Relation between prognostics predictor evaluation metrics and local interpretability SHAP values

Baptista, Marcia L.; Goebel, Kai; Henriques, Elsa M.P.

DOI
10.1016/j.artint.2022.103667

Publication date
2022

Document Version
Final published version

Published in
Artificial Intelligence

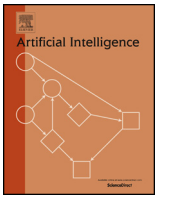
Citation (APA)
Baptista, M. L., Goebel, K., & Henriques, E. M. P. (2022). Relation between prognostics predictor evaluation metrics and local interpretability SHAP values. *Artificial Intelligence*, *306*, Article 103667. https://doi.org/10.1016/j.artint.2022.103667

Artificial Intelligence 306 (2022) 103667

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Relation between prognostics predictor evaluation metrics and local interpretability SHAP values

Marcia L. Baptista [a,*], Kai Goebel [b,c], Elsa M.P. Henriques [d]

[a] Delft University of Technology (TU Delft), Mekelweg 5, 2628 CD Delft, the Netherlands
[b] Luleå University of Technology, 971 87 Luleå, Sweden
[c] Palo Alto Research Center (PARC), Palo Alto CA 94304, USA
[d] University of Lisbon - Instituto Superior Tecnico (IST), Av. Rovisco Pais nº1, 1049-001 Lisbon, Portugal

ARTICLE INFO

Article history:
Received 6 October 2020
Received in revised form 24 December 2021
Accepted 20 January 2022
Available online 15 February 2022

Keywords:
Local interpretability
Model-agnostic interpretability
SHAP values
Monotonicity
Trendability
Prognosability

ABSTRACT

Maintenance decisions in domains such as aeronautics are becoming increasingly dependent on being able to predict the failure of components and systems. When data-driven techniques are used for this prognostic task, they often face headwinds due to their perceived lack of interpretability. To address this issue, this paper examines how features used in a data-driven prognostic approach correlate with established metrics of monotonicity, trendability, and prognosability. In particular, we use the SHAP model (SHapley Additive exPlanations) from the field of eXplainable Artificial Intelligence (XAI) to analyze the outcome of three increasingly complex algorithms: Linear Regression, Multi-Layer Perceptron, and Echo State Network. Our goal is to test the hypothesis that the prognostics metrics correlate with the SHAP model's explanations, i.e., the SHAP values. We use baseline data from a standard data set that contains several hundred run-to-failure trajectories for jet engines. The results indicate that SHAP values track very closely with these metrics with differences observed between the models that support the assertion that model complexity is a significant factor to consider when explainability is a consideration in prognostics.

1. Introduction

Over the last decades, developments in storage and acquisition technologies have permitted access to large volumes of data. The continual growth in computing power, followed by a corresponding decrease in costs, has also come to meet the requirements of more advanced decision-making systems. These systems have started to revolutionize the way we think about data and modeling, but have also brought additional challenges, especially at the interpretability level. Decision systems based on machine learning are well-known for their promising results [79] but also for their complexity and lack of transparency [142,76]. An accuracy-interpretability trade-off [42] is true for almost all machine learning methods. For example, deep learning networks, an advanced form of machine learning, typically combine the activities of several hundred or even thousands of neurons. Despite each neural unit's relative simplicity, the network's structure can be so intricate that it may not be fully understood, even by its designer. Mostly due to this reason, neural network systems tend to be seen as black-boxes, where the user is typically only aware of input-output relationships, but not the underlying reasoning.

* Corresponding author.
E-mail addresses: m.lbaptista@tudelft.nl (M.L. Baptista), kgoebel@parc.com (K. Goebel), elsa.h@ist.utl.pt (E.M.P. Henriques).

https://doi.org/10.1016/j.artint.2022.103667

Machine learning algorithms have already revolutionized fields such as image recognition or natural language processing [103]. However, several obstacles hinder their adoption in other fields. In highly regulated environments, strict requirements on the audit and verifiability of decisions have limited their acceptance. For example, in aerospace, certification by regulatory bodies requires the applicant to demonstrate that the system meets minimal safety criteria. Accountability and trust are essential properties in many applications. As noted by Wilkinson et al. [135], from the Federal Aviation Administration (FAA) and National Aero Space Agency (NASA), "understanding the mechanisms used (...) is essential to understanding the impact on software assurance". The General Data Protection Regulation (GDPR) approved by the European Parliament in 2016 has also imposed restrictions on automated decision-making by establishing the human right to obtain explanations about the logic involved in algorithmic decisions that influence their lives [47,132].

Recognizing the importance of interpretability to accelerate machine learning progress, the Artificial Intelligence (AI) research community has started to pay increasing attention to the explainability topic. Researchers from different backgrounds and experiences have started to produce a significant body of research about explanations and intelligibility. A project of note is the eXplainable AI (XAI) initiative [52] led by the Defense Advanced Research Projects Agency (DARPA) of the United States. The XAI initiative aims at creating machines that can operate in their environments while also providing explanations for their behavior.

Researchers from different fields have researched interpretability, and given the complexity of the subject, there is no agreement on a single definition or taxonomy. As noted by Lipton [84], the concept of interpretability is not a monolithic one, but it reflects several distinct ideas, such as trust or transparency. Given the lack of a "formal technical meaning" [84], it is important to establish a definition for interpretability. Here, we adopt the definition of Biran and Cotton [19], as the level that an observer can understand the cause of a decision. Following the work of Miller [92], and for simplicity, we equate *interpretability* with *explainability*.

As described in the work of Arrieta et al. [8], there are many approaches to interpretability. One such approach is the SHAP model (SHapley Additive exPlanations) [85]. This kind of XAI model uses Shapley values from game theory to characterize the input variables' relative importance. The approach is *model-agnostic* [107] as it only requires knowing the black-box model's output for the neighbor instances of an input sample. When using SHAP, each observed value of a feature gets its SHAP value. The focus is on explaining what the model locally depends on, instead of learning the full mapping. In other words, the goal is to achieve *local interpretability* [43].

The technique of *local interpretability* contrasts with *global interpretability*. Global interpretability consists of all the techniques that are able to explain the structure of a model using a macro perspective. This type of approach is most often used for simpler methods since as the complexity of the models increases it can become gradually more difficult to understand them [93]. Global interpretability methods typically examine the black-box model's input-output relationships to infer an equivalent logical structure that can describe or simulate the black-box model's behavior. In other words, the goal is to build a surrogate model that is more transparent [60]. Local interpretability concerns the provision of independent explanations for individual model responses. Models such as SHAP focus on calculating the importance of the different features for a specific prediction. The goal is to isolate a single instance and build a surrogate model in the neighborhood (locally) of that instance to explain how the model processes it. Because there is typically no explicit concern in maintaining the correlation between the diverse independent local models, this work aims to understand better how the SHAP local models relate to each other.

In this work, we are interested in understanding how SHAP can benefit Remaining Useful Life (RUL) estimation in aeronautics. To this end, we study three increasingly complex prognostics models: Linear Regression (LR), Multi-Layer Perceptron (MLP), and the more recent algorithm of Echo State Network (ESN) [64,37]. The ESN is a recurrent neural network where only the connections to the output are computed, and this is done with regression instead of gradient-based methods, which simplifies and accelerates the training process. These networks have the additional capability of learning multidimensional temporal patterns. As an ESN is fed with input signals, past signals can influence new ones due to the network's feedback loops. This kind of memory enables an ESN to capture the temporal dimension of the data explicitly. There are other architectures with memory, such as Long-Short Term Memory Network (LSTM) [59] or Gated Recurrent Unit (GRU) [29]. The ESN is, however, a simple and efficient alternative that has shown promising results in prognostics [101,95,110,114,109].

This paper discusses the need for XAI in prognostics by providing a comprehensive literature review and investigating the SHAP model according to the classical metrics of PHM (monotonicity, trendability, and prognosability) proposed in [32]. It is advantageous that the trajectories of explanatory values produced by SHAP exhibit these properties. **Monotonic** SHAP values imply that the weight associated with a given feature is changing monotonically over the unit's lifecycle. Monotonicity is desirable as it means that sensor features exhibit either increasing or decreasing importance over time. Having fluctuating SHAP values would most likely mean that the SHAP model is unstable and probably not the most appropriate model to analyze the importance of a prognostics feature over time. **Trendability** is also relevant as trendable SHAP values imply that the SHAP trajectories, or SHAP sequences, of different units, follow the same trend line. In prognostics, the importance of a feature should be consistent across different units. This consistency facilitates the prognostics and the interpretation of results. Prognosable SHAP values imply that the weight associated with a specific feature at the end of life has a slight variation when considering different units. Prognosability is also a desirable characteristic as it means that the explainability at the different end of life points is consistent across units. This work's main contribution is to show that SHAP values exhibit the desirable properties of monotonicity, trendability, and prognosability. A secondary contribution is showing that model complexity influences interpretability.

The remainder of this article is organized as follows. Section 2 reviews related work in the field of Prognostics and Health Management (PHM) and eXplainable Artificial Intelligence (XAI). Section 3 describes the approach. Section 4 focuses on the case study and the methodology. The results of the experiments are presented and discussed in Section 5. Section 6 concludes the article.

2. Background and related work

In this section we describe the field of Prognostics and Health Management (PHM), review work on interpretability techniques, and discuss some contributions to PHM in the interpretability domain.

2.1. Prognostics and health management

Prognostics and Health Management (PHM) is the engineering discipline that studies how to improve the system lifecycle based on current health status and future condition [102]. PHM seeks to prevent unexpected failure based on real-time monitoring technologies, improve control and maintenance operations, and use condition monitoring data to promote better system design. Within PHM, *prognostics* focuses on predicting the future health state and failure modes of the equipment based on condition monitoring, historical trends, and anticipated usage profiles [44]. The field's significance comes from its potential to enable more reliable operations and enhance understanding of aging factors and safety margins. Some benefits of prognostics are less unscheduled maintenance, optimized lifecycle management, and increased availability of engineered systems and infrastructure. These goals are particularly encouraged in aerospace engineering [63], where issues such as performance, reliability, and safety are of concern.

Zio [149] makes a distinction between three approaches to prognostics: first principles model-based, reliability model-based, and data-driven approaches. In the approach of **first principles model-based**, the prediction algorithm bases its estimates on a mathematical model derived from first principles to describe engineering behavior. Mostly due to promising results in the field [25,35,74,94], there is an almost established notion that these methods are superior in performance to the remaining prognostics approaches. This notion premises that it is possible to derive a rigorous model of the degradation process. In practice, defining a complete physical model is not easy and sometimes not even possible. Most complex systems are subject to multiple, nonlinear, stochastic processes of degradation. In such cases, it may only be possible to partially describe the actual physics, with much of the underlying phenomena being represented as a black-box or simplification. Modeling errors can be minimized by optimizing the model parameters given experimental or field data. However, such a design can be faulty if based on inadequate test-benches.

Reliability model-based approaches depend on classical reliability theory (e.g., bathtub curve and product failure behavior) to estimate the time to failure of the equipment. In this approach, failure (or repair) time distributions are described by statistical properties estimated from failure (or repair) records. The Weibull distribution is most often the preferred choice for this analysis. Generally, these models do not include environmental (e.g., weather conditions) or operational parameters (e.g., temperature, pressure, vibration, load). Several authors such as Peng and Huang [100], Rocchetta et al. [111], Alvehag and Soder [5], and Naseri et al. [97] have worked on this limitation using techniques such as accelerated life models or proportional hazard models. Nevertheless, even these advanced methods can be too simple to capture the full range of system's change and its complicated effect on deterioration.

Compared to the previous methods, **data-driven** methods do not rely on explicit domain knowledge. There is no reliance on reliability theory nor the explicit physical representation of aging processes. There is often the misconception that there is no underlying mathematical model in this kind of approach [71, p. 244]. This claim is, however, not entirely valid. Data-driven models also build a mathematical model to describe the observed relationships between input data and target variables. The way these models are built depends on the utilized artificial intelligence technique. For example, and in simple terms, decision trees exploit causality, neural networks optimize function composition, and support vector techniques are kernel-based estimation methods. The target output is an analytical and measurable model that relates input and output variables irrespective of the procedure applied by the data-driven approach. Even though this kind of model does not have easily obtainable physical meaning, data-driven modeling produces abstract yet useful physical phenomena representations. The generalization capabilities these models bring may provide solutions to some significant issues in prognostics. For example, they can help cluster data, address complex numerical problems, and capture nonlinear relationships automatically [112].

In prognostics, and as noted by Zio and Maio [150], there has been some skepticism about data-driven methods. Deep learning, a complex class of data-driven methods, has come to raise even more questions [144]. Fields such as video games, computer vision, or natural language processing have hastened to adopt deep learning and have seen these techniques surpass classical methods' performance beyond most expectations [79]. The same trend has not been observed in prognostics, not in such a manner. Two substantial differences between prognostics and these fields can explain such a discrepancy. The first is a lack of failure data in prognostics, particularly for highly-reliable assets or new equipment. This issue is a critical one [45], but several options exist to address it, from unsupervised anomaly detection [126] to data simulation [55]. The second difference, a major cause of mistrust in deep learning for critical systems [151], relates to the general lack of interpretability of these models.

Due to several factors, prognostics technologies still have a low Technology Readiness Level (TRL), and PHM may still be considered an emerging field [116]. Over the last decades, a great deal of effort has been invested into improving the accurate prediction of the Remaining Useful Life (RUL) of different systems and components. The focus has been on accuracy rather than certification. However, increased concerns about trust, accountability, and auditing in diverse fields such as nuclear energy [11, pp. 151-152], or aerospace [63,135], are bringing increased attention to this area. Recent works [141,87,6,136,69,80] have started to apply methods from eXplainable Artificial Intelligence (XAI) to PHM. Before the latest developments in "explainable" prognostics, several authors [150,38,133,82] used the fuzzy set theory of Zadeh [139] to promote interpretability in PHM. This form of multi-valued logic was particularly successful in diagnostics applications (see the review in [133]), showing that it is possible to capture nonlinear complexity while maintaining some degree of transparency. However, and as Mencar [91] notes, fuzzy logic by itself does not guarantee interpretability. There are several open questions in the field, namely, the difficulty to define exact fuzzy rules, membership functions, and optimize fuzzy systems. As noted by Chimatapu et al. [28], fuzzy systems are often not perceived as an XAI technique. However, it is interesting to observe that the original motivation [10,88] for fuzzy control systems came from artificial intelligence. Regardless of the classification, the contribution of fuzzy techniques to advance the field of XAI and "explainable" prognostics is of notice. In the next subsections, we review other important contributions to the field of interpretability.

2.2. Pre-model interpretability

Some authors, such as Carvalho et al. [23], consider the existence of pre-model, or data, interpretability. This kind of interpretability consists of applying independent techniques to understand the data used to train or build the model. Such approaches only depend on the data itself and are, therefore, model-agnostic. Principal Component Analysis (PCA), Distributed Stochastic Neighbor Embedding (t-SNE), and clustering methods are examples of exploratory data analysis methods [53] that can be classified under pre-model interpretability. These techniques often do not have a high interpretability power but they are considered by some authors [23,8,131] to be part of the XAI field. This follows from these techniques being able to promote a better comprehension of the model and being able to aid experts to understand and gain insights into the prognostics process. They can also work in combination with more advanced techniques to provide a more holistic overview of the model.

In prognostics, pre-model interpretability has been subject to extensive study. For example, PCA is the preferred choice of several authors in prognostics, such as Zhang et al. [146], Benkedjouh et al. [17], Mosallam et al. [96], Lasheras et al. [78], and Yongxiang et al. [138] to reduce data dimensionality. A relevant study is that of Lall and Thomas [77], who compared the utility of PCA and Indendepent Component Analysis (ICA) in capturing the damage evolution of electronic assemblies. The authors reported that ICA could help discriminate between the before and after failure even though it did not clearly indicate damage progression. The PCA helped to distinguish between the healthy and failure stage and the variance of the principal components of the instantaneous frequency of the strain signals allowed to follow failure progression.

Another data exploration technique is t-SNE, a technique proposed by Maaten and Hinton [86], which allows visualizing high-dimensional data in a two or three-dimensional map. The technique is typically used in PHM [27,56] to help separate different failure modes. For example, Chen et al. [27] apply the dimension reduction methods of t-SNE, PCA, and Locality Preserving Projections (LPP) to a PHM dataset related to bearings, with t-SNE achieving the best accuracy of the three methods. The authors trained classifiers on the features derived from the different visualization techniques and from that outcome it was possible to estimate accuracy.

2.3. In-model interpretability

The field of in-model interpretability [23] focuses on intrinsically interpretable models. These "transparent" [84] models naturally, and by design, provide some degree of interpretability. Lipton [84] classifies transparency in three dimensions, namely simulatability, decomposability, and algorithmic transparency. Simulatability relates to the ability to understand the entire model. Lipton [84] notes that simulatability is not a direct consequence of the use of a particular model. For example, and even though models such as linear regression, rule-based systems, and decision trees are typically easier to interpret [8], in some cases, a compact neural network may be more transparent than the former alternatives. Note that even simple methods such as linear regression can become very challenging as the number of predictors increases. In expert systems based on if-then rules, it may not be possible to grasp all the rules and their interactions. Seemingly, decision trees can become too deep or too broad for graphical visualization and comprehension. Lipton [84]'s second notion of transparency, decomposability, defines to which degree the user can understand the model components — input data, parameters, and calculation rules. The third notion of Lipton [84] is algorithmic transparency, which relates to the ability to understand the inferential process. It is important to consider these three notions when designing "transparent" machine learning models. We hereafter review some of the approaches proposed to achieve in-model transparency.

In their work, Fellous et al. [39] identifies four classes of approaches to achieve in-model interpretability: 1) hybrid models, 2) architecturally explainable models, 3) explainable convolutional networks, and 4) models with regularization. Arrieta et al. [8] review **hybridization** in XAI. Hybrid models in XAI combine simple and more interpretable models with more complex models. One modeling trend that is becoming popular is to propose deep formulations of classical machine learning models. A work of mention is Deep k-Nearest Neighbours (DkNN) by Papernot and McDaniel [99], a hybrid classifier

that runs the K-Nearest Neighbors (KNN) algorithm on the data learned by each layer of a DNN. In a DkNN, the neighbor instances can be used as human-interpretable explanations of the prediction.

Another hybrid model is the Deep Weighted Averaging Classifier (DWAC) by Card et al. [22]. The DWAC bases its explanations of predictions on examples or prototypes [70]; presenting the user with the training samples similar to the given input instance. Deep models are also often combined with probabilistic graphical models as in Deep Kalman filters (DKFs) by Krishnan et al. [72], conditional random fields as RNNs by Zheng et al. [147], Deep Variational Bayes Filters (DVBFs) by Karl et al. [68], and Structural Variational Autoencoders (SVAE) by Johnson et al. [67].

Other approaches to hybridization use transparency mechanisms inside the black-box models. Bennetot et al. [18] use, for instance, a knowledge-base to enhance a neural network. Ensemble techniques have also been used by authors such as Zhou et al. [148] to create hybrid models that integrate transparent and black-box models. Probabilistic graphical models are another choice given their interpretability advantages: the learned graphical structures can often reveal relevance-independence and causal relationships.

A survey of techniques to enrich neural networks with transparency techniques, extract symbolic rules from neural networks, and utilize ANNs to define rule-based systems is provided by Andrews et al. [7]. Other examples of hybrid models include Self-Explaining Neural Networks (SENN) by Melis and Jaakkola [90], Contextual Explanation Networks (CEN) by Al-Shedivat et al. [3], and BagNets by Brendel and Bethge [20].

Architecturally explainable models display architecture adjustments that enhance their interpretability. For example, the interpretable convolutional network proposed by Zhang et al. [145] is included in this class. The proposed architecture differs from the conventional convolutional architecture in that a loss function is added to each filter in a convolutional layer, which results in more meaningful representations. Another contribution of note is that of Alain and Bengio [4], who propose using linear classifier "probes" to extract information from the intermediate layers of a neural network. The general idea is to use each layer's information to fit a linear classifier function and then observe how well the function can predict the output classes. Joint prediction-explanation models are machine learning models explicitly trained to explain their predictions. An example of such an approach is the Teaching Explanations for Decisions (TED) framework proposed by Hind et al. [58]. TED's underlying idea is to augment the training dataset to include the rationale for the outcome; the explanation is provided explicitly to the algorithm.

Under model transparency, there are also **regularization techniques**. Note the difference between architectural change and regularization techniques. Regularization techniques [73] are specific architectural schemes to reduce overfitting (e.g., weight decay, dropout, and data augmentation). Architectural modifications typically entail more complexity, such as a change of network or altering the model's structural components. As Zhang et al. [143] note, regularization has its significance, however, architectural changes may hold increased interpretability potential.

2.4. Post-model interpretability

In addition to pre-model and in-model interpretability, there is post-model interpretability [23]. Post-model techniques analyze the model after its creation (post-hoc); they are devised as independent methods that can interpret the final decisions. There approaches can be model-specific or model-agnostic [8]. Post-hoc model-specific interpretability consists of methods specifically designed for a given machine learning algorithm. In contrast, post-hoc model-agnostic interpretability is agnostic to the analyzed machine learning model.

Several **model-specific** studies substitute the original model with a simplified version of it. For example, authors such as Barakat and Diederich [14], Martens et al. [89], Barakat and Bradley [15] proposed Support Vector Machines (SVM) rule extraction techniques to enhance the comprehensibility of the model. Assche and Blockeel [9] proposed a method to learn a single decision tree from an ensemble of decision trees.

Aside from model simplification, there are other research directions in model-specific interpretability. For example, DeepLIFT is a method proposed by Shrikumar et al. [123] that tries to compute internal neuron importance. The method inspects deep learning models comparing the activation of a neuron to a "reference activation" and assigns a score to the neuron contribution accordingly. The "reference activation" corresponds to a default input selected by the designer. Another post-hoc model-specific technique is Layer-wise Relevance Propagation (LRP) proposed by Bach et al. [12]. The method produces a heatmap highlighting the pixels responsible for the predicted class in an image classification task.

In addition to LRP, other model-specific visualization techniques have received considerable attention. For example, the use of Saliency Maps (SM) [124] based on the parameters or gradients of neural networks is common. Saliency maps are heatmaps that help visualize the importance of different regions of some visual input. Examples of works using these techniques are by Springenberg et al. [127], Shrikumar et al. [123], Selvaraju et al. [119], Sundararajan et al. [129], Gu et al. [51], Gu and Tresp [50,49].

A popular post-hoc **model-agnostic** interpretability approach consists in generating a neighborhood around the instance. By observing the black-box model's behavior in this neighborhood, it is possible to characterize the relative importance of the input variables. Typically, this is done by fitting an interpretable surrogate model (e.g., linear regression) to the new instances. This kind of approach is model-agnostic [107] as it only requires knowing the black-box model's output for the neighbor instances. The focus is on explaining what the model locally depends on instead of learning the full mapping.

Examples of models that follow the model-agnostic approach include Local Interpretable Model-Agnostic Explanations (LIME) [108] and its variants. The variants of LIME attempt to address its limitations. For example, NormLime proposed by

Ahern et al. [2] tackles the issue of deriving global interpretability from local explanations. LIME-Aleph by Rabold et al. [105] combines the Inductive Logic Programming system Aleph with LIME to provide enriched visual and verbal explanations. GraphLime by Huang et al. [62] is a model-specific (note that the vast majority of LIME approaches are model-agnostic) LIME approach tailored to graph neural networks.

Some of LIME alternatives address the topic of neighborhood generation. By neighborhood generation, we mean creating a set of synthetic instances around the instance to explain. These instances serve to train an interpretable local model from which to extract an explanation. The synthetic instances are classified during the training process utilizing the original black-box model. A variant that uses clustering techniques to address neighborhood generation is KLIME by Hall et al. [54]. This variant of LIME partitions the training set into K clusters and then fits local models to each cluster. Zafar and Khan [140] proposed an alternative neighborhood generation scheme to LIME called Deterministic Local Interpretable Model-Agnostic Explanations (DLIME). In DLIME, instead of random perturbation, a clustering algorithm is combined with K-Nearest Neighbors to discover each instance's relevant cluster. This method's advantage is that it provides stable explanations; however, the quality of the clusters and the local predictions' accuracy depends on the number of samples in the training dataset. Shankaranarayana and Runje [120] proposed the autoencoder-based local interpretability model ALIME. The authors use an autoencoder as data generator and weighting function. Instead of computing the Euclidean distance between generated data and the instance to be explained, ALIME uses the distance on the latent vector space.

Another popular post-hoc interpretability model is SHAP (SHapley Additive exPlanations) [85]. SHAP works by assigning a SHAP value [121] to each predictor to indicate its contribution to the final outcome. Lundberg and Lee [85] show that SHAP provides guarantees of accuracy and stability and that LIME is actually a subset of SHAP lacking those properties. SHAP is one of the most consistent approaches to post-hoc interpretability [85] and therefore, the method subject to investigation in this paper. In the next section, we review some of the most important contributions to the field of interpretability in prognostics.

2.5. Interpretability in prognostics

In prognostics, interpretability methods are starting to be used more extensively. Some authors have proposed model-transparent methods for prognostics, such as Xie et al. [136], who explain hard disk failure predictions by performing a series of feature replacement tests to determine failure causes. Keneni et al. [69] propose an explainable model for the decisions of an Unmanned Aerial Vehicle (UAV). The explainable model is based on the Sugeno-type fuzzy inference. Other authors, such as Amruthnath and Gupta [6], perform fault diagnosis using factor analysis classical techniques. In the work of Amruthnath and Gupta [6], Gaussian mixture clustering is used to partition the data into significant groups, and spectrum analysis to diagnose each cluster to a specific state of the machine. The significant features are identified with a random forest classification scheme.

Recently, Lee et al. [80] proposed an explainable deep learning approach to estimate the remaining useful lives of rotating machinery. The model first learns high-level features using an autoencoder. The features are used as input to a feedforward neural network to estimate the remaining useful life. Octave-band filtering simplifies the model and improves its interpretability.

Regarding post-hoc XAI modeling in prognostics, there are a few important contributions. For example, in the work of Zeldam [141], the authors work with sensor data extracted from a 2.4L diesel engine. LIME is used to identify the critical sensors concerning anomaly detection. Seemingly, LIME is utilized by Madhikermi et al. [87] to explain fault detection in the heat recovery of an air handling unit.

A comparative study of interpretability techniques in prognostics that includes SHAP and LIME is by Jalali et al. [65]. This work differs from our own in that we do not compare different interpretability methods from XAI but instead we compare SHAP performance against classical pre-interpretability methods from prognostics. More specifically, we aim to investigate the relationship between SHAP values and the metrics of monotonicity, trendability, and prognosability [32].

Despite the importance of these contributions, there is still considerable work to be done in the field of eXplainable prognostics. XAI could help answer pressing industrial questions of data-driven prognostics models such as "why is the model giving this prediction", "which sensor is triggering the next failure" or also importantly "can a given model prediction be trusted". With this work we aim not only to analyze the SHAP modeling approach but also to motivate other researchers to investigate the previously reviewed methods and how to apply them to the black-box models of PHM.

3. Approach

In this paper, we correlate three popular prognostics metrics of predictor importance with the SHAP values. We start by describing the utilized prognostics metrics. We briefly review SHAP, the XAI method used in this work. We then describe the denoising solution used and the three data-driven models studied in the paper.

3.1. Prognostics feature selection metrics

Metrics such as monotonicity, trendability, and prognosability are often used in Prognostics and Health Management (PHM) applications [81] to compare potential predictors. Nevertheless, different authors often use distinct methods to compute these evaluation indicators [81]. In this work, we adopt the definitions and formalization proposed by Coble and

Hines [32]. Extensive work has been done in PHM [32,31,33,98,109] based on the concepts proposed by these authors. We hereafter explain each metric in detail.

Monotonicity characterizes the increasing or decreasing trend of a predictor. Formally, the monotonicity of a predictor (feature) is defined as

$$\text{monotonicity} = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{N_j-1}\frac{\text{sgn}\big(x_j(k+1)-x_j(k)\big)}{N_j-1}\right| \tag{1}$$

where:

M = number of units
N_j = number of measurements of a feature on unit j
x_j = vector of measurements of a feature on unit j
$x_j(k+1)$ = a measurement of a feature of unit j at time $k+1$
$x_j(k)$ = a measurement of a feature of unit j at time k
sgn = sign function

With this metric we measure the degree of monotonicity of a signal. In PHM, if a predictor shows an obvious increasing or decreasing trend over time, more accurate Remaining Useful Life (RUL) prediction results are expected. The monotonicity is in the range [0, 1]. A monotonicity of 1 means that the feature is strictly monotonic, whereas a monotonicity of zero means the feature has the least possible monotonicity and it is usually a non-desirable predictor for PHM applications.

Trendability measures the extent to which the predictor displays the same shape across a group of units. It is a measure of similarity between all the damage trajectories of the population of units. The metric of trendability is devised for run-to-failure data. Formally, the trendability of a predictor (feature) is defined as

$$\text{trendability} = \min_{j,k}\big|\text{corr}(x_j, x_k)\big|, \quad j,k = 1,\ldots,M \tag{2}$$

where:

M = number of units
x_j = vector of measurements of a feature on unit j
x_k = vector of measurements of a feature on unit k
corr = Pearson correlation function

When x_j and x_j have different lengths we (linearly) interpolate the smallest vector to match the length of the longer vector. The linear interpolation between two points (x_0, y_0) and (x_1, y_1) is given by the formula

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} \tag{3}$$

The trendability metric is in the range $[0, 1]$ and, as the monotonicity metric, it is positively correlated with the importance of the predictor. This metric measures how much the signals of different units resemble one another.

Prognosability measures the variance of a predictor at the End of Life (EoL) for the set of units. Formally, the prognosability of a predictor (feature) is defined as

$$\text{prognosability} = \exp\left(-\frac{\text{std}_j(x_j(N_j))}{\text{mean}_j|x_j(1) - x_j(N_j)|}\right) \tag{4}$$

where:

M = number of units
j = index of unit ($j = 1,\ldots,M$)
N_j = number of measurements of a feature on unit j
x_j = vector of measurements of a feature on unit j
mean_j = average function of all units
std_j = standard deviation function of all units

Prognosability measures the degree to which failure occurs at the same measurement level, for a population of units. It measures the dispersion of the fault critical level for each predictor. Prognosability is within the range of $[0, 1]$. When it is close to 1, the failure measurements at the EoL are similar and when it is close to zero it indicates that failure measurements are different from each other. The performance of the predictor is expected to be higher when the prognosability is close to 1.

Monotonicity and trendability metrics are widely used in PHM literature. For example, Javed et al. [66] apply trigonometric functions and cumulative transformation to vibration signals (decomposed by Discrete wavelet transform) to enhance their monotonic and trendable traits. The final features are selected based on their monotonicity and trendability scores. In the work of Saidi et al. [113], monotonicity and trendability are used to evaluate the performance of time-domain predictors derived from spectral Kurtosis applied to bearing degradation health data. She and Jia [122] use monotonicity and trendability to evaluate wear indicators. Other works that use monotonicity and trendability to assess their predictors or health indexes' fitness can be found in Ref. [61,21,75,130,26].

Prognosability is a less popular but also important metric that some authors use in PHM. For example, He et al. [57] compare distinct time-domain health indexes using the performance indicators of monotonicity, trendability, and prognosability. Baraldi et al. [16] optimize a health indicator based on the three properties of monotonicity, trendability, and prognosability. The indicators of monotonicity, trendability, and prognosability are used by Qiu et al. [104] to assess the performance of a bearing health index.

3.2. SHAP model

Machine learning can help establish the sometimes complex and occasionally counter-intuitive patterns observed between predictor variables and the residual life of engineering equipment. However, the lack of a deep understanding of machine learning models prevents its widespread use in the Prognostics and Health Management (PHM) community. This paper studies a recently proposed method from eXplainable Artificial Intelligence (XAI), the SHapley Additive exPlanations (SHAP) approach, evaluating the quality of the produced explanatory trajectories using the classical prognostics metrics described in Section 3.1.

SHAP [85] is a model-agnostic approach from XAI that draws its foundations from game theory [128]. The goal of SHAP is to explain a prediction $f(x)$ of an instance x by computing the relative contribution of each feature value to the specific outcome. The explanation function $g(.)$ receives as input a coalition vector $z' \subset \{0, 1\}^N$ where N is the number of features in the original instance vector x. The coalition vector represents the presence or absence of each feature in a binary format: an entry of 1 means that the corresponding feature contributes to the explanation, while an entry of 0 means that the feature is considered to have no contribution. We have that the explanation function $g(z')$ can be decomposed as follows:

$$g(z') = \phi_0 + \sum_{i=1}^{N} \phi_i z'_i, \quad \phi_i \in \mathbb{R} \tag{5}$$

where:

N = number of input features in x, the instance vector
g = explanation model
z' = coalition vector such that $z' \subset \{0, 1\}^N$
ϕ_i = decomposition factor

Several methods match the definition in Equation (5), namely LIME [108], DeepLIFT [123] and Layer-wise Relevance Propagation (LRP) [12]. These are all additive feature attribution methods that, as SHAP, attribute an effect (or importance) ϕ_i to each predictor (feature), and the sum of these effects, $g(z')$, approximates the output $f(x)$ of the original model. As an example, consider Fig. 1. The picture displays the relationship between an input vector and the corresponding prediction. Here, the feature values, x_i lead to the prediction $f(x)$. SHAP, and the other referred models, work by assigning a decomposition factor ϕ_i, to each feature value, which aims to reflect the importance of the feature to that particular prediction.

Assuming the four axioms of efficiency, symmetry, dummy and additivity, the previous decomposition has been shown [85] to have a unique solution known as Shapley value, proposed by Lloyd Shapley [121] in cooperative game theory:

$$\phi_i(f, x) = \frac{1}{N!} \sum_{S \subseteq P \setminus \{x_i\}} \Big[|S|!(N - |S| - 1)! \Big] \Big[f(S \cup \{x_i\}) - f(S) \Big] \tag{6}$$

where:

x = instance vector
N = number of input features in x

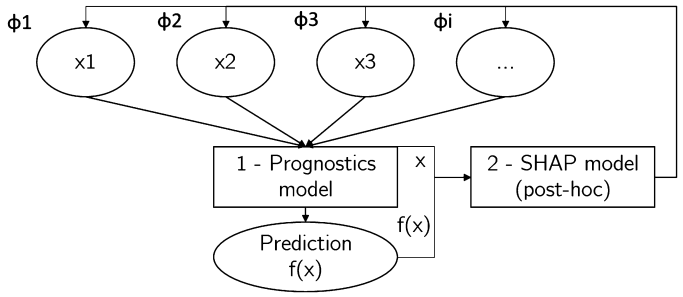


Fig. 1. Overview of the relationship between an input instance vector and the corresponding prediction, when the (prognostics) model is a black box machine learning algorithm. The inputs lead to the prediction without explicit causal relationships. The SHAP model is able to output decomposition factors (SHAP values) facilitating the understanding of the importance of each feature value to the prediction.

f = original prediction model
ϕ_i = decomposition factor (Shapley value)
P = set with all the feature values used in the model
S = subset of the features used in the model that does not include the feature value x_i
$|S|$ = number of non-zero entries in S

The Shapley value of a feature i characterizes the gain of adding feature i, weighted and summed over all possible feature value combinations where the feature i is not present. Note the difference between feature value and the Shapley value. The feature value is the numerical or categorical value of a feature instance; the Shapley value is the feature contribution to the prediction. The formula can be rewritten and understood as:

$$\phi_i(f, x) = \frac{1}{\text{number of coalitions}} \sum_{\text{coalitions excluding i}} \text{weight} * \Big[\text{marginal contribution of i to coalition}\Big] \tag{7}$$

In simple terms, the correct interpretation of a Shapley value is the contribution of the feature value to the difference between the actual prediction and the mean prediction. Unfortunately, the calculation of the Shapley values is often time-consuming, and only an approximate solution is feasible in such cases. An approximation to Equation (6) was proposed by Štrumbelj and Kononenko [128] that can be obtained with Monte-Carlo sampling. The SHAP model assigns each feature (and each local prediction) an approximate Shapley value, the SHAP value, that represents the change in the expected model prediction calculating the effect of observing or not observing the feature. Features with large absolute SHAP values are more important. The range of the SHAP values is bounded by the prediction range.

Note that from the previous description, the Shapley values take into account the interdependence of the features. However, the methods that calculate approximations to the Shapley values, the SHAP values, might produce imprecise explanations. Only recently, have authors [1] started to address this kind of problematic. The utility of this work is to provide means to measure the quality of individual SHAP values assuming that the used SHAP method provides reasonably good estimations.

3.3. Denoising generative adversarial network

Sensor data are typically contaminated with noise and also (often non-normally) distributed outliers. Such signal corruption makes the problem of computing the residual life of the engineering equipment a challenging task [125]. To filter the data and mitigate noise and outliers we use a Generative Adversarial Network (GAN). Deep generative adversarial neural networks have been shown to outperform other methods in noise removal, especially in computer vision (for a review, see Ref. [48]) and more recently, for 1-dimensional signal processing, see the works of Creswell and Bharath [34] and Casas et al. [24].

In this work, we adopt a standard GAN architecture as our denoising model. We provide a brief introduction to the topic and refer readers to follow Ref. [46] for more details. In broad terms, a GAN consists of two neural networks that compete to reach a learning solution. In our case and during training, the GAN takes as input a pair of synthetic sensor signals – one synthetic signal subject to noise and the same signal without such artifacts. The generator network "learns" how to reduce the noise by minimizing the difference between the recovered signal and the noise-free signal. The loss function of the generator, i.e., the function that measures the difference between the two signals, is the discriminator neural network. After the training phase, the GAN generator can receive a new set of (non-synthetic) signals and denoise them.

A challenge of our work was how to generate synthetic pairs of signals with and without noise for the GAN's training phase. In this work, and to train the GAN, we generate synthetic sensor data according to the same methodology of Baptista et al. [13]. We refer the reader to that work for more details. The result of the synthetic data generation process is a collection of synthetic trajectories. These synthetic data are used for the sole purpose of training the GAN. The most important training hyperparameters of the GAN are outlined in Table 1.

Table 1
Principal hyperparameters of Denoising Generative Adversarial Network.

Parameter	Value
2D representation width	20
2D representation height	20
2D number of channels	1
Base number of filters	64
Kernel size	(3, 3)
Strides	(2, 2)
Dropout rate	0.5
Learning rate	2e-4
Max epochs	2000
Regularization L1 lambda	100

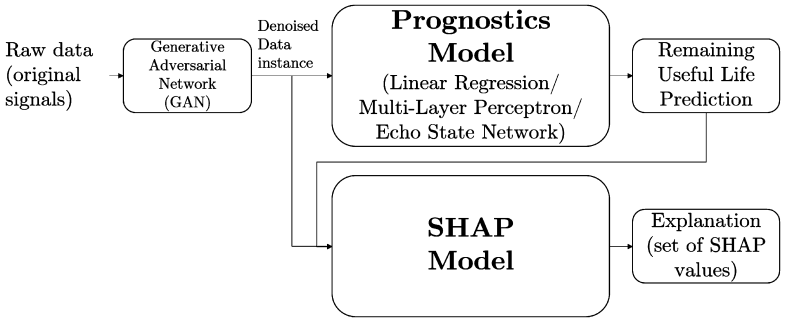


Fig. 2. Overview of the interpretability process of a prognostics model by the SHAP approach.

It is also important to note that sensor data differ from image data. Each sensor trajectory of a unit is a 1-dimensional time-series data. In contrast, images are 2-dimensional representations of a given point in time. For these reasons, and since we use the traditional convolutional GAN, our signals are transformed into 2D representations before they are fed into the network architecture. A simple reshape operation accomplishes this transformation. This operation transforms a 1D vector representation (the original signal) to a 2D image representation. It is not expected that the configuration of this operation, i.e., the setting of the image width and height, influences the results significantly. Other works have followed this representation choice when dealing with one-dimensional time series as can be found in [36,83].

3.4. Remaining useful life estimation

The objective of a prognostic model is to estimate (predict) the Remaining Useful Life (RUL) of an engineering equipment as accurately as possible. The model's focus is typically not causality (explanation or interpretability) but the accuracy of the estimation. How the prognostics models are developed is, therefore, different from explanatory methods such as SHAP. SHAP investigates the causal relationships between the prognostic model's input and its output. Fig. 2 depicts the flow to obtain a local explanation for a prediction. The SHAP model receives as input both the observation instance and the resulting prediction. This model tries to establish a reasonable connection between input and output in a simplified way that is more transparent and less complex than the original prognostics algorithm. The RUL is the outcome of the prognostics model, while the corresponding explanation is the outcome of the SHAP model.

In this work, we study three prognostics approaches with an increasing level of complexity, namely: Linear Regression (LR), Multi-Layer Perceptron (MLP), and the Echo State Network (ESN). We briefly describe each of these techniques. Linear regression (LR) is the most straightforward machine learning algorithm, while the Multi-Layer Perceptron (MLP) or Feed Forward Neural Network (FFNN) is the classical neural network. An MLP is composed of one or more layers. In this work, and for simplicity, we use a single layer network. A more complex neural network is the Echo State Network (ESN) [64]. This network is a Recurrent Neural Network due to its feedback loops. It is also a Reservoir Computing technique [64] with the particularity that the reservoir is sparsely connected. The ESN has been shown to be well fitted for time series forecasting [30,137]. It is one promising approach in prognostics due to its dynamic properties, generalization, and training speed. Some works in PHM have used this approach with positive results. For example, Morando et al. [95] apply the ESN to the Remaining Useful Life (RUL) estimation of industrial fuel cells. Fink et al. [40] combine an ESN and Conditional Restricted Boltzmann Machines to predict railway speed restrictions. Wang et al. [134] estimate the health state of lithium-ion batteries using the ESN. Rigamonti et al. [110] use a single ESN and Rigamonti et al. [109] an ensemble of optimized ESNs for RUL estimation.

The three models are optimized during the training process. The ESN is optimized with Differential Evolution (DE) as in the work of Rigamonti et al. [110]. Further details on the ESN configuration can be found in our previous work in [13].

Table 2
Non-flat predictors (features) in C-MAPSS data.

Predictor	Description	Units
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	–
htBleed	Bleed Enthalpy	–
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

4. Methodology

This section explains the case study, the hypotheses of the paper, and the metrics used to assess them. We also refer the indicators used to evaluate the performance of the Remaining Useful Life (RUL) estimation.

4.1. Case study

The data used in this case study is from the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) developed at the NASA Glenn Research Center [41]. C-MAPSS simulator mimics the operation of a large, high-bypass ratio turbofan engine similar to GE90. The C-MAPSS data consists of a collection of measurements taken at cruise. Each degradation trajectory is characterized by a series of predictors (features), including sensor and operational variables, changing over time from some nominal condition to failure. The data is described in detail by Saxena et al. [117]. This paper studies the C-MAPSS training dataset 1, which comprises simulated data of 100 jet engines. The recorded useful life ranged from 128 to 362 cycles. We selected this dataset as it is the simplest, including only one operating regime and one fault mode.

In C-MAPSS, an engine unit is characterized by 21 prognostics sensors and three additional indicators (Altitude, Mach Number, and Throttle Resolver Angle). From the 21 available features, we selected 14 features that were not steady state signals (the features with non-constant value). Table 2 describes each selected feature.

4.2. Metrics and hypotheses

We advance three hypotheses to test on C-MAPSS data. The first hypothesis concerns how SHAP values change over time for each unit:

H1: SHAP values tend to exhibit monotonic behavior.

We use the monotonicity metric (Equation (1)) to assess this hypothesis. The second hypothesis concerns how SHAP values change over time for a population of units:

H2: SHAP values tend to exhibit trendability behavior.

We use the trendability metric (Equation (2)) to assess this hypothesis. We also investigate prognosability. We use the prognosability metric (Equation (4)) to assess the following hypothesis:

H3: SHAP values tend to exhibit prognosability behavior.

We accept these hypotheses even if, for some instances, the behavior might not perfectly monotonic, trendable, or prognosable. It is also accepted (and expected) that some features will not exhibit these traits. Therefore, we are aiming to prove that, in general, the SHAP trajectories tend to exhibit these properties. In order to investigate the hypotheses, we apply the following methodology:

- *Pre-processing*: The data is pre-processed using the denoising Generative Adversarial Network (GAN).
- *Prediction*: A set of randomly selected units (40% of total units) is used to train three different models (LR, MLP, and ESN). The remaining units are used for configuration, validation, and testing (10%, 10% and 40% of all units).
- *Interpretability*: A SHAP model is built on top of each of the prognostics models LR, MLP, and ESN. A set of SHAP values, one for each feature (or predictor), is generated for an input instance.

Table 3
RUL estimation performance (on C-MAPSS first dataset).

	PHM'08 Score	Accuracy	FPR	FNR	MAE	RMSE	MAPE
Linear Regression	$2055\pm2.5\times10^3$	24.81±1.8	36.97±4.8	38.21±3.7	30.49±1.4	39.53±3.2	82.36±5.9
Multi-Layer Perceptron	$12077\pm2.3\times10^4$	33.85±2.5	37.33±8.7	28.82±6.3	26.37±1.7	35.92±2.8	46.41±5.6
Echo-State Network	829.47±960.7	26.72±3.2	36.81±7.9	36.48±7.3	29.7±2.9	38.54±4.2	65.58±16.9

- *Validation*: The monotonicity, trendability and prognosability of the SHAP values are evaluated according to Equations (1), (2), and (4) respectively. We rerun the experience several times (K=10) and retrieve appropriate statistics from the results.

We are also interested in correlating the features (predictors), along the dimensions of monotonicity, trendability, and prognosability with the SHAP values. To this effect we advance another three hypotheses:

H4: The monotonicity of the SHAP values for one predictor is correlated with the predictor's monotonicity.
H5: The trendability of the SHAP values for one predictor is correlated with the predictor's trendability.
H6: The prognosability of the SHAP values for one predictor is correlated with the predictor's prognosability.

Correlation is measured using the Spearman rank correlation coefficient. Spearman's rank correlation coefficient or Spearman's ρ, is a statistical measure of nonparametric (no requirement of normality) correlation. It aims at evaluating how well the relationship between two variables can be described by a monotonic function. Several attempts have been made to translate the Spearman's correlation coefficient into descriptive labels such as "weak", "moderate", or "strong" relationship but there is actually no formal consensus [118]. Nevertheless, when the correlation is at 0.50 authors often consider it to be a moderate relationship. We use this correlation cut-off value to discuss the previous hypotheses. We are stating in H4 to H6 that the quality of SHAP trajectories is directly related to the feature quality. For example, if a feature changes its behavior in a non-monotonic way, it is expected that the marginal contribution of the feature to prognostics will also fluctuate. Put differently, if the feature is of no quality, then the explanations related to the feature are expected to be not to be trusted since its quality will also be low. These hypotheses imply that SHAP values can be combined with classical prognostics metrics to evaluate predictors' significance (importance).

4.3. Evaluation of RUL estimation performance

We compute the Remaining Useful Life (RUL) for Linear Regression (LR), Multi-Layer Perceptron (MLP), and Echo State Network (ESN). The following metrics are used to evaluate model performance: Scoring function of the PHM Challenge in 2008 (S), Accuracy (A), False Positive Rate (FPR) and False Negative Rate (FNR), Mean Average Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Saxena et al. [115], Ramasso and Saxena [106] provide more detail on these metrics.

5. Results

In this section we present and discuss the results of our experiments. We discuss the obtained scores of monotonicity, trendability, and prognosability of the SHAP values. We also examine the rankings of predictor importance produced by these scores.

5.1. Remaining useful life performance evaluation

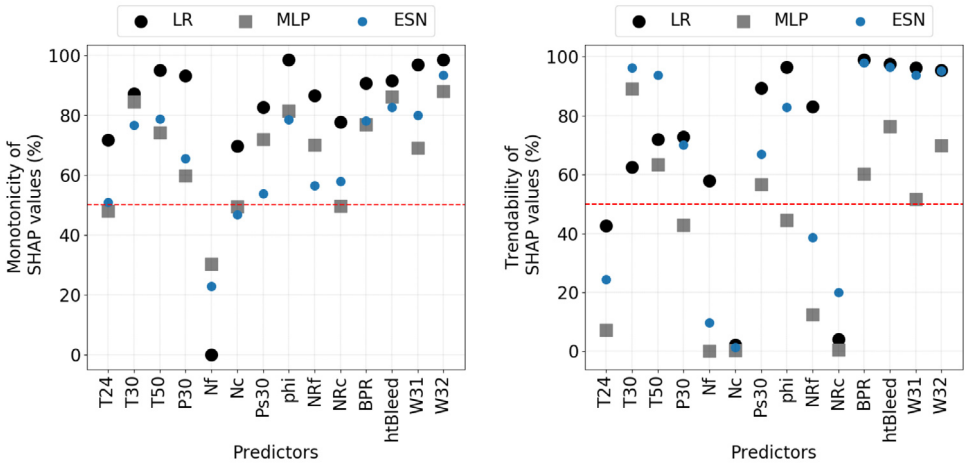
The results of evaluating the increasingly complex prognostics models are presented in Table 3. We show the average results of the ten runs of the cross-validation method with the corresponding standard deviation. As it is shown, there was no clear winning model as the different approaches ranked differently according to the utilized evaluation metric.

The best performing model in regards to the averaging metrics of MAE, RMSE, and MAPE is the Multi-Layer Perceptron (MLP). The MAE, RMSE and MAPE results of the MLP are the lowest of the three tested models. In regards to the PHM'08 metric, the MLP model has the worse performance but this follows from the tendency of the model to overestimate instead of underestimate. Since this metric penalizes late predictions (late predictions of failure) the MLP scores low in this respect.

Importantly, the model that scored the best at the PHM'08 Score was the Echo-State Network. This was the metric selected at the original C-MAPSS competition to evaluate the quality of the prognostics exercise. The interesting trait of this metric is that it penalizes late predictions over early predictions.

5.2. Monotonicity of SHAP values

The evolution of the SHAP values over time for different features (or predictors) is shown in Figs. 4, 5 and 6. Each Figure corresponds to one model namely, Regression (LR) (Fig. 4), Multi-Layer Perceptron (MLP) (Fig. 5), and Echo-State Network



(a) Monotonicity of SHAP values

(b) Trendability of SHAP values

(c) Prognosability of SHAP values

Fig. 3. Monotonicity, trendability and prognosability of the trajectories formed by the SHAP values of different predictors in different models (Linear Regression (LR), Multi-Layer Perceptron (MLP), and Echo State Network (ESN)).

(ESN) (Fig. 6). Each plot represents the SHAP values of a feature for an engine. We selected the presented four features: T24, P30, Nc, and Nf, as these features correspond to representative cases of the different trajectories formed by the SHAP values. The aim here was to show that the shapes of SHAP trajectories depend not only on the feature but also on the model. Interestingly, as time goes by, we can observe monotonic trends for a significant number of predictors in all the studied models (LR, MLP, and ESN).

The performance of the three models in Remaining Useful Life (RUL) estimation is presented in Table 3. A quantitative analysis of the trajectories formed by the SHAP values resulted in Fig. 3. Each plot of Fig. 3 shows the monotonicity, trendability, and prognosability scores (y-axis) of the SHAP value trajectories for different predictors (x-axis) in each of the three considered models (LR, MLP, and ESN).

As depicted in Fig. 3a, the predictors showed a marked tendency to score high ($> 50\%$) in monotonicity across the models. Only 4 out of 42 predictors (9.52%) had scores below the 50%. This property of SHAP values is important to prognostics as it means that SHAP values provide consistent information over the equipment's life. Had the values fluctuated with no clear trend, it would have been a sign that the explanations were not coherent/connected over time. It is desirable that the significance (or contribution) of a predictor changes over time in a slow manner towards the equipment's end of life.

Comparing the monotonicity shown by the SHAP values for the different models (see Fig. 3a), Linear Regression (LR) consistently presents the highest scores. This result can also be examined in the plots of Fig. 4 which illustrate how the SHAP values of LR exhibit monotonic trends over time. In LR, except for Nf, all the other predictor signals are increasing/decreasing time functions.

SHAP assigns Nf with a zero contribution in the LR model for most of the time. This non-significance attributed by SHAP to the Nf predictor in the LR model may be explained by the low monotonicity and trendability of the original features (see

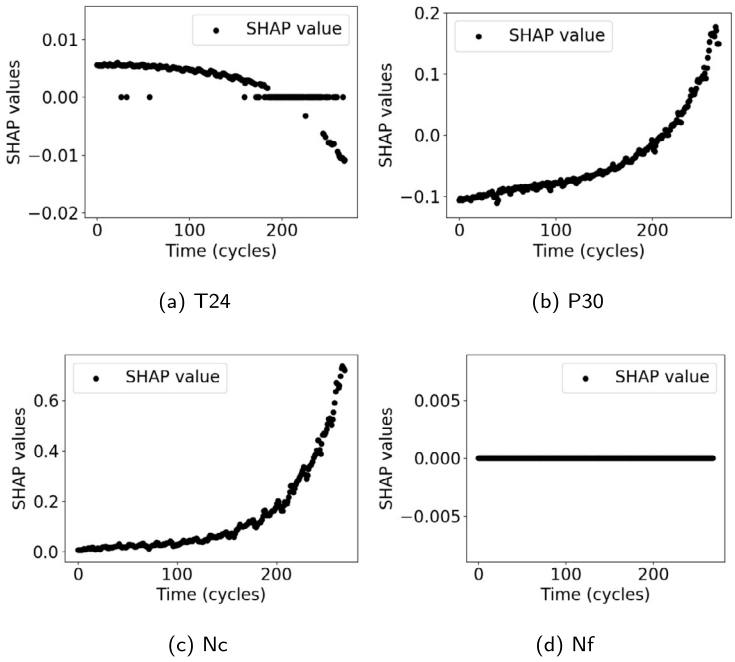


Fig. 4. Evolution of SHAP values over time for different predictors of the same unit (Linear Regression - LR).

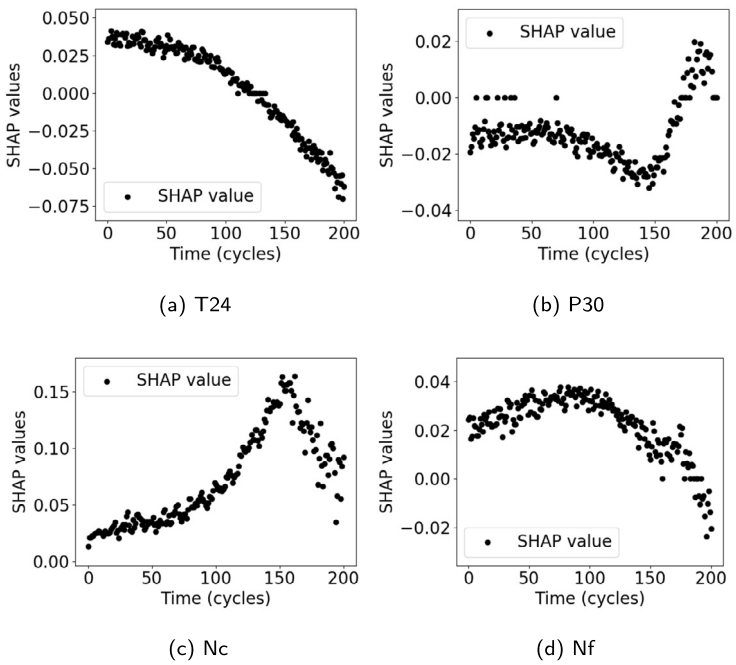


Fig. 5. Evolution of SHAP values over time for different predictors of the same unit (MultiLayer Perceptron - MLP).

Table 4). From all the predictors, Nf is the feature with the lowest trendability. Seemingly, and still analyzing the LR model, T24, Nc, and NRc experience low SHAP monotonicity (see Fig. 3a) while also scoring low in original predictor trendability (see Table 4).

The Multi-Layer Perceptron (MLP) is a more complex model than Linear Regression (LR), but it also signals the predictors Nf, T24, Nc, and NRc as the less SHAP monotonic. As shown in Figs. 5 and 3a, the monotonicity of the MLP SHAP values is not as high as in the LR case. Note that the SHAP values represent the impact of a feature in the outcome. Since the SHAP values of different predictors interfere with each other, a predictor's SHAP values may change its direction over time. This

Table 4
Monotonicity, trendability and prognosability of the selected predictors (features) from C-MAPSS data. Scores (average ± standard deviation) are shown as percentages.

Predictor	Monotonicity	Trendability	Prognosability
T24	81.48±1.6	68.11±25	92.64±0.7
T30	79.54±1.1	97.11±0.4	91.47±0.6
T50	88.18±1.1	96.41±1.1	94.4±0.5
P30	80.98±0.5	96.65±1	88.13±0.9
Nf	73.42±3.2	3.01±7.7	68.43±2
Nc	50.6±2.9	5.83±10.7	32.69±3
Ps30	69.22±1.4	88.75±1.5	92.94±0.5
phi	95.34±0.9	97.47±0.2	88.87±0.6
NRf	80.05±1.8	77.36±7	69.33±1.4
NRc	68.16±3.1	7.33±14.8	30.54±2.5
BPR	81.62±0.7	98.44±0.4	94.15±0.3
htBleed	85.09±0.8	95.7±0.7	91.32±0.6
W31	92.43±0.8	96.23±0.5	94.96±0.5
W32	94.09±1.5	94.73±1	94.38±0.5

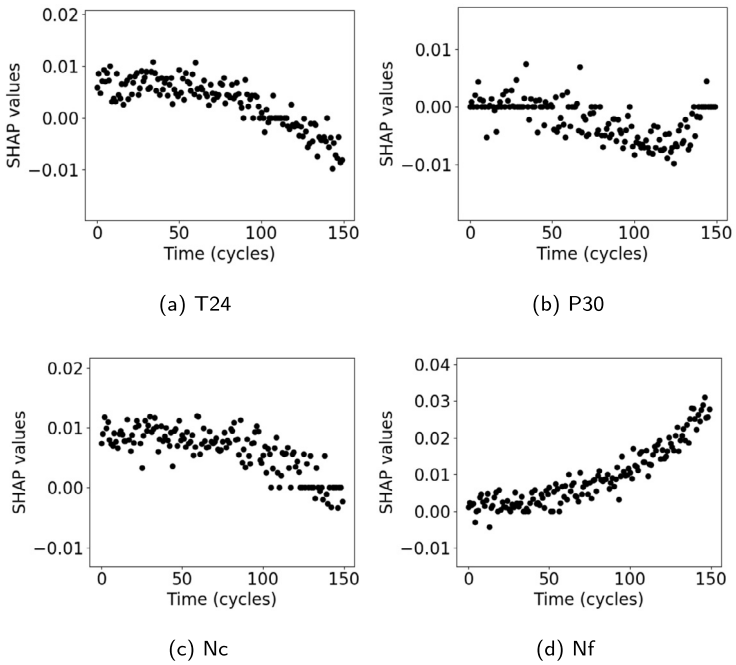


Fig. 6. Evolution of SHAP values over time for different predictors of the same unit (Echo State Network - ESN).

lack of trendability is not necessarily alarming; however, it is more desirable and understandable to have strictly monotonic SHAP values.

The most complex model examined is the Echo State Network (ESN). The SHAP values of ESN have monotonicity scores comparable to that of the Multi-Layer Perceptron (MLP) (see Fig. 3a). Nevertheless, observing the SHAP values' trajectories in Fig. 6, it can be seen that there are fewer changes in direction. This lack of changes in direction can be explained by the fact that the ESN is a Recurrent Neural Network (RNN), and hence, its predictions are linked in time. The ESN model produces, however, more noise in the SHAP trajectories compared to the MLP. The increased level of noise in the ESN's SHAP values has probably to do with the increased complexity of the model. For example, a linear regression model is a weighted sum of variables. Given its simplicity, in linear regression, the output is less likely to fluctuate than a neural network. As a result, the computed SHAP values (contributions to the output) are less oscillatory for simpler models, such as linear regression.

Given that the SHAP monotonicity of our three models was high, this reinforces the notion in **H1** that:

H1: SHAP values tend to exhibit monotonic behavior.

We were also interested in testing if the SHAP monotonic ranking of the predictors by the different models was similar. To this effect, we compared the predictor rankings using the Spearman rank correlation coefficient (ρ). Table 5 shows how

Table 5
Spearman rank correlation analysis of monotonicity scores obtained from the SHAP values of different models (Linear Regression (LR), Multi-Layer Perceptron (MLP), and Echo State Network (ESN)).

		Monotonicity		
	Model:	LR	MLP	ESN
	LR		0.70	0.89
Monotonicity	MLP	0.70		0.80
	ESN	0.89	0.80	

Table 6
Spearman rank correlation analysis of trendability scores obtained from the SHAP values of different models (Linear Regression (LR), Multi-Layer Perceptron (MLP), and Echo State Network (ESN)).

		Trendability		
	Model:	LR	MLP	ESN
	LR		0.60	0.78
Trendability	MLP	0.60		0.69
	ESN	0.78	0.69	

the rankings based on SHAP monotonicity correlated well with each other for the different models. As presented, all the models correlated well with each other ($\rho > 0.70$ with p-value[1] < 0.01). In particular, the LR model and the ESN showed the highest correlation. These results are important as they indicate that the rankings of the most significant features, according to SHAP monotonicity, are similar regardless of the model. To a certain extent, this outcome attests the integrity of the monotonicity rankings produced by SHAP.

5.3. Trendability of SHAP values

Comparing the trendability (see Fig. 3b) shown by the trajectories of the SHAP values, we can see that the MLP has lower scores than the ESN and LR. The model has seven predictors below the 50% trendability score. In contrast, the ESN has only five predictors and the LR three predictors below the 50% trendability score. The desirable outcome is a high trendability score. High SHAP trendability means the SHAP values tend to have the same functional form across different units. Traditionally, trendability is expected of good features. This result suggests that SHAP values can be used to reflect damage degradation over the equipment's life. They can be used, not to perform the prediction itself, as SHAP values are calculated after the forecasting, but to expose damage progression of the equipment and elicit a better understanding of the equipment's actual health state at each moment.

In general, it can be seen that 14 predictors are below the 50% mark, but the majority of the predictors, 28 of them, which amount to 67% of the predictors, score above the 50% threshold, some with very high trendability scores above 80%. Given that the SHAP trendability of the three models was high, this reinforces the notion in **H2** that:

H2: SHAP values tend to exhibit trendability behavior.

As in the case of SHAP monotonicity, we were interested in analyzing if the different models' SHAP trendability rankings were correlated. To this end, we compared the predictor rankings using the Spearman Rank correlation coefficient (ρ). Table 6 shows how the rankings based on SHAP trendability correlated with each other for the different models. Even though the correlations were not as high as for SHAP monotonicity, still, the models correlated reasonably well with each other ($\rho > 0.60$ with p-value < 0.01). These results are important as they indicate that the rankings of the most significant features, according to SHAP trendability, are similar for different models. This result again attests to the integrity of the rankings generated by SHAP for prognostics.

5.4. Prognosability of SHAP values

Concerning the prognosability of the SHAP values (see Fig. 3c), most predictors exhibited high prognosability for all models. This result means that the variance of the SHAP values at the end of life is low for the population of engine units. This outcome is an expected result, as we are dealing with C-MAPSS dataset 1, which concerns only one failure mode. A predictor's impact at the end of life is likely to be the same across different units. It would be interesting to check how

[1] In null hypothesis significance testing, the p-value is the likelihood of obtaining results confirming the null hypothesis.

Table 7
Spearman rank correlation analysis of prognosability scores obtained from the SHAP values of different models (Linear Regression (LR), Multi-Layer Perceptron (MLP), and Echo State Network (ESN)).

		Prognosability		
	Model:	LR	MLP	ESN
Prognosability	LR		0.56	0.78
	MLP	0.56		0.69
	ESN	0.78	0.69	

Table 8
Spearman Rank correlation analysis of relationship between monotonicity (Mon), trendability (Tren) and prognosability (Prog) rankings by SHAP values and classical monotonicity, trendability and prognosability rankings.

	ρ-Mon	ρ-Trend	ρ-Prog
Linear Regression (LR)	0.93	0.83	0.98
MultiLayer Perceptron (MLP)	0.58	0.52	0.58
Echo State Network (ESN)	0.71	0.65	0.94

these results change for other C-MAPSS datasets with more failure modes. With these results, we confirm the **H3** notion that:

H3: SHAP values tend to exhibit prognosability behavior.

We note that there are, however, some predictors with low SHAP prognosability for LR, MLP, and ESN. These predictors are T24, Nf, Nc, and NRc. This low SHAP prognosability seems to be linked to the original predictors' low trendability scores (see Table 4).

As for SHAP monotonicity and SHAP trendability, we also studied if the SHAP importance rankings produced by the prognosability indicator were correlated. We compared the rankings using the Spearman rank correlation coefficient (ρ). Table 7 shows how the rankings based on SHAP prognosability correlated well with each other for the different models ($\rho > 0.56$ with p-value< 0.01). This result implies that SHAP ranks features according to SHAP prognosability in a similar way for the considered models (LR, MLP, and ESN). This finding validates the utility and generality of the importance rankings derived from SHAP for prognostics.

5.5. Measuring predictor importance with SHAP

To investigate hypothesis **H4**, **H5**, and **H6**, we used Spearman rank correlation analysis. Fisher transformation was used to average the results of several experiments. The analysis is similar to the one in Tables 5, 6, and 7, but in this case, we correlate the importance rankings produced by the originals predictors with the importance rankings produced by the SHAP values. The goal was not to check if different models had similar SHAP rankings but to analyze if the significance assigned to each predictor by the classical metrics on top of SHAP was similar to the significance computed using the classical metrics on top of the original predictors. Results are summarized in Table 8.

As can be seen in Table 8, applying the classical metrics to the SHAP values of linear regression (LR) and Echo State Network (ESN) produced rankings that correlated well with applying the same metrics directly on top of the predictors. The Multi-Layer Perceptron (MLP) had slightly worse results in this respect, but except for the trendability metric, the monotonicity and prognosability indicators had acceptable correlation scores. These results reinforce the notions that

H4: The monotonicity of the SHAP values for one predictor is correlated with the predictor's monotonicity.
H5: The trendability of the SHAP values for one predictor is correlated with the predictor's trendability.
H6: The prognosability of the SHAP values for one predictor is correlated with the predictor's prognosability.

These results are important as they indicate that we can compute the classical metrics of monotonicity, trendability, and prognosability on top of SHAP values with some certainty that they will indicate the best features from the available predictor set.

6. Conclusion

Prognostics approaches based on machine learning are typically perceived as black-box models that are severely dependent on the quality of the training data. The application of interpretability methods to these prognostics algorithms is a way to engage in complex questions, from transparency to safety and ethics. Discussion and debate over the explainability

of the Prognostics and Health Management (PHM) models can bring additional information to different engineering fields, from aerospace to energy, bringing tacit knowledge about which and how the different sensor features contribute to the estimations along the life cycle of different equipment. Acquiring this knowledge is an essential step in moving towards interpretability and explainability in engineering.

Monotonicity, trendability and prognosability [32] are important metrics often used to evaluate health monitoring trajectories. In this work, we applied the same metrics to evaluate the quality of SHAP values. In prognostics, SHAP values form explanatory trajectories for each engineering unit. If an explainable trajectory is **monotonic**, it means that the associated sensor signal provides increasingly important information from the start to the end of life of the equipment. If the explainable trajectory is **trendable** it means that the importance of the sensor feature is consistent across the lifetimes of different engineering units. If the explainable trajectory is **prognosable**, the importance of the sensor feature at the end of life of the equipment is the same across different units. All these qualities are desirable in SHAP values trajectories of prognostics applications. The goal of this work is therefore to measure the effectiveness and the quality of the interpretability of a temporal prognostics sequence resorting to these three metrics.

In this paper, we investigate how the explanations produced by SHapley Additive exPlanations (SHAP), an eXplainable Artificial Intelligence (XAI) post-hoc method, can be classified according to monotonicity, trendability and prognosability. This exercise was carried out for three models of increasing complexity: Linear Regression (LR), Multi-Layer perceptron (MLP), and Echo State Network (ESN).

The results of this study indicate that SHAP values form monotonic trajectories that exhibit trendability and prognosability traits. These findings are important outcomes as they mean that we can trust SHAP values. The SHAP variables also have the added value of bringing additional information to the prognostics exercise, to be explored in the future. A close examination of SHAP values might help answer questions such as “why did the RUL estimation changed so much from one instant to the other” or “what was the most important factor for a given RUL forecast at a given time”.

We studied three models in this study of increasing complexity. We showed that the most complex models had the best performance in Remaining Useful Life (RUL) estimation. Concretely, the Linear Regression (LR) had worse performance than the Multi-Layer Perceptron (MLP) and the Echo State Network (ESN). However, the Linear Regression (LR) had the best SHAP monotonicity scores. This finding raises the question if it may be sufficient to use a simple model, in combination with SHAP, to identify the best predictors for a prognostics application. It will also be of relevance to study the influence of each of the pre-processing steps of prognostics (e.g. denoising, monotonicity constraints, feature selection, health stage classification) to the interpretability result of SHAP or other XAI method.

Theoretically, we can also apply SHAP to understand physics-based models. SHAP is a post-hoc method and can hence be applied to any kind of RUL forecasting method. It would be interesting to know if the same results can be observed for physics-based approaches in future work. It might also be interesting to study other metrics from the literature [81] and understand how they can be applied to SHAP values. Another direction of future work is to develop new indicators for SHAP in the context of prognostics. For example, it can be interesting to explore indicators of feature importance that, instead of analyzing features independently from each other, consider that prognostics features are interrelated. Also, feature importance can change over the life of the equipment. Deriving indicators capable of measuring feature importance over time is a topic yet to be fully explored.

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.

References

[1] K. Aas, M. Jullum, A. Løland, Explaining individual predictions when features are dependent: more accurate approximations to Shapley values, Artif. Intell. 298 (2021) 103502, https://doi.org/10.1016/j.artint.2021.103502.
[2] I. Ahern, A. Noack, L. Guzman-Nateras, D. Dou, B. Li, J. Huan, Normlime: a new feature importance metric for explaining deep neural networks, preprint, arXiv:1909.04200, 2019.
[3] M. Al-Shedivat, A. Dubey, E.P. Xing, Contextual explanation networks, preprint, arXiv:1705.10301, 2017.
[4] G. Alain, Y. Bengio, Understanding intermediate layers using linear classifier probes, preprint, arXiv:1610.01644, 2016.
[5] K. Alvehag, L. Soder, A reliability model for distribution systems incorporating seasonal variations in severe weather, IEEE Trans. Power Deliv. 26 (2011) 910–919, https://doi.org/10.1109/tpwrd.2010.2090363.
[6] N. Amruthnath, T. Gupta, Factor analysis in fault diagnostics using random forest, preprint, arXiv:1904.13366, 2019.
[7] R. Andrews, J. Diederich, A.B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowl.-Based Syst. 8 (1995) 373–389.
[8] A.B. Arrieta, N. Díaz-Rodríguez, J.D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115, https://doi.org/10.1016/j.inffus.2019.12.012.
[9] A.V. Assche, H. Blockeel, Seeing the forest through the trees: Learning a comprehensible model from an ensemble, in: Machine Learning: ECML 2007, Springer Berlin Heidelberg, 2007, pp. 418–429.
[10] S. Assilian, Artificial intelligence in control of real dynamic systems, Ph.D. thesis, 1974.
[11] T. Aven, E. Zio (Eds.), Knowledge in Risk Assessment and Management, John Wiley & Sons, Ltd., 2018.
[12] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (2015) e0130140, https://doi.org/10.1371/journal.pone.0130140.

[13] M.L. Baptista, E.M. Henriques, K. Goebel, More effective prognostics with elbow point detection and deep learning, Mech. Syst. Signal Process. 146 (2021) 106987, https://doi.org/10.1016/j.ymssp.2020.106987.
[14] N. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, Int. J. Comput. Intell. 2 (2005) 59–62.
[15] N.H. Barakat, A.P. Bradley, Rule extraction from support vector machines: a sequential covering approach, IEEE Trans. Knowl. Data Eng. 19 (2007) 729–741, https://doi.org/10.1109/tkde.2007.190610.
[16] P. Baraldi, G. Bonfanti, E. Zio, Differential evolution-based multi-objective optimization for the definition of a health indicator for fault diagnostics and prognostics, Mech. Syst. Signal Process. 102 (2018) 382–400.
[17] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Fault prognostic of bearings by using support vector data description, in: 2012 IEEE Conference on Prognostics and Health Management, IEEE, 2012.
[18] A. Bennetot, J.L. Laurent, R. Chatila, N. Díaz-Rodríguez, Towards explainable neural-symbolic visual reasoning, in: NeSy Workshop IJCAI, 2019.
[19] O. Biran, C. Cotton, Explanation and justification in machine learning: a survey, in: IJCAI-17 Workshop on Explainable AI (XAI), 2017.
[20] W. Brendel, M. Bethge, Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet, preprint, arXiv:1904.00760, 2019.
[21] F. Cannarile, P. Baraldi, M. Compare, D. Borghi, L. Capelli, M. Cocconcelli, A. Lahrache, E. Zio, An unsupervised clustering method for assessing the degradation state of cutting tools used in the packaging industry, in: Safety and Reliability – Theory and Applications, CRC Press, 2017.
[22] D. Card, M. Zhang, N.A. Smith, Deep weighted averaging classifiers, in: Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT'19, ACM Press, 2019.
[23] D.V. Carvalho, E.M. Pereira, J.S. Cardoso, Machine learning interpretability: a survey on methods and metrics, Electronics 8 (2019) 832, https://doi.org/10.3390/electronics8080832.
[24] L. Casas, A. Klimmek, N. Navab, V. Belagiannis, Adversarial signal denoising with encoder-decoder networks, preprint, arXiv:1812.08555, 2018.
[25] C. Chen, M. Pecht, Prognostics of lithium-ion batteries using model-based and data-driven methods, in: Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing), IEEE, 2012.
[26] C. Chen, T. Xu, G. Wang, B. Li, Railway turnout system RUL prediction based on feature fusion and genetic programming, Measurement 151 (2020) 107162, https://doi.org/10.1016/j.measurement.2019.107162.
[27] J. Chen, D. Zhou, C. Lyu, C. Lu, An approach to fault diagnosis for rotating machinery based on feature reconstruction with LCD and t-SNE, in: Vibroengineering PROCEDIA, vol. 11, 2017, pp. 40–45.
[28] R. Chimatapu, H. Hagras, A. Starkey, G. Owusu, Explainable AI and fuzzy logic systems, in: Theory and Practice of Natural Computing, Springer International Publishing, 2018, pp. 3–20.
[29] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder–decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, 2014.
[30] N. Chouikhi, B. Ammar, N. Rokbani, A.M. Alimi, PSO-based analysis of echo state network parameters for time series forecasting, Appl. Soft Comput. 55 (2017) 211–225.
[31] J. Coble, An Automated Approach for Fusing Data Sources to Identify Optimal Prognostic Parameters, Ph.D. thesis, Dissertation, University of Tennessee Knoxville, TN, 2010.
[32] J. Coble, J.W. Hines, Identifying optimal prognostic parameters from data: a genetic algorithms approach, in: Annual Conference of the Prognostics and Health Management Society, 2009.
[33] J. Coble, J.W. Hines, Identifying suitable degradation parameters for individual-based prognostics, in: Diagnostics and Prognostics of Engineering Systems, IGI Global, 2013, pp. 135–150.
[34] A. Creswell, A.A. Bharath, Denoising adversarial autoencoders, IEEE Trans. Neural Netw. Learn. Syst. 30 (2018) 968–984.
[35] M. Daigle, B. Saha, K. Goebel, A comparison of filter-based approaches for model-based prognostics, in: 2012 IEEE Aerospace Conference, IEEE, 2012.
[36] J. Debayle, N. Hatami, Y. Gavet, Classification of time-series images using deep convolutional neural networks, in: J. Zhou, P. Radeva, D. Nikolaev, A. Verikas (Eds.), Tenth International Conference on Machine Vision (ICMV 2017), SPIE, 2018.
[37] H. Dinsdag, H. Jaeger, A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF, and the echo state network, Technical Report, German National Research Center for Information Technology, 2001.
[38] M. El-Koujok, R. Gouriveau, N. Zerhouni, A neuro-fuzzy self built system for prognostics: a way to ensure good prediction accuracy by balancing complexity and generalization, in: 2010 Prognostics and System Health Management Conference, IEEE, 2010.
[39] J.M. Fellous, G. Sapiro, A. Rossi, H. Mayberg, M. Ferrante, Explainable artificial intelligence for neuroscience: behavioral neurostimulation, Front. Neurosci. 13 (2019), https://doi.org/10.3389/fnins.2019.01346.
[40] O. Fink, E. Zio, U. Weidmann, Predicting time series of railway speed restrictions with time-dependent machine learning techniques, Expert Syst. Appl. 40 (2013) 6033–6040.
[41] D.K. Frederick, J.A. DeCastro, J.S. Litt, User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), Technical Report, 2007.
[42] A.A. Freitas, Comprehensible classification models, ACM SIGKDD Explor. Newsl. 15 (2014) 1–10, https://doi.org/10.1145/2594473.2594475.
[43] S.A. Friedler, C.D. Roy, C. Scheidegger, D. Slack, Assessing the local interpretability of machine learning models, preprint, arXiv:1902.03501, 2019.
[44] K. Goebel, M.J. Daigle, A. Saxena, I. Roychoudhury, S. Sankararaman, J.R. Celaya, Prognostics: the science of making predictions, 2017.
[45] K. Goebel, B. Saha, A. Saxena, A comparison of three data-driven techniques for prognostics, in: 62nd Meeting of the Society for Machinery Failure Prevention Technology (MFPT), 2008, pp. 119–131.
[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[47] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Mag. 38 (2017) 50–57, https://doi.org/10.1609/aimag.v38i3.2741.
[48] B. Goyal, A. Dogra, S. Agrawal, B. Sohi, A. Sharma, Image denoising review: from classical to state-of-the-art approaches, Inf. Fusion 55 (2020) 220–244, https://doi.org/10.1016/j.inffus.2019.09.003.
[49] J. Gu, V. Tresp, Saliency methods for explaining adversarial attacks, preprint, arXiv:1908.08413, 2019.
[50] J. Gu, V. Tresp, Semantics for global and local interpretation of deep neural networks, preprint, arXiv:1910.09085, 2019.
[51] J. Gu, Y. Yang, V. Tresp, Understanding individual decisions of CNNs via contrastive backpropagation, in: Computer Vision – ACCV 2018, Springer International Publishing, 2019, pp. 119–134.
[52] D. Gunning, D. Aha, DARPA's explainable artificial intelligence (XAI) program, AI Mag. 40 (2019) 44–58, https://doi.org/10.1609/aimag.v40i2.2850.
[53] B.D. Haig, Exploratory Data Analysis, Oxford University Press, 2018.
[54] P. Hall, N. Gill, M. Kurka, W. Phan, Machine learning interpretability with H2O driverless AI, H2O.ai, http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf, 2017.
[55] D. He, N. Hu, M. Wang, Study on real-time fault injection and simulation of mechanic-electronic-hydraulic control system based on AMESim and LabVIEW, in: 2014 Prognostics and System Health Management Conference (PHM-2014 Hunan), IEEE, 2014.
[56] W. He, Y. He, B. Li, C. Zhang, Analog circuit fault diagnosis via joint cross-wavelet singular entropy and parametric t-SNE, Entropy 20 (2018) 604, https://doi.org/10.3390/e20080604.
[57] W. He, Q. Miao, M. Azarian, M. Pecht, Health monitoring of cooling fan bearings based on wavelet filter, Mech. Syst. Signal Process. 64 (2015) 149–161.

[58] M. Hind, D. Wei, M. Campbell, N.C.F. Codella, A. Dhurandhar, A. Mojsilović, K.N. Ramamurthy, K.R. Varshney, TED: teaching AI to explain its decisions, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, ACM, 2019.
[59] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[60] S.R. Hong, J. Hullman, E. Bertini, Human factors in model interpretability: industry practices, challenges, and needs, in: Proceedings of the ACM on Human-Computer Interaction, vol. 4, 2020, pp. 1–26.
[61] Y. Hu, T. Palmé, O. Fink, Deep health indicator extraction: a method based on auto-encoders and extreme learning machines, in: PHM 2016, Denver, USA, 3–6 October, 2016, PMH Society, 2016, pp. 446–452.
[62] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, Y. Chang, Graphlime: local interpretable model explanations for graph neural networks, preprint, arXiv:2001.06216, 2020.
[63] S. Jacklin, J. Schumann, P. Gupta, M. Richard, K. Guenther, F. Soares, Development of advanced verification and validation procedures and tools for the certification of learning systems in aerospace applications, in: Infotech@Aerospace, American Institute of Aeronautics and Astronautics, 2005.
[64] H. Jaeger, The Echo State Approach to Analyzing and Training Recurrent Networks, Technical Report, German National Research Center for Information Technology, 2001.
[65] A. Jalali, A. Schindler, B. Haslhofer, A. Rauber, Machine learning interpretability techniques for outage prediction: a comparative study, in: PHM Society European Conference, 2020, p. 10.
[66] K. Javed, R. Gouriveau, N. Zerhouni, P. Nectoux, Enabling health monitoring approach based on vibration data for accurate prognostics, IEEE Trans. Ind. Electron. 62 (2015) 647–656, https://doi.org/10.1109/tie.2014.2327917.
[67] M.J. Johnson, D.K. Duvenaud, A. Wiltschko, R.P. Adams, S.R. Datta, Composing graphical models with neural networks for structured representations and fast inference, in: Advances in Neural Information Processing Systems, 2016, pp. 2946–2954.
[68] M. Karl, M. Soelch, J. Bayer, P. Van der Smagt, Deep variational Bayes filters: unsupervised learning of state space models from raw data, preprint, arXiv:1605.06432, 2016.
[69] B.M. Keneni, D. Kaur, A.A. Bataineh, V.K. Devabhaktuni, A.Y. Javaid, J.D. Zaientz, R.P. Marinier, Evolving rule-based explainable artificial intelligence for unmanned aerial vehicles, IEEE Access 7 (2019) 17001–17016, https://doi.org/10.1109/access.2019.2893141.
[70] B. Kim, C. Rudin, J.A. Shah, The Bayesian case model: a generative approach for case-based reasoning and prototype classification, in: Advances in Neural Information Processing Systems, 2014, pp. 1952–1960.
[71] N.H. Kim, D. An, J.H. Choi, Study on attributes of prognostics models, in: Prognostics and Health Management of Engineering Systems: An Introduction, Springer International Publishing, 2016, pp. 243–280.
[72] R.G. Krishnan, U. Shalit, D. Sontag, Deep Kalman filters, preprint, arXiv:1511.05121, 2015.
[73] J. Kukačka, V. Golkov, D. Cremers, Regularization for deep learning: a taxonomy, preprint, arXiv:1710.10686, 2017.
[74] C.S. Kulkarni, M. Daigle, G.S. Gorospe, K. Goebel, Application of model-based prognostics framework to pneumatic valves on cryogenic testbed, in: AIAA Infotech @ Aerospace, American Institute of Aeronautics and Astronautics, 2015.
[75] P. Kundu, A.K. Darpe, M.S. Kulkarni, A correlation coefficient based vibration indicator for detecting natural pitting progression in spur gears, Mech. Syst. Signal Process. 129 (2019) 741–763, https://doi.org/10.1016/j.ymssp.2019.04.058.
[76] H. Kuwajima, M. Tanaka, M. Okutomi, Improving transparency of deep neural inference process, Prog. Artif. Intell. 8 (2019) 273–285, https://doi.org/10.1007/s13748-019-00179-x.
[77] P. Lall, T. Thomas, PCA and ICA based prognostic health monitoring of electronic assemblies subjected to simultaneous temperature-vibration loads, in: ASME 2017 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems, American Society of Mechanical Engineers, 2017.
[78] F. Lasheras, P. Nieto, F. de Cos Juez, R. Bayón, V. Suárez, A hybrid PCA-CART-MARS-based prognostic approach of the remaining useful life for aircraft engines, Sensors 15 (2015) 7062–7083, https://doi.org/10.3390/s150307062.
[79] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.
[80] N. Lee, M.H. Azarian, M.G. Pecht, An explainable deep learning-based prognostic model for rotating machinery, preprint, arXiv:2004.13608, 2020.
[81] 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, https://doi.org/10.1016/j.ymssp.2017.11.016.
[82] X. Li, Remaining useful life prediction of bearings using fuzzy multimodal extreme learning regression, in: 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), IEEE, 2017.
[83] X. Li, Q. Ding, J.Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliab. Eng. Syst. Saf. 172 (2018) 1–11, https://doi.org/10.1016/j.ress.2017.11.021.
[84] Z.C. Lipton, The mythos of model interpretability, Commun. ACM 61 (2018) 36–43, https://doi.org/10.1145/3233231.
[85] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017, pp. 4765–4774, http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.
[86] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[87] M. Madhikermi, A.K. Malhi, K. Främling, Explainable artificial intelligence based heat recycler fault detection in air handling unit, in: Lecture Notes in Computer Science, Springer International Publishing, 2019, pp. 110–125.
[88] E. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Mach. Stud. 7 (1975) 1–13, https://doi.org/10.1016/s0020-7373(75)80002-2.
[89] D. Martens, B. Baesens, T.V. Gestel, J. Vanthienen, Comprehensible credit scoring models using rule extraction from support vector machines, SSRN Electron. J. (2006), https://doi.org/10.2139/ssrn.878283.
[90] D.A. Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 7775–7784.
[91] C. Mencar, Interpretability of fuzzy systems, in: Fuzzy Logic and Applications, Springer International Publishing, 2013, pp. 22–35.
[92] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. 267 (2019) 1–38, https://doi.org/10.1016/j.artint.2018.07.007.
[93] Y. Ming, H. Qu, E. Bertini, RuleMatrix: visualizing and understanding classifiers with rules, IEEE Trans. Vis. Comput. Graph. 25 (2019) 342–352, https://doi.org/10.1109/tvcg.2018.2864812.
[94] M. Mishra, J. Martinsson, M. Rantatalo, K. Goebel, Bayesian hierarchical model-based prognostics for lithium-ion batteries, Reliab. Eng. Syst. Saf. 172 (2018) 25–35, https://doi.org/10.1016/j.ress.2017.11.020.
[95] S. Morando, S. Jemei, R. Gouriveau, N. Zerhouni, D. Hissel, Fuel Cells prognostics using echo state network, in: IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2013.
[96] A. Mosallam, K. Medjaher, N. Zerhouni, Nonparametric time series modelling for industrial prognostics and health management, Int. J. Adv. Manuf. Technol. 69 (2013) 1685–1699, https://doi.org/10.1007/s00170-013-5065-z.
[97] M. Naseri, P. Baraldi, M. Compare, E. Zio, Availability assessment of oil and gas processing plants operating under dynamic arctic weather conditions, Reliab. Eng. Syst. Saf. 152 (2016) 66–82, https://doi.org/10.1016/j.ress.2016.03.004.

[98] S.A. Niknam, J. Kobza, J.W. Hines, Techniques of trend analysis in degradation-based prognostics, Int. J. Adv. Manuf. Technol. 88 (2016) 2429–2441, https://doi.org/10.1007/s00170-016-8909-5.
[99] N. Papernot, P. McDaniel, Deep k-nearest neighbors: towards confident, interpretable and robust deep learning, preprint, arXiv:1803.04765, 2018.
[100] L. Peng, Y. Huang, Survival analysis with temporal covariate effects, Biometrika 94 (2007) 719–733, https://doi.org/10.1093/biomet/asm058.
[101] Y. Peng, H. Wang, J. Wang, D. Liu, X. Peng, A modified echo state network based remaining useful life estimation approach, in: 2012 IEEE Conference on Prognostics and Health Management, IEEE, 2012.
[102] A. Popov, W. Fink, A. Hess, PHM for astronauts – a new application, in: Annual Conference of the Prognostics and Health Management Society, 2013, pp. 566–572.
[103] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M.P. Reyes, M.L. Shyu, S.C. Chen, S.S. Iyengar, A survey on deep learning, ACM Comput. Surv. 51 (2019) 1–36, https://doi.org/10.1145/3234150.
[104] G. Qiu, Y. Gu, J. Chen, Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model, Measurement 150 (2020) 107097.
[105] J. Rabold, H. Deininger, M. Siebers, U. Schmid, Enriching visual with verbal explanations for relational concepts – combining LIME with aleph, in: Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, 2020, pp. 180–192.
[106] E. Ramasso, A. Saxena, Performance benchmarking and analysis of prognostic methods for cmapss datasets, 2014.
[107] M.T. Ribeiro, S. Singh, C. Guestrin, Model-agnostic interpretability of machine learning, preprint, arXiv:1606.05386, 2016, 91–95.
[108] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016.
[109] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, S. Poll, Ensemble of optimized echo state networks for remaining useful life prediction, Neurocomputing 281 (2018) 121–138, https://doi.org/10.1016/j.neucom.2017.11.062.
[110] M. Rigamonti, P. Baraldi, E. Zio, et al., Echo state network for the remaining useful life prediction of a turbofan engine, in: European Conference of the Prognostics and Health Management Society (PHME), 2016, pp. 255–270.
[111] R. Rocchetta, Y. Li, E. Zio, Risk assessment and risk-cost optimization of distributed power generation systems considering extreme weather conditions, Reliab. Eng. Syst. Saf. 136 (2015) 47–61, https://doi.org/10.1016/j.ress.2014.11.013.
[112] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (2017) e1602614, https://doi.org/10.1126/sciadv.1602614.
[113] L. Saidi, J.B. Ali, E. Bechhoefer, M. Benbouzid, Wind turbine high-speed shaft bearings health prognosis through a spectral kurtosis-derived indices and SVR, Appl. Acoust. 120 (2017) 1–8.
[114] S.B. Salah, I. Fliss, M. Tagina, Echo state network and particle swarm optimization for prognostics of a complex system, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), IEEE, 2017.
[115] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, Metrics for evaluating performance of prognostic techniques, in: International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–17.
[116] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Metrics for offline evaluation of prognostic performance, Intern. J. Prog. Health Manag. 1 (2010) 4–23.
[117] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: International Conference on Prognostics and Health Management, IEEE, 2008, pp. 1–9.
[118] P. Schober, C. Boer, L.A. Schwarte, Correlation coefficients, Anesth. Analg. 126 (2018) 1763–1768, https://doi.org/10.1213/ane.0000000000002864.
[119] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017.
[120] S.M. Shankaranarayana, D. Runje, ALIME: autoencoder based approach for local interpretability, in: Intelligent Data Engineering and Automated Learning – IDEAL 2019, Springer International Publishing, 2019, pp. 454–463.
[121] L.S. Shapley, A value for n-person games, Contrib. Theory Games 2 (1953) 307–317.
[122] D. She, M. Jia, Wear indicator construction of rolling bearings based on multi-channel deep convolutional neural network with exponentially decaying learning rate, Measurement 135 (2019) 368–375, https://doi.org/10.1016/j.measurement.2018.11.040.
[123] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: 34th International Conference on Machine Learning, 2017, pp. 3145–3153, JMLR.org.
[124] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, preprint, arXiv:1312.6034, 2013.
[125] M. Soleimani, F. Campean, D. Neagu, Diagnostics and prognostics for complex systems: a review of methods and challenges, Qual. Reliab. Eng. Int. (2021), https://doi.org/10.1002/qre.2947.
[126] W. Song, L. Wen, L. Gao, X. Li, Unsupervised fault diagnosis method based on iterative multi-manifold spectral clustering, in: IET Collaborative Intelligent Manufacturing, vol. 1, 2019, pp. 48–55.
[127] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net, preprint, arXiv:1412.6806, 2014.
[128] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowl. Inf. Syst. 41 (2013) 647–665, https://doi.org/10.1007/s10115-013-0679-x.
[129] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 3319–3328, JMLR.org.
[130] D.D. Susilo, A. Widodo, T. Prahasto, M. Nizam, Remaining useful life estimation of the motor shaft based on feature importance and state-space model, in: Proceedings of the 6th International Conference and Exhibition on Sustainable Energy and Advanced Materials, Springer, Singapore, 2020, pp. 675–688.
[131] E. Tjoa, C. Guan, A survey on explainable artificial intelligence (XAI): toward medical XAI, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–21, https://doi.org/10.1109/tnnls.2020.3027314.
[132] E. Union, Date subjects' rights, in: EU General Data Protection Regulation (GDPR), third edition, IT Governance Publishing, 2019, pp. 62–77.
[133] Z. Viharos, K. Kis, Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement, Measurement 67 (2015) 126–136, https://doi.org/10.1016/j.measurement.2015.02.001.
[134] J. Wang, Z. Li, X. Li, Y. Zhao, A novel SOH prediction framework for the lithium-ion battery using echo state network, in: International Conference on Neural Information Processing, Springer, 2014, pp. 438–445.
[135] C. Wilkinson, J. Lynch, R. Bharadwaj, K. Woodham, Verification of adaptive systems, Federal Aviation Administration, DOT/FAA/TC-16/4, 2016.
[136] Y. Xie, D. Feng, F. Wang, X. Tang, J. Han, X. Zhang, DFPE: explaining predictive models for disk failure prediction, in: 2019 35th Symposium on Mass Storage Systems and Technologies (MSST), IEEE, 2019.
[137] C. Yang, J. Qiao, Z. Ahmad, K. Nie, L. Wang, Online sequential echo state network with sparse RLS algorithm for time series prediction, Neural Netw. (2019).
[138] L. Yongxiang, S. Jianming, W. Gong, L. Xiaodong, A data-driven prognostics approach for RUL based on principle component and instance learning, in: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2016.
[139] L. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353, https://doi.org/10.1016/s0019-9958(65)90241-x.

[140] M.R. Zafar, N.M. Khan, DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems, preprint, arXiv:1906.10263, 2019.
[141] S. Zeldam, Automated failure diagnosis in aviation maintenance using explainable artificial intelligence (XAI), Master's thesis, University of Twente, 2018.
[142] J. Zerilli, A. Knott, J. Maclaurin, C. Gavaghan, Transparency in algorithmic and human decision-making: is there a double standard?, Philos. Technol. 32 (2018) 661–683, https://doi.org/10.1007/s13347-018-0330-6.
[143] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, preprint, arXiv:1611.03530, 2016.
[144] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, M. Wei, A review on deep learning applications in prognostics and health management, IEEE Access 7 (2019) 162415–162438, https://doi.org/10.1109/access.2019.2950985.
[145] Q. Zhang, Y.N. Wu, S.C. Zhu, Interpretable convolutional neural networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018.
[146] X. Zhang, R. Xu, C. Kwan, S. Liang, Q. Xie, L. Haynes, An integrated approach to bearing fault diagnostics and prognostics, in: Proceedings of the 2005 American Control Conference, IEEE, 2005.
[147] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P.H. Torr, Conditional random fields as recurrent neural networks, in: IEEE international Conference on Computer Vision, 2015, pp. 1529–1537.
[148] Z.H. Zhou, Y. Jiang, S.F. Chen, Extracting symbolic rules from trained neural network ensembles, AI Commun. 16 (2003) 3–15.
[149] E. Zio, Prognostics and health management of industrial equipment, in: Diagnostics and Prognostics of Engineering Systems, IGI Global, 2013, pp. 333–356.
[150] E. Zio, F.D. Maio, A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system, Reliab. Eng. Syst. Saf. 95 (2010) 49–57, https://doi.org/10.1016/j.ress.2009.08.001.
[151] E. Zio, F.D. Maio, M. Stasi, A data-driven approach for predicting failure scenarios in nuclear systems, Ann. Nucl. Energy 37 (2010) 482–491, https://doi.org/10.1016/j.anucene.2010.01.017.