuals” (alternative positive scenarios), we can change the input data in different ways, such as by doing light maintenance. For example, a water wash on the engine can decrease the temperature of the engine and prolong its life. Also, the sensor data can also be contaminated by noise and inaccuracies. Applying a data denoising can also result in an adjustment of certain sensors, resulting in a more realistic set of trajectories and a better estimated RUL.

In the third visualization tool we propose, the focus is on evaluating the uncertainty of the generated CFEs. The tool shown in Fig. 13(b) enables us to observe how confident the model is in its suggested CFEs

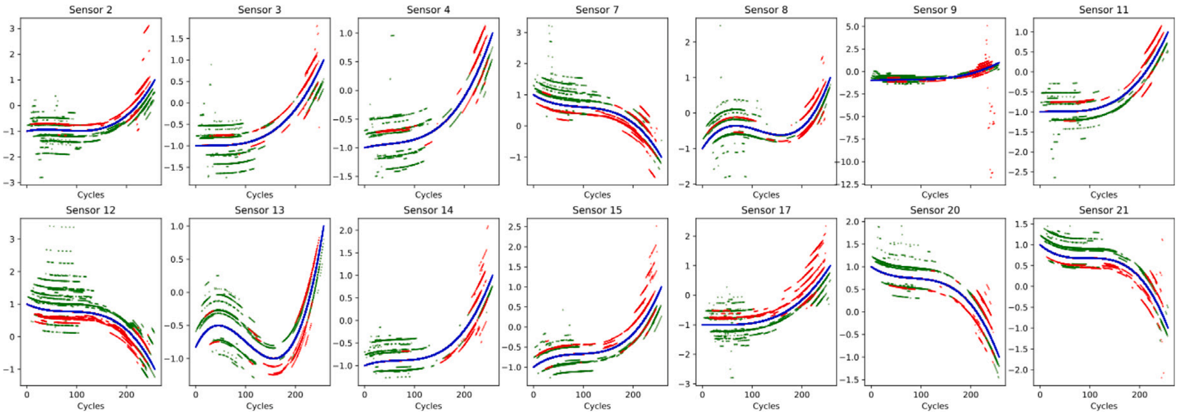
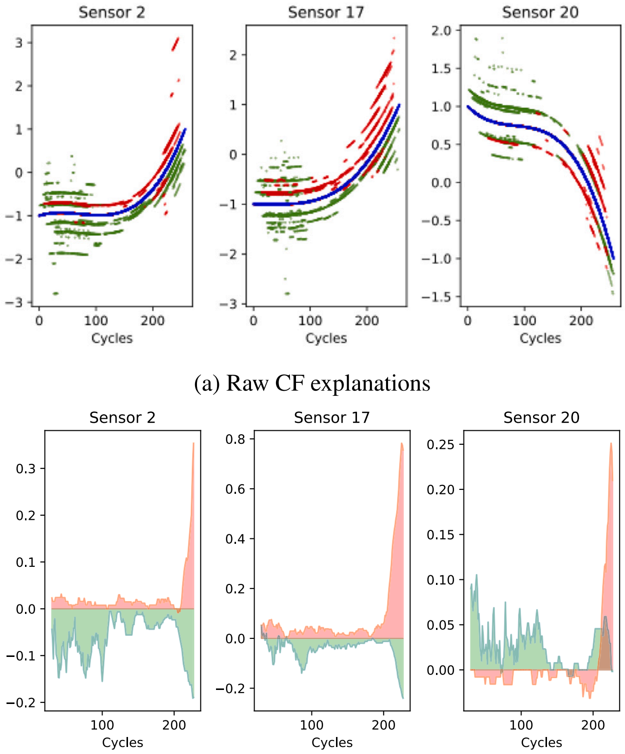


Fig. 12. CF input of engine 4. Center line = original input, red = 10 cycles lower RUL, green = 10 cycles higher RUL.

Table 4
Comparison of the Baseline B-LSTM vs Augmented by Counterfactuals B-LSTM.

	Baseline	Augmented uncertainty CF	Augmented RUL CF (increasing)	Augmented RUL CF (decreasing)	Augmented RUL CF (combined)
RMSE	9.56	**8.47**	9.60	9.01	10.26
RMSE Std	8.68	7.38	7.78	7.42	**7.19**
Accuracy α=0.2	74%	**79%**	70%	78%	68%
Score	2802.69	**2312.69**	2815.19	2391.36	3148.24



(a) Raw CF explanations

Fig. 13. Counterfactual explanations (CFEs) for sensor 2, 17 and 20 of engine 4. Center line = original input, red = 10 cycles lower RUL, green = 10 cycles higher RUL.

by visualizing the degree of uncertainty associated with each explanation. Larger uncertainty bands indicate that the model is less certain about the impact of the input change on the predicted Remaining Useful Life (RUL), while narrower bands suggest a higher confidence in the prediction.

In our results, the diversity of the generated counterfactual explanations (CFEs) was significant with a result of 0.43 (between 0 and 1) in line with the charts of Fig. 13(b). This diversity level can be attributed to the approach used to generate the counterfactuals. Specifically, the optimization process focused on finding counterfactuals that closely resemble the original instance while still being moderatively varied. This strategy ensures that counterfactuals remain realistic and feasible, while exploring the different regions of the feature space.

5.3. Data augmentation by counterfactuals

In this subsection, we present the findings of our data augmentation experiments, which utilized various types of CFEs (RUL CFEs and Uncertainty CFEs). To this aim, we compared the results of the several models explained in Section 3.3. The results are shown in Table 4 and Figs. 14 and 15. The model augmented with Uncertainty CFEs can be seen as the optimal choice, outperforming all other models across three out of four metrics. We attribute this performance to the decreased uncertainty and hence the superior quality of the Uncertainty CFEs.

Surprisingly, the model augmented with all RUL CFEs (both decreasing and increasing) did not exceed the performance of its counterparts, except for RMSE Standard deviation. In this last dimension, the model is the best (RMSE std = 7.19 cycles) but the RMSE value itself is significantly high (RMSE = 10.26 cycles). This result suggests that effective data augmentation using counterfactuals may require a more targeted approach, possibly focusing on similar types of counterfactuals to enhance overall model performance.

5.4. Counterfactuals and causality

It is important to note that distinguishing between correlation and causation in counterfactual explanations remains a challenge, especially when the features in question are correlated with, but not causal for, the RUL. Several general approaches to address this problem have been proposed over the years. A work of note is by Xu and Dang [123] who propose a data-driven framework to discover and represent causal relationships between quality issues and production factors using a causal knowledge graph.

This approach, as well as other methodologies in graphical causal modeling [124], have the limitation of depending on the availability of causal knowledge. A topic to explore in future work is instrumental

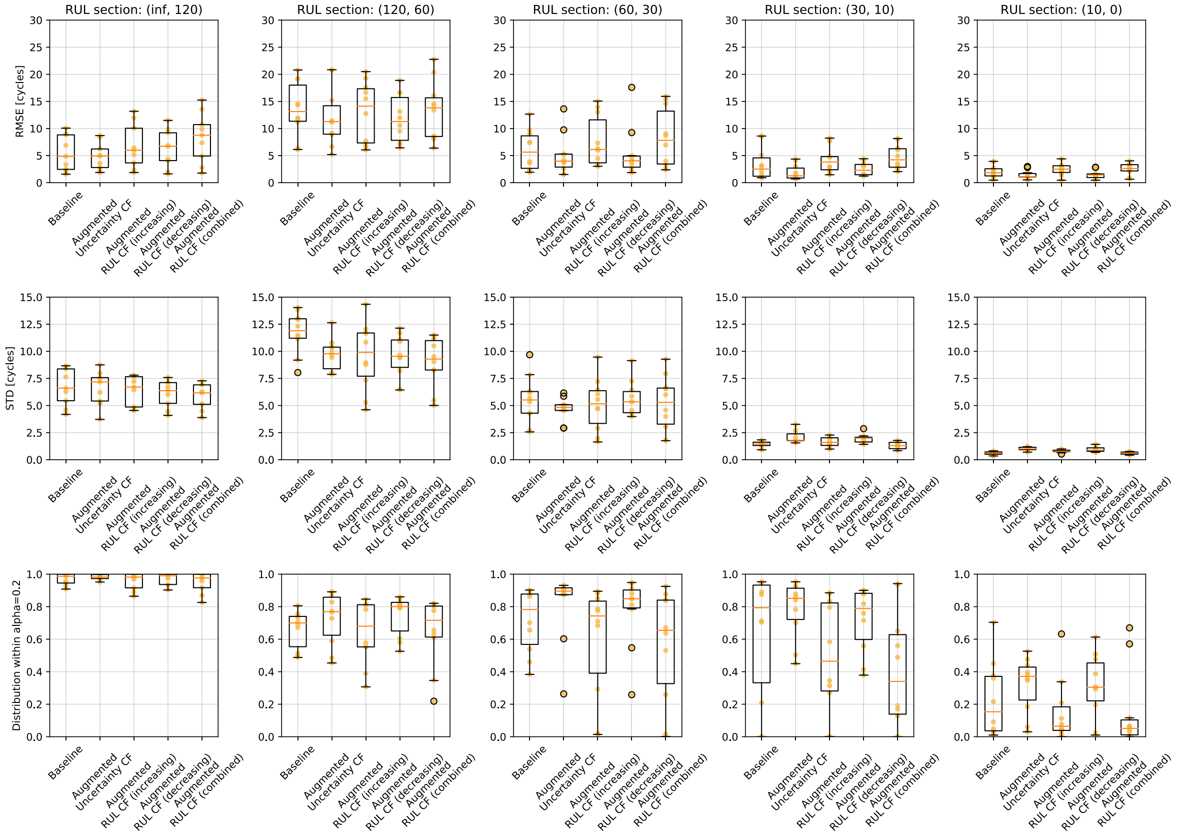


Fig. 14. Overall performance of the 5 trained models per section of the life cycle.

variable methods for causal inference [125]. These methods are beneficial when causal knowledge is incomplete as they rely on structural assumptions or external sources of variation rather than expert causal understanding.

In real-world applications, domain expertise will continue to play a critical role. For example, in the case of high temperature due to poor lubrication, a domain expert would be able to point out that temperature is merely a symptom and the underlying cause (lubrication) should be adjusted. Incorporating domain knowledge into the model or validating counterfactuals through domain expert feedback is going to be important to avoid misleading suggestions.

6. Conclusions & recommendations

The goal of this research is to find and apply methods that combine Counterfactual Explanations (CFEs) with Bayesian uncertainty to improve the interpretability and performance of RUL estimation models. To achieve this, we set out the answer the following research question:

How can Bayesian uncertainty and Counterfactual Explanations be used in predictive maintenance to improve interpretability and predictive performance?

This study involved the development and implementation of a Bayesian LSTM model (B-LSTM). This model was shown capable of accurately predicting the Remaining Useful Life (RUL) throughout the lifespan of a series of simulated synthetic engines, while also providing a measure of uncertainty.

The use of a Bayesian model enabled the generation of various types of counterfactual explanations, namely RUL CFEs (increasing/decreasing) and uncertainty CFEs. The generation of uncertainty CFEs would be unattainable through a deterministic model lacking stochastic modeling capabilities.

The CFEs, derived from manipulated sensor inputs, showed trends and gave practical guidance for interpreting engine lifespan. The approach provided valuable information on the correlation between sensor data and engine health, laying the foundations for maintenance strategies and further exploration in predictive maintenance modeling. RUL CFEs can elucidate how much a given maintenance repair can affect the RUL, ahead of time. For instance, if a counterfactual tells us that raising the HPC temperature by x leads to a new RUL of RUL_x, and our programmed maintenance repair causes an increase of x, it becomes clear how many cycles remain until failure and the effectiveness of the maintenance repair.

The second part of this research attempted to use CFEs as a data augmentation method to generate more data points for model training. Among the CFEs used we used the Bayesian uncertainty of the B-LSTM to generate CFEs with a reduced measure of uncertainty. These uncertainty CFE inputs, along with the RUL CFE inputs, were added to the training data. Analyzing the performance of five models differently augmented we could observe that the addition of CFE with reduced uncertainty improved the overall model performance. This finding confirms that this CFE data augmentation method is a viable approach to model performance enhancement.

In conclusion, our study contributes to the ongoing debate regarding the trade-off between accuracy and explainability in AI models. While traditional perspectives often suggest a compromise between these two aspects, our findings demonstrate that it is possible to enhance both interpretability and model performance simultaneously. By generating counterfactual explanations and augmenting the dataset, we have shown that models can not only achieve higher accuracy but also provide insights into decision processes.

For applying this method in future research, it is advised to implement a CF generation model tailored specifically for reducing uncertainty, such as the Counterfactual Latent Uncertainty Explanations (CLUE) model [126], in order to get a more consistent uncertainty

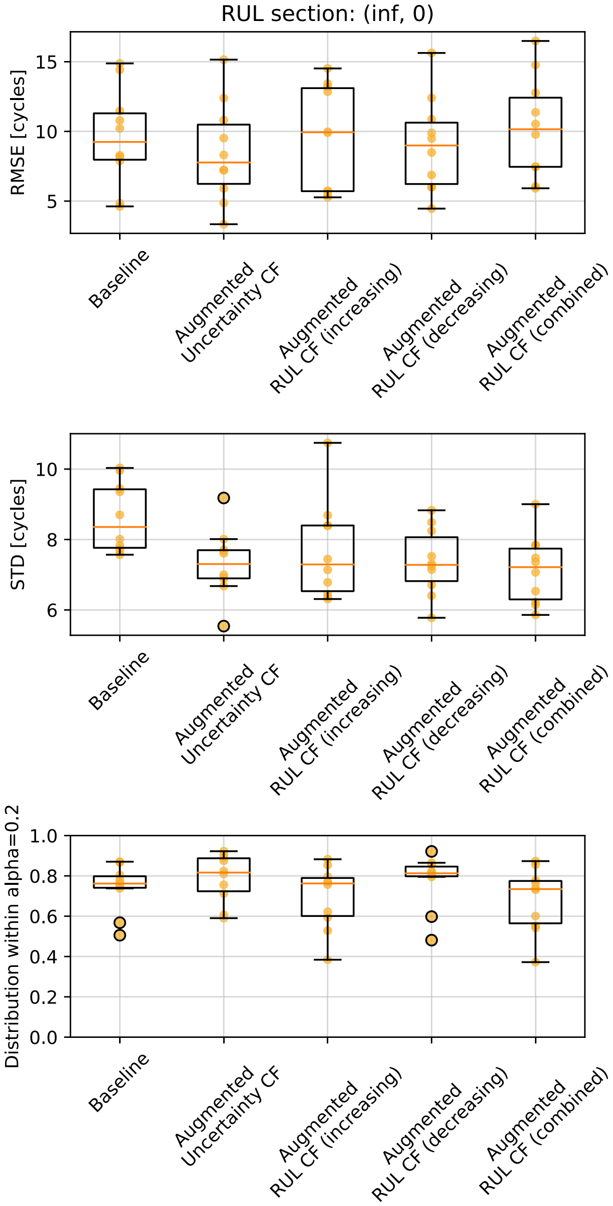


Fig. 15. Performance of 5 trained models: RMSE (top), STD (middle), $\alpha - \lambda$ score (bottom).

reduction over the engine life cycle. Also, we recommend applying this method to a more complex dataset, as all the tested models evaluated significantly well due to the relatively simple dataset. We also recommend looking into what is the best fraction of CFE inputs to real inputs in order to find the optimal amount of augmented CF data to add to the training set in order to maximize performance.

CRediT authorship contribution statement

Jilles Andringa: Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Marcia L. Baptista:** Writing – review & editing, Writing – original draft, Supervision, Methodology. **Bruno F. Santos:** Supervision.

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.

Acknowledgments

This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UIDB/04152/2020 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS) (https://doi.org/10.54499/UIDB/04152/2020).

Appendix. Acronyms

AI	Artificial Intelligence
ARIMA	Auto-Regressive Integrated Moving Average
B-LSTM	Bayesian Long Short-Term Memory
BNN	Bayesian Neural Network
BNNs	Bayesian Neural Networks
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CFEs	Counterfactual Explanations
CLUE	Counterfactual Latent Uncertainty Explanations
CMAPSS	Dataset of Turbofan Data
DiCE	Diverse Counterfactual Explanations
DNNs	Deep Neural Networks
DRL	Deep Reinforcement Learning
EOL	End of Life
GM	Gray Model
HPC	high-pressure compressor
HPT	High-pressure turbine
KL	Kullback–Leibler
LIME	Local Interpretable Model-Agnostic Explanations
LPC	Low-Pressure Compressor
LPT	Low-Pressure Turbine
LR	Learning Rate
LSTM	Long Short Term Memory
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
NASA	National Aeronautics and Space Administration
NDS	Normalized Diversity Score
PHM	Prognostics and Health Management
PrcM	Prescriptive Maintenance
PrdM	Predictive Maintenance
PrvM	Preventive Maintenance
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network

RUL Remaining Useful Life

SHAP Shapley Additive Explanations

STD Standard Deviation

VI Variational Inference

WP Wiener Process

XAI Explainable Artificial Intelligence

Data availability

The CMAPSS dataset is a public dataset made available by NASA.

References

[1] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, et al., Explainable AI (XAI): Core ideas, techniques, and solutions, ACM Comput. Surv. 55 (9) (2023) 1–33.

[2] R.K. Mobley, An Introduction to Predictive Maintenance, Elsevier, 2002.

[3] A.D. Cho, R.A. Carrasco, G.A. Ruz, Improving prescriptive maintenance by incorporating post-prognostic information through chance constraints, IEEE Access 10 (2022) 55924–55932.

[4] M.A.M. Esa, M. Muhammad, Adoption of prescriptive analytics for naval vessels risk-based maintenance: A conceptual framework, Ocean Eng. 278 (2023) 114409.

[5] J. Gama, S. Nowaczyk, S. Pashami, R.P. Ribeiro, G.J. Nalepa, B. Veloso, XAI for predictive maintenance, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5798–5799.

[6] M.L. Baptista, K. Goebel, E.M. Henriques, Relation between prognostics predictor evaluation metrics and local interpretability SHAP values, Artificial Intelligence 306 (2022) 103667.

[7] A. Singhal, P. Pratap, K.K. Dixit, K. Kathuria, Advancements in explainable AI: Bridging the gap between model complexity and interpretability, in: 2024 2nd International Conference on Disruptive Technologies, ICDT, IEEE, 2024, pp. 675–680.

[8] T. Zonta, C.A. Da Costa, R. da Rosa Righi, M.J. de Lima, E.S. da Trindade, G.P. Li, Predictive maintenance in the Industry 4.0: A systematic literature review, Comput. Ind. Eng. 150 (2020) 106889.

[9] E. Zio, Some challenges and opportunities in reliability engineering, IEEE Trans. Reliab. 65 (4) (2016) 1769–1782.

[10] S. Kumar, S. Sarraf, A.K. Kar, P.V. Ilavarasan, A study of explainable artificial intelligence: A systematic literature review of the applications, in: IoT, Big Data and AI for Improving Quality of Everyday Life: Present and Future Challenges: IOT, Data Science and Artificial Intelligence Technologies, Springer, 2023, pp. 243–259.

[11] Q. Zhou, F. Liao, C. Mou, P. Wang, Measuring interpretability for different types of machine learning models, in: Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2018 Workshops, BDASC, BDM, ML4Cyber, PAISI, DaMEMO, Melbourne, VIC, Australia, June 3, 2018, Revised Selected Papers 22, Springer, 2018, pp. 295–308.

[12] D. Lewis, Counterfactuals and comparative possibility, in: IFS: Conditionals, Belief, Decision, Chance and Time, Springer, 1973, pp. 57–85.

[13] J. van der Waa, E. Nieuwburg, A. Cremers, M. Neerincx, Evaluating XAI: A comparison of rule-based and example-based explanations, Artificial Intelligence 291 (2021) 103404.

[14] M.T. Keane, B. Smyth, Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI), in: Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28, Springer, 2020, pp. 163–178.

[15] D. Ley, U. Bhatt, A. Weller, Diverse, global and amortised counterfactual explanations for uncertainty estimates, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 7, 2022, pp. 7390–7398.

[16] M.Z. Espinosa, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, Z. Shams, F. Precioso, S. Melacci, A. Weller, et al., Concept embedding models: Beyond the accuracy-explainability trade-off, Adv. Neural Inf. Process. Syst. 35 (2022) 21400–21413.

[17] C. Mathys, J. Daunizeau, K.J. Friston, K.E. Stephan, A Bayesian foundation for individual learning under uncertainty, Front. Hum. Neurosci. 5 (2011) 39.

[18] R. Malhotra, H. Kaur, Reliability of a manufacturing plant with scheduled maintenance, inspection, and varied production, in: Manufacturing Engineering and Materials Science, CRC Press, 2024, pp. 254–264.

[19] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, J. Barbosa, Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges, Comput. Ind. 123 (2020) 103298.

[20] P. Nunes, J. Santos, E. Rocha, Challenges in predictive maintenance–A review, CIRP J. Manuf. Sci. Technol. 40 (2023) 53–67.

[21] H.M. Elattar, H.K. Elminir, A. Riad, Prognostics: a literature review, Complex Intell. Syst. 2 (2) (2016) 125–154.

[22] V.D. Nguyen, M. Kefalas, K. Yang, A. Apostolidis, M. Olhofer, S. Limmer, T. Bäck, A review: Prognostics and health management in automotive and aerospace, Int. J. Progn. Heal. Manag. 10 (2) (2019).

[23] M.A. Chao, C. Kulkarni, K. Goebel, O. Fink, Fusing physics-based and deep learning models for prognostics, Reliab. Eng. Syst. Saf. 217 (2022) 107961.

[24] H.M. Elattar, H.K. Elminir, A.M. Riad, A.M. Riad, Prognostics: a literature review, Complex & Intell. Syst. 2 (2) (2016) 125–154.

[25] Z. Zhao, B. Liang, X. Wang, W. Lu, Remaining useful life prediction of aircraft engine based on degradation pattern learning, Reliab. Eng. Syst. Saf. 164 (2017) 74–83.

[26] C.M. García, T. Escobet, J. Quevedo, PHM techniques for condition-based maintenance based on hybrid system model representation, in: Annual Conference of the PHM Society, Vol. 2, No. 1, 2010.

[27] M.Y.L. Chew, K. Yan, Enhancing interpretability of data-driven fault detection and diagnosis methodology with maintainability rules in smart building management, J. Sensors 2022 (2022) 1–48.

[28] S. Ansari, A. Ayob, M.S. Hossain Lipu, A. Hussain, M.H.M. Saad, Remaining useful life prediction for lithium-ion battery storage system: A comprehensive review of methods, key factors, issues and future outlook, Energy Rep. 8 (2022) 12153–12185.

[29] X. Li, L. Zhang, Z. Wang, P. Dong, Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and elman neural networks, J. Energy Storage 21 (2019) 510–518.

[30] L. Chen, L. Xu, Y. Zhou, Novel approach for lithium-ion battery on-line remaining useful life prediction based on permutation entropy, Energies 11 (4) (2018) 820.

[31] Y. Zhou, M. Huang, Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and ARIMA model, Microelectron. Reliab. 65 (2016) 265–273.

[32] Z. Zhou, Y. Huang, Y. Lu, Z. Shi, L. Zhu, J. Wu, H. Li, Lithium-ion battery remaining useful life prediction under grey theory framework, in: 2014 Prognostics and System Health Management Conference, PHM-2014 Hunan, IEEE, 2014, pp. 297–300.

[33] W. Gu, Z. Sun, X. Wei, H. Dai, A new method of accelerated life testing based on the Grey System Theory for a model-based lithium-ion battery life evaluation system, J. Power Sources 267 (2014) 366–379.

[34] D. Zhou, L. Xue, Y. Song, J. Chen, On-line remaining useful life prediction of lithium-ion batteries based on the optimized gray model GM (1, 1), Batteries 3 (3) (2017) 21.

[35] 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.

[36] S. Tang, C. Yu, X. Wang, X. Guo, X. Si, Remaining useful life prediction of lithium-ion batteries based on the wiener process with measurement error, Energies 7 (2) (2014) 520–547.

[37] J. Feng, P. Kvam, Y. Tang, Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: A case study on lithium-ion batteries used in electric vehicles, Eng. Fail. Anal. 70 (2016) 323–342.

[38] C. Xu, T. Cleary, D. Wang, G. Li, C. Rahn, D. Wang, R. Rajamani, H.K. Fathy, Online state estimation for a physics-based lithium-sulfur battery model, J. Power Sources 489 (2021) 229495.

[39] X. Hu, J. Jiang, D. Cao, B. Egardt, Battery health prognosis for electric vehicles using sample entropy and sparse Bayesian predictive modeling, IEEE Trans. Ind. Electron. 63 (4) (2015) 2645–2656.

[40] S.S. Ng, Y. Xing, K.L. Tsui, A naive Bayes model for robust remaining useful life prediction of lithium-ion battery, Appl. Energy 118 (2014) 114–123.

[41] M. Jafari, L.E. Brown, L. Gauchia, A Bayesian framework for EV battery capacity fade modeling, in: 2018 IEEE Transportation Electrification Conference and Expo, ITEC, IEEE, 2018, pp. 304–308.

[42] M.A. Galal, W.M. Hussein, E. El-din abdel Kawy, M.M. Sayed, Satellite battery fault detection using Naive Bayesian classifier, in: 2019 IEEE Aerospace Conference, IEEE, 2019, pp. 1–11.

[43] H. Wang, J. Li, F. Yang, Overview of support vector machine analysis and algorithm, Appl. Res. Comput. 31 (5) (2014) 1281–1286.

[44] M.A. Patil, P. Tagade, K.S. Hariharan, S.M. Kolake, T. Song, T. Yeo, S. Doo, A novel multistage support vector machine based approach for Li ion battery remaining useful life estimation, Appl. Energy 159 (2015) 285–297.

[45] D. Wang, Q. Miao, M. Pecht, Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model, J. Power Sources 239 (2013) 253–264.

[46] D. Liu, J. Zhou, D. Pan, Y. Peng, X. Peng, Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning, Measurement 63 (2015) 143–151.

[47] L. Li, P. Wang, K.-H. Chao, Y. Zhou, Y. Xie, Remaining useful life prediction for lithium-ion batteries based on Gaussian processes mixture, PLoS One 11 (9) (2016) e0163004.

[48] J. Liu, Z. Chen, Remaining useful life prediction of lithium-ion batteries based on health indicator and Gaussian process regression model, Ieee Access 7 (2019) 39474–39484.

[49] X. Li, C. Yuan, Z. Wang, Multi-time-scale framework for prognostic health condition of lithium battery using modified Gaussian process regression and nonlinear regression, J. Power Sources 467 (2020) 228358.

[50] K. Park, Y. Choi, W.J. Choi, H.-Y. Ryu, H. Kim, LSTM-based battery remaining useful life prediction with multi-channel charging profiles, Ieee Access 8 (2020) 20786–20798.

[51] Y. Choi, S. Ryu, K. Park, H. Kim, Machine learning-based lithium-ion battery capacity estimation exploiting multi-channel charging profiles, Ieee Access 7 (2019) 75143–75152.

[52] B. Chinomona, C. Chung, L.-K. Chang, W.-C. Su, M.-C. Tsai, Long short-term memory approach to estimate battery remaining useful life using partial data, Ieee Access 8 (2020) 165419–165431.

[53] Y. Liu, G. Zhao, X. Peng, Deep learning prognostics for lithium-ion battery based on ensembled long short-term memory networks, IEEE Access 7 (2019) 155130–155142.

[54] R. Rouhi Ardeshiri, C. Ma, Multivariate gated recurrent unit for battery remaining useful life prediction: A deep learning approach, Int. J. Energy Res. 45 (11) (2021) 16633–16648.

[55] M. Wei, H. Gu, M. Ye, Q. Wang, X. Xu, C. Wu, Remaining useful life prediction of lithium-ion batteries based on Monte Carlo dropout and gated recurrent unit, Energy Rep. 7 (2021) 2862–2871.

[56] B. Zraibi, C. Okar, H. Chaoui, M. Mansouri, Remaining useful life assessment for lithium-ion batteries using CNN-LSTM-DNN hybrid method, IEEE Trans. Veh. Technol. 70 (5) (2021) 4252–4261.

[57] J. Hong, D. Lee, E.-R. Jeong, Y. Yi, Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning, Appl. Energy 278 (2020) 115646.

[58] D. Zhou, Z. Li, J. Zhu, H. Zhang, L. Hou, State of health monitoring and remaining useful life prediction of lithium-ion batteries based on temporal convolutional network, IEEE Access 8 (2020) 53307–53320.

[59] R. Jiao, K. Peng, J. Dong, Remaining useful life prediction of lithium-ion batteries based on conditional variational autoencoders-particle filter, IEEE Trans. Instrum. Meas. 69 (11) (2020) 8831–8843.

[60] L. Ren, L. Zhao, S. Hong, S. Zhao, H. Wang, L. Zhang, Remaining Useful Life Prediction for Lithium-Ion Battery: A Deep Learning Approach, IEEE Access 6 (2018) 50587–50598.

[61] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[62] G.I. Webb, E. Keogh, R. Miikkulainen, Naive Bayes, Encycl. Mach. Learn. 15 (2010) 713–714.

[63] T. Minka, Bayesian Linear Regression, Technical Report, Citeseer, 2000.

[64] J. Pearl, Bayesian networks, 2011.

[65] J. Wang, An intuitive tutorial to Gaussian processes regression, 2020, arXiv preprint arXiv:2009.10862.

[66] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (Jun) (2001) 211–244.

[67] 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 Health Monit. 28 (10) (2021).

[68] S. Choubey, R.G. Benton, T. Johnsten, A holistic end-to-end prescriptive maintenance framework, Data- Enabled Discov. Appl. 4 (2020) 1–20.

[69] A. Ucar, M. Karakose, N. Kırımça, Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends, Appl. Sci. 14 (2) (2024) 898.

[70] S. Vollert, M. Atzmueller, A. Theissler, Interpretable machine learning: A brief survey from the predictive maintenance perspective, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, IEEE, 2021, pp. 01–08.

[71] T. Speith, A review of taxonomies of explainable artificial intelligence (XAI) methods, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 2239–2250.

[72] P. Umiliacchi, D. Lane, F. Romano, A. SpA, Predictive maintenance of railway subsystems using an ontology based modelling approach, in: Proceedings of 9th World Conference on Railway Research, May, Citeseer, 2011, pp. 22–26.

[73] Q. Cao, A. Samet, C. Zanni-Merk, F.D.B. De Beuvron, C. Reich, An ontology-based approach for failure classification in predictive maintenance using fuzzy C-means and SWRL rules, Procedia Comput. Sci. 159 (2019) 630–639.

[74] A. Canito, J. Corchado, G. Marreiros, A systematic review on time-constrained ontology evolution in predictive maintenance, Artif. Intell. Rev. 55 (4) (2022) 3183–3211.

[75] S. Kaparthi, D. Bumblauskas, Designing predictive maintenance systems using decision tree-based machine learning techniques, Int. J. Qual. Reliab. Manag. 37 (4) (2020) 659–686.

[76] G. Dorgo, A. Palazoglu, J. Abonyi, Decision trees for informative process alarm definition and alarm-based fault classification, Process. Saf. Environ. Prot. 149 (2021) 312–324.

[77] N. Li, Y. Lei, J. Lin, S.X. Ding, An improved exponential model for predicting remaining useful life of rolling element bearings, IEEE Trans. Ind. Electron. 62 (12) (2015) 7762–7773.

[78] X.-S. Si, W. Wang, C.-H. Hu, M.-Y. Chen, D.-H. Zhou, A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation, Mech. Syst. Signal Process. 35 (1–2) (2013) 219–237.

[79] A. Simões, J.M. Viegas, J.T. Farinha, I. Fonseca, The state of the art of hidden markov models for predictive maintenance of diesel engines, Qual. Reliab. Eng. Int. 33 (8) (2017) 2765–2779.

[80] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[81] M. Kozielski, Contextual explanations for decision support in predictive maintenance, Appl. Sci. 13 (18) (2023) 10068.

[82] F. Giobergia, E. Baralis, M. Camuglia, T. Cerquitelli, M. Mellia, A. Neri, D. Tricarico, A. Tuninetti, Mining sensor data for predictive maintenance in the automotive industry, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2018, pp. 351–360.

[83] A. Lekidis, A. Georgakis, C. Dalamagkas, E.I. Papageorgiou, Predictive maintenance framework for fault detection in remote terminal units, Forecasting 6 (2) (2024) 239–265.

[84] 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.

[85] J.S. Dhaliwal, I. Benbasat, The use and effects of knowledge-based system explanations: theoretical foundations and a framework for empirical evaluation, Inf. Syst. Res. 7 (3) (1996) 342–362.

[86] M. Atzmueller, N. Hayat, A. Schmidt, B. Kloepper, Explanation-aware feature selection using symbolic time series abstraction: approaches and experiences in a petro-chemical production context, in: 2017 IEEE 15th International Conference on Industrial Informatics, INDIN, IEEE, 2017, pp. 799–804.

[87] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 1–42.

[88] O.K. Aimiyekagbon, L. Muth, M. Wohlleben, A. Bender, W. Sextro, Rule-based diagnostics of a production line, in: PHM Society European Conference, Vol. 6, No. 1, 2021, pp. 10–10.

[89] C.W. Hong, C. Lee, K. Lee, M.-S. Ko, D.E. Kim, K. Hur, Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction, Sensors 20 (22) (2020) 6626.

[90] S. Sundar, M.C. Rajagopal, H. Zhao, G. Kuntumalla, Y. Meng, H.C. Chang, C. Shao, P. Ferreira, N. Miljkovic, S. Sinha, et al., Fouling modeling and prediction approach for heat exchangers using deep learning, Int. J. Heat Mass Transfer 159 (2020) 120112.

[91] D.M. Onchis, G.-R. Gillich, Stable and explainable deep learning damage prediction for prismatic cantilever steel beam, Comput. Ind. 125 (2021) 103359.

[92] S.L. Morgan, C. Winship, Counterfactuals and Causal Inference, Cambridge University Press, 2015.

[93] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harv. JL & Tech. 31 (2017) 841.

[94] S.R. Pfohl, T. Duan, D.Y. Ding, N.H. Shah, Counterfactual reasoning for fair clinical risk prediction, in: Machine Learning for Healthcare Conference, PMLR, 2019, pp. 325–358.

[95] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Min. Knowl. Discov. (2022) 1–55.

[96] P. Pileggi, E. Lazovik, R. Snijders, L.-U. Axelsson, S. Drost, G. Martinelli, M. de Grauw, J. Graff, A lesson on operationalizing machine learning for predictive maintenance of gas turbines, in: Turbo Expo: Power for Land, Sea, and Air, vol. 84966, American Society of Mechanical Engineers, 2021, V004T05A006.

[97] J. Jakubowski, P. Stanisz, S. Bobek, G.J. Nalepa, Roll wear prediction in strip cold rolling with physics-informed autoencoder and counterfactual explanations, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2022, pp. 1–10.

[98] J.F. Barraza, E.L. Droguett, M.R. Martins, FS-SCF network: Neural network interpretability based on counterfactual generation and feature selection for fault diagnosis, Expert Syst. Appl. 237 (2024) 121670.

[99] A.K.M. Nor, S.R. Pedapati, M. Muhammad, V. Leiva, Overview of explainable artificial intelligence for prognostic and health management of industrial assets based on preferred reporting items for systematic reviews and meta-analyses, Sensors 21 (23) (2021) 8020.

[100] N. Wiratunga, A. Wijekoon, I. Nkisi-Orji, K. Martin, C. Palihawadana, D. Corsar, Discern: Discovering counterfactual explanations using relevance features from neighbourhoods, in: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2021, pp. 1466–1473.

[101] Z. Chen, F. Silvestri, J. Wang, H. Zhu, H. Ahn, G. Tolomei, Relax: Reinforcement learning agent explainer for arbitrary predictive models, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 252–261.

[102] F. Ezzeddine, O. Ayoub, D. Andreoletti, S. Giordano, SAC-FACT: Soft actor-critic reinforcement learning for counterfactual explanations, in: World Conference on Explainable Artificial Intelligence, Springer, 2023, pp. 195–216.

[103] F. Hamman, E. Noorani, S. Mishra, D. Magazzeni, S. Dutta, Robust counterfactual explanations for neural networks with probabilistic guarantees, in: International Conference on Machine Learning, PMLR, 2023, pp. 12351–12367.

[104] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, FACE: feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 344–350.

[105] E. AlJalaud, M. Hosny, Counterfactual explanation of AI models using an adaptive genetic algorithm with embedded feature weights, IEEE Access (2024).

[106] S. Dandl, C. Molnar, M. Binder, B. Bischl, Multi-objective counterfactual explanations, in: International Conference on Parallel Problem Solving from Nature, Springer, 2020, pp. 448–469.

[107] C.S. Han, K.M. Lee, Gradient-based counterfactual generation for sparse and diverse counterfactual explanations, in: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, 2023, pp. 1240–1247.

[108] J. Del Ser, A. Barredo-Arrieta, N. Díaz-Rodríguez, F. Herrera, A. Saranti, A. Holzinger, On generating trustworthy counterfactual explanations, Inform. Sci. 655 (2024) 119898.

[109] V. Guyomard, F. Fessant, T. Guyet, T. Bouadi, A. Termier, VCNet: A self-explaining model for realistic counterfactual generation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2022, pp. 437–453.

[110] K. Sarathi, S. Mitra, P. Deepak, S. Chakraborti, Counterfactuals as explanations for monotonic classifiers, in: 4th Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems, 2023.

[111] A. Kuratomi, I. Miliou, Z. Lee, T. Lindgren, P. Papapetrou, JUICE: JUstIfied counterfactual explanations, in: International Conference on Discovery Science, Springer, 2022, pp. 493–508.

[112] R.R. Fernández, I.M. De Diego, V. Aceña, A. Fernández-Isabel, J.M. Moguerza, Random forest explainability using counterfactual sets, Inf. Fusion 63 (2020) 196–207.

[113] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 607–617.

[114] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, J. Jorge, Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications, Inf. Fusion 81 (2022) 59–83.

[115] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 607–617.

[116] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Mach. Learn. 20 (1995) 197–243.

[117] K. Goebel, A. Saxena, S. Saha, B. Saha, J. Celaya, Prognostic performance metrics, Mach. Learn. Knowl. Discov. Eng. Syst. Heal. Manag. 147 (2011) 20.

[118] 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, PHM 2008, 2008.

[119] D.K. Frederick, J.A. Decastro, J.S. Litt, User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), ISBN: 3016210134, 2007.

[120] N.B. Gallagher, Savitzky-golay smoothing and differentiation filter, Eig. Res. Inc. (2020).

[121] M. Benker, L. Furtner, T. Semm, M.F. Zaeh, Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo, J. Manuf. Syst. 61 (2021) 799–807.

[122] L.D. Libera, A comparative study between Bayesian and frequentist neural networks for remaining useful life estimation in condition-based maintenance, 2019, arXiv preprint arXiv:1911.06256.

[123] Z. Xu, Y. Dang, Data-driven causal knowledge graph construction for root cause analysis in quality problem solving, Int. J. Prod. Res. 61 (10) (2023) 3227–3245.

[124] D. Cheng, J. Li, L. Liu, J. Liu, T.D. Le, Data-driven causal effect estimation based on graphical causal modelling: A survey, ACM Comput. Surv. 56 (5) (2024) 1–37.

[125] M. Baiocchi, J. Cheng, D.S. Small, Instrumental variable methods for causal inference, Stat. Med. 33 (13) (2014) 2297–2340.

[126] J. Antoran, U. Bhatt, T. Adel, A. Weller, J.M. Hernández-Lobato, Getting a {clue}: A method for explaining uncertainty estimates, in: International Conference on Learning Representations, 2021.