Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics
The case of turbofan engines

Mitici, Mihaela; de Pater, Ingeborg; Barros, Anne; Zeng, Zhiguo

DOI
10.1016/j.ress.2023.109199

Publication date
2023

Document Version
Final published version

Published in
Reliability Engineering and System Safety

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines

Mihaela Mitici [a,*], Ingeborg de Pater [b], Anne Barros [c], Zhiguo Zeng [c]

[a] Faculty of Science, Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands
[b] Faculty of Aerospace Engineering, Delft University of Technology, HS 2926 Delft, The Netherlands
[c] Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, 3 Rue Joliot Curie, Gif-sur-Yvette 91190, France

ARTICLE INFO

Keywords:
Predictive maintenance planning
Probabilistic remaining useful life prognostics
Aircraft
Maintenance scheduling
C-MAPSS turbofan engines

ABSTRACT

The increasing availability of condition-monitoring data for components/systems has incentivized the development of data-driven Remaining Useful Life (RUL) prognostics in the past years. However, most studies focus on point RUL prognostics, with limited insights into the uncertainty associated with these estimates. This limits the applicability of such RUL prognostics to maintenance planning, which is per definition a stochastic problem. In this paper, we therefore develop probabilistic RUL prognostics using Convolutional Neural Networks. These prognostics are further integrated into maintenance planning, both for single and multiple components. We illustrate our approach for aircraft turbofan engines. The results show that the optimal replacement time for the engines is close to the lower bound of the 99% confidence interval of the RUL estimates. We also show that our proposed maintenance approach leads to a cost reduction of 53% compared to a traditional Time-based maintenance strategy. Moreover, compared with the ideal case when the true RUL is known in advance (perfect RUL prognostics), our approach leads to a limited number of failures. Overall, this paper proposes an end-to-end framework for data-driven predictive maintenance for multiple components, and showcases the potential benefits of data-driven predictive maintenance on cost and reliability.

1. Introduction

Modern systems are currently monitored by multiple sensors that generate large volumes of data. As an example, for a Boeing 787, approximately 1000 parameters are continuously monitored for the engine, leading to a total of 20 terabytes of data per flight hour [1]. Using such datasets, several AI algorithms have been developed in the past years to estimate the Remaining Useful Life (RUL) of components and systems [2].

The integration of data-driven RUL prognostics into maintenance planning has been shown to significantly reduce maintenance costs and the number of failures [3–5]. However, most existing studies focus either on developing RUL prognostics only, or on developing predictive maintenance planning models where the probability of failure is based on simple, generic probability distributions of the time-to-failure/RUL [6].

Data-driven RUL prognostics for mechanical components/systems are often obtained using AI algorithms and sensor measurements. Examples of such studies are [7], where point RUL prognostics are developed for batteries using a deep neural network. Point RUL prognostics are also developed in [4,8] for turbofan engines using a convolutional neural network. Only few studies also estimate the distribution of the RUL. In [9,10], the probability distribution of the RUL of turbofan engines is predicted using deep Gaussian processes and neural networks with Monte Carlo dropout, respectively.

For predictive maintenance planning, many studies propose advanced planning models, but assume that the degradation of components/systems follow a generic distribution with fixed parameters, instead of component-specific data-driven RUL prognostics. For instance, such studies assume that the degradation of components/systems follows a generic Gamma process [11–14], a Wiener process [15], or a non-homogeneous Poisson process [16]. With these assumptions, the maintenance planning of the components/systems is posed as a renewal process [13], a Markov decision process [17,18], or a dynamic grouping heuristic [14]. In practice, however, the degradation of components/systems rarely follows such generic degradation processes. Rather, the degradation trends exhibited by the sensor measurements are often noisy and reactive to the dynamic environment in which the components/systems are operated.

Few studies develop data-driven RUL prognostics by tracking the specific degradation of a component/system via sensor measurements,

[*] Corresponding author.
E-mail address: m.a.mitici@uu.nl (M. Mitici).

https://doi.org/10.1016/j.ress.2023.109199
Received 28 November 2022; Received in revised form 24 January 2023; Accepted 22 February 2023
Available online 25 February 2023

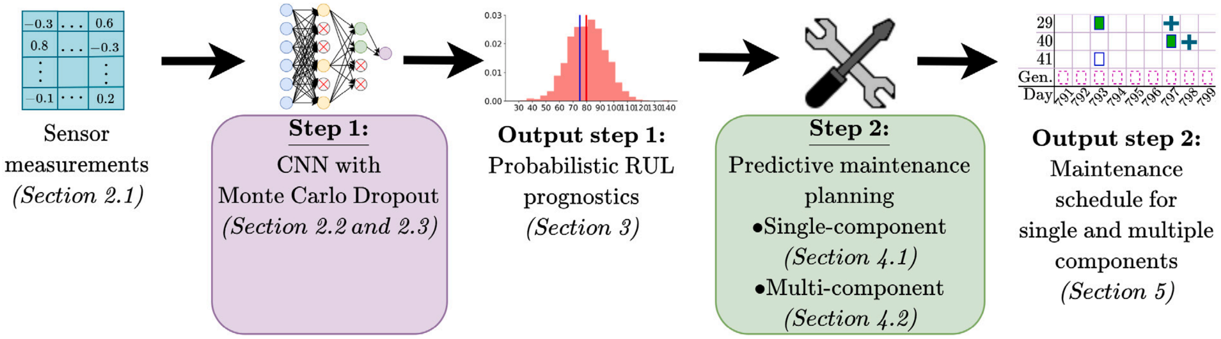


Fig. 1. Overview of the roadmap for predictive maintenance, from raw sensor measurements to probabilistics RUL prognostics using Convolution Neural Networks to maintenance planning.

and actually integrate these prognostics into maintenance planning. As the degradation trends are specific for each component/system, maintenance planning should consider RUL prognostics specific to each component/system and propose customized maintenance tasks. In [5], data-driven probabilistic RUL prognostics for an aircraft engine are integrated into a deep reinforcement learning (RL) model to optimize the moment of engine replacement. The RL approach, however, provides little insight into the maintenance decision process (black-box). In contrast, in this paper we propose a renewal process (white-box) that considers data-driven RUL prognostics to specify the replacement time of engines. In [4,19], data-driven point RUL prognostics are developed for aircraft engines using a Convolutional Neural Network and a bidirectional LSTM neural network, respectively. These point RUL prognostics are integrated into maintenance planning using an integer linear program and a threshold-based approach, respectively. Having point RUL prognostics, however, maintenance planning is done without any insight into the uncertainty of the estimated RUL. In contrast, we propose a maintenance planning framework that integrates *probabilistic* RUL prognostics into maintenance planning. In [20], the probability of failure of an engine within a pre-defined time-window is determined using a Short Long-Term Memory (LSTM) neural network. With this estimate, the engine replacement and ordering of spare parts is optimized. The choice of the time window, however, limits the applicability of the estimates for maintenance planning. Instead, we directly estimate the probability density function (PDF) of the RUL. Moreover, we propose models for predictive maintenance for both a single engine, as well as multiple engines. In [21] the distribution of the RUL for aircraft cooling units is estimated using a physics-based model. Similarly, in [22], the future degradation of a railway track is predicted using a physics-based model. These predictions are further used to plan the maintenance of the cooling units and railway tracks, respectively. Physic-based models for RUL prognostics, however, require a high-fidelity modeling of the degradation, which is not often obtainable in practice.

In this paper, we propose an end-to-end framework for multi-component predictive maintenance starting from raw sensor measurements, to probabilistic RUL prognostics, to maintenance planning (see Fig. 1). We develop data-driven *probabilistic* RUL prognostics using Convolutional Neural Networks with Monte Carlo dropout. The obtained distributions of the RUL are shown to be accurate and reliable. The prognostics are integrated into a single-component maintenance planning model using renewal–reward processes. These results are further extended to a multi-component maintenance planning model, where additional constraints regarding maintenance resources are considered. According to [14], the use of prognostic/predictive information in maintenance decisions for complex multi-component systems is a relatively underexplored area. With this paper, we address the area of data-driven predictive maintenance, from the development of RUL prognostics to the integration of these prognostics into maintenance planning for multiple components. Moreover, we show the utility of having probabilistic RUL prognostics for maintenance planning. For example, knowing that there is a high probability that a component will fail soon (high probability that the RUL is small) incentivizes the soon replacement of this component. In fact, the probability associated with the predicted RUL weighs in the decision to replace or not a component, together with the costs of replacement.

We illustrate our approach for turbofan engines. We compare our maintenance approach with a traditional, Time-based Maintenance strategy. The results show that our approach leads to a 53% reduction of maintenance costs compared to the Time-based Maintenance strategy. Compared with the case when perfect RUL prognostics are available, our approach leads to only a slight increase in the number of failures.

The main contributions of this paper are:

- We obtain *reliable*, data-driven probabilistic RUL prognostics (PDF of RUL) for turbofan engines using Convolutional Neural Networks with Monte Carlo dropout. The availability of PDFs of RUL opens up the opportunity to plan maintenance tasks taking into account the uncertainty associated with RUL estimates.
- We show how a renewal–reward process is specified for data-driven probabilistic RUL prognostics, instead of using a one-fits-all (exponential/Weibull) distribution of failure times assumed for all components.
- For maintenance planning, a maintenance cost is proposed that integrates the data-driven RUL prognostics, instead of using a generic, scalar value (which is often proposed in existing studies). Moreover, for the multi-component maintenance planning, the model is extended to consider the availability of resources (e.g., hangar availability, capacity of maintenance slots).
- While most existing studies focus only on single component maintenance planning, we propose maintenance planning models both for a single and *multiple* components that integrate data-driven probabilistic RUL prognostics.

The remainder of this paper is structured as follows. In Section 2 we develop probabilistic RUL prognostics using Convolutional Neural Networks with Monte Carlo dropout. We analyze the accuracy and reliability of these prognostics in Section 3. In Section 4 we propose an optimization model for the maintenance planning of components, taking into account estimates of the PDF of their RUL. In Section 5 we illustrate our approach for a set of turbofan engines. We also analyze the performance of our approach relative to a Time-based maintenance strategy, and relative to the case when perfect RUL prognostics are available. Conclusions are provided in Section 6.

2. Probabilistic RUL prognostics for turbofan engines using convolutional neural networks

In this section, we estimate the PDF of the RUL of aircraft engines after each flight cycle using a Convolutional Neural Network with Monte Carlo dropout.

Table 1
C-MAPSS data subsets for aircraft engines [23].

	FD 001	FD 002	FD 003	FD 004
# of training instances	100	260	100	249
# of test instances	100	259	100	248
# of operating conditions	1	6	1	6
# of fault conditions	1	1	2	2

2.1. Description of the dataset

In this study, we consider the aircraft turbofan engines in the C-MAPSS dataset [23]. In C-MAPSS, the degradation of engines is simulated using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) program developed by NASA.

The C-MAPSS dataset consists of four subsets: FD001, FD002, FD003 and FD004. In turn, each subset consists of a training and a test set. For each engine in the training set, one measurement per sensor per flight cycle is generated, from the installation of the engine until the failure (i.e., run-to-failure instances). In the test set, one measurement per sensor per flight cycle is generated as well. However, the sensor measurements stop at some time before failure. The goal is to predict the RUL at that moment, i.e., the number of flight cycles until the engine fails. For subset FD002 and FD004, six different flight conditions are present, where each flight is performed under one flight condition. Moreover, in subset FD003 and FD004, two different fault conditions are present (see Table 1).

There are 21 sensors in C-MAPSS. Seven of these 21 sensors have a constant sensor measurement over the flight cycles. We therefore only consider the remaining 14 sensors with non-constant measurements. We normalize the sensor measurements in each subset with min–max normalization with respect to the operating condition as follows [24,25]:

$$\hat{m}_{ij} = \frac{2(m^{o}_{ij} - m^{\min}_{jo})}{m^{\max}_{jo} - m^{\min}_{jo}} - 1, \tag{1}$$

with m^{o}_{ij} the sensor measurement of sensor j during flight cycle i, where flight cycle i was performed under operating condition o, $m^{\min}_{jo}$ and $m^{\max}_{jo}$ are the minimum and maximum value in the training set of sensor j under operating condition r respectively, and $\hat{m}_{ij}$ the normalized measurement of sensor j during flight cycle i.

2.2. Architecture of the Convolutional Neural Network

Fig. 2 shows the proposed architecture of the CNN. At flight cycle f of an engine v, we consider data sample X^{v}_{f} as input:

$$X^{v}_{f} = [x^{v}_{f-N}, x^{v}_{f-N+1}, \dots, x^{v}_{f}]. \tag{2}$$

Here, N denotes the number of past flight cycles included (i.e., the window size), and x^{v}_{i} denotes the normalized sensor measurements of engine v at flight cycle i:

$$x^{v}_{i} = [\hat{m}^{v}_{i1}, \hat{m}^{v}_{i2}, \dots, \hat{m}^{v}_{iH}], \tag{3}$$

with H the total number of considered sensors, and $\hat{m}^{v}_{ij}$ the normalized sensor measurement of flight cycle i of engine v from sensor j (see Eq. (1)).

The CNN consists of L convolutional layers (see also Fig. 2). Each convolutional layer consists of K filters, where each kernel has a size of $1 \times S$, i.e., we use one-dimensional kernels. The convolutional operation in the lth convolutional layer for the nth filter k^{l}_{n} is [26]:

$$z^{l}_{n} = tanh(k^{l}_{n} * z^{l-1} + b^{l}_{n}) \tag{4}$$

where z^{l}_{n} is the nth feature map of layer l, $*$ is the convolutional operator, z^{l-1} are the feature maps in layer $l-1$, b^{l}_{n} is the bias of the nth filter of layer l, and $tanh(\cdot)$ denotes the tanh (hyperbolic tangent) activation function. Next, we consider a single convolutional layer with one filter, where each kernel has a size of $1 \times S'$. This layer combines all K feature maps in one single feature map. We denote the output of this last convolutional layer by z^{L}.

Last, we add two fully connected layers to the CNN. These layers predict the RUL based on the extracted features of the last convolutional layer. The output z^{f} of the first fully connected layer is [26]:

$$z^{f} = tanh(w^{f} z^{L} + b^{f}), \tag{5}$$

where b^{f} is the bias and w^{f} be the weights of the first fully connected layer. Last, a second fully connected layer with one neuron and the ReLU activation function outputs the final RUL prediction.

2.3. Monte Carlo dropout

We apply a dropout rate ρ in each layer of the CNN, except the first (to avoid the loss of input information). We perform M forward passes through the neural network for each test sample. During each forward pass, different, randomly selected neurons (ρ percent) are dropped (see Fig. 3). Thus, a different RUL prediction is obtained with each forward pass. In [27], it is shown that a neural network with Monte Carlo dropout approximates a Bayesian neural network representing a deep Gaussian process. Below, we give a short overview of this result and how we apply it to construct a PDF of the RUL.

Let X be the samples with sensor measurements in the training set of CNN, and let Y be the corresponding RUL values. In a Bayesian neural network, the goal is to predict the posterior distribution $p(y|x, X, Y)$ of the RUL y belonging to a test sample x, given the training samples X and Y:

$$p(y|x, X, Y) = \int p(y|x, \omega)p(\omega|X, Y)d\omega, \tag{6}$$

where ω denotes all the weights in the neural network. Here, $p(y|x, \omega)$ is the probability that the RUL equals y, given test sample x and the weights of the neural network ω. Moreover, $p(\omega|X, Y)$ is the posterior distribution of the weights, and denotes the probability that the weights are ω, given the training samples X and Y.

It usually is computationally very expensive to analyze the posterior distribution $p(\omega|X, Y)$ exactly. In variational inference, the posterior distribution $p(\omega|X, Y)$ is therefore approximated with a distribution $q(\omega)^*$ instead. Here, we first define a family (i.e., set) Q of possible posterior distributions $q(\omega)$. The goal is then to find the distribution $q(\omega)^* \in Q$ that minimizes Kullback–Leibler divergence KL with the true posterior distribution $p(\omega|X, Y)$ [28]:

$$q(\omega)^* = \arg\min_{q(\omega)\in Q} \{KL(q(\omega)|p(\omega|X, Y))\}. \tag{7}$$

This is equivalent to finding the distribution $q(\omega)^* \in Q$ that maximizes the evidence lower bound (ELBO) $\mathcal{L}_{\text{ELBO}}$ [27,28]:

$$\mathcal{L}_{\text{ELBO}} = \int q(\omega)\log(p(Y|X, \omega))\, d\omega - KL(q(\omega)|p(\omega)), \tag{8}$$

where $p(\omega)$ is the prior of the weights. We assume that the prior is the standard multivariate normal distribution. Using $q(\omega)^*$, we approximate the posterior distribution $p(y|x, X, Y)$ of the RUL of a test sample by:

$$q(y|x) = \int p(y|x, \omega)q(\omega)^* d\omega, \tag{9}$$

where $q(y|x)$ is the approximation of $p(y|x, X, Y)$.

In [27], the family Q of possible distributions $q(\omega)$ is defined as all Gaussian mixture distributions with two components. The authors of [27] show that this mixture can be approximated by setting $q(\omega)$ in each layer i as:

$$\omega_i = \omega^{\text{original}}_{i} \cdot \text{diag}([\theta_{i,j}]^{R_i}_{j=1}), \tag{10}$$

$$\theta_{ij} \sim \text{Bernoulli}(1-\rho), j = 1, \dots, R_i \tag{11}$$

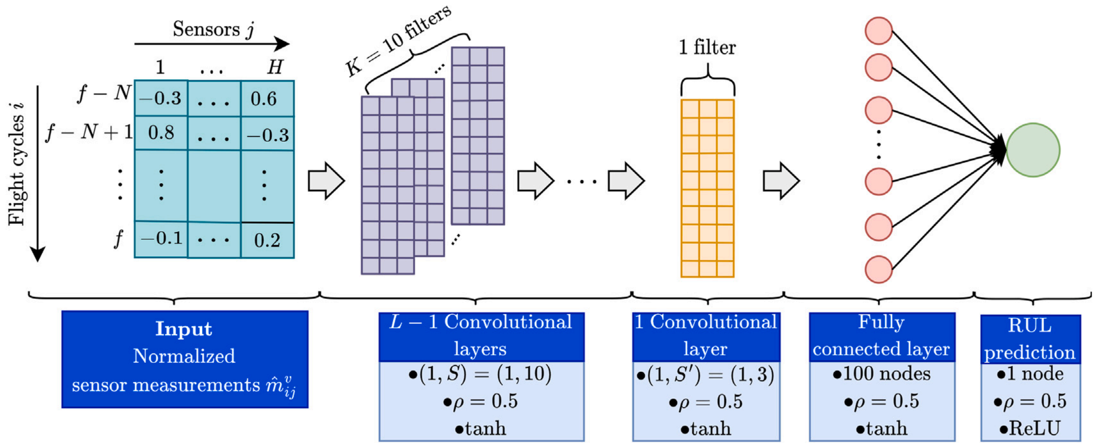
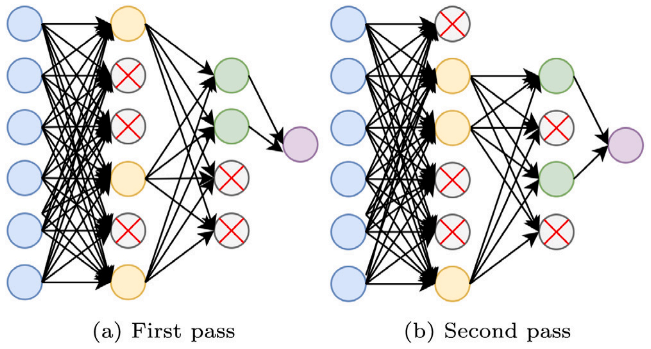


Fig. 2. A schematic overview of the considered Convolutional Neural Network.


(a) First pass

(b) Second pass

Fig. 3. A schematic example of Monte Carlo dropout for a neural network with three fully connected layers, during two passes of a sample through a neural network.

where ω_i are the weights of layer i, R_i the number of nodes in layer i, and $\omega_i^{\text{original}}$ are the weights of layer i without dropout. Moreover, $\text{diag}(z)$ denotes the diagonal matrix constructed with a vector z and θ_{ij} is zero when node j of layer i is dropped out, and one if not. Here, we minimize Eq. (8) by changing the weights ω of the neural network. With this family Q, we obtain the following estimator of the $-\mathcal{L}_{\text{ELBO}}$ [27]:

$$\hat{\mathcal{L}}_{\text{MC}} = \frac{1}{T}\sum_{i=1}^{T}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{T}\|\omega_i\|_2^2, \tag{12}$$

with T the number of training samples, y_i and $\hat{y}_i$ the actual and predicted RUL of training sample i respectively, and λ a weight decay parameter. This is the same objective as minimized when training a neural network.

With this result, we approximate the expected value $\hat{y}$ of the RUL of a test sample as:

$$\hat{y} = E_{q(y|x)}(y) = \frac{1}{M}\sum_{j=1}^{M}\hat{y}_j(x,\omega^j), \tag{13}$$

where M is the number of forward passes through the neural network, ω^j are the weights of the neural network belonging to the jth forward pass (i.e., where some neurons are dropped out), and $\hat{y}(x,\omega^j)$ is the resulting RUL prediction from the jth forward pass through the neural network. For the PDF of the RUL, we give each individual RUL prediction $\hat{y}_j(x,\omega^j)$ a probability of $\frac{1}{M}$.

3. Results - Probabilistic RUL prognostics for aircraft turbofan engines

In this section, we present probabilistic RUL prognostics for turbofan engines.

Table 2
Considered hyperparameters of the CNN.

Hyperparameter	Value
Hyperparameters — architecture	
Window-size N	30
Convolutional layers L	5
Number of filters K	10
Kernel size S	10
Kernel size S' last convolutional layer	3
Number of nodes fully connected layer	100
Monte Carlo dropout rate ρ	0.5
Number of passes M	1000
R_{early}	125
Hyperparameters — optimization	
Optimizer	Adam [32]
Number of epochs	250
Training–Validation split	80%–20%
Initial learning rate	0.001
Decrease learning rate when no improvement in validation loss for ... epochs in a row	10
Decrease learning rate by	$\frac{1}{2}$

3.1. Hyperparameter tuning

The considered hyperparameters of the neural network are in Table 2, optimized with as starting point the hyperparameters of the CNN in [4,24]. In contrast with these papers, the window size equals 30 for all four data subsets. For data subsets FD002 and FD004, however, some test instances do not have 30 historical flight cycles. For these test instances we apply zero padding [29], i.e., we set all the sensor measurements of the missing flight cycles to zero. This technique is very common in image processing. Moreover, we use a piece-wise linear RUL target function [24,30,31] with $R_{\text{early}} = 125$ flight cycles, i.e, the target RUL is $R_{\text{early}} = 125$ flight cycles when the actual RUL is larger than 125 flight cycles.

3.2. Mean RUL prognostics

Table 3 shows the Root Mean Square Error (RMSE) [33] with the mean RUL prediction (see Eq. (13)) for the four data subsets. The RMSE of the mean RUL prediction is a higher for subset FD002 and FD004, probably due to the multiple operating modes.

Table 3 also shows the results of existing studies employing various machine learning algorithms for the same dataset. The performance of

Table 3
RMSE for RUL prognostics using C-MAPSS and various machine learning algorithms. Here, $R_{\text{early}} = 130$ in [34], and $R_{\text{early}} = 125$ in the other considered studies. The best results are denoted in bold.

	FD 001	FD 002	FD 003	FD 004
Our approach	12.42	**13.72**	12.16	15.95
CNN [24]	12.61	22.36	12.64	23.31
LSTM-MLSA [31]	11.57	14.02	12.13	17.21
CNN-LSTM [35]	**11.17**	–	**9.99**	–
HAGCN [34]	11.93	15.05	11.53	**15.74**
HDNN [36]	13.02	15.24	12.22	18.17
MPHD-NN [37]	–	14.25	–	16.44

Table 4
α-Coverage (α-C) and α-Mean width (α-MW, in flight cycles) for the RUL prognostics of the engines in the C-MAPSS dataset.

		FD 001	FD 002	FD 003	FD 004
$\alpha = 0.50$	α-C	0.54	0.51	0.57	0.52
	α-MW	16.3	15.2	16.7	16.8
$\alpha = 0.90$	α-C	0.91	0.85	0.92	0.85
	α-MW	39.2	36.1	40.3	40.1
$\alpha = 0.95$	α-C	0.95	0.89	0.97	0.90
	α-MW	46.4	42.4	47.6	47.3

our RUL prognostic method with respect to the mean RUL is comparable to the state-of-the art solutions, especially for data subset FD002 and FD004. Potential contributing factors to the good performance of our approach are that we consider larger sizes of the window for FD002 and FD004 than most existing studies, and that we normalize the measurements with respect to the operating conditions (see Eq. (1)).

3.3. PDF of the RUL prognostics

Instead of predicting only one number for the RUL, however, we predict the PDF of the RUL. Fig. 4 shows the PDF of the RUL for two test instances of data subset FD004. For test instance 67 (Fig. 4(a)), the mean RUL prediction is close to the actual RUL. The PDF, however, is very wide. For test instance 38 (Fig. 4(b)), the mean RUL prediction is far away from the actual RUL. Moreover, the actual RUL falls outside the predicted PDF, even though this PDF is very wide as well.

The RMSE only evaluates the mean RUL prediction. We therefore use the α-Coverage and the reliability diagram to evaluate the reliability of the PDF of the RUL, i.e., how well the predicted probabilities match with the observed outcomes [10,38]. The α-Coverage is defined as [10]:

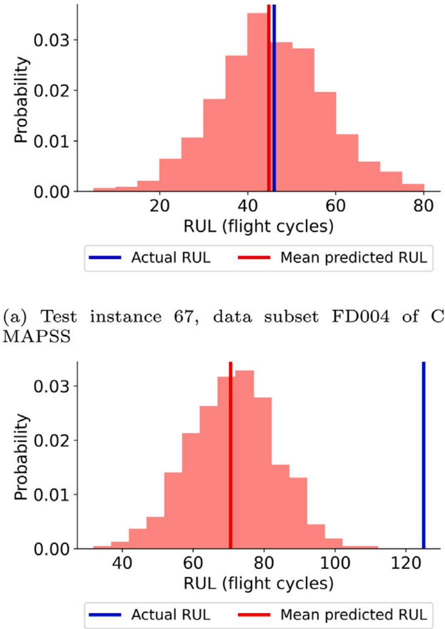
$$\alpha\text{-Coverage} = \frac{1}{D}\sum_{i=1}^{D} \mathcal{I}(\alpha)_i, \tag{14}$$

$$\text{with } \mathcal{I}(\alpha)_i = \begin{cases} 1, & y_i \in [\hat{y}_i^{0.5-0.5\alpha}, \hat{y}_i^{0.5+0.5\alpha}] \\ 0, & \text{Otherwise,} \end{cases}$$

where $\alpha \in [0,1]$ is an user-defined parameter, $\hat{y}_i^k$ is the kth percentile of the estimated RUL distribution for test instance i, and D is the number test instances. $[\hat{y}_i^{0.5-0.5\alpha}, \hat{y}_i^{0.5-0.5\alpha}]$ is thus the α percent confidence interval around the median of the PDF of test instance i. The closer the α-Coverage is to α, the more reliable the RUL predictions are. Related to this metric is the α-Mean width, which is the mean width in flight cycles of the confidence intervals at α [10,38]:

$$\alpha\text{-Mean width} = \frac{1}{D}\sum_{i=1}^{D} (\hat{y}_i^{0.5+0.5\alpha} - \hat{y}_i^{0.5-0.5\alpha}). \tag{15}$$

Table 4 shows the α-Coverage for $\alpha \in \{0.5, 0.9, 0.95\}$. As an example, we illustrate the $\alpha = 0.9$ coverage. With $\alpha = 0.9$, we obtain the confidence interval $[\hat{y}_i^{0.05}, \hat{y}_i^{0.95}]$, where $\hat{y}_i^{0.05}$ and $\hat{y}_i^{0.95}$ are the RUL predictions belonging to the 5%th and the 95%th percentile respectively. If we estimate the RUL and the corresponding 90% confidence interval for test instance i a large number of times, we thus expect that $\alpha = 90\%$ of the resulting confidence intervals contains the actual RUL. To calculate the α-Coverage, we construct this confidence interval with width $\alpha = 0.9$ for all D test instances of each data subset. It is expected that for $\alpha = 90\%$ of the test instances, the actual RUL y_i falls within the considered confidence interval. For data subset FD003, the actual RUL y_i falls within the confidence interval for 92% of the test instances. This means that the *uncertainty* for data subset FD003 and $\alpha = 0.9$ is slightly *overestimated*. In contrast, for data subset FD002 and FD004, the α-Coverage equals 85%, i.e., the actual RUL y_i of 85% of the test instances falls within the considered confidence interval. This means that the *uncertainty* for data subsets FD002 and FD004 and $\alpha = 0.9$ is *underestimated*.



(a) Test instance 67, data subset FD004 of C-MAPSS

(b) Test instance 38, data subset FD004 of C-MAPSS

Fig. 4. Histogram of the predicted RUL of two test instances.

In general, the α-Coverage is close to α for all subsets and all the considered values for α (see Table 4). This means that the predicted probabilities match well with the observed outcomes, and the RUL prognostics are thus reliable. However, the mean widths of the confidence interval are quite large, as also observed in Fig. 4. The uncertainty of the RUL prognostics is thus large, despite the reliability of the prognostics.

Fig. 5 shows the reliability diagram of the four subsets of C-MAPSS [10]. Here, $C(\alpha)_i = \{\alpha\text{-Coverage}, \alpha \in \{0.00, 0.01, 0.02, \ldots, 1.00\}\}$ is the reliability curve of subset $i \in$ {FD001, FD002, FD003, FD004}. Moreover $\tilde{C}(\alpha) = \alpha, \alpha \in \{0.00, 0.01, \ldots, 1.00\}$ is the ideal curve, i.e., the curve where the coverage $C(\alpha) = \alpha$. When the reliability curve is beneath the ideal curve for a certain α, then the uncertainty is underestimated at this value for α. In contrast, when the reliability curve is above the ideal curve for a certain α, the uncertainty is overestimated at this value for α. Fig. 5 shows that the reliability curves of all four data subsets are close to the ideal curve. Thus, the uncertainty of the RUL prognostics is well estimated.

4. Maintenance scheduling

In this section, we propose a model for optimal replacement of a component based on the probabilistic RUL prognostics and the expected

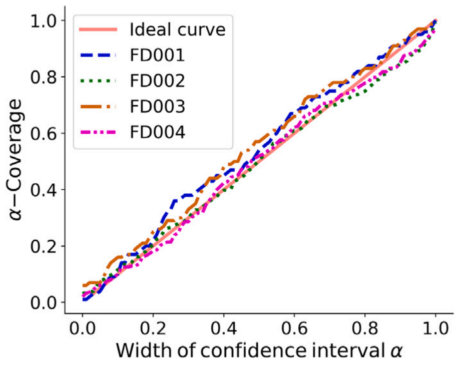


Fig. 5. Reliability diagram for the four subsets of C-MAPSS.

costs associated with maintenance. For the maintenance planning of a single component, we pose the problem of identifying an optimal replacement time as a renewal–reward process. For the maintenance planning of multiple components, we propose an integer linear program that additionally takes into account the availability of maintenance slots and the capacity of these slots.

4.1. Single-component replacement using probabilistic RUL prognostics and renewal–reward processes

We pose the problem of single-component replacement as a renewal–reward process. Let $\{N(t), t \geq 0\}$ be a renewal process where the process regenerates when a component is replaced [39]. Let C_n be the cost incurred during the nth renewal cycle, due to a replacement of the component, and let L_n be the length of the nth cycle, i.e., the time between the nth and the $(n-1)$ replacement. In our case, an nth component is thus used for a L_n amount of time. Defining $C(t), t \in \{1, 2, \ldots\}$ as the cumulative cost incurred up to time t, we have that [39]:

$$\lim_{t \to \infty} \frac{C(t)}{t} = \frac{\mathbb{E}[C_1]}{\mathbb{E}[L_1]}.$$

To determine an optimal replacement time, we thus analyze the long-term average cost per unit of time:

$$\frac{\mathbb{E}(\text{cost incurred during one cycle})}{\mathbb{E}(\text{length of one cycle})}. \tag{16}$$

We assume that at some present moment, a component (aircraft engine) has been used for k time steps. At this present moment k, an optimal replacement moment that minimizes Eq. (16) is determined. If a preventive replacement is scheduled at $k + t_k$ time steps (i.e., in t_k time steps from the present moment), and the component does not fail until time $k + t_k$, then a cost c_r is incurred for this preventive replacement. Here, the risk is that the component may fail at some time $k + j, 0 \leq j < t_k$. If the component indeed fails after j time steps, than a cost $c_\mathrm{f} > c_\mathrm{r}$ is incurred and this component is immediately replaced by a new one. The component is thus either (i) replaced upon failure at a cost c_f, or (ii) is preventively replaced at cost a c_r after using it for $k + t_k$ time-steps.

The component is continuously monitored by sensors. The sensor measurements of the first k time steps of usage are available. In Section 2, $x_i^v, i \in \{1, 2, \ldots, k\}$ denotes this series of measurements up to time step k. Based on these sensor measurements, the probability that the RUL of the component is i time steps, $i \geq 0$, is estimated using Convolutional Neural Networks with Monte Carlo dropout (see Section 2). Let $\phi_k(i)$ denote the probability that, after being used for k time steps, the component has a RUL of exactly i time steps, $i \geq 0$, as estimated in Section 5.1.

Let $C(k, t_k)$ denote the replacement costs of the component and let $L(k, t_k)$ denote the lifetime of the component, given that this component has already been used for k time steps. Here, the component is replaced(i) upon failure, or (ii) preventively after being used for $k + t_k$ time-steps. We are interested in identifying the optimal value of t_k, i.e., an optimal time to replace the component, which minimizes the long-term average cost per unit of time (see Eq. (16)):

$$\frac{\mathbb{E}[C(k, t_k)]}{\mathbb{E}[L(k, t_k)]}, \tag{17}$$

where the expected replacement cost of the component is:

$$\mathbb{E}[C(k, t_k)] = c_\mathrm{f} \sum_{i=0}^{t_k - 1} \phi_k(i) + c_\mathrm{r} \left(1 - \sum_{i=0}^{t_k - 1} \phi_k(i)\right), \tag{18}$$

and the expected lifetime of the component is:

$$\mathbb{E}[L(k, t_k)] = k + \sum_{i=0}^{t_k - 1} i \cdot \phi_k(i) + t_k \left(1 - \sum_{i=0}^{t_k - 1} \phi_k(i)\right). \tag{19}$$

Let t_k^* denote the optimal value for t_k, that minimizes Eq. (17). In general, if there is a high probability that the RUL is zero, then t_k^* is also expected to tend to zero. Conversely, if there is a high probability that the RUL is large, then t_k^* is also expected to be large.

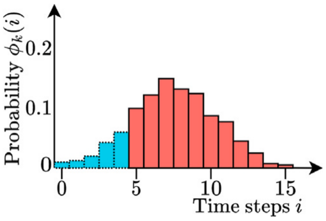


Fig. 6. An example of the probability $\phi_k(i)$ for a single component, as estimated at time $k = 100$.

Fig. 6 shows an example of the probability $\phi_k(i)$ of a component of having a RUL of i time steps, given that it has already been used for $k = 100$ time steps. We evaluate the expected costs over the expected lifetime if $t_k = 5$, i.e., if we replace the component at day $k + t_k = 105$. The probability that the component fails in the next $t_k = 5$ days equals $\sum_{i=0}^{4} \phi_k(i) = 0.14$ (blue, dotted area in Fig. 6). The expected cost when replacing the component at day 105 is $c_\mathrm{f} \cdot 0.14 + c_\mathrm{r}(1 - 0.14)$. The expected lifetime is $100 + 0.41 + 5 \cdot (1 - 0.14) = 104.71$.

4.2. Multi-component replacement using probabilistic RUL prognostics

We now consider the maintenance planning for multiple components. Let V denote the set of components (aircraft engines). Let d_p denote the present day, and d_0^v denote the installation day of a component $v \in V$. Let $k_\mathrm{p} = d_\mathrm{p} - d_0^v$ denote the usage time of component $v \in V$ at present day d_p.

At present day d_p, a probabilistic RUL prognostic, i.e., the estimated PDF of the RUL, is available for each component $v \in V$ (see Section 2). Let $\phi_{k_\mathrm{p}}^v(i)$ denote the estimated probability that the RUL of component $v \in V$ is exactly i flight cycles, after being used for k_p flight cycles. Again, $\phi_{k_\mathrm{p}}^v(i)$ is estimated using CNN with Monte Carlo dropout in Section 2. We assume that each engine performs one flight cycle per day.

A component can be replaced in dedicated maintenance slots, i.e. days when the component is available for maintenance, and the maintenance facility and required equipment are available. In the case of an aircraft engine, the aircraft is on the ground during these maintenance slots, and the aircraft maintenance hangar and equipment are available [4]. At present day d_p, maintenance slots are known up to l days in advance. Let S^v be the set of maintenance slots available for a component $v \in V$ in the period $[d_\mathrm{p}, d_\mathrm{p} + l)$. The set S^v is specific to component v.

We also consider generic maintenance slots, i.e., a day during which any component $v \in V$ can be replaced, but at an additional high cost

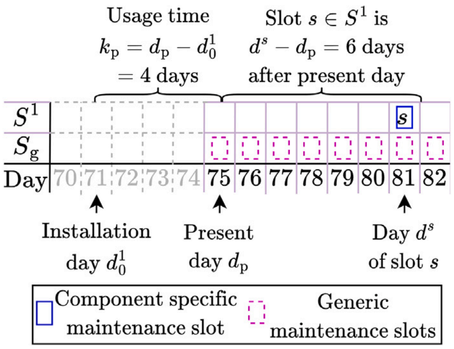
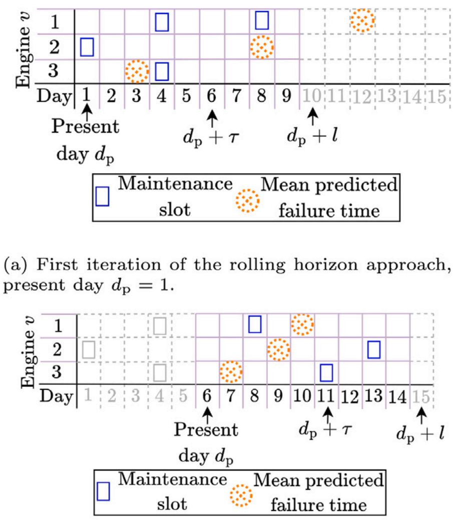


Fig. 7. Example of a multi-component replacement problem for a single engine $v = 1$.



(a) First iteration of the rolling horizon approach, present day $d_p = 1$.

(b) Second iteration of the rolling horizon approach, present day $d_p = 6$.

Fig. 8. Schematic example of two iterations of the rolling horizon approach, with $|V| = 3$ engines, $\tau = 5$ days, $l = 10$ days, and present days $d_p = 1$ and $d_p = 6$.

c_g. One generic maintenance slot is available per day. Let S_g be the set of generic maintenance slots available in the period $[d_p, d_p + l)$.

Let d^s be the day belonging to slot $s, s \in S_g \cup S^v, v \in V$, and let S^d be the set of all slots at day $d \in [d_p, d_p + l)$, i.e., all $s \in S_g \cup S^v, v \in V$: $d^s = d$. Let $t_p^s = d^s - d_p$ denote the number of days a maintenance slot s is available after present day d_p. Last, due to limited resources, at most h components can be replaced per day. Fig. 7 shows an example of the notation in the multi-component maintenance planning for an engine $v = 1$.

We analyze the maintenance planning for a period of T days using a rolling horizon approach. First, at present day $d_p = 1$, the maintenance planning is made for the time-window $[d_p, d_p + l)$. Here, the RUL prognostics are obtained at present day $d_p = 1$. Then, the replacements planned in the first $\tau, \tau < l$ days of this maintenance planning are executed. Also, it is assumed that components that fail in the first τ days are immediately replaced. Next, we roll to the next time window. We update the present day $d_p := 1 + \tau$. We also update the RUL prognostics. With these new RUL prognostics, a maintenance planning is now made for the time-window $[d_p, d_p + l)$, i.e., for $[1 + \tau, 1 + \tau + l)$. This is repeated until the maintenance planning of T days is executed.

An example of two iterations of a rolling horizon approach is in Fig. 8, with $|V| = 3$ engines. At the first present day $d_p = 1$, we know the maintenance slots of the next $l = 10$ days ahead. Using the RUL prognostics, we thus make a maintenance planning for the next $l = 10$ days (Fig. 8(a)). Then, we fix the decisions of the first $\tau = 5$ days, update the RUL prognostics and move forward to present day $d_p = 1 + \tau = 6$ (Fig. 8(b)). Note that the mean predicted failure time changes between present day $d_p = 1$ and $d_p = 6$, due to the updating of the RUL prognostics.

We propose the following Integer Linear Program (ILP) to plan component replacements at present day d_p for the time-window $[d_p, d_p + l)$.

Costs

Let c_s^v denote the expected costs over the expected lifetime of the component (see Eq. (17)) when component v is replaced in slot s, with $d_p \leq d^s < d_p + l$. We calculate c_s^v by dividing the expected costs of this replacement by the expected lifetime, given that component v is replaced(i) upon failure or (ii) preventively in slot s, whichever comes first. The expected costs consists of the expected failure costs, the expected preventive replacement costs, and the expected costs of using the generic slot. Formally,

$$c_s^v = \frac{c_f \sum_{i=0}^{t_p^s - 1} \phi_{k_p}^v(i) + \left(c_r + c_g \mathcal{I}_g(s)\right)\left(1 - \sum_{i=0}^{t_p^s - 1} \phi_{k_p}^v(i)\right)}{(d_p - d_0^v) + \sum_{i=0}^{t_p^s - 1} i \phi_{k_p}^v(i) + t_p^s \left(1 - \sum_{i=0}^{t_s^p - 1} \phi_{k_p}^v(i)\right)}, \tag{20}$$

where

$$\mathcal{I}_g(s) = \begin{cases} 1 & s \in S_g \\ 0 & \text{Other,} \end{cases}. \tag{21}$$

Let c_{DN}^v denote the expected cost over the expected lifetime of component $v \in V$ within the period $[d_p, d_p + l)$, when no replacement is planned for this component in this period, i.e., when we do nothing (DN). In other words, the replacement of this component is postponed to after day $d_p + l$, and we thus only incur costs in the period $[d_p, d_p + l)$ if the component fails in this period. Formally,

$$c_{DN}^v = \frac{c_f \sum_{i=0}^{l-1} \phi_{k_p}^v(i)}{(d_p - d_0^v) + \sum_{i=0}^{l-1} i \phi_{k_p}^v(i) + l \left(1 - \sum_{i=0}^{l-1} \phi_{k_p}^v(i)\right)}. \tag{22}$$

Decision variables

We consider the following decision variable:

$$x_{vs} = \begin{cases} 1, & \text{component } v \in V \text{ is replaced} \\ & \text{in slot } s \in S^v \cup S_g \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Objective

We aim to minimize the expected costs over the expected lifetime, i.e.,:

$$\text{min.} \sum_{v \in V} \left(\sum_{s \in S^v \cup S_g} c_s^v x_{vs} + c_{DN}^v \left(1 - \sum_{s \in S^v \cup S_g} x_{vs} \right) \right) \tag{24}$$

Constraints

A component $v \in V$ can be scheduled for replacement at most once in a time-window, i.e,:

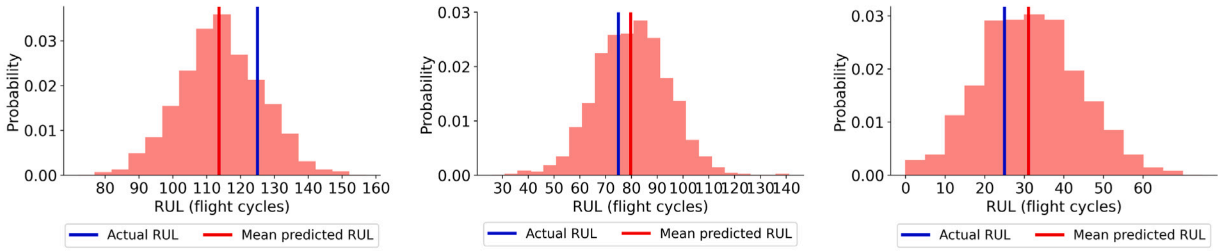
$$\sum_{s \in S^v \cup S_g} x_{vs} \leq 1, \quad \forall v \in V \tag{25}$$

At most h components can be scheduled for replacement during one day:

$$\sum_{v \in V} \sum_{s \in S^d} x_{vs} \leq h, \quad \forall d \in [d_p, d_p + l) \tag{26}$$

Lastly,

$$x_{vs} \in \{0, 1\} \quad \forall v \in V, \forall s \in \left(S^v \cup S_g\right). \tag{27}$$



(a) Actual RUL = 125 flight cycles

(b) Actual RUL = 75 flight cycles

(c) Actual RUL = 25 flight cycles

Fig. 9. Predicted PDF of the RUL of engine 2 of subset FD001, C-MAPSS. This is the first engine selected from subset FD001 for maintenance planning.

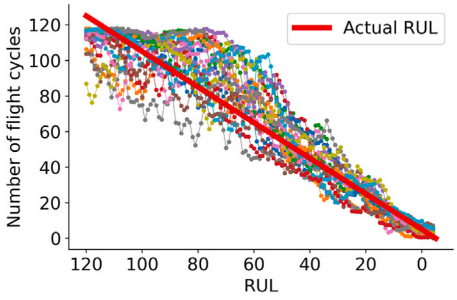


Fig. 10. The mean RUL prediction for the last 125 flight cycles for the engines selected for maintenance planning from subset FD001, C-MAPSS.

Table 5
The RMSE, α-Coverage (α-C) and α-Mean width (α-MW) for the RUL prognostics of the engines selected for maintenance planning from the C-MAPSS training sets.

		FD 001	FD 002	FD 003	FD 004
	RMSE	13.06	15.15	13.58	15.93
$\alpha = 0.50$	α-C	0.50	0.47	0.60	0.60
	α-MW	16.3	15.5	16.9	16.5
$\alpha = 0.90$	α-C	0.90	0.82	0.89	0.88
	α-MW	39.2	37.5	40.7	39.9
$\alpha = 0.95$	α-C	0.94	0.88	0.93	0.91
	α-MW	46.4	44.4	48.1	47.2

5. Results — maintenance planning for turbofan engines

In this section, we illustrate our maintenance planning methodology proposed in Sections 2 and 4 for aircraft turbofan engines.

5.1. Results — probabilistic RUL prognostics for the maintenance planning of turbofan engines

In Section 3 we have presented the RUL prognostics for the engines in the C-MAPSS test sets. For these test engines, the measurements stop at some moment before failure, i.e., these are not complete series until the moment of failure. For maintenance planning, however, we need complete series of measurements up to the moment of failure, i.e., we need run-to-failure instances. The engines in the C-MAPSS training set have such complete run-to-failure series of measurements. Thus, we use the engines in the C-MAPSS training set for maintenance planning. We first randomly select 80% of the engines of each training set (568 engines in total) [4,20] to train the CNNs (see Section 2). For the remaining 20% of engines (a total of 141 engines), we generate RUL prognostics using the trained CNNs and Monte Carlo dropout. The RUL prognostics of these 141 engines are then used to analyze the maintenance planning model proposed in Section 4.

Fig. 9 shows the obtained PDF of the RUL of engine 2 of subset FD001 when the actual RUL is 125, 75 and 25 flight cycles. Engine 2 is the first engine randomly selected from FD001 for maintenance planning. The PDF of the RUL of this engine is centered around the actual RUL for all three moments in time. However, the mean RUL prediction is closer to the actual RUL when the actual RUL is 75 or 25 flight cycles.

Fig. 10 shows the mean predicted RUL for the last 125 flight cycles before failure for all engines selected for maintenance planning from FD001. Here, each colored line shows the RUL predictions belonging to one engine. After each flight, the mean predicted RUL is updated for each engine. Fig. 10 shows the mean predicted RUL from 125 flight cycles before failure (i.e., the actual RUL is 125 flight cycles), until failure (i.e., the actual RUL is 0). For all engines, the mean RUL prediction slightly underestimates the actual RUL when the actual RUL is 125 flight cycles. For a few engines, moreover, the mean predicted RUL deviates substantially from the actual RUL when the actual RUL is still large. However, the mean predicted RUL always converges to the actual RUL when an engine becomes closer to failure.

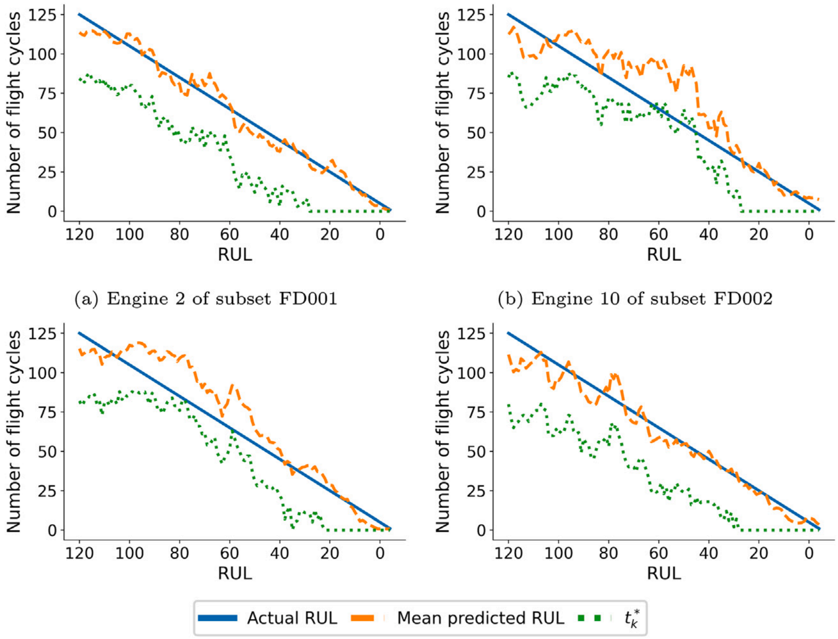
Table 5 shows the metrics of the RUL prognostics with all the 141 engines selected for maintenance planning. The RMSE is higher than for the test instances in Section 2, while the α-Coverage diverges more from α. This is as expected, since there are less failure instances available to train the CNN.

5.2. Results — Single-engine maintenance planning

In this section, we discuss the optimal moment for replacement $k+t_k^*$ from Section 4.1, Eq. (17). Table 6 shows for four engines the optimal moment for replacement $k+t_k^*$, at five moments during the life of each engine. Engine 2, 10, 2, and 3 is the first randomly selected engine for maintenance planning from C-MAPPS subset FD001, FD002, FD003, and FD004, respectively. As an example, consider engine 2 of subset FD001 when the actual RUL is 125 flight cycles. At this moment, the engine has been used for $k = 162$ flight cycles. The mean predicted RUL is 113.7 flight cycles, and the 99% confidence interval of the predicted RUL is $[83, 144]$. Given the current usage of 162 flight cycles, it is optimal to replace this engine after an additional of $t_k^* = 83$ flight cycles, i.e., at the lower bound of the 99% confidence interval of the predicted RUL. The optimal moment of a preventive replacement of this engine is thus $k + t_k^* = 162 + 83 = 245$ flight cycles.

The optimal moment of replacement $k + t_k^*$ varies over time, since the RUL prognostics are updated after each flight cycle. In Table 6, t_k^* is close to the lower bound of the 99% confidence interval of the RUL prediction. Here, t_k^* is thus smaller if the RUL predictions are more uncertain, i.e., if the confidence intervals are wider, and the lower bound of the 99% confidence interval is thus smaller. On the other hand, the more certain the RUL predictions are, the closer t_k^* is to the mean RUL prediction.

Fig. 11 further illustrates the optimal number of flight cycles t_k^* for the engines in Table 6. For all four engines, t_k^* follows the same trend as the mean predicted RUL. Moreover, when the actual RUL is 25 flight cycles or less, it is optimal to immediately perform a preventive



(a) Engine 2 of subset FD001

(b) Engine 10 of subset FD002

(c) Engine 2 of subset FD003

(d) Engine 3 of subset FD004

Fig. 11. Actual RUL, Mean Predicted RUL and t_k^*. Engine 2, 10, 2 and 3 are the first randomly selected engines for maintenance planning from FD001, FD002, FD003 and FD004, respectively.

Table 6
The actual RUL, the number of cycles the engine has already been in use k, the mean predicted RUL, the optimal number of flight cycles to use the engine before preventive replacement t_k^*, and the optimal replacement moment of the engine $k + t_k^*$. The first engines randomly selected for maintenance planning from FD001, FD002, FD003, FD004 are chosen for illustration.

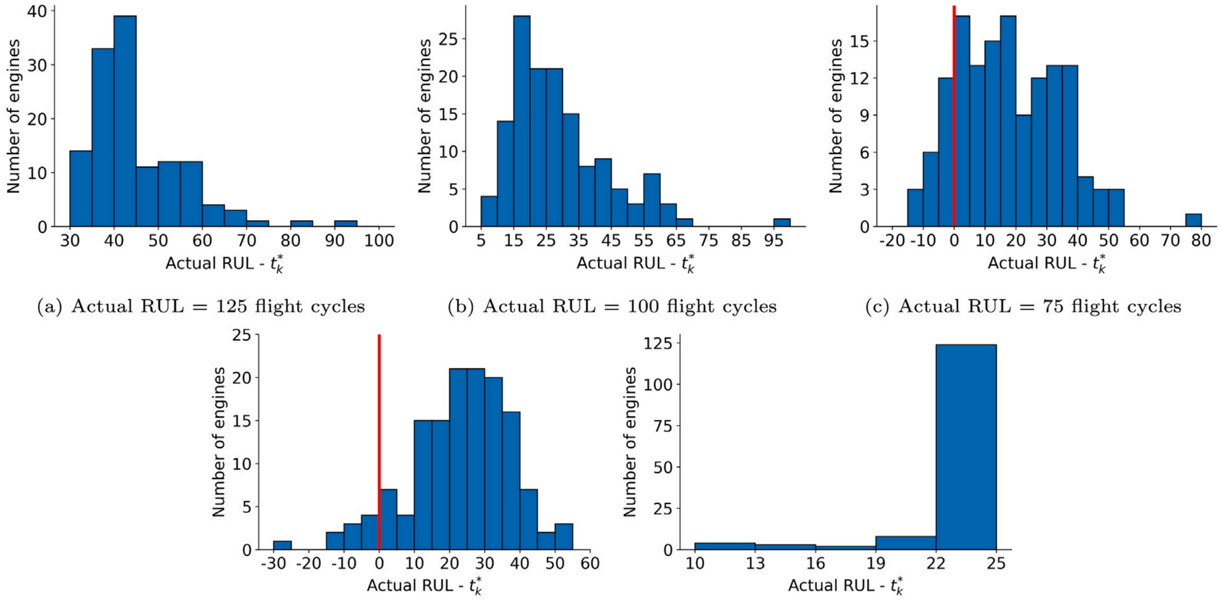
Actual RUL	k	Mean predicted RUL	99% CI of the RUL	t_k^*	$k + t_k^*$
Engine 2 - subset FD001 **True lifetime = 287 flight cycles**					
125	162	113.7	[83,144]	83	245
100	187	99.3	[66, 128]	66	253
75	212	79.7	[40,114]	40	252
50	237	45.5	[14,75]	13	250
25	262	31	[0,64]	0	262
Engine 10 of subset FD002 **True lifetime = 184 flight cycles**					
125	59	112.5	[82, 141]	85	144
100	84	114.2	[85, 146]	85	169
75	109	91.6	[55, 127]	58	167
50	134	81.1	[49, 114]	48	182
25	159	30.4	[0,60]	0	159
Engine 2 of subset FD003 **True lifetime = 253 flight cycles**					
125	128	115.3	[83, 146]	82	210
100	153	118.4	[89,151]	88	241
75	178	92.9	[67,125]	62	240
50	203	59.3	[27,92]	27	230
25	228	28.4	[0,57]	0	227
Engine 3 of subset FD004 **True lifetime = 307 flight cycles**					
125	182	111.5	[81, 144]	80	262
100	207	101.2	[67,135]	72	279
75	232	68.1	[40,100]	40	272
50	257	50.4	[23,79]	21	278
25	282	23.2	[0,53]	0	282

replacement for all four engines, i.e., $t_k^* = 0$. This is because, at this moment, we predict with a quite high probability that the RUL of the engine is 0 flight cycles (see the 99% confidence intervals of the RUL in Table 6).

For engine 10 of subset FD002 (Fig. 11(b)), the mean predicted RUL is larger than the actual RUL, when the actual RUL is around 60 flight cycles. Based on these prognostics, Eq. (17) is minimized when the engine is replaced after the failure time. For example, the mean predicted RUL is 93 flight cycles, with a 100% confidence interval of $[62, 123]$, while the actual RUL is only 55 flight cycles. The predicted probability that the engine fails on or before the actual RUL is thus zero. The optimal replacement moment, based on the RUL prognostics and Eq. (17), is therefore $t_k^* = 65$ flight cycles, i.e., after the failure time of the engine. Around 40 flight cycles before failure, it is again optimal to replace engine 10 of FD002 before the failure time.

Fig. 12 shows the Actual RUL $- t_k^*$ for all 141 engines considered for maintenance planning, at five moments during the engines' life. As an example, for engine 2 of subset FD001 in Table 6, it is optimal to replace the engine after $t_k^* = 83$ flight cycles, while the actual RUL is 125 flight cycles. The Actual RUL $- t_k^* = 125 - 83 = 42$ flight cycles, i.e., the optimal moment of replacement is 42 flight cycles before engine failure.

When the actual RUL is 125 cycles or when the actual RUL is 100 cycles, then the optimal moment of replacement is always before the engine fails. When the actual RUL is 125 cycles, it is optimal to replace each engine at least 30 cycles before its failure. This is because the RUL is slightly underestimated when the actual RUL is 125 cycles (see Fig. 10), and the lower bounds of the 95% and 99% confidence intervals are always much smaller than 125 cycles. When the actual RUL is 75 or 50 cycles, however, the RUL prognostics sometimes overestimate the actual RUL: For some engines, the predicted probability that the RUL is equal to or smaller than the actual RUL is even zero. The optimal moment of replacement for some engines therefore falls after the engine failure date when the actual RUL is 75 or 50 cycles. When the actual RUL is 25 cycles (Fig. 12(e)), however, it is optimal to immediately perform a preventive replacement for most engines, i.e., $t_k^* = 0$. Moreover, the optimal maintenance moment is at least 10 days before the engine fails.



(a) Actual RUL = 125 flight cycles (b) Actual RUL = 100 flight cycles (c) Actual RUL = 75 flight cycles

(d) Actual RUL = 50 flight cycles (e) Actual RUL = 25 flight cycles

Fig. 12. Histogram of (Actual RUL - t_k^*), for all engines selected for maintenance planning from the C-MAPSS dataset.

5.3. Results - Multi-engine maintenance planning

In this section, we plan maintenance for multiple engines using the methodology in Section 4.2. We consider a fleet of $|V| = 50$ engines, which are randomly selected from the 141 engines considered for maintenance. Maintenance slots are randomly sampled with a frequency of one per 10–20 days [4]. We assume that at most $h = 1$ engine per day can be maintained, and that maintenance slots are known $l = 50$ days ahead. We consider a preventive replacement cost $c_r = 10$, a cost of using a generic slot $c_g = 10$ and a cost of a failure are $c_f = 50$. As mentioned before, we also assume that each engine performs one flight cycle per day.

We analyze the maintenance planning of the engines for a period of $T = 10 \times 365$ days (i.e., ten years). We consider time windows of $\tau = 10$ days for maintenance planning. Last, we assume that a failed engine is immediately replaced with a new engine. We implement the ILP in Python using Gurobi, on a computer with an Intel Core i7 processor at 2.11 GHz and 8Gb RAM. It requires 109 s to create a maintenance planning for a period of ten years.

Fig. 13 shows the resulting maintenance planning at present days $d_p = 791$ and $d_p = 801$. At day 791, a replacement is planned for 6 engines, namely engines 9, 14, 16, 29, 40 and 41. For each engine with a planned replacement, we show the available maintenance slots (blue squares), the actual failure time (red cross in a circle) and the mean predicted failure time (orange cross in a dotted circle). For an engine $v \in V$ at present day d_p, we use $t^*_{k_{p,v}}$ to denote the optimal t^*_k from Eq. (17) in Section 4.1. Here, $d_p + t^*_{k_{p,v}}$ thus denotes the optimal replacement moment from the renewal–reward process with a single-component (blue cross).

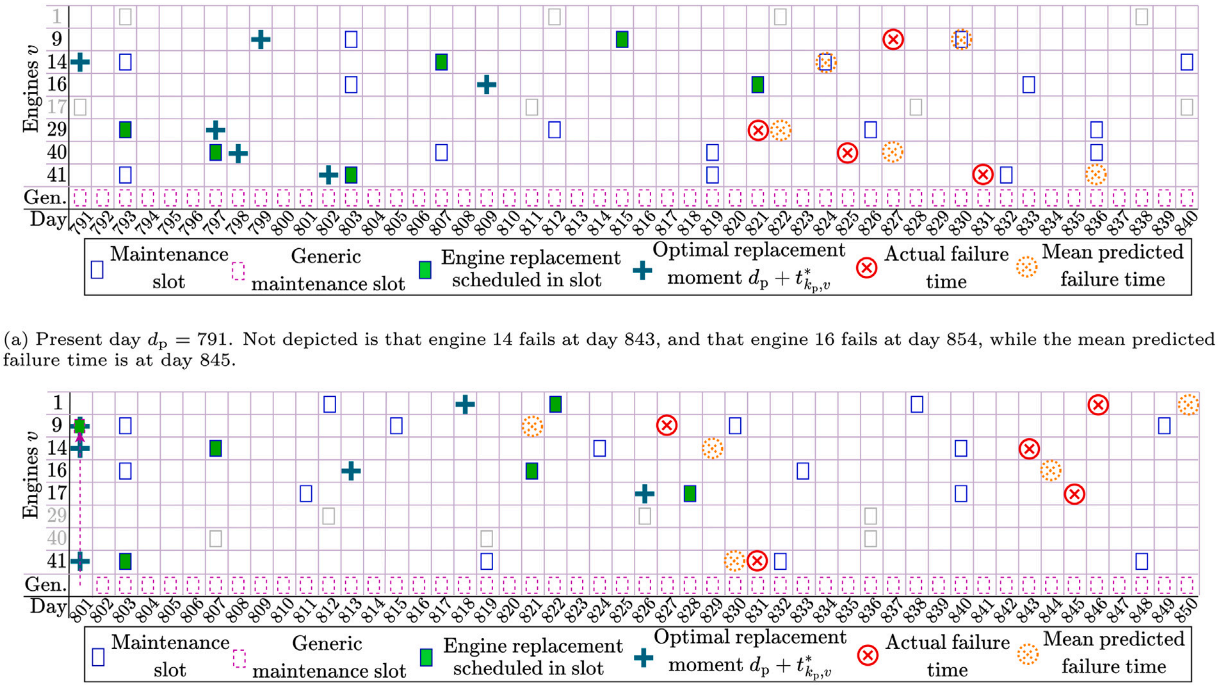
For engine 29, 40 and 41, the engine replacement is planned close to the optimal moment from the renewal–reward process. For engines 29 and 40, the replacement is planned 4 and 1 day(s) before this optimal moment, respectively. For engine 41, the replacement is planned 1 day after this optimal moment. For engine 9, 14 and 16, however, the replacement is planned 16, 16 and 12 days after the optimal moment from the single-component problem, respectively. For all these three engines, an earlier maintenance slot closer to the optimal moment from the single-component is available. However, engine 29 or engine 41 is already replaced at the days of these maintenance slots. Since we assume that at most one engine per day can be maintained, we cannot maintain engine 9, 14 and 16 at these days as well.

Using the rolling horizon approach, the maintenance decisions of the first $\tau = 10$ days of the maintenance planning are executed. Engine 29 and engine 40 are replaced at day 793 and day 797 respectively, and no engine failure occurs. The RUL prognostics are updated, and a new maintenance planning is then made at present day 801.

At day 801, the replacements for engine 14, 16 and 41 remain planned in the same maintenance slots. The updated prognostics of engine 9, however, indicate that this engine could fail very soon: At present day 791, the first available maintenance slot that fulfilled the capacity constraint was at day 815. The probability that engine 9 would fail before day 815 was estimated at 7.7%. The costs of using a generic slot (10) thus exceeded the expected failure costs ($c_f \cdot 0.077 = 50 \cdot 0.077 = 7.7$). The replacement was therefore scheduled at day 815. At present day 801, however, the estimated probability that engine 9 fails before day 815 is 26.5%. The expected failure costs ($50 \cdot 0.265 = 13.25$) are thus higher than the cost of using a generic slot. A generic slot (purple, dotted square) is therefore used to replace this engine immediately at day 801. Additionally, a replacement is planned for engine 1 at day 822, four days after the optimal moment for a single component, and for engine 17 at day 829, two days after the optimal moment for a single component.

Let $d^{\text{ILP}}_{p,v}$ be the day of a scheduled replacement of engine $v \in V$, as planned at present day d_p using the ILP for multi-component maintenance planning. Fig. 14 gives a histogram of the optimal moment of replacement $d_p + t^*_{k_{p,v}}$ for the case of a single component, minus the scheduled replacement time $d^{\text{ILP}}_{p,v}$ from the multi-component problem. For example, in Fig. 13(a), it is optimal at present day $d_p = 791$ to replace engine $v = 41$ at day 802, i.e. $d_p + t^*_{k_{p,v}} = 791 + 11 = 802$. However, it is planned to replace engine $v = 41$ at day $d^{\text{ILP}}_{p,v} = 803$ instead. The replacement of engine 41 is thus scheduled $802 - 803 = 1$ day after the optimal moment.

Fig. 14 shows that the results obtained for the maintenance of multiple engines are similar to the results for single-engine maintenance: Most of the time, the optimal moment of replacement is the same in the single-component and the multi-component problem. The optimal replacement time obtained from the multi-component problem, however, can also deviate up to 20 days from the optimal replacement



(a) Present day $d_p = 791$. Not depicted is that engine 14 fails at day 843, and that engine 16 fails at day 854, while the mean predicted failure time is at day 845.

(b) Present day $d_p = 801$. Not depicted is that engine 16 fails at day 854, and that the mean predicted failure time of engine 17 is at day 852.

Fig. 13. Maintenance planning at present day $d_p = 791$ and day $d_p = 801$. Only the engines for which a replacement is planned during these two present days are depicted.

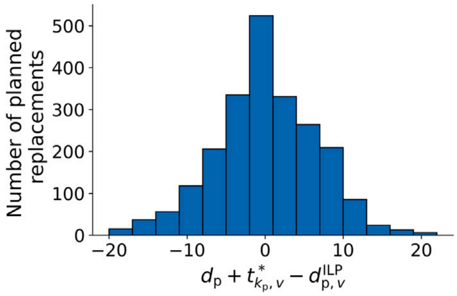


Fig. 14. The optimum replacement time $d_p + t^*_{k_p,v}$ for single-component replacement (see Eq. (17)) minus the scheduled replacement time $d^{\mathrm{ILP}}_{p,v}$ in the multi-component replacement problem.

time obtained from the single-component problem due to the limited availability of slots and the limit of h replacements per day.

5.4. Long-term performance: Maintenance with probabilistic RUL prognostics vs maintenance with perfect RUL prognostics and time-based maintenance

In this section, we compare three maintenance strategies: (i) the proposed maintenance strategy with probabilistic RUL prognostics (Section 4), (ii) a maintenance strategy with perfect RUL prognostics, and (iii) a Time-based maintenance strategy. The long-term performance of these maintenance strategies is analyzed by means of a Monte Carlo simulation.

Perfect RUL prognostics. For this maintenance strategy, we assume that we exactly know the failure time of the engine, i.e., that we have perfect RUL prognostics. For this strategy, we use the same ILP as in Section 4.2. However, we now input the actual RUL in Eqs. (20) and (22), instead of the probabilistic RUL prognostics.

Time-based maintenance. For this strategy, we determine the probability of the failure of an engine based on its current usage. In Section 5.1, we selected 568 engines to train the CNN, while the remaining 141 engines were used for maintenance planning. For the Time-based Maintenance strategy, we use the 568 engines to determine a *generic* histogram of the lifetime of engines (see Fig. 16).

For any of the 141 engines selected for maintenance planning, the probability that the lifetime of this engine is i flight cycles at the moment of installation, is the proportion of the 568 engines for which the lifetime is i flight cycles. For example, 2 out of the 568 engines have a lifetime of 231 flight cycles. When planning maintenance, we therefore estimate that, at the moment of installation, the probability that the lifetime of an engine is 231 flight cycles is $\frac{2}{568}$. Let $\tilde{\phi}(i)$ denote the probability that the lifetime of an engine is exactly i flight cycles. In our example, $\tilde{\phi}(231) = \frac{2}{568} \approx 0.0035$.

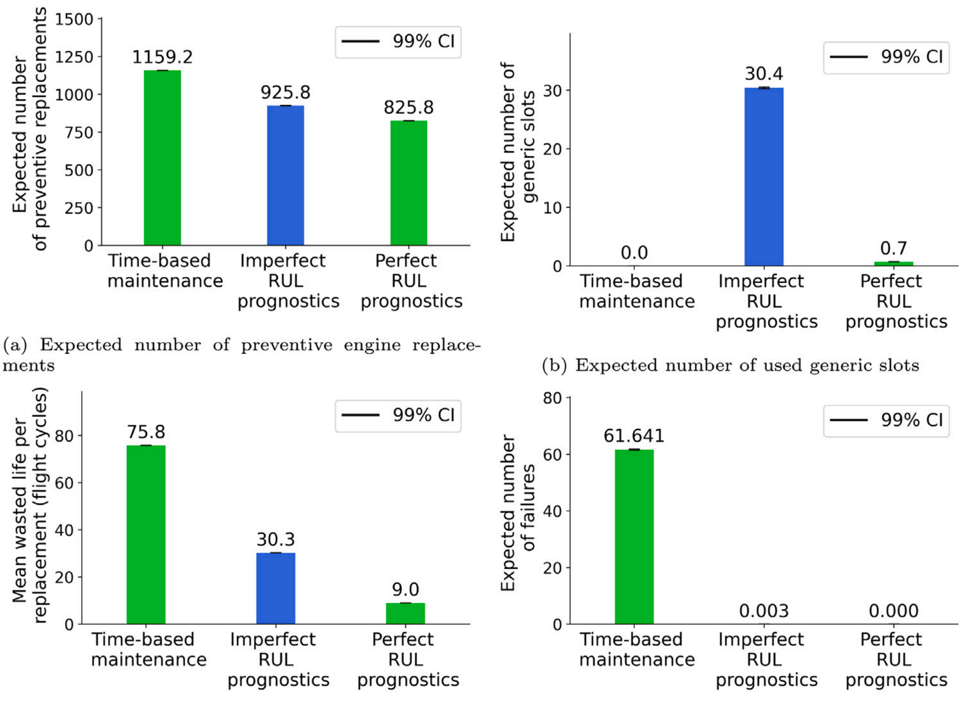
Consider that any engine of the total 141 engines has been used for k flight cycles. The lifetime L of this engine is thus larger than or equal to k cycles, i.e., $L \geq k$. Given that $L \geq k$, the conditional probability $\tilde{\phi}(k+i|L \geq k)$ that the lifetime $L = k+i$ flight cycles is solely based on the histogram in Fig. 16:

$$\tilde{\phi}(k+i|L \geq k) = \frac{\tilde{\phi}(L = k+i \cap L \geq k)}{\tilde{\phi}(L \geq k)} \tag{28}$$

$$= \frac{\tilde{\phi}(k+i)}{1 - \sum_{i=0}^{k-1} \tilde{\phi}(i)} \tag{29}$$

The difference between Section 5.1 and the Time-based Maintenance strategy is that under the Time-based Maintenance strategy, the probability of failure for *all* 141 engines is the *same, given an usage period of k cycles*.

As another example, assume that an engine has been used for $k = 220$ flight cycles. The probability $1 - \sum_{i=0}^{219} \tilde{\phi}(i)$ that the lifetime of an engine is 220 flight cycles or larger is 0.4296. The conditional probability that the lifetime L of this engine equals 231 flight cycles is thus $\tilde{\phi}(231|L \geq 220) = \frac{0.0035}{0.4296} = 0.008$. Using these conditional probabilities



(a) Expected number of preventive engine replacements

(b) Expected number of used generic slots

(c) Mean wasted life per replacement

(d) Expected number of failures

Fig. 15. The expected number of engine replacements, used generic slots and failures per ten years, and the mean wasted life per replacement, for(i) Time-based maintenance, (ii) maintenance with probabilistic, imperfect RUL prognostics and (iii) maintenance with perfect RUL prognostics. The 99% confidence interval of the mean (CI) is also given.

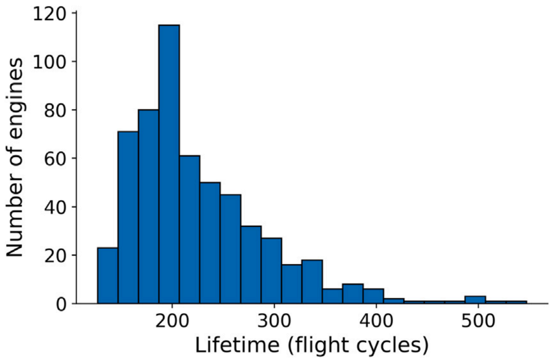


Fig. 16. Histogram of the lifetime of the 568 historical failure instances selected for training the CNN in Section 5.1.

$\tilde{\phi}(k+i|L \geq k)$, we plan the single-component and multi-component replacements.

Time-based single-component replacement

Now that $\tilde{\phi}(k+i|L \geq k)$ is determined based on the histogram in Fig. 16, Eq. (18) and (19) become:

$$\mathbb{E}[C(k,t_k)] = c_\mathrm{f} \sum_{i=0}^{t_k-1} \tilde{\phi}(k+i|L \geq k) + c_\mathrm{r}\left(1 - \sum_{i=0}^{t_k-1} \tilde{\phi}(k+i|L \geq k)\right),$$

$$\mathbb{E}[L(k,t_k)] = k + \sum_{i=0}^{t_k-1} i \cdot \tilde{\phi}(k+i|L \geq k) + t_k\left(1 - \sum_{i=0}^{t_k-1} \tilde{\phi}(k+i|L \geq k)\right).$$

Time-based multi-component replacement

We again plan the multi-component replacement using the ILP in Section 4.2. Let $k_\mathrm{p}^v = d_\mathrm{p} - d_0^v$ be the number of days engine $v \in V$ is in use at present day d_p.

Now that $\tilde{\phi}(k+i|L \geq k)$ is determined based on the histogram in Fig. 16, Eq. (20) becomes:

$$\frac{\text{Nominator}}{\text{Denominator}},$$

where

$$\text{Nominator} = c_\mathrm{f} \sum_{i=0}^{t_\mathrm{p}^s-1} \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v) + \left(c_\mathrm{r} + c_\mathrm{g}\mathcal{I}_\mathrm{g}(s)\right) \cdot \left(1 - \sum_{i=0}^{t_\mathrm{p}^s-1} \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v)\right),$$

$$\text{Denominator} = k_\mathrm{p}^v + \sum_{i=0}^{t_\mathrm{p}^s-1} i \cdot \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v) + t_\mathrm{p}^s\left(1 - \sum_{i=0}^{t_\mathrm{p}^s-1} \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v)\right).$$

Also,

$$c_\mathrm{DN}^v = \frac{c_\mathrm{f} \sum_{i=0}^{l-1} \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v)}{k_\mathrm{p}^v + \sum_{i=0}^{l-1} i \cdot \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v) + l\left(1 - \sum_{i=0}^{l-1} \tilde{\phi}(k_\mathrm{p}^v + i|L \geq k_\mathrm{p}^v)\right)}.$$

Long-term results. We evaluate the long-term performance of the three maintenance strategies by performing a Monte Carlo simulation with 10,000 simulation runs, where each run lasts 10 years. Fig. 15 shows the expected results of this simulation.

The expected number of replacements decreases by 32% when considering imperfect RUL prognostics instead of Time-based Maintenance. When using perfect instead of imperfect RUL prognostics, the expected number of replacements further decreases by 11% (see Fig. 15(a)).

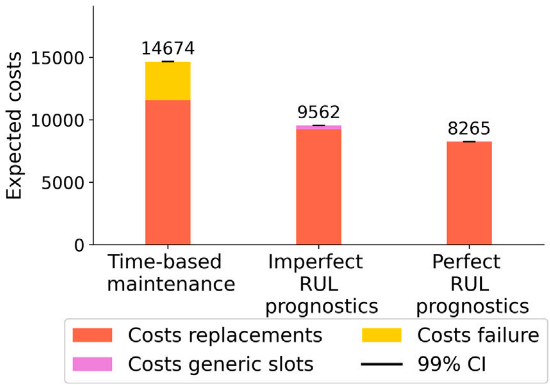


Fig. 17. The expected costs over a period of ten year, for (i) Time-based maintenance, (ii) maintenance with probabilistic, imperfect RUL prognostics and (iii) maintenance with perfect RUL prognostics. The 99% confidence interval of the mean (CI) is also given.

This difference is because the mean wasted life per replacement is highest (75.8) when using Time-based Maintenance, and lowest when considering perfect RUL prognostics (9.0, see Fig. 15(c)).

However, when using imperfect RUL prognostics, wasting the life of the engines prevents many engine failures: Only 26 engines fail in the 10,000 simulations of ten years. When using perfect RUL prognostics, the exact failure time is known, and no engine thus fails in any simulation. With Time-based maintenance, on average 62 engines fail per ten years, even though an engine is replaced 75.8 flight cycles before failure on average.

Last, we expect to use 30 generic slots when considering imperfect RUL prognostics, while we expect to use less than one generic slot with perfect RUL prognostics or Time-based Maintenance (see Fig. 15(d)). This is because imperfect RUL prognostics are updated over time. The predicted failure time of an engine may therefore suddenly decrease when new sensor measurements become available. In this case, it is sometimes more cost efficient to replace an engine in a generic slot than to risk a failure. In contrast, the predictions for perfect prognostics and Time-based Maintenance are constant.

Fig. 17 shows the expected maintenance cost per ten years for the three maintenance strategies. For all three strategies, most costs come from replacing the engines. When considering Time-based Maintenance, moreover, the expected costs of engine failures are 3082. In contrast, these costs are negligible when considering perfect or imperfect RUL prognostics. With imperfect RUL prognostics, however, the expected costs of using generic slots are 304, while these costs are negligible for the other two strategies. Overall, the costs decrease with 53% when considering imperfect RUL prognostics instead of Time-based Maintenance. Moreover, the maintenance costs further decrease by 14% when considering perfect RUL prognostics.

6. Conclusions

This paper proposes an end-to-end framework for predictive maintenance for complex components/ systems, from sensor measurements, to probabilistic RUL prognostics, to maintenance planning for single and multiple components/systems that integrate RUL estimates. Our proposed approach is illustrated for C-MAPSS turbofan engines. Probabilistic RUL prognostics (pdf of the RUL) are obtained using a Convolutional Neural Network with Monte Carlo dropout. The resulting probabilistic prognostics are shown to be reliable, with a high α-Coverage for all values of α.

Using these probabilistic RUL prognostics and renewal–reward processes, we determine an optimal moment to replace an engine. The optimal moment of engine replacements is shown to be close to the lower bound of the 99% confidence interval of the RUL prognostics. The results also show that more uncertain the RUL prognostics are, i.e., the wider the confidence intervals are, the earlier a component is preventively replaced to avoid a failure.

We further plan the replacements of multiple engines using an integer linear programming model that integrates RUL estimates. Here, the planning is further constrained by the availability of maintenance slots and the limited capability of a maintenance hangar to perform multiple replacements in the same day. In the long-run, we show that our approach leads to low expected number of engine failures. Compared with the ideal case when the true RUL is known in advance (ideal RUL prognostics), our approach leads to only 11% more engine replacements and a cost increase of only 14%. We also analyze the long-term performance of our approach vs a traditional, Time-based maintenance strategy. The results show that by using our probabilistic RUL prognostics, the number of replacements decreases with 32% per year. Moreover, the expected number of failure decreases from over 61 to 0.003 when considering a period of ten years.

As future work, we aim to further analyze the impact of cost choices on the maintenance planning results. We also aim to further improve our RUL prognostics by considering additional features such as attention mechanisms integrated into the neural network.

CRediT authorship contribution statement

Mihaela Mitici: Writing – review & editing, Writing – original draft, Supervision, Methodology, Formal analysis, Conceptualization. **Ingeborg de Pater:** Writing – original draft, Methodology, Formal analysis, Data curation. **Anne Barros:** Writing – review & editing, Visualization. **Zhiguo Zeng:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgment

The research contribution of Anne Barros and Zhiguo Zeng is partially supported by Chair of Risk and Resilience of Complex Systems (Chair EDF, SNCF and Orange), France.

References

[1] Badea VE, Zamfiroiu A, Boncea R. Big data in the aerospace industry. Inform Econ 2018;22(1):17–24.

[2] Hu Y, Miao X, Si Y, Pan E, Zio E. Prognostics and health management: A review from the perspectives of design, development and decision. Reliab Eng Syst Saf 2022;217:108063.

[3] Alaswad S, Xiang Y. A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliab Eng Syst Saf 2017;157:54–63.

[4] de Pater I, Reijns A, Mitici M. Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics. Reliab Eng Syst Saf 2022;221:108341.

[5] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. Reliab Eng Syst Saf 2022;108908.

[6] Keizer MCO, Flapper SDP, Teunter RH. Condition-based maintenance policies for systems with multiple dependent components: A review. European J Oper Res 2017;261(2):405–20.

[7] Ren L, Zhao L, Hong S, Zhao S, Wang H, Zhang L. Remaining useful life prediction for lithium-ion battery: A deep learning approach. Ieee Access 2018;6:50587–98.

[8] de Pater I, Mitici M. Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder. Eng Appl Artif Intell 2023;117:105582.

[9] Biggio L, Wieland A, Chao MA, Kastanis I, Fink O. Uncertainty-aware prognosis via deep Gaussian process. IEEE Access 2021;9:123517–27.

[10] de Pater I, Mitici M. Novel metrics to evaluate probabilistic remaining useful life prognostics with applications to turbofan engines. In: PHM society European conference, vol. 7. 2022, p. 96–109.

[11] Lee J, Mitici M. An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic Petri nets. Reliab Eng Syst Saf 2020;202:107052.

[12] Do P, Assaf R, Scarf P, Iung B. Modelling and application of condition-based maintenance for a two-component system with stochastic and economic dependencies. Reliab Eng Syst Saf 2019;182:86–97.

[13] Van Horenbeek A, Pintelon L. A dynamic predictive maintenance policy for complex multi-component systems. Reliab Eng Syst Saf 2013;120:39–50.

[14] Shi Y, Zhu W, Xiang Y, Feng Q. Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement. Reliab Eng Syst Saf 2020;202:107042.

[15] Zhang Z, Si X, Hu C, Lei Y. Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods. European J Oper Res 2018;271(3):775–96.

[16] Caballé NC, Castro IT, Pérez CJ, Lanza-Gutiérrez JM. A condition-based maintenance of a dependent degradation-threshold-shock model in a system with multiple degradation processes. Reliab Eng Syst Saf 2015;134:98–109.

[17] Schöbi R, Chatzi EN. Maintenance planning using continuous-state partially observable Markov decision processes and non-linear action models. Struct Infrastruct Eng 2016;12(8):977–94.

[18] Andriotis C, Papakonstantinou K. Managing engineering systems with large state and action spaces through deep reinforcement learning. Reliab Eng Syst Saf 2019;191:106483.

[19] Chen C, Zhu ZH, Shi J, Lu N, Jiang B. Dynamic predictive maintenance scheduling using deep learning ensemble for system health prognostics. IEEE Sens J 2021;21(23):26878–91.

[20] Nguyen KT, Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab Eng Syst Saf 2019;188:251–62.

[21] de Pater I, Mitici M. Predictive maintenance for multi-component systems of repairables with remaining-useful-life prognostics and a limited stock of spare components. Reliab Eng Syst Saf 2021;214:107761.

[22] Consilvio A, Di Febbraro A, Sacco N. A rolling-horizon approach for predictive maintenance planning to reduce the risk of rail service disruptions. IEEE Trans Reliab 2020;70(3):875–86.

[23] Saxena A, Goebel K. Turbofan engine degradation simulation data set. NASA Ames Prognostics data repository 2008;878–87.

[24] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.

[25] Babu GS, Zhao P, Li X-L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications. Springer; 2016, p. 214–28.

[26] Wang B, Lei Y, Li N, Yan T. Deep separable convolutional network for remaining useful life prediction of machinery. Mech Syst Signal Process 2019;134:106330.

[27] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. PMLR; 2016, p. 1050–9.

[28] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: A review for statisticians. J Amer Statist Assoc 2017;112(518):859–77.

[29] Hashemi M. Enlarging smaller images before inputting into convolutional neural network: Zero-padding vs. interpolation. J Big Data 2019;6(1):1–13.

[30] Li H, Zhao W, Zhang Y, Zio E. Remaining useful life prediction using multi-scale deep convolutional neural network. Appl Soft Comput 2020;89:106113.

[31] Xia J, Feng Y, Lu C, Fei C, Xue X. LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems. Eng Fail Anal 2021;125:105385.

[32] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[33] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834.

[34] Li T, Zhao Z, Sun C, Yan R, Chen X. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction. Reliab Eng Syst Saf 2021;215:107878.

[35] Peng C, Chen Y, Chen Q, Tang Z, Li L, Gui W. A remaining useful life prognosis of turbofan engine using temporal and spatial feature fusion. Sensors 2021;21(2):418.

[36] Al-Dulaimi A, Zabihi S, Asif A, Mohammadi A. A multimodal and hybrid deep neural network model for remaining useful life estimation. Comput Ind 2019;108:186–96.

[37] Al-Dulaimi A, Asif A, Mohammadi A. Multipath parallel hybrid deep neural networks framework for remaining useful life estimation. In: 2020 IEEE international conference on prognostics and health management. IEEE; 2020, p. 1–7.

[38] Baraldi P, Mangili F, Zio E. A prognostics approach to nuclear component degradation modeling based on Gaussian Process Regression. Prog Nucl Energy 2015;78:141–54.

[39] Tijms HC. Stochastic modelling and analysis: a computational approach. John Wiley & Sons, Inc.; 1986.