Classification prognostics approaches in aviation

Baptista, Marcia L.; Henriques, Elsa M.P.; Prendinger, Helmut

DOI
10.1016/j.measurement.2021.109756

Publication date
2021

Document Version
Final published version

Published in
Measurement: Journal of the International Measurement Confederation

Contents lists available at ScienceDirect

Measurement

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

Classification prognostics approaches in aviation

Marcia L. Baptista [a,*], Elsa M.P. Henriques [b], Helmut Prendinger [c]

[a] *Section of Air Transport and Operations, Delft University of Technology, Netherlands*
[b] *Instituto Superior Tecnico, Universidade de Lisboa, Lisbon, 1049-001, Portugal*
[c] *National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Predictive maintenance
Prognostics
Classification
Deep learning
Recurrent neural networks
Aeronautics
Case study

ABSTRACT

Traditionally, prognostics approaches to predictive maintenance have focused on estimating the remaining useful life of the equipment. However, from an industrial point of view, the goal is often not to predict the residual life but to determine the need for a maintenance action at a given time window. This approach allows us to frame the data-driven prognostics problem as a binary classification task rather than a regression one. To address this problem, we propose in this paper to explore the relative strengths and limitations of a set of classifier approaches such as random forests, support vector machines, nearest neighbors, and deep learning techniques. We evaluate the models using metrics such as sensitivity, specificity, accuracy, receiver operating characteristic curve, and F-score. This work's novelty lies in adopting a modeling approach with a natural probabilistic interpretation of the prognostics exercise. The comparison of an extensive range of classifier models is performed on two real-world datasets from the aeronautics sector. Results indicate that deep learning classifier methods are well suited for this kind of prognostics and can outperform by a significant margin the traditional classification techniques. Importantly, the proposed modeling approach aims to generate an alternative prognostics representation that goes in line with the expectations of aeronautical engineers.

1. Introduction

There have been several definitions of prognostics [1], each alluding to their specific industrial context. For instance, in the case of unmanned aerial vehicles (UAVs), the definition of prognostics relates to the estimation of the end-of-charge or end-of-life of the equipment based on the state of health (SOH) assessment [2]. Seemingly, in the field of structural health monitoring, prognostics relates to damage progression, such as crack growth in civil infrastructure [3]. In the aeronautics sector, prognostics deals with health monitoring information to predict the onset of failures of the many components and subsystems of avionics and aerospace systems [4]. Often, and in many of these fields, prognostics is used intertwined with the idea of Remaining Useful Life (RUL) [5–8]. The concept of RUL is defined as the usage life of a system or component, measured in usage units (e.g., calendar time, usage time, number of cycles), at a given instant in time. Sankararaman and Goebel [9] provide a formal definition of the concepts of RUL and end-of-life.

Importantly, the RUL and the remaining Time To Failure (TTF) are not necessarily the same concepts. These concepts depend on how the end-of-life is expressed [9]. For example, in aviation, the end-of-life may be the point where the system reliability has degraded from "six-nines" to "three-nines" and requires a preventive maintenance action. In contrast, the remaining TTF refers to when the system no longer meets its functional specifications or design standards.

Notwithstanding the importance of RUL prediction, in the industry, it is often more relevant to perform *fault detection*, i.e., to identify if a fault or anomaly is going to happen in the next few days than to perform RUL estimation. Fault detection differs from RUL estimation in that it involves the setting of a time window and the binary classification of failure within that moving window. The goal of these two approaches is fundamentally the same: to signal the end-of-life before it occurs [10] and they are both in the field of prognostics. We concur with [11] that the definition of prognostics is essentially the science of making predictions about engineering systems.

The field of RUL estimation is well established [12,13] and there are many evaluation instruments [14,15]. The field could, however, benefit from the development of methodologies to understand the impact of RUL estimation on the maintenance processes [16]. Adding industrial interpretability to RUL estimates could significantly improve their understanding and overall acceptance. In this paper, we aim to address this issue by exploring how to perform fault detection.

This work aims to assess the performance of classification methods using traditional metrics from machine learning and to discuss the merits and limitations of these methods. Enhanced interpretability and

[*] Corresponding author.
E-mail address: m.lbaptista@tudelft.nl (M.L. Baptista).

https://doi.org/10.1016/j.measurement.2021.109756
Received 8 April 2021; Received in revised form 23 May 2021; Accepted 11 June 2021
Available online 17 June 2021

more natural handling of the uncertainty of the estimates are the two main advantages of the proposed approaches. We argue that the categorical label proposed in the paper is an interpretable outcome that responds to the needs of most industrial maintenance cases in aeronautics. We claim that by generating probabilities instead of deterministic RUL estimates, the proposed models can better handle the uncertainty of the prognostics exercise.

In this work, we frame the problem of prognostics as a classification task in which the goal is to estimate, at each prediction step, the probability of the need for a maintenance action in the next time window of length d using the discrete sequence of observations up to the current time. The goal is to compute at each time the probability distribution of a fault event in a future prediction window. To achieve this goal, we compare a set of representative classifier models.

We take a data-driven approach to the prognostics exercise. The models proposed do not rely on an explicit representation of physical phenomena but capture damage patterns from historical data using advanced statistical and machine learning methods. Data-driven methods such as the ones used (Naive Bayes, random forests, support vector machines, neural networks, etc.) have the advantage of adapting to different problems with more ease than physics-of-failure models. These latter models are built on physics equations that describe the equipment behavior and require adjusting the equations for each new case study. This dependence on a physics-of-failure description is not necessary for machine learning models. They optimize the relationship between sensor data and the state of the component or system using data and statistics. These approaches “learn” a specific task without having to be coded explicitly for it.

The tested data-driven approaches include the models of K-Nearest Neighbors (KNN), Gaussian Support Vector Machines (GSVM), Random Forests (RF), Multi-Layer Perceptron (MLP), Gaussian Naive Bayes (NB), and advanced models relying on Deep Recurrent Neural Networks (RNNs). We also compared these approaches against the baseline models of the frequent, uniform, and stratified classifiers.

Grid search and evolutionary search are used to optimize the parameters of the algorithms. We also use signal processing and feature engineering techniques such as Principle Component Analysis (PCA). In addition to the classical classifier methods, we explore three deep learning RNNs: the standard network [17,18], the state-of-the-art Long-Short Term Memory (LSTM) network [19] and the more recent Gated Recurrent Unit (GRU) network [20].

The novelty of our work lies in the use of the classification approach to the problem of prognostics, with a comprehensive comparison of models on two real-world industrial case studies, both involving large-scale datasets from the aeronautics sector: (1) one describing the damage progression of a critical component from a modern gas turbine engine and (2) another describing the reliability of the engine. Note the difficulty and the value of obtaining results on real-world datasets from the aeronautical sector. It should also be noted that, from an industrial point of view, the ability to anticipate the occurrence of a failure in a future time window allows for more efficient planning of maintenance actions.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the approach and classification models. Section 4 presents the two case studies. In Section 5 we provide a detailed treatment of the two real-world datasets, which are used to establish the validity of the results. Finally, conclusions and future work are addressed in Section 6.

2. Background

This section reviews previous work and starts in Section 2.1 with a general overview of remaining useful life (RUL) prognostics. Section 2.2 reviews works that have also used classification methods to estimate system health degradation. Along with the text, we introduce some formal notation and essential concepts.

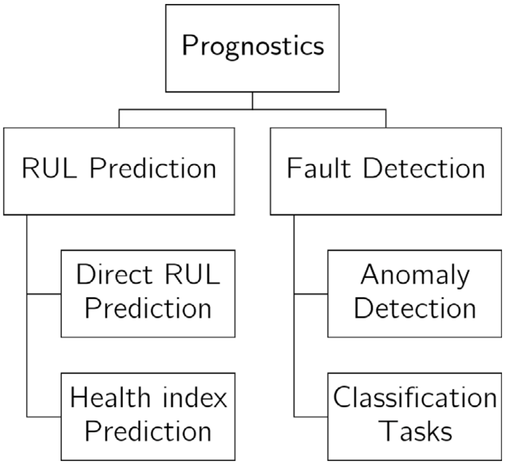


Fig. 1. Taxonomy of prognostics approaches. We distinguish between prognostics based on remaining useful life prediction and fault detection. Remaining useful life prognostics focuses on estimating remaining useful life given health monitoring information (features and conditions) up to the prediction time. Fault detection consists of detecting a prognostics event. It is related to disciplines such as anomaly detection and machine learning classification.

2.1. Remaining useful life estimation

Remaining useful life (RUL) models can be classified into model-based, data-driven, and hybrid methods [16]. Model-based methods exploit domain knowledge of the system and its failure mechanisms [6, 21]. Data-driven approaches, rather than relying on system and domain knowledge to predict the RUL, use large amounts of sensor data to train algorithms that aim at capturing degradation trends [22]. Hybrid modeling combines data-driven and physics of failure techniques.

In data-driven RUL estimation, the problem of capturing the onset of failure can be seen from two major perspectives (see Fig. 1). The first focuses on direct RUL estimation. This form of prognostics is particularly efficient when enough run-to-failure data be used can extrapolate the RUL estimates from the set of sensor readings. The second perspective consists of the development of a health index. By health index, we mean a health indicator describing the system’s current condition [14]. In this approach, the RUL is predicted in two steps: (1) first from sensor signals to health index, and then (2) mapping the index to RUL. We review a few works on each of these topics to establish a basis for comparing this work and previous contributions.

In data-driven direct RUL estimation models, algorithms learn the relation between the sensor data and the end-of-life point directly. There are a few published works [1,23,24] that follow this approach. The drawback of using these methods is that in the face of missing data, the extrapolation mechanisms of the algorithms may lead to significant performance errors [25]. Several transfer learning strategies have been used to improve the prediction accuracy of the RUL, such as [26,27]. Also, to address the lack of run-to-failure data, simulation data from a dynamic model has been shown to help improve the modeling accuracy [28].

The problem of predicting a health trajectory proceeds in two steps. First, estimates are made on the health monitoring domain. The trajectory may take a non-linear form to better capture damage evolution over time, with the end-of-life being reached when the predicted trajectory goes beyond a predefined threshold. This methodology has the advantage of being easier to interpret. Health index prognostics is a popular approach in the data-driven community with works of note such as in [29–31].

2.2. Fault detection

In the context of prognostics, fault can be defined as any change in the nominal operation of a system that makes it unable to perform its function satisfactorily or unavailable to meet its functional requirements [32]. The concept of fault is often used interchangeably with the idea of failure. In this paper, we interchangeably use both concepts.

Some authors refer to fault detection as determining the source of the problems after a fault has been detected [33]. This process occurs when there is the need to diagnose and isolate the faults of the system. In this paper, we refer to fault detection as the predictive exercise before fault isolation that involves using anomaly detection or machine learning classification techniques to indicate the eminence of failure. The difference is that in fault diagnostics, detection concerns which fault has occurred, while in prognostics, fault detection concerns if a fault is set to happen.

In general, anomaly detection focuses on discovering patterns in the data that do not conform to expected normal behavior [34,35]. In prognostics, anomaly detection can be defined as the process of developing evidence to reject the null hypothesis that the component is nominal. Early detection of anomalies can translate to significant future actionable insights. Despite the advantages of this approach, the nature of the data, availability of labeled data, and the type of anomalies to be detected can make this approach challenging.

An important contribution to anomaly detection in prognostics is the work in [36], which proposes a neural net anomaly detector to predict faults and other off-nominal operations that were not anticipated nor found before. An application in the military aviation sector is used to validate the approach. Other works of note include the work of Ellefsen et al. [37] who used spectral anomaly detection combined with a variational autoencoder to detect faults in autonomous ferries. Jin et al. [38] proposed a moving window-based statistical test to detect anomalies in bearing data. Zhang et al. [39] compared different statistical methods to isolate rare events that may affect how the equipment condition evolves in time.

Classification models are not standard practice in prognostics, but these could significantly help tackle important industrial problems. Several contributions are of mention. Ramasso et al. [40] propose an approach based on case-based reasoning and belief functions to predict an observation trajectory and classify the trajectory onto several discrete degradation modes. Importantly, the work required the training of a classification model to distinguish among the different states. These states were then used to predict the RUL of the equipment.

Javed [41] proposed using classifier techniques to assess the health state of engineering systems. The goal here is to cluster data into homogeneous groups in which intragroup similarity is maximized, and intergroup similarity is minimized, resulting in compact and separate clusters. Based on distance metrics, data are assigned labels corresponding to the closest cluster's discrete state, and the RUL is estimated when the transition from degrading to faulty condition occurs. This approach is different from ours in that we aim at classifying fault and no-fault states.

Another work of note is the one of Patil et al. [42]. The authors propose a two-stage prognostics model in which a gross RUL estimation is made on the first stage using a classification technique. In the second stage, a regression technique is used to estimate a more accurate RUL prediction. Other contributions that follow the same modeling approach are in [43,44].

Note the difference between the previous approaches and the proposed work. Our approach is based exclusively on classification methods from machine learning. We address how to map sensor data into two classes that can support maintenance planning and scheduling decisions. Here, the degradation processes are classified using a binary label: the label is set to one if a maintenance action is required in the next time window of size d, or set to zero, otherwise. The authors believe that this study is one of the most extensive and comprehensive efforts in the field that uses real-world data obtained from two different original equipment manufacturers in aeronautics. The connection with the industrial mindset is another significant and relevant contribution of this paper to prognostics and health management, and aeronautics.

3. Methodology

The problem of fault detection consists of detecting a future health event of a system or component. In this section, we formally define the problem of fault detection. We then describe a general data-driven approach to prognostics. We briefly review the selected models and describe the performance evaluation metrics.

3.1. Problem formulation

We assume the fault detection system can be described by:

$$\boldsymbol{y}(t) = f\left(\boldsymbol{x}(t_P), t_{P:P+d}\right) = \begin{cases} 0, & \text{if fault} \in [t_P, t_{P+d}] \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

where $t_P \in \mathbb{R}$ is the continuous time variable with P being the current time index, $\boldsymbol{x}(t_P) \in \mathbb{R}^{n_x}$ is the feature input vector, $t_{P:P+d}$ is the future prediction interval of size d, $f(.)$ is the classification function, $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ is the binary outcome vector. This representation considers a general classification model with no restrictions on the functional form of the function $f(.)$.

Prognostics is framed as a classification task in which the goal is to estimate at each prediction time t_P the probability of a health event in the next time interval using the discrete sequence of observations up to time t_P, denoted as $x_{0:t_P}$. The interest is not directly on the end-of-life of the equipment, but on the need for a corrective intervention in the prediction interval $t_{P:P+d}$. We formally express the variable of interest as the outcome of the function $f(.)$ where $f(.)$ is function of the system state $x(t_P)$, and interval of prediction $t_{P:P+d}$. Concretely, the function f estimates the need for a maintenance action, when $f\left(\boldsymbol{x}(t_P), t_{P:P+d}\right) = 1$ and zero otherwise.

Since many sources of uncertainty can influence prediction, rather than a zero or one output, the fault detection system computes at each time t_P, the probability distribution of the random variable X = *Fault in the interval* $[t_P, T_{T+d}]$. The goal is therefore to compute, at time t_P, a probabilistic outcome from function $f(.)$. The resulting probabilities are transformed into binary states (close/far from the end-of-life) with predefined thresholds. These thresholds are determined such that the generated receiver operating curve (ROC) in respect to the window size (d) parameter has the maximum area under the curve. When above the threshold, predictions are considered to belong to class 1 (close to end-of-life), or else, they belong to class 0 (far from the end-of-life). The higher the probability, the more likely it is that the failure event will occur soon. The lower the probability, the more likely it is that the equipment is far from its end-of-life.

3.2. Prognostics architecture

We follow a data-driven approach, wherein machine learning methods are used to extract actionable insights from data. The architecture consists of two sequential phases, as illustrated in Fig. 2:

- **Offline phase:** in the offline or training phase the goal is to construct the model f from data that can capture the evolution of damage over time. The relation between the extracted information $\boldsymbol{x}(t)$ and the classification labels $\boldsymbol{y}(t)$ is characterized in order to predict future fault events.
- **Online phase:** in the online or testing phase the model f outputs the probabilities of the binary labels $\boldsymbol{y}(t_{P:t_{P+d}})$ for each set of input features $\boldsymbol{x}(t_P)$.

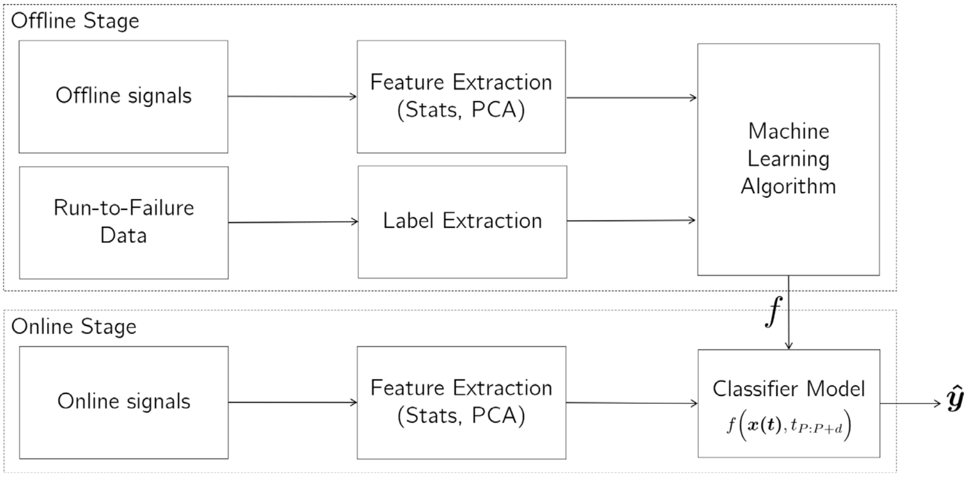

Offline Stage
Offline signals
Feature Extraction (Stats, PCA)
Machine Learning Algorithm
Run-to-Failure Data
Label Extraction
f
Online Stage
Online signals
Feature Extraction (Stats, PCA)
Classifier Model $f\big(\boldsymbol{x(t)}, t_{P:P+d}\big)$
$\hat{\boldsymbol{y}}$

Fig. 2. Fault detection architecture. We use a supervised fault detection architecture that proceeds in two steps. In the first step, offline signals and past run-to-failure data are fed into a machine learning classification algorithm such as support vector machines or K-nearest neighbors to generate a model $f(.)$. In the second stage, online signals are processed to detect future fault events. A pre-processing step can take place for both stages to improve the input signals, such as Principal Component Analysis (PCA) or general statistical functions.

The purpose of the offline or training phase is to learn a mapping between the data representing the evolution of damage and the binary outcomes (labels). Before the mapping, feature extraction is used to convert each sensor readings group to a feature set. This stage consists of the feature extraction module shown in Fig. 2. Statistical techniques such as kurtosis, skewness analysis, and the feature reduction method of Principal Component Analysis (PCA) are considered in this module. Other techniques can be applied without loss of generality. This task of feature extraction can be simplified in deep learning classifiers. When utilizing these methods, more complex features/predictors can be learned from data automatically [45]. This characteristic is beneficial in our case, where performance is dependent on the quality of the predictors and feature engineering is non-trivial.

Another important step of the offline phase is the extraction of the classification labels $\boldsymbol{y}(t)$ from the run-to-failure data. As shown in Fig. 3, this task is performed by first calculating for each time t_P, the true RUL as $RUL = t_{EOL} - t_P$. Then, the RUL of each t_P is mapped into a label of zero or one using a threshold d. If the RUL is below threshold d then the label is one. Otherwise, it is zero. In this way, a label of one indicates that a fault event will occur in the window $[t_P, t_{P+d}]$:

$$\boldsymbol{y}(t) = \begin{cases} 1, & t_{EOL} - t_P \leq d \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The labeling threshold d is the time limit at which the binary classification label changes from zero to one. When the label is zero, the equipment is far from the end-of-life, and a maintenance action is not needed. When the label is one, the equipment is close to the end-of-life, and a maintenance action is required within the following time window of d days. This is a new approach to prognostics that follows the industrial procedures in commercial aviation.

Note that the labeling threshold is different from the thresholds used in works such as [46–48]. In those works, cut-off values are applied to multidimensional sensor data (or to a health index) to determine the end-of-life of the equipment. In this paper, we use direct classification methods.

The labeling threshold d should be small but not as small as to generate an imbalanced dataset. Note that as d increases, the larger the prediction interval $[t_P, t_{P+d}]$ becomes, and the more positive labels are included in the dataset.

After feature and label extraction, the feature sets, which capture the basic information about each input, and the labels, which classify the input set in close or far from the end-of-life, are fed into a machine learning classification algorithm to generate model $f(.)$. This step completes the offline phase. The same feature extraction module is used to convert unseen inputs to feature sets during the following online phase. These feature sets are then fed into model $f(.)$, which is used to map the input into class zero, indicating that the equipment is far from its end-of-life and does not require a maintenance action within a d time horizon or into class one, indicating the opposite. In addition to this classification, the model provides the probability (uncertainty) associated with the prediction.

3.3. Fault detection models

The relevant features extracted from sensor signals can be used for fault prediction by applying a non-linear pattern classifier. In this work, we selected five classifiers and three variations of a deep recurrent neural network classifier. The selected classifiers are K-Nearest Neighbors (KNN), Random forests (RF), Naive Bayes (NB), Gaussian Support Vector Machines (GSVM), Multi-Layer Perceptron (MLP), and three versions of deep Recurrent Neural Networks (RNNs): the standard Recurrent Neural Network (RNN), the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU). Please note that we only experiment with deep versions (more than one layer) of recurrent neural networks. To emphasize this fact, we refer to the used architectures as DSRNN, DLSTM, and DGRU.

3.4. Performance metrics

A considerable number of performance metrics have been proposed for the field of prognostics [14]. Most of these metrics are, however, specific to regression models, such as root mean square error or relative accuracy [49], as the goal of prediction is typically the remaining useful life (RUL). For this case, however, general-purpose evaluation metrics for classification algorithms were used. The most commonly used ones are built from a confusion matrix [50]. Table 1 presents the traditional confusion matrix for a 2-class classifier [51]. In our case, we define True Positive (TP) as a correctly predicted abnormal condition (fault) and a True Negative (TN) as a correctly predicted normal condition (no fault). In contrast, False Positives (FP) and False Negatives (FN) are defined as false predicted abnormal and false predicted normal equipment condition. The classification performance is measured based on sensitivity, specificity, precision, accuracy, recall and F-score, which are given by [52]:

$$\text{Sensitivity/Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

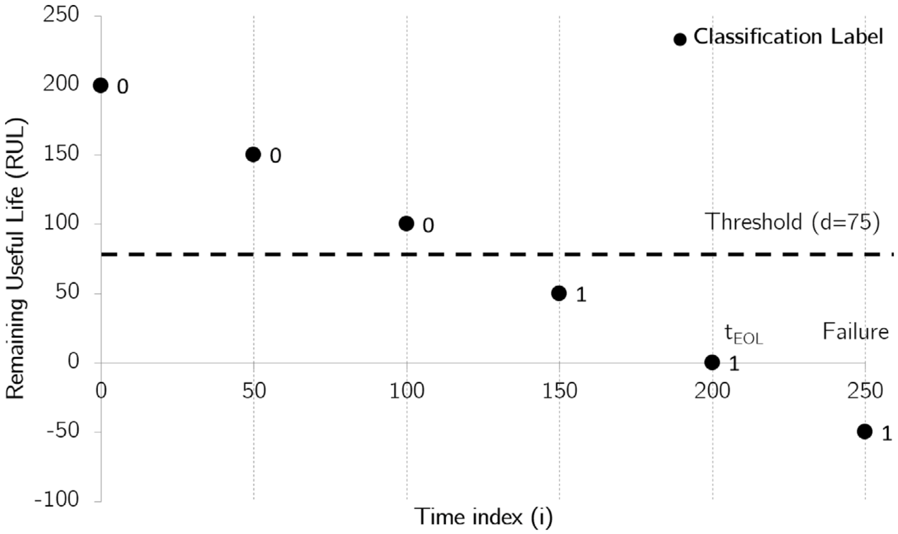


Fig. 3. Label extraction performed during the offline (training) stage. Classification labels are extracted from run-to-failure data using a threshold based approach. First, for each time t_P the remaining useful life (RUL) is calculated as $RUL = T_{EOL} - t_P$. A threshold d (set arbitrarily at 75 in this example) is used to classify each measurement in class zero or one.

Table 1
Confusion matrix representation.

Actual	Predicted	
	Negative	Positive
Negative	True negative (TN)	False positive (FP)
Positive	False negative (FN)	True positive (TP)

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

$$\text{F-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

Sensitivity or recall relates to the ability of the system to predict fault events correctly. In contrast, specificity relates to predicting the negative class correctly, that is, to assess that the equipment will be operating still far from its end-of-life. Precision is a function of true positives and false positives. Accuracy is the correct classification ratio of normal and abnormal samples. F-score is a measure of an evenly balanced precision and recall.

Receiver Operating Characteristic (ROC) analysis is an evaluation method from statistical decision theory to compare and measure classification performance. It captures the relationship between the fraction of true-positive samples (sensibility) and the fraction of false-positive samples (1-specificity) as the decision criterion changes [53]. The Area Under the ROC Curve (AUC) is a helpful index for measuring the classifiers' performance.

4. Case studies

This work is based on two real-world case studies from the aeronautical sector: one related to a gas turbine engine (DS-1) and another specific to a critical valve-subsystem of the engine (DS-2). These two datasets were obtained as part of a collaborative effort with two original equipment manufacturers from aeronautics. The manufacturers were responsible for collecting the data. A detailed description of the datasets can be found in references [54–57]. In this section, we only provide an overview of the data.

Regarding dataset DS-1, the data describe the performance of gas turbine engines over approximately ten years. Each measurement consists of a multidimensional signal taken at three different flight phases: take-off, climb, and cruise. Overall, these multidimensional time series represent 3GB of data. These data consisted of one single cruise snapshot per flight. In addition to the monitoring data, dataset DS-1 comprises information about the engine overhauls. An overhaul is a comprehensive engine inspection that involves removing and disassembling the system, testing all its sub-systems, cleaning and replacing parts as needed, and then reassembling the engine [58]. Both fixed-interval and condition-based interventions are considered in dataset DS-1.

Dataset DS-2 describes the reliability of engine bleed valves. These valves are critical systems of the air management system of the aircraft [59]. Primarily due not to the valve itself but to the complexity of the system where the valve operates, it is not easy to recognize faults and failure patterns [54]. We study the unscheduled removals recorded between 2010 and 2015 for commercial aircraft of 3 airlines. By *removal*, it is meant a maintenance and repair action where the equipment is removed from the airplane and restored to its original condition or replaced by a new or repaired unit. In addition to the valves' removal times, dataset DS-2 comprises 100 GB of data collected from aircraft sensors. These data, recorded with a sampling frequency of 1 Hz, contain monitoring information of the bleed system and environmental conditions during flight.

Given the large volume of data on DS-2, the dataset is submitted to a pre-processing stage. The functions shown in Table 2 [60] are applied on equal-size time segments to produce the time domain features of the models. Four functions (p_1, p_3, p_4, p_7) were used to capture the amplitude and energy of each signal while the remaining ones reflect the distribution of each signal over the time domain.

5. Results

In this section, we empirically evaluate the performance of the proposed models. Details about the experimental setup are provided, and results are presented and subject to discussion.

5.1. Research question

The goal of our experiments is to show the effectiveness of the proposed classifier approach to prognostics. The research hypothesis of the paper is the following:

> H1: Classifier approaches to prognostics can be used as effective means to predict fault events.

Table 2
Pre-processing functions of dataset DS-2. Each sensor signal $s(t)$ of dataset DS-2, recorded with a sampling frequency of 1 Hz, was subject to different pre-processing functions to generate the input features of the classification models..

Feature	Description	Equation
p_1	Average amplitude	$\frac{1}{k}\sum_{i=1}^{k} s(i)$
p_2	Standard deviation	$\left(\frac{\sum_{i=1}^{k}(s(i)-p_1)^2}{k-1}\right)^{\frac{1}{2}}$
p_3	Root mean square amplitude	$\left(\frac{1}{k}\sum_{i=1}^{k} s(i)^2\right)^{\frac{1}{2}}$
p_4	Squared mean root abs amplitude	$\left(\frac{1}{k}\sum_{i=1}^{k} \lvert s(i)\rvert^{\frac{1}{2}}\right)^2$
p_5	Kurtosis coefficient	$\frac{\sum_{i=1}^{k}\left(s(i)-p_1\right)^4}{(k-1)p_2^4}$
p_6	Skewness coefficient	$\frac{\sum_{i=1}^{k}\left(s(i)-p_1\right)^3}{(k-1)p_2^3}$
p_7	Peak value	$\max \lvert s(i)\rvert$
p_8	Peak factor	$\frac{p_7}{p_3}$
p_9	Margin factor	$\frac{p_7}{p_4}$
p_{10}	Waveform factor	$\frac{p_3}{\frac{1}{k}\sum_{i=1}^{k}\lvert s(i)\rvert}$
p_{11}	Impulse factor	$\frac{p_7}{\frac{1}{k}\sum_{i=1}^{k}\lvert s(i)\rvert}$

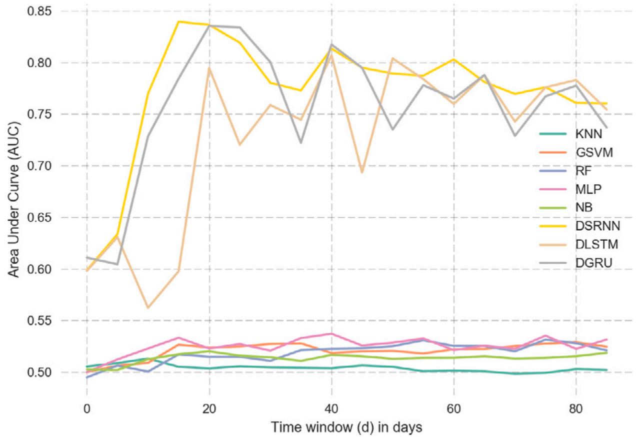


Fig. 4. Area under the Curve (AUC) shown as a function of the window size d for dataset DS-1.

The following sections present the results of testing and comparing different classifier algorithms on dataset DS-1 and dataset DS-2. First, AUC performance is shown as a function of the window size d to demonstrate the classifiers' performance for different prediction intervals. Second, we evaluate the models quantitatively according to the metrics in Section 3.4. Third, the algorithms' performance is shown for a set of randomly selected samples to give a qualitative idea of how they forecast a health event.

To support this hypothesis, we propose to compare a set of classifier models. Concretely, we empirically evaluate the performance of K-Nearest Neighbors (KNN), Random forests (RF), Naive Bayes (NB), Support Vector Machines (SVM), Multi-Layer Perceptron (MLP) and three versions of deep Recurrent Neural Networks (RNNs), namely the standard RNN, the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU). More details about these models can be found in [61]. As baseline models, we choose classifiers that make decisions based on simple rules: the uniform random classifier, the stratified classifier, and the most frequent item classifier. The random classifier generates predictions uniformly at random. The stratified classifier draws predictions from the training set's class distribution. The most frequent classifier predicts the most frequent label in the training set.

5.2. Area Under the Curve (AUC) analysis

Figs. 4 and 5 illustrate the Area Under the Curve (AUC) performance of the classifier algorithms for different values of parameter d. The parameter d sets the size of the prediction window, i.e., the days in advance that the algorithm should alert the occurrence of a health event of interest, an engine overhaul (DS-1), or a valve removal (DS-2). For example, if $d = 10$, ten days before the health event, the algorithm should signify the occurrence. The range of the d parameter varied between zero and the maximum prediction window that made sense for each dataset: a value that allowed maintenance staff to intervene with some planning time in advance but not too large as to promote too soon maintenance actions. Maximum acceptable window sizes of 80 days and 40 days were set for DS-1 and DS-2 datasets, respectively. This configuration resulted from the time between engine overhauls being as long as four years for dataset DS-1 and one year long for the valve removals in dataset DS-2.

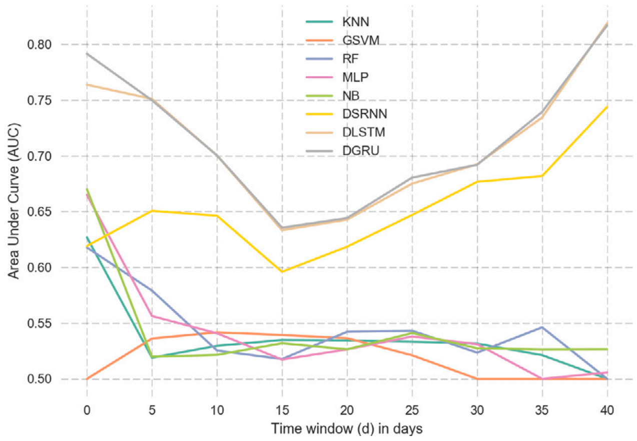


Fig. 5. Area under the Curve (AUC) shown as a function of the window size d for dataset DS-2.

From Fig. 4, it can be seen that the AUC performance of the traditional classifier methods of KNN, GSVM, RF, MLP, and NB is approximately the same over the range of the parameter d for both datasets. For all values of d, the mentioned classifiers' performance is around 0.50, which is equal to a random classifier's performance. Such low AUC values indicate that traditional classifiers could not distinguish between the two degradation states in the tested dataset (DS-1). The referred models performed poorly, with none showing better results than the other.

The results on dataset DS-2 were similar to the ones on dataset DS-1, but for $d \in [0, 10[$, results differed. These shorter intervals leave the least space for imprecision as to when the removal will happen. In the case of DS-1, the models show the same performance for this interval as for other values of d (0.50). However, in DS-2, it seems easier for traditional models (except the GSVM) to identify health events on these shorter intervals. This result might indicate a more evident degradation pattern on the interval of zero to ten days before the valve removal, less precise for other time windows. This assumption is supported by the fact that experts usually look at a window of 10 days to make a removal decision.

As shown in Fig. 4, the deep recurrent neural network classifiers of DSRNN, DLSTM, and DGRU exhibit performance superior to that of a random classifier for the considered range of d values − the discriminative ability of the models is acceptable for most values of d in DS-1. For dataset DS-1 and after $d = 20$, the deep models show acceptable AUC values ranging between 0.69 (lowest value at $d = 44$) and 0.84 (highest value at $d = 20$). Please note that for $d < 20$, all models show significantly low AUC values. This finding may be explained by the fact that the dataset is too unbalanced with such low d values, i.e., negative samples outnumber the positive samples. After this analysis of the AUC with the d parameter, we selected $d = 30$ as the value of interest for dataset DS-1. This value allows for acceptable algorithmic performance and gives enough time to arrange for the following maintenance action.

For dataset DS-2, we can see a trend different from dataset DS-1 for the deep models. Here and as show in Fig. 5, the AUC of the DSRNN, DLSTM, and DGRU tends to decrease until reaching its lowest value for $d = 15$ and then tends to increase again. Once again, this trend can be explained by the fact that it may be easier to discern fault events for smaller prediction windows ($d < 15$). The desirable window of prediction for valve removal, from an industrial standpoint, is around 10 and 30 days. Taking this into consideration, we selected $d = 10$ for our experiments in dataset DS-2. This choice was made to respect the window sizes usually used in the industry.

5.3. Receiver operating characteristic curve (ROC) analysis

Several insights arise from an analysis of the ROC curves in Fig. 6 ($d = 30$) and Fig. 7 ($d = 10$). As illustrated, the performance of the traditional classifiers of KNN, GSVM, RF, MLP, and NB is low, as measured by the closest point to the ROC's top left corner and by the AUC. The performance of these approaches is, however, slightly better for dataset DS-1 than for DS-2. This result can be explained by the lower quality of the data (i.e., missing data, outliers) and the limits of predictability of dataset DS-2. As previously mentioned, DS-2 is a very challenging dataset where fault patterns are difficult to recognize, even by experts [54]. Note that this dataset consists solely of unscheduled removals, that is, fault events that experts were unable to trace. In contrast, DS-1 only includes overhauls scheduled by human intervention, automatic detection, or fixed intervals, with more traceable causes.

The Figs. 6 and 7 also show that the performance of the deep models is acceptable and superior to the traditional approaches, with the points that represent the hit and the false-alarm rate tracing out curvilinear bow-shaped ROC curves almost symmetric about the negative diagonal.

5.4. Performance evaluation

After examining the AUC performance of the proposed approaches and selecting the appropriate window size (d) values, we compared the models numerically. Tables 3 and 4 illustrate this comparison for the two datasets. The baseline models exhibit, as expected, the worse AUC results (around 0.50 for all algorithms). The frequent baseline always predicts that the equipment is far from the end-of-life, which is of no use to our case. The uniform classifier randomly predicts one of the two labels. The stratified classifier attempts to generate predictions by drawing on the 2-class distribution of the training set.

Despite the apparent advantage of the stratified approach over the random (uniform) classifier, this algorithm gives worse F-Score results than the later model. On DS-1, the uniform classifier's performance is low, as measured by its AUC, but its F-Score is better than the score of the traditional classifiers. Concretely, the F-Score of 18.14% is better than the F-Score of the KNN, NB, GSVM, and the MLP on DS-1. Seemingly, the uniform classifier's F-Score of 48.37% surpasses the NB, KNN, MLP, RF, and GSVM on DS-2. Only the RF shows better F-Score performance for dataset DS-1, even though the improvement is not significant. These results reinforce the notion that traditional classifier approaches are not suited to this kind of prognostics task.

A comparison between the uniform classifier and the deep models shows that the latter approaches are better for prognostics purposes. Here, the deep models surpass the baseline in almost all metrics. For example, on DS-1, with an AUC of 0.81, the DGRU model is better than the random classifier, which only exhibits an AUC of 0.50. The F-Score of 32.88% is almost doubling the baseline score of 18.14%. On DS-2, differences are not as significant, with a difference of 0.3 between the models' AUC and 0.10 between the models' F-Score.

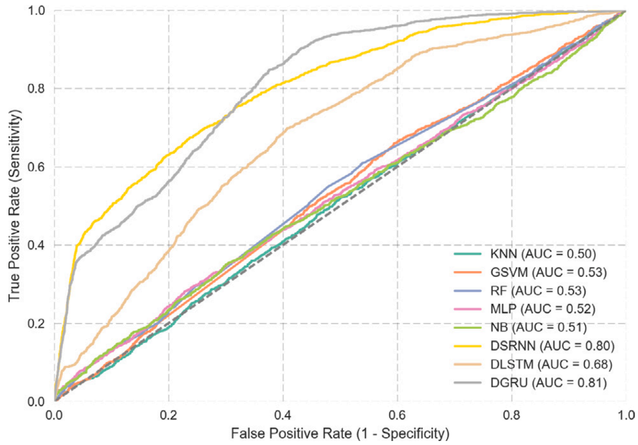


Fig. 6. The receiver operating characteristic curve (ROC) for dataset DS-1 (label threshold $d = 30$ days). ROC curve showing hit rate as a function of false-alarm rate for different probability cut-off values.

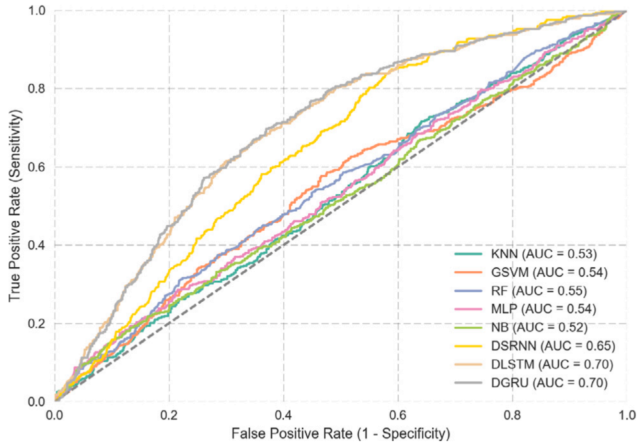


Fig. 7. The receiver operating characteristic curve (ROC) for dataset DS-1 (label threshold $d = 10$ days). ROC curve showing hit rate as a function of false-alarm rate for different probability cut-off values.

5.5. Illustrative examples

We end this section with an illustrative example of how the classifiers generate predictions for different samples of datasets DS-1 and DS-2. Fig. 8 illustrates the predictions of four randomly selected overhauls from dataset DS-1. In the charts, we show the time to overhaul on the x-axis. On the y-axis, we show the predicted probability of a fault event (overhaul in this case). It is expected that the predicted probability increases as the time to the overhaul advances. The predictions over time of each model are shown in different colors. Horizontal lines are drawn to indicate the probability thresholds of each model. These thresholds are calculated from the ROC curve of the models as the values that maximize the AUC. When above these lines, the predictions are considered to belong to class 1, signaling that the equipment is far from the end-of-life and no maintenance action is needed. When below the cut-off values, the equipment is far from the end-of-life, and a maintenance action is not required within the next d days. The optimal expected behavior is to have predictions of class 1 only after the vertical line of $x = 30$ days.

The best models in Fig. 8 are the models of DSRNN, DLSTM, and DGRU as these tend to consistently output predictions above their thresholds only after the $x = 30$ and $x = 10$ days line. For example, from Figs. 8(a) and 8(b) it can be seen how the models predictions

Table 3
Results of dataset DS-1 (ordered by AUC and F-Score).

Classifier	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F-Score (%)	AUC
Frequent (Baseline)	0.00	100.00	ND	90.03	ND	0.50
Stratified (Baseline)	9.21	91.39	10.60	83.19	9.86	0.50
Random (Baseline)	100.00	0.00	9.97	9.97	18.14	0.50
KNN	50.84	50.86	10.28	50.86	17.11	0.50
NB	51.34	51.30	10.46	51.31	17.38	0.51
MLP	51.76	51.78	10.63	51.78	17.63	0.52
GSVM	52.43	52.47	10.89	52.47	18.04	0.53
RF	51.26	54.95	11.19	54.58	18.38	0.53
DLSTM	63.40	63.48	16.13	63.47	25.72	0.68
DSRNN	71.36	71.35	21.62	71.35	33.19	0.80
DGRU	71.02	71.09	21.39	71.08	32.88	0.81

Table 4
Results of dataset DS-2 (ordered by AUC and F-Score).

Classifier	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F-Score (%)	AUC
Frequent (Baseline)	0.00	100.00	ND	68.10	ND	0.50
Stratified (Baseline)	30.18	69.62	31.75	57.04	30.95	0.50
Random (Baseline)	100.00	0.00	31.90	31.90	48.37	0.50
NB	50.90	50.95	32.71	50.93	39.82	0.52
KNN	51.35	51.48	33.14	51.44	40.28	0.53
MLP	51.80	52.11	33.63	52.01	40.78	0.54
GSVM	54.73	54.96	36.27	54.89	43.63	0.54
RF	54.05	54.43	35.71	54.31	43.01	0.55
DSRNN	60.36	60.65	41.81	60.56	49.40	0.65
DGRU	65.77	65.72	47.33	65.73	55.04	0.70
DLSTM	65.77	65.82	47.40	65.80	55.09	0.70

increase considerably after the $x = 30$ and $x = 10$ lines crossing their probability thresholds close to the engine overhaul.

Given these results, we find enough evidence to support hypothesis H1 partially: it is possible to perform fault prognostics using classifier approaches, albeit only if advanced classifier methods such as deep recurrent neural networks are used. Also, and as can be seen in the different plots of Fig. 8, the weak learners, i.e., the models that perform slightly better than random guessing, output approximately the same probability outcomes at each moment in time. This finding suggests that it would be challenging to increase overall performance by fusing these approaches since they operate similarly.

6. Conclusion

In this paper, a novel approach to prognostics based on classification algorithms has been proposed. The proposed models intend to be an alternative solution to remaining useful life estimation methods. Importantly, the proposed methodology does not make use of remaining useful life estimates to perform the prognostics but directly estimates the probabilities of a binary classification label. This label provides information to help industry experts decide if a maintenance action is needed within a given time window of fixed size. To this aim, we have evaluated the use of several machine learning classifiers on two real-world case studies from aeronautics.

The tested data-driven approaches were devised so that they could learn the prognostics task at hand without having to be coded explicitly for it. Only minimal changes were required to adapt the techniques to the two case studies. The configuration, pre-processing, and training stages of the algorithms were the same in both scenarios. The adaptive capabilities of these techniques make them useful methods of general-purpose.

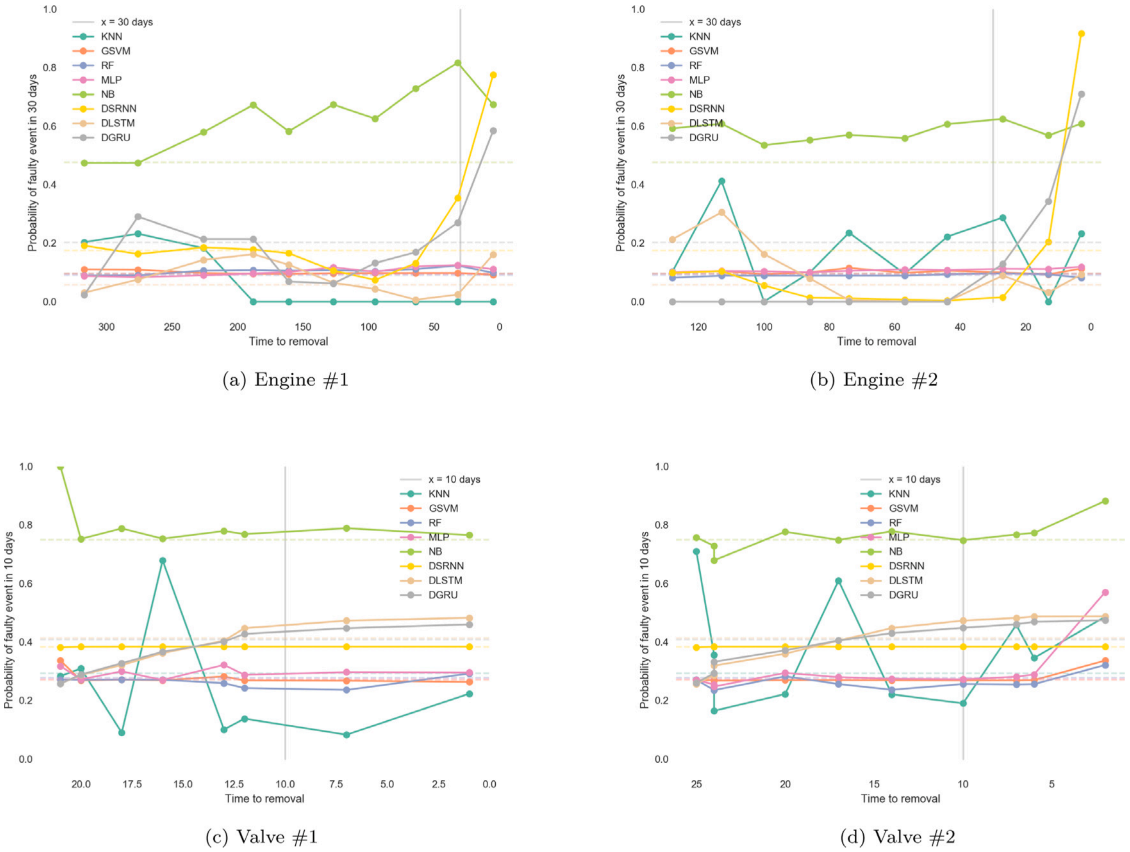
The proposed classification models have been validated through comparison with simple classifiers. Traditional classification models, such as Naive Bayes or support vector machines, have shown only a slight improvement over random guessing. This type of classification model might not be the most appropriate for this kind of task. On the contrary, the deep learning classifiers exhibited a very satisfactory performance. According to our experimental results, the most suitable prognostics models are deep learning methods. The best performing deep model seems to depend on the specific application and significantly affects the classification performance.

In this work, we utilized a sliding time window to classify each data point as being close or far from the end-of-life. The smaller the size of this forecasting window, the lower the performance of the classification algorithms tended to be. After performing a sensitivity analysis, the window sizes of 30 and 10 days were selected for the two case studies, respectively. The industry experts referred to these time windows as acceptable forecasting intervals in which the deep learning methods also exhibited satisfactory performance. Maintenance engineers typically want to know 10 days in advance if a maintenance action of the bleed system is needed. The same holds for 30 days for aircraft engines.

The positive results of the deep neural networks in this work align with previous studies from other fields [62]. These methods try to overcome some of the problems of techniques such as those based on statistical hypothesis testing, which require the configuration of fixed parameters and are significantly dependent on the application. For example, the popular Z-score test [63] has its limitations, such as when the data are not normally distributed or when the data contain extreme values. In these situations, statistical testing may fail to screen outliers appropriately. Statistical techniques may be sufficient for anomaly detection, but there is often the need to investigate more sophisticated algorithms.

The proposed approach in this work is an alternative interpretation of data-driven prognostics. To a certain extent, it may even be easier to interpret and manage an array of probabilities describing the needs for a maintenance intervention in the following days than to deal with a collection of RUL estimates. Note that classification techniques deal with probabilities in a natural way as opposed to most regression methods. By classifying and associating a confidence level to the maintenance needs, we obtain more interpretable models that can better support decision-making in maintenance.

In the future, we intend to study how to set the model's probability cut-off values dynamically. In theory, this will allow us to generate better predictions and eventually improve traditional and more advanced classifiers' performance. It is also our goal to explore other advanced deep models such as convolution networks and auto-encoders. Finally, we aim to propose new evaluation metrics for these models.



(a) Engine #1

(b) Engine #2

(c) Valve #1

(d) Valve #2

Fig. 8. Performance of classifiers on randomly selected overhauls on dataset DS-1 and randomly selected valves on dataset DS-2.

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] J. Sikorska, M. Hodkiewicz, L. Ma, Prognostic modelling options for remaining useful life estimation by industry, Mech. Syst. Signal Process. 25 (5) (2011) 1803–1836.

[2] B. Saha, E. Koshimoto, C.C. Quach, E.F. Hogge, T.H. Strom, B.L. Hill, S.L. Vazquez, K. Goebel, Battery health management system for electric UAVs, in: Aerospace Conference, 2011 IEEE, IEEE, 2011, pp. 1–9.

[3] C.R. Farrar, N.A. Lieven, Damage prognosis: the future of structural health monitoring, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 365 (1851) (2007) 623–632.

[4] C. Wilkinson, D. Humphrey, B. Vermeire, J. Houston, Prognostic and health management for avionics, in: Aerospace Conference, 2004. Proceedings. 2004 IEEE, 5, IEEE, 2004, pp. 3435–3447.

[5] Z. Chen, Y. Yang, Z. Hu, A technical framework and roadmap of embedded diagnostics and prognostics for complex mechanical systems in prognostics and health management systems, IEEE Trans. Reliab. 61 (2) (2012) 314–322.

[6] M.J. Daigle, K. Goebel, A model-based prognostics approach applied to pneumatic valves, Int. J. Progn. Health Manag. 2 (2) (2011) 84–99.

[7] C.S. Kulkarni, J.R. Celaya, K. Goebel, G. Biswas, Physics based Degradation Modeling and Prognostics of Electrolytic Capacitors under Electrical Overstress Conditions, in: AIAA Infotech@ Aerospace Conference, 2013, p. 5137.

[8] K. Goebel, B. Saha, Prognostics applied to electric propulsion UAV, in: Handbook of Unmanned Aerial Vehicles, Springer, 2015, pp. 1053–1070.

[9] S. Sankararaman, K. Goebel, Why is the remaining useful life prediction uncertain? in: Annual Conference of the PHM Society, 5, (1) 2013.

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

[11] K. Goebel, M. Daigle, A. Saxena, S. Sankararaman, R. I., J. Celaya, Prognostics: The Science of Making Predictions, 2017.

[12] S. Uckun, K. Goebel, P.J. Lucas, Standardizing research methods for prognostics, in: Prognostics and Health Management, 2008. PHM 2008. International Conference on, IEEE, 2008, pp. 1–10.

[13] K. Javed, R. Gouriveau, N. Zerhouni, State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels, Mech. Syst. Signal Process. 94 (2017) 214–236.

[14] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, Metrics for evaluating performance of prognostic techniques, in: Prognostics and Health Management, 2008. Phm 2008. International Conference on, IEEE, 2008, pp. 1–17.

[15] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Metrics for offline evaluation of prognostic performance, Int. J. Progn. Health Manag. 1 (1) (2010) 4–23.

[16] A.K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510.

[17] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.

[18] M.I. Jordan, Serial order: A parallel distributed processing approach, Adv. Psychol. 121 (1997) 471–495.

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv Preprint arXiv:1406.1078.

[21] M.J. Roemer, C.S. Byington, G.J. Kacprzynski, G. Vachtsevanos, An overview of selected prognostic technologies with application to engine health management, in: ASME Turbo Expo 2006: Power for Land, Sea, and Air, American Society of Mechanical Engineers, 2006, pp. 707–715.

[22] M. Schwabacher, A survey of data-driven prognostics, in: Infotech@ Aerospace, 2005, p. 7002.

[23] A. Mosallam, K. Medjaher, N. Zerhouni, Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction, J. Intell. Manuf. 27 (5) (2016) 1037–1048.

[24] J.-M. Bai, G.-S. Zhao, H.-J. Rong, Novel direct remaining useful life estimation of aero-engines with randomly assigned hidden nodes, Neural Comput. Appl. (2019) 1–12.

[25] T. Wang, J. Yu, D. Siegel, J. Lee, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, in: Prognostics and Health Management, 2008. PHM 2008. International Conference on, IEEE, 2008, pp. 1–6.

[26] X. Li, W. Zhang, H. Ma, Z. Luo, X. Li, Data alignments in machinery remaining useful life prediction using deep adversarial neural networks, Knowl.-Based Syst. 197 (2020) 105843.

[27] D. Xiao, C. Qin, H. Yu, Y. Huang, C. Liu, J. Zhang, Unsupervised machine fault diagnosis for noisy domain adaptation using marginal denoising autoencoder based on acoustic signals, Measurement 176 (2021) 109186.

[28] K. Yu, Q. Fu, H. Ma, T.R. Lin, X. Li, Simulation data driven weakly supervised adversarial domain adaptation approach for intelligent cross-machine fault diagnosis, Struct. Health Monit. (2020) 1475921720980718.

[29] J. Sun, H. Zuo, W. Wang, M.G. Pecht, Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance, Mech. Syst. Signal Process. 28 (2012) 585–596.

[30] F. Yang, M.S. Habibullah, T. Zhang, Z. Xu, P. Lim, S. Nadarajan, Health index-based prognostics for remaining useful life predictions in electrical machines, IEEE Trans. Ind. Electron. 63 (4) (2016) 2633–2644.

[31] C. Song, K. Liu, Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach, IISE Trans. 50 (10) (2018) 853–867.

[32] P. Jayaswal, A. Wadhwani, K. Mulchandani, Machine fault signature analysis, Int. J. Rotating Mach. 2008 (2008).

[33] R. Isermann, Model-based fault-detection and diagnosis – status and applications, Annu. Rev. Control 29 (1) (2005) 71–85.

[34] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15.

[35] M.G. Pecht, M. Kang, Machine learning: Anomaly detection, Wiley-IEEE Press, 2019.

[36] T. Brotherton, T. Johnson, Anomaly detection for advanced military aircraft using neural networks, in: Aerospace Conference, 2001, IEEE Proceedings., 6, IEEE, 2001, pp. 3113–3123.

[37] A.L. Ellefsen, P. Han, X. Cheng, F.T. Holmeset, V. Æsøy, H. Zhang, Online fault detection in autonomous ferries: Using fault-type independent spectral anomaly detection, IEEE Trans. Instrum. Meas. 69 (10) (2020) 8216–8225.

[38] X. Jin, Z. Que, Y. Sun, Y. Guo, W. Qiao, A data-driven approach for bearing fault prognostics, IEEE Trans. Ind. Appl. 55 (4) (2019) 3394–3401.

[39] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. Orchard, G. Vachtsevanos, Anomaly detection: A robust approach to detection of unanticipated faults, in: Prognostics and Health Management, 2008. PHM 2008. International Conference on, IEEE, 2008, pp. 1–8.

[40] E. Ramasso, M. Rombaut, N. Zerhouni, Prognostic by classification of predictions combining similarity-based estimation and belief functions, in: Belief Functions: Theory and Applications, Springer, 2012, pp. 61–68.

[41] K. Javed, A robust & reliable data-driven prognostics approach based on extreme learning machine and fuzzy clustering., Ph.D. thesis, Université de Franche-Comté, 2014.

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

[43] J.B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, F. Fnaiech, Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network, Mech. Syst. Signal Process. 56 (2015) 150–172.

[44] M. Yan, X. Wang, B. Wang, M. Chang, I. Muhammad, Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model, ISA Trans. 98 (2020) 471–482.

[45] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[46] F. Camci, System maintenance scheduling with prognostics information using genetic algorithm, IEEE Trans. Reliab. 58 (3) (2009) 539–552.

[47] K. Javed, R. Gouriveau, N. Zerhouni, Novel failure prognostics approach with dynamic thresholds for machine degradation, in: IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2013, pp. 4404–4409.

[48] K. Medjaher, N. Zerhouni, Framework for a hybrid prognostics., Chem. Eng. Trans. 33 (2013) 91–96.

[49] L. Tang, M.E. Orchard, K. Goebel, G. Vachtsevanos, Novel metrics and methodologies for the verification and validation of prognostic algorithms, in: Aerospace Conference, 2011 IEEE, IEEE, 2011, pp. 1–8.

[50] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, 2011.

[51] R. Kohavi, F. Provost, Glossary of terms, appl, Mach. Learn. Knowl. Discov. Process 30 (2/3) (1998).

[52] M. Sokolova, N. Japkowicz, S. Szpakowicz, Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2006, pp. 1015–1021.

[53] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.

[54] M. Baptista, I.P. de Medeiros, J.P. Malere, C. Nascimento Jr, H. Prendinger, E.M. Henriques, Comparative case study of life usage and data-driven prognostics techniques using aircraft fault messages, Comput. Ind. 86 (2017) 1–14.

[55] M. Baptista, C.L. Nascimento Jr, H. Prendinger, E. Henriques, A case for the use of data-driven methods in gas turbine prognostics, in: Annual Conference of the PHM Society, 9, (1) 2017.

[56] M.L. Baptista, I.P. de Medeiros, J.P. Malere, C.L. Nascimento, H. Prendinger, E. Henriques, Aircraft on-condition reliability assessment based on data-intensive analytics, in: Aerospace Conference, 2017 IEEE, IEEE, 2017, pp. 1–12.

[57] M. Baptista, H. Prendinger, E. Henriques, Prognostics in Aeronautics with Deep Recurrent Neural Networks, in: PHM Society European Conference, 5, (1) 2020, p. 11.

[58] R. Seemann, S. Langhans, T. Schilling, V. Gollnick, Modeling the life cycle cost of jet engine maintenance, Technische UniversitäT Hamburg-Harburg (TUHH), Hamburg, 2010.

[59] R. de Pádua Moreira, C.L. Nascimento, Prognostics of aircraft bleed valves using a SVM classification algorithm, in: Aerospace Conference, IEEE, 2012, pp. 1–8.

[60] Q. Wu, X. Yang, Q. Zhou, Pattern recognition and its application in fault diagnosis of electromechanical system, J. Inf Comput. Sci. 9 (8) (2012) 2221–2228.

[61] C.M. Bishop, Pattern recognition, Mach. Learn. 128 (2006) 1–58.

[62] S.B. Kotsiantis, I.D. Zaharakis, P.E. Pintelas, Machine learning: a review of classification and combining techniques, Artif. Intell. Rev. 26 (3) (2006) 159–190.

[63] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, N. Sharma, Detection of spatial outlier by using improved Z-score test, in: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2019, pp. 788–790.