Taylor & Francis
Taylor & Francis Group

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Optimal predictive selective maintenance for fleets of mission-oriented systems

R. O'Neil, A. Khatab & C. Diallo

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2024, VOL. 62, NO. 24, 8790–8817
https://doi.org/10.1080/00207543.2024.2349257

Taylor & Francis
Taylor & Francis Group

Check for updates

Optimal predictive selective maintenance for fleets of mission-oriented systems

R. O'Neil[a], A. Khatab [a,b] and C. Diallo [a]

[a]Dalhousie University, Industrial Engineering, Halifax, Nova Scotia, Canada; [b]Laboratory of Industrial Engineering, Production and Maintenance (LGIPM), Lorraine University, Metz, France

ABSTRACT
In many settings, fleets of assets must perform series of missions with in-between finite breaks. For such fleets, a widely used maintenance strategy is the fleet selective maintenance (FSM). Under resource constraints, the FSM problem selects an optimal subset of feasible maintenance actions to be performed on a subset of components to minimise the maintenance cost while ensuring high system reliability during the upcoming mission. The majority of extant FSMP models are focussed on traditional physics-based reliability models. With recent advances in Machine Learning (ML) and Deep Learning (DL) algorithms, data-driven methods have shown accuracy in predicting remaining useful life (RUL). This paper proposes a predictive FSM strategy for fleets of complex and large multi-component systems. It relies on a concurrent ML/DL and optimisation approach where a clustering algorithm is used to determine the health states of components and a probabilistic RUL prognostics model is used for component reliability assessment. An optimisation model is developed to solve the predictive FSM problem to ensure high reliability of all systems within the fleet. An efficient two-phase solution approach is developed to solve this complex optimisation problem. Numerical experiments show the validity of the approach and highlight the improved maintenance plans achieved.

ARTICLE HISTORY
Received 18 May 2023
Accepted 22 April 2024

KEYWORDS
Predictive selective maintenance; fleet selective maintenance; mission-oriented systems; deep learning; reliability optimisation; artificial intelligence

Acronyms

C-MAPSS	Commercial modular aero-propulsion system simulation
DL	Deep Learning
FSM	Fleet selective maintenance
PdSM	Predictive selective maintenance
FSMP	Fleet selective maintenance problem
FSMO	Fleet selective maintenance optimisation
LSTM	Long short-term memory
Bi-LSTM	Bidirectional long short-term memory
ML	Machine Learning
RNN	Recurrent neural network
RUL	Remaining useful life
SM	Selective maintenance
FCM	Fuzzy c-means

1. Introduction

The functioning of contemporary society relies heavily on complex and interdependent networks for the production and distribution of goods, energy and services. These networks must be continuously monitored and proactively maintained to ensure minimal disruptions, exceptional dependability, and a high degree of preparedness to respond to unforeseen events such as pandemics, natural calamities, and weather disturbances. Many of these production and distribution assets can be classified as mission-oriented, meaning they are designed to operate in cycles, alternating between missions and maintenance breaks. Selective maintenance (SM) is a strategy that is applicable to these mission-oriented systems that aims to identify an optimal set of maintenance actions to be performed on the different system components. The ultimate goal of SM is usually to maximise the system's reliability for the upcoming mission or to minimise maintenance expenses (Diallo et al. 2018).

Industry 4.0 is the latest trend in automation that employs sensors, Cyber-Physical Systems (CPS), and the Industrial Internet of Things (IIoT) to connect the physical and virtual world. As a result of implementing these innovative technologies, predictive maintenance is increasingly being recognised as an appealing maintenance strategy in numerous industries. A key element of predictive maintenance is prognostics, which deals with fault prediction before it occurs (Jardine, Lin, and Banjevic 2006). There are two main prediction types in prognostics. The first is remaining useful life (RUL) prediction, and the second deals with predicting the probability

CONTACT C. Diallo claver.diallo@Dal.ca Dalhousie University, Industrial Engineering, 5269 Morris Street, PO Box 15000, Halifax, Nova Scotia, B3H 4R2, Canada

that a component operates without failure up to some future time. Recently, deep learning (DL) algorithms have made tremendous strides in performance and have improved the state of the art in prognostics and predictive maintenance (Namuduri et al. 2020). Effective predictive maintenance plans have the capability of reducing maintenance cost, extending equipment life, reducing machine downtime, and improving production. Prognostic methodologies encompass traditional machine learning (ML) techniques, including logistic regression, support vector machines, and ensemble methods, as well as deep learning (DL) techniques, such as multilayer perceptron models, convolutional neural networks, and recurrent neural networks. DL algorithms require tens to hundreds of thousands of training samples to achieve a desirable level of accuracy. IIoT sensors can produce massive amounts of data thus enabling DL algorithms to be applied. ML and DL methods can identify a relationship between sensor values and dependant variables such as time of failure, future sensor value, or the detection of anomalies.

Almost all selective maintenance problem (SMP) models proposed in the literature are based on statistical approaches for reliability assessment. Components are modelled using lifetime distributions and the hazard rate function is used to describe the ageing process. A limitation of using lifetime distributions is the assumption that the failure rate of a system depends only on time and a limited number of environmental factors/co-variates. However, many industrial systems degrade or deteriorate with both time, usage and multiple other factors. It is therefore more realistic to base components failure on their health condition (Khatab, Diallo, Aghezzaf et al. 2018). The application of new technologies such as wireless sensors and IIoT, and the recent advancements in Artificial Intelligence (AI), have led to data-driven methods based on DL algorithms to become very popular for RUL prediction and reliability assessment. Two major advantages of utilising these data-driven approaches are: (i) the current health condition of the component is considered and (ii) they do not rely on knowing the physical nature of the degradation mechanism model. These data-driven approaches utilise data collected from similar systems to train a model, from which the RUL or reliability can be predicted (An, Choi, and Kim 2018).

In this paper, a data-driven predictive selective maintenance strategy is proposed for fleets of mission-oriented systems, where a concurrent DL and optimisation approach is developed to identify optimal maintenance decisions. The strategy is referred to as predictive selective maintenance (PdSM) because a DL approach is utilised to predict a RUL distribution, an approach that is commonly used in predictive maintenance for machine prognostics. To the best of our knowledge, the first PdSM framework was introduced by Hesabi, Nourelfath, and Hajji (2022) where a probabilistic classifier is used for reliability prediction and a basic SM optimisation model is used to identify maintenance decisions. The authors consider a relatively small series-system and utilise a full enumeration approach to solve the optimisation problem. For large and complex systems comprised of many components and subsystems their proposed solution method would be unable to provide optimal decisions within a reasonable time frame. This highlights the need to develop a more intelligent and efficient solution approach. In the work of Hesabi, Nourelfath, and Hajji (2022), it is assumed that maintenance actions can restore a component to a previous state where the RUL is higher, and more expensive maintenance actions lead to a better RUL enhancement. In practice, the level of RUL improvement of a given maintenance action is an extremely difficult if not impossible quantity to determine.

The goal of this paper is to extend and improve upon the PdSM approach proposed by Hesabi, Nourelfath, and Hajji (2022). Rather than a basic series system, we consider fleets of systems each composed of multiple components connected according to complex reliability structures. The present work also considers the impact of component degradation level on the time required to perform maintenance. Such a consideration is indeed necessary to meet practical maintenance plans requirements. Unlike the work of Hesabi, Nourelfath, and Hajji (2022), the optimisation model developed here not only identifies the optimal set of components to be maintained, but also jointly provides optimal repairperson maintenance task assignment decisions. To overcome the limitations of the full enumeration solution method used in Hesabi, Nourelfath, and Hajji (2022), the present paper develops an efficient solution method based on a two-phase decomposition approach. Finally, rather than using a probabilistic classifier to predict component reliability, a DL algorithm combined with Monte Carlo dropout is used. The advantages of this approach will be highlighted and discussed later in the paper.

This paper is also a major extension of its preliminary version published in O'Neil, Diallo, and Khatab (2022). Compared to O'Neil, Diallo, and Khatab (2022), the problem is extended to a fleet of mission-oriented systems, a new prediction framework is proposed, new deep learning methods are applied and a new binary integer programming model is developed. The remainder of this paper is structured around five additional sections. Section 1 provides a review examining papers related to selective and predictive maintenance, with an emphasis on the utilisation of DL techniques for the purpose of prognostics. The system investigated is presented and fully described in Section 2 along with the notation and the main working assumptions used. Section 3 presents the concurrent DL and optimisation approach used to solve the fleet predictive selective maintenance strategy. The clustering algorithm used for health state identification and the probabilistic DL classifier used for reliability assessment are both discussed. The FSMO model and the solution method are also presented in Section 3. In Section 4, various numerical experiments are carried out to validate the ML and DL models and algorithms, and demonstrate the effectiveness and the accuracy of the proposed PdSM approach. This Section also includes a description of the dataset used to train and test our ML and DL algorithms, the data preprocessing steps, and the evaluation metrics used to evaluate the effectiveness of our models. Conclusions and future research extensions are presented in Section 5.

2. Literature review

This review section deals with three topics related to the objective of our proposed maintenance strategy: the SMP, DL methods for prognostics and state of the art (SOTA) data-driven maintenance strategies.

2.1. The selective maintenance problem

Many military and industrial systems are designed to complete a sequence of missions with finite breaks in between. Maintenance is necessary during these breaks to ensure these systems perform acceptably during missions. Maintenance actions range from minimal repair for failed components to perfect replacement. Resource constraints, such as time, cost, and availability of repairpersons, dictate that only a subset of system components can be selected for maintenance. The goal of the SMP is to determine the optimal subset of components and maintenance actions that will minimise maintenance cost subject to a reliability requirement being met for the upcoming mission(s), given these resource constraints.

The original selective maintenance model introduced in Rice, Cassady, and Nachlas (1998) dealt with an elementary series system comprised of components with exponentially distributed lifetimes. The only maintenance action considered was the replacement of failed components. An enumeration approach was devised to identify the optimal maintenance decisions that would maximise the system reliability for the upcoming mission. Many researchers have since expanded upon the original SMP to include complex system configurations (Cassady, Pohl, and Murdock 2001; Diallo et al. 2018), multistate systems (Liu and Huang 2010; Pandey, Zuo, and Moghaddass 2013), component dependence and failure propagation (Dao and Zuo 2017; Maaroufi, Chelbi, and Rezg 2013; Xu et al. 2016), fleet level SM (Khatab et al. 2022; Schneider and Cassady 2015), multiple repair channels (Diallo et al. 2019; Khatab, Diallo, Venkatadri et al. 2018), stochastic mission, break, and maintenance durations (Al-Jabouri et al. 2024; Khatab, Aghezzaf, Diallo et al. 2017; Khatab, Aghezzaf, Djelloul et al. 2017; Liu, Chen, and Jiang 2018), stochastic maintenance quality (Duan et al. 2018; Khatab and Aghezzaf 2016), and joint selective maintenance and orienteering (O'Neil, Khatab et al. 2023).

There have been significant improvements in the solution methods and approaches used to solve the SMP. Rajagopalan and Cassady (2006) proposed refinements to the total enumeration approach developed by Rice, Cassady, and Nachlas (1998). Pandey et al. (2013) utilise a differential evaluation approach as a solution method, multiple studies (Dao, Zuo, and Pandey 2014), Chaabane et al. (2020) use genetic algorithms as a solution technique, and O'Neil, Diallo et al. (2023) propose a hybrid column generation and genetic algorithm. Al-Jabouri, Saif, Diallo et al. (2023) propose a branch-and-price approach to solve large-scale instances of the SMP. In addition, several papers have used reinforcement learning to handle this complex optimisation problem (Achamrah and Attajer 2023; Liu, Chen, and Jiang 2020; Liu and Qian 2022). Most of the papers reviewed thus far focus on simple series-parallel system architectures, although more complicated structures, such as k-out-of-n:G, are common in various industrial applications. Diallo et al. (2018) is the first to examine the SMP in the context of serial k-out-of-n:G systems. The authors propose two new nonlinear SMP formulations. To solve the resulting SM optimisation problems, a two-phase approach using a binary integer programming (BIP) model is developed: the first phase generates all feasible maintenance patterns for each k-out-of-n:G subsystem, while the second phase solves a multidimensional multiple-choice knapsack problem to select the optimal mix of maintenance patterns. Recent literature reviews on the SMP can be found in Al-Jabouri et al. (2022) and Al-Jabouri, Saif, Khatab et al. (2023).

The majority of SMP models proposed in the literature use probability distributions to model component lifetimes, where the hazard rate function is used to describe the ageing process. A limitation of using lifetime distributions is the assumption that the failure rate of a system depends only on time, when in reality many systems degrade with both time and usage. Khatab, Diallo, Aghezzaf et al. (2018) introduced the first SM strategy that considers the actual health condition of components, and refer to the strategy as condition-based SM. The authors assume that each component degrades following a stochastic process and the actual health condition of a component is revealed only through periodic inspections performed during a mission. Hesabi, Nourelfath, and Hajji (2022) introduce a predictive SM framework, where system components are assumed to be continuously monitored by multiple sensors. A trained DL algorithm is used to predict the mission failure probability of each component. Meanwhile, only a simple series system is considered and a basic solution strategy is used to solve the optimisation problem. As was previously stated, for large and complex systems, the full enumeration approach proposed by Hesabi, Nourelfath, and Hajji (2022) would be unable to provide solutions within a reasonable time frame. Thus, there is a need to develop predictive selective maintenance approaches that are capable of efficiently solving systems of practical size.

2.2. Data-driven maintenance strategies

Prognostics is a central aspect of predictive maintenance and usually deals with the prediction of RUL. Different techniques have been proposed for RUL prediction such as physics based, data-driven, and hybrid methods. In recent years, however, there has been a surge in the use of data-driven methods based on DL techniques. These methods have shown exceptional performance in their ability to learn patterns and useful information from multi-variate sensor data and text data from maintenance logs to make accurate predictions (Fu et al. 2024; Usuga-Cadavid et al. 2022; Zhang, Liu, and Ye 2024). This review section will focus on data-driven prognostic models and the SOTA data-driven maintenance strategies.

Li, Ding, and Sun (2018) propose a new data-driven approach for prognostics that implements a deep convolutional neural network (DCNN) for RUL prediction of turbofan engines. The DCNN is compared to other popular approaches like the multi-layer perceptron (MLP), recurrent neural network (RNN), and the LSTM network. The proposed DCNN was shown to outperform the other algorithms in terms of both root-mean-square error (RMSE) and a popular scoring function employed by the International Conference on Prognostics and Health Management Data Challenge. Yang, Liu, and Zio (2019) propose a double-CNN architecture for the purpose of RUL prediction. The first CNN is used to identify the incipient fault point and the second is used for RUL prediction. Li et al. (2020) propose a multi-scale CNN (MS-CNN) and multiple recent studies implement temporal convolutions with attention mechanisms (Fu et al. 2024; Zhang, Liu, and Ye 2024).

The application of RNNs for RUL prediction has also been studied extensively in the literature. Zhang et al. (2018) utilise a LSTM neural network for RUL prediction of Lithium-Ion batteries. The experiment results showed that the LSTM generally predicts more accurate RULs than simple RNNs and support vector machines. There have also been studies that have used bi-directional LSTMs (Zhang et al. 2018) and convolutional bi-directional LSTMs (Zhao et al. 2017) for RUL prediction. Su, Li, and Wen (2020) propose an integrated variational autoencoder (VAE) and time-window-based sequence neural network (twSNN) approach for RUL prediction. The VAE is used to reduce data dimension and extract meaningful data features, the twSNN is then used to predict the RUL. Song et al. (2022) define the problem of RUL estimation as a bi-level prediction problem. The lower-level uses an LSTM to predict the time-series sensor data in the next several time-steps. The upper-level takes in the predicted future time-series data as input and uses an LSTM to estimate the RUL. The proposed method was shown to outperform state-of-the-art DL and ML algorithms. Ruan, Wu, and Yan (2021) compare CNNs, LSTMs, and a combined CNN-LSTM architecture for predicting RUL of turbofan engines. The results obtained show that the LSTM network with 2 LSTM layers and 2 dense layers performs the best in terms of RUL prediction. Other studies that propose hybrid approaches incorporating multiple DL algorithms can be found in Niu et al. (2019), Li, Li, and He (2019) and Ren et al. (2020).

The aforementioned studies are mainly focussed on RUL point prediction. However, the focus of the present work lies in the application of the prognostics results to guide and inform decisions related to SM planning. Dealing with the recent data-driven maintenance approaches, Nguyen and Medjaher (2019) use an LSTM classifier to predict the probability that the RUL of a system falls within certain time intervals. These reliability predictions are then used to optimise maintenance and spare parts ordering decisions. Zhou et al. (2023) propose a PdSM strategy for multi-state series-parallel systems using an adaptive C-Transformer for RUL prediction, and a CNN for reliability prediction. Dehghan Shoorkand, Nourelfath, and Hajji (2023) and Shoorkand, Nourelfath, and Hajji (2024) develop integrated predictive maintenance and production planning frameworks that leverage DL algorithms for component reliability prediction and mathematical programming.

All papers mentioned above rely on training a neural network for the task of predicting failure probability within a given time-window or mission length. The main drawback of these approaches is the need to train different models for varying window-sizes or mission lengths, resulting in a lack of flexibility in handling different operational scenarios without having to retrain a new model. In the literature, several works have been proposed for data-driven maintenance using RUL predictions instead of reliability predictions (Chen, Zhu et al. 2021; de Pater, Reijns, and Mitici 2022; Zhang and Zhang 2020). However, a shortcoming of these works resides in their inability to consider the uncertainty inherent to RUL predictions. Chen et al. (2022) propose an approach for RUL uncertainty quantification using a Local Uncertainty Estimation (LUE) model together with a Bi-LSTM. This approach allows to construct a RUL prediction interval from which a RUL probability distribution is derived and used to deal with maintenance optimisation. Zeng and Liang (2023) propose a predictive maintenance framework that implements probabilistic RUL prognostics into maintenance planning using a Gaussian distribution to model prediction uncertainties. Rather than training a model solely for RUL point prediction, the model proposed in Zeng and Liang (2023) is designed to predict the mean and standard deviation of a Gaussian distribution, thus providing a distribution of the RUL. An integer program is then proposed to optimise the maintenance planning problem. Recently, several other works (Lee and Mitici 2023; Mitici et al. 2023; Zhuang, Xu, and Wang 2023) have implemented Monte Carlo dropout to predict the RUL probability distribution. Monte Carlo is a computationally efficient method for estimating the uncertainty in neural network outputs (Abdar et al. 2021). Thus, a DL algorithm combined with Monte Carlo dropout will be used in the present paper to predict the RUL probability distribution.

2.3. Paper contributions

The main contributions of the paper are as follows:

- A novel PdSM approach is developed for a fleet of systems with multiple components connected via complex reliability architectures. The impact of component degradation on maintenance time is accounted for. The approach is designed to jointly provide optimal maintenance actions and repairperson task assignments decisions.
- Rather than using a probabilistic classifier for reliability prediction as in Hesabi, Nourelfath, and Hajji (2022), an advanced DL algorithm with Monte Carlo dropout is implemented. This reliability prediction approach holds several key advantages including more accurate predictions in terms of mission success probability and mission length variability. Furthermore, the ability to predict a RUL distribution allows for the development of a more realistic and practical data-driven selective maintenance model as it takes into account the impact of component failure in the optimisation process.
- In contrast to the full enumeration solution method used in Hesabi, Nourelfath, and Hajji (2022), a more efficient solution method is proposed on the basis of decomposition techniques.
- In contrast to most existing data-driven maintenance approaches, our approach leverages a predicted RUL distribution to compute both component reliability and expected downtime during the upcoming mission. The predicted reliability of components is used to accurately determine the overall system reliability. Indeed, calculating the expected component downtime allows to guide the maintenance selection process toward components that are likely to fail early in the mission, thus contributing to an improved fleet performance.
- This paper provides a comparative analysis of different methods used for reliability prediction. This comparison focuses specifically on different recent DL techniques as well as traditional statistical approaches. The advantage of using DL models to make predictions based on the current condition is clearly demonstrated. The results also highlight the advantages and disadvantages of using predicted RUL distributions opposed to point predictions or classification models for reliability assessment.
- Finally, the proposed predictive models are general and can indeed be applied to any mission-oriented systems such as production lines in manufacturing, turbines in hydroelectricity generation, and turbofan engines in aerospace to name just a few.

3. System description

The following notation system is used.

I	Number of systems within the fleet
i	Index of systems, $i = 1, \ldots, I$
N_i	Number of subsystems within system i
j	Index of subsystems in system $i, j = 1, \ldots, N_i$
M_{ij}	Number of components in subsystem j of system i
k	Index components in subsystem j of system i, $k = 1, \ldots, M_{ij}$
E_{ijk}	The k^{th} component of subsystem j of system i
Π_{ijk}	The penalty cost per unit time for component E_{ijk}
L_{ijk}	The highest maintenance level available for component E_{ijk}
l	Index of maintenance levels for E_{ijk}, $l = 1, \ldots, L_{ijk}$
Q	Number of repairpersons

r	Index of repairpersons, $r = 1, \ldots, Q$
t^p_{ijkl}	Time to perform preventive maintenance (PM) of level l on component E_{ijk}
t^c_{ijkl}	Time to perform corrective maintenance (CM) of level l on component E_{ijk}
c^p_{ijkl}	Cost of PM of level l on component E_{ijk}
c^c_{ijkl}	Cost of CM of level l on component E_{ijk}
c^f_r	Fixed cost of hiring or using repairperson r
c^v_r	Variable cost rate of using repairperson r
D	Maintenance break duration
U	Mission duration
$\mathcal{R}_{ijk}$	Probability/reliability of component E_{ijk} to operate the next mission
$\mathcal{R}_{ij}$	Reliability of subsystem j of system i during the next mission
$\mathcal{R}_i$	Reliability of system i during the next mission

3.1. Main working assumptions

(1) The system and components degrade with both age and usage (Khatab, Diallo, Aghezzaf et al. 2018).
(2) During the maintenance break the systems are assumed to be switched off and therefore not experiencing any degradation. This is reasonable and is commonly used (Diallo et al. 2018).
(3) Maintenance activities are allowed only during the break duration. This is also reasonable and commonly assumed (Khatab, Diallo, Aghezzaf et al. 2018; Khatab et al. 2022)
(4) After a maintenance action is performed, the component is brought back to an as good as new state.
(5) The time and cost of a maintenance action are tightly dependant on the degradation level of the component to be maintained. A maintenance action carried out on a component with higher degradation level will induce a higher cost and require longer time.
(6) Individual components contribute to the overall performance of the system. Accordingly, the failure of a component is penalised at a cost proportional to its duration as such failures lead to a decrease in system productivity or efficiency. This is a reasonable assumption that aligns with existing literature (Khatab, Diallo, Aghezzaf et al. 2018).

3.2. Fleet description and reliability computation

A fleet of I multi-component systems is investigated. Without loss of generality, components and systems are binary. Each system i $(i = 1, \ldots, I)$ is comprised of N_i subsystems in series, and each subsystem is composed of M_{ij} statistically independent components E_{ijk} $(j = 1 \cdots N_i; k = 1 \cdots M_{ij})$. We consider K_{ij}-out-of-M_{ij}:G subsystems, which implies that for the subsystem to operate correctly, at least K_{ij} out of the M_{ij} constituent components must be in working order. Finally, we assume components degrade/deteriorate with both operational time and usage.

The fleet of systems being examined is required to operate in sequences, alternating between active missions and maintenance breaks. It is assumed that the fleet has recently accomplished a mission and is entering the first maintenance break. During the break, all systems of the fleet are switched off for a duration D during which maintenance actions can be performed on their components. There is a time and cost associated with performing maintenance actions. To monitor the degradation of components, a common set of S sensors are available. New sensor measurements are recorded after the completion of each operating cycle. An operating cycle may refer to a defined period of time or a set of operations that must be executed. For example, the lifetime of a jet engine can be measured in flight cycles, where a flight cycle refers to all operational activities from take-off to landing. The overall degradation measurements corresponding to component E_{ijk} are represented as a matrix $\mathbf{G}_{ijk}$:

$$\mathbf{G}_{ijk} = \begin{bmatrix} \boldsymbol{g}^1 \\ \boldsymbol{g}^2 \\ \vdots \\ \boldsymbol{g}^{T_{ijk}} \end{bmatrix} = \begin{bmatrix} g_1^1 & g_2^1 & \cdots & g_S^1 \\ g_1^2 & g_2^2 & \cdots & g_S^2 \\ \vdots & \vdots & & \vdots \\ g_1^{T_{ijk}} & g_2^{T_{ijk}} & \cdots & g_S^{T_{ijk}} \end{bmatrix}.$$

where $\boldsymbol{g}^t$ is a vector of all sensor measurements during cycle t $(t = 1, \ldots, T_{ijk})$, and T_{ijk} refers to the number of cycles that component E_{ijk} has completed.

When the fleet starts the maintenance break, the sensor data and the number of cycles completed by components are available. The state of each component S_{ijk} is known and defined as follows:

$$S_{ijk} = \begin{cases} 1 & \text{if component } E_{ijk} \text{ is functioning at the start of the break,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

By first training a DL algorithm on the historical sensor data, the RUL of each component E_{ijk} can be evaluated and compared with the next mission duration U. Accordingly, the probability $\mathcal{R}_{ijk}$ of component E_{ijk} to successfully achieve the next mission can be determined; this probability is the reliability of component E_{ijk} for the subsequent mission. Similarly, the reliability $\mathcal{R}_i$ of system i is also defined as the probability that it can successfully complete the upcoming mission. The reliability $\mathcal{R}_i$ is function of both the system structure and the corresponding component reliability functions $\mathcal{R}_{ijk}$.

Given that systems are comprised of N_i k-out-of-n:G subsystems in series, the reliability $\mathcal{R}_i$ is then computed as:

$$\mathcal{R}_i = \prod_{j=1}^{N_i} \left[\sum_{e_{k_{ij}}=1}^{M_{ij}} \sum_{e_{k_{ij}-1}=1}^{e_{k_{ij}}-1} \cdots \sum_{e_1=1}^{e_2-1} \left(\prod_{v \in \{e_1,\ldots,e_{k_{ij}}\}} \mathcal{R}_{ijv} \right) \times \left(\prod_{\substack{u=1 \\ u \notin \{e_1,\ldots,e_{k_{ij}}\}}}^{e_{k_{ij}}} (1 - \mathcal{R}_{iju}) \right) \right]. \quad (2)$$

The exact formulation of the k-out-of-n:G subsystems was first presented by Arulmozhi (2002), efficient algorithms like the one proposed by Kuo and Zuo (2003) exist to solve the complex equation.

In this paper, we consider that a component subjected to a maintenance action becomes as good as new. There are q repair crews to whom maintenance actions are assigned with fixed cost c_r^f and variable cost rate c_r^v ($r = 1, \ldots, Q$). A maintenance action carried out on component E_{ijk} requires t_{ijk} units of time. In the proposed predictive selective maintenance (PdSM) approach, maintenance time is considered as dependent on the component degradation level. This is a reasonable assumption that aligns with previous work (Khatab, Diallo, Aghezzaf et al. 2018; Tai and Chan 2010). Components that are in more degraded states are likely to require more thorough inspections and repairs/replacements. The effective maintenance time estimation will be further discussed in the subsequent section.

4. Predictive selective maintenance optimisation approach

In this section, the proposed predictive selective maintenance approach is developed and fully discussed. The general principle of this approach is summarised in Figure 1 and mainly relies on three phases: offline, online, and maintenance decision making phases. During the offline phase, a fuzzy c-means (FCM) algorithm and bidirectional long short-term memory network (Bi-LSTM) are trained on historical components run-to-failure data. Before the training procedure for the unsupervised and supervised algorithms are initiated, data preprocessing and denoising steps are conducted.

After the completion of the preprocessing steps, the unsupervised clustering algorithm can be trained on the processed data. The FCM clustering algorithm is used to partition the historical run-to-failure data into discrete health states. The output of the clustering algorithm are

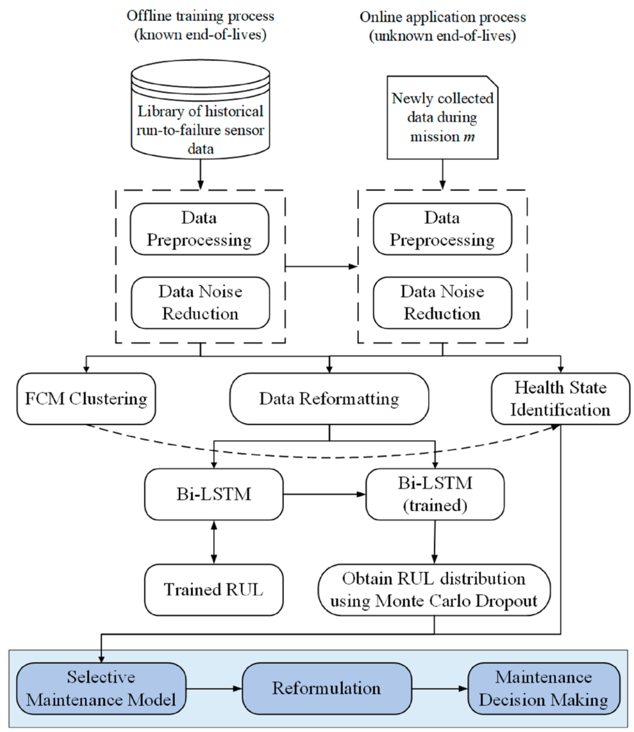


Figure 1. Flowchart of the proposed predictive selective maintenance framework for fleets of mission-oriented systems.

cluster centroids which can be used to identify the health state of components that are currently in operation. This is a crucial step in the proposed predictive selective maintenance approach as maintenance time depend on component's degradation level.

Before initiating the training process for the DL model, the historical components run-to-failure dataset is reformatted into 3-dimensional tensors of the form (N_s, N_{ts}, N_f), where N_s, N_{ts} and N_f refer to the number of samples, the time sequence dimension, and the number of selected features, respectively. Unlike probabilistic classifier-based approaches (Hesabi, Nourelfath, and Hajji 2022), our approach adopts a DL model with Monte Carlo dropout to predict a RUL distribution. Reliability computation involves conducting $\mathcal{M}$ forward passes of the trained network and generating $\mathcal{M}$ unique RUL predictions. The probability that a component will successfully complete the upcoming mission of length U can then be determined by assessing the proportion of these predictions that are greater than U.

The second phase of the proposed approach (*i.e.* the online phase) deals with the application of the trained DL model and clustering algorithm for online prognostics and health state identification. At the start of the maintenance break, the collected real-time monitoring data for each component are used as input to the trained DL model to predict a RUL distribution. The predicted distribution is used to compute the reliability and the expected down time of each component for the next mission. The centroids identified through training the FCM clustering algorithm serve to identify the current health state of each system component. It should be noted that the second phase does not repeat the time-consuming training process which reduces the time of failure prediction and health state identification.

The final phase of the proposed PdSM approach utilises the predicted failure probability and expected downtime along with health state for each component as input data for maintenance decision making. Reliability, expected downtime and degradation levels of components are used as input to a FSMO model to select the components that should be maintained in a fleet of heterogeneous systems with the objective of minimising the total maintenance cost while ensuring that each system in the fleet operates at a required minimum reliability level. The following subsections provide a detailed description of the three phases of the proposed PdSM approach.

4.1. FCM clustering for health state identification

The FCM clustering algorithm is implemented in this work to determine the hidden structure within the historical component failure data representing states of degradation, ranging from normal to near failure. FCM is trained in an unsupervised way and aims to find similarities within the data. In the literature, FCM has been successfully implemented for the purpose of health state division (Chen, Lu et al. 2021; Chen, Zhu et al. 2021; Javed, Gouriveau, and Zerhouni 2015; Wang et al. 2019). The main advantage of FCM algorithm over hard clustering algorithms is its ability to allow gradual memberships of data points to clusters measured as degrees in the interval $[0, 1]$. This gives the flexibility to express the data points as belonging to more than one cluster (Bora, Gupta, and Kumar 2014). After training the FCM algorithm on historical component run-to-failure data, the identified cluster centroids can then be used to estimate components health state membership.

The objective of the following FCM algorithm is to find the optimal cluster centres and components membership degrees such that the following constrained optimisation problem is minimised:

$$\min \quad \mathcal{J} = \sum_{c=1}^{\mathcal{C}} \sum_{i=1}^{\mathcal{M}} \sum_{j=1}^{T_i} u_{cij}^m \cdot d^2\left(\boldsymbol{g}_i^j, w_c\right) \tag{3}$$

$$\text{s.t.} \quad \sum_{c=1}^{\mathcal{C}} u_{cij} = 1, \quad \forall i, j \tag{4}$$

$$u_{cij} \in [0, 1], \quad \forall c, i, j \tag{5}$$

where m is the degree of fuzziness, u_{cij} is the membership of sample $\boldsymbol{g}_i^j$ to cluster w_c, $\mathcal{M}$ is the number of training components, T_i is the time to failure (in cycles) of training component i, $\mathcal{C}$ is the number of clusters, and $d^2(\boldsymbol{g}_i^j, w_c)$ is the euclidean distance between sample $\boldsymbol{g}_i^j$ and cluster w_c. Elements of vector $\boldsymbol{g}_i^j$ refer to all sensor measurements for training component i during cycle j.

The above optimisation problem is solved by converting it into an unconstrained problem using Lagrange multipliers. An iterative approach is then used where firstly, the cluster centres are fixed to find the membership degrees; secondly, the membership degrees are fixed to find the cluster centres (Benaichouche, Oulhadj, and Siarry 2013). The two steps are repeated until a convergence criteria is met.

Once the optimal cluster centroids have been found through training, the membership degree of operating component E_{ijk} to each cluster (or health state) is computed using Equation (6).

$$u_{c,E_{ijk}} = \frac{1}{\sum_{v=1}^{\mathcal{C}} \left(\frac{d^2\left(\boldsymbol{g}^{T_{ijk}}, w_c\right)}{d^2\left(\boldsymbol{g}^{T_{ijk}}, w_v\right)} \right)^{\frac{2}{m-1}}}, \quad \forall c \in \{1 \cdots \mathcal{C}\}. \tag{6}$$

The expected times to perform a PM (t_{ijkl}^p) and a CM (t_{ijkl}^c) action on component E_{ijk} are then computed as:

$$t_{ijkl}^p = \sum_{c=1}^{\mathcal{C}} u_{c,E_{ijk}} \cdot t_{cl}^p \tag{7}$$

and

$$t_{ijkl}^c = \sum_{c=1}^{\mathcal{C}} u_{c,E_{ijk}} \cdot t_{cl}^c \tag{8}$$

where t_{cl}^p and t_{cl}^c are respectively the expected times to perform a PM and a CM of level l on components whose health states belong to cluster w_c.

4.2. Probabilistic DL approach for RUL and reliability prediction

The LSTM network is a type of recurrent neural network (RNN) that is designed to overcome the vanishing and exploding gradient problem frequently encountered when training standard RNNs. LSTMs are particularly well-suited for processing sequential or time-series data, where the network needs to retain information about past events to make accurate predictions. At the core of an LSTM network are 'memory cells', which are connected via 'gates' that allow the cells to selectively add or remove information. In an LSTM, there are three types of gates:

- The forget gate, comprised of the first sigmoid layer in each unit, determines what information from the previous cell state should be discarded. It does this by taking as input the previous hidden state h_{t-1} and the current input g_t, passing them through a sigmoid function, and generating a vector f_t of values between 0 and 1. The vector f_t is then element-wise multiplied by the previous cell state to "forget–the irrelevant information. The vector f_t is given by:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, g_t] + b_f\right). \tag{9}$$

- The input gate, formed by the subsequent sigmoid and *tanh* layers, determine the new information that will be added to the cell state. The sigmoid layer produces a vector i_t of values between 0 and 1 to determine which values in the cell to update. The *tanh* layer produces a vector s_t of new candidate values. These two vectors can be mathematically expressed as:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, g_t] + b_i\right) \tag{10}$$
$$s_t = tanh\left(W_s \cdot [h_{t-1}, g_t] + b_s\right). \tag{11}$$

- Finally, the output gate determines what information should be provided as input to the next hidden state. It takes as input the current hidden state and the current input, passes them through a sigmoid function to generate a vector o_t of values between 0 and 1. The vector o_t is multiplied by the hyperbolic tangent of the current cell state to produce the new hidden state h_t. The vector o_t and h_t are calculated as follows:

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, g_t] + b_o\right). \tag{12}$$
$$h_t = o_t \cdot tanh(c_t). \tag{13}$$

The Bi-LSTM network, which comprises two LSTM cells, is an extended version of the standard LSTM. It has demonstrated exceptional performance in predicting remaining useful life (Chen et al. 2022; Zhang et al. 2018; Zhao et al. 2017), as well as other tasks involving sequential and time series data, such as natural language processing. The main advantage of using Bi-LSTMs instead of standard RNNs is that they can learn the temporal relationships in both the forward and backward directions simultaneously. The hidden state and output result for the Bi-LSTM at time t are computed according to the following equations:

$$\overrightarrow{h}_t = \mathrm{LSTM}\left(g_t, \overrightarrow{h}_{t-1}\right) \tag{14}$$
$$\overleftarrow{h}_t = \mathrm{LSTM}\left(g_t, \overleftarrow{h}_{t+1}\right) \tag{15}$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y. \tag{16}$$

The Bi-LSTM is followed by multiple fully connected dense layers, where the last one is comprised of a single neuron used to output the predicted RUL. Similar to the LSTM, the Bi-LSTM network is trained using the back-propagation through time (BPTT) algorithm. The loss function to minimise is the mean squared error with L2 regularisation:

$$\mathcal{L} = \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(y_n - \widehat{y}_n\right)^2 + \mu \cdot ||\mathbf{W}||_2^2. \tag{17}$$

where $\mathcal{N}$ is the number of samples, y_n and $\widehat{y}_n$ are the true and predicted targets, $\mathbf{W}$ are the weights of the neural network and μ is a hyperparameter that controls the strength of the regularisation.

The above model is designed for RUL point prediction, however, to account for uncertainty in the prediction, Monte Carlo dropout is implemented. Typically, dropout is a regularisation technique that is applied only during the training phase of a neural network. Monte Carlo dropout is an approach that extends dropout to the inference phase (Gal and Ghahramani 2016). During inference, rather than using the trained model to make a single prediction, Monte Carlo dropout involves performing $\mathcal{M}$ forward passes with dropout applied each time. This produces $\mathcal{M}$ different RUL predictions. Let $(\widehat{\mathrm{RUL}}_{ijk}^1, \ldots, \widehat{\mathrm{RUL}}_{ijk}^{\mathcal{M}})$ represent the $\mathcal{M}$ RUL predictions obtained for component E_{ijk} and let $\widehat{F}_{\mathrm{RUL}_{ijk}}(x)$ represent the empirical distribution of the RUL of E_{ijk}. Accordingly, the reliability $\mathcal{R}_{ijk}(U)$ and the expected downtime Δ_{ijk} of E_{ijk} for a mission of length U can then be computed as follows:

$$\mathcal{R}_{ijk}(U) = \frac{1}{\mathcal{M}}\sum_{m=1}^{\mathcal{M}} \mathbb{1}_{\left(\widehat{\mathrm{RUL}}_{ijk}^m > U\right)}, \tag{18}$$

$$\begin{aligned}\Delta_{ijk} &= \int_0^U (U - x)\,\mathrm{d}\widehat{F}_{\mathrm{RUL}_{ijk}}(x) \\ &= U\int_0^U \mathrm{d}\widehat{F}_{\mathrm{RUL}_{ijk}}(x) - \int_0^U x\,\mathrm{d}\widehat{F}_{\mathrm{RUL}_{ijk}}(x) \\ &= U\frac{\sum_{m=1}^{\mathcal{M}} \mathbb{1}_{\left(\widehat{\mathrm{RUL}}_{ijk}^m \le U\right)}}{\mathcal{M}} \\ &\quad - \frac{\sum_{m=1}^{\mathcal{M}} \widehat{\mathrm{RUL}}_{ijk}^m \cdot \mathbb{1}_{\left(\widehat{\mathrm{RUL}}_{ijk}^m \le U\right)}}{\mathcal{M}},\end{aligned} \tag{19}$$

where $\mathbb{1}_{(A)}$ is the indicator function taking a value of 1 if event A occurs and 0 otherwise.

4.3. Fleet predictive SM optimisation model and solution method

Due to resource limitations such as time and budget, it is not always possible to perform all desirable maintenance actions during the break. Thus, leading to the FSMO problem which corresponds to the third phase of the proposed predictive selective maintenance approach (see Figure 1). The FSMO model uses predicted health state, reliability and expected downtime of components provided by the Bi-LSTM and FCM algorithms respectively to jointly select the maintenance actions to be performed on selected components, and the assignment of repair crews to selected maintenance actions. The objective of the optimisation model is to minimise the grand total maintenance cost while ensuring the fleet can operate at a target reliability for the upcoming mission. Before defining the optimisation model, the following decision variables and notation are introduced:

$$x_{ijkrl} = \begin{cases} 1 & \text{if repairperson } r \text{ performs maintenance level } l \text{ on } E_{ijk}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

$$z_r = \begin{cases} 1 & \text{if repaircrew } r \text{ is hired,} \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

Two levels of maintenance for each component are considered in this work (i.e. $L_{ijk} = 2$). The $l = 1$ case refers to 'Do-nothing' and the $l = 2$ case refers to a maintenance action that will restore a component to the 'as-good-as-new' state. In addition to the above decision variables, we denote by $R_{ijkl}(U)$ the reliability of component E_{ijk} if it undergoes maintenance level l. Accordingly, $R_{ijk1}(U)$ is given by Equation (22), while $R_{ijk2}(U)$ is the reliability of a brand new component.

$$R_{ijk1}(U) = \begin{cases} \mathcal{R}_{ijk}(U) & \text{if } S_{ijk} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

where $\mathcal{R}_{ijk}(U)$ is given by Equation (18).

It is reasonable to assume that components within a system contribute to the overall system performance, as highlighted by Khatab, Diallo, Aghezzaf et al. (2018). Thus, a penalty cost due to components' downtime is included in the objective function. This cost is, without loss of generality, assumed to be proportional to the failure duration. Let Δ_{ijkl} represents the expected downtime of component E_{ijk} when maintenance level l is selected. Δ_{ijk1} is given by Equation (23) and Δ_{ijk2} is the expected downtime of a brand new component.

$$\Delta_{ijk1} = \begin{cases} \Delta_{ijk} & \text{if } S_{ijk} = 1, \\ U & \text{otherwise.} \end{cases} \tag{23}$$

where Δ_{ijk} is given by Equation (19).

The FSMO model can be formulated as the following mixed-integer non-linear optimisation problem:

$$\begin{aligned} \min\ \mathcal{Z} = & \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}}\sum_{r=1}^{Q}\sum_{l=1}^{L_{ijk}} \Pi_{ijk} \cdot \Delta_{ijkl} \cdot x_{ijkrl} \\ & + \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}}\sum_{r=1}^{Q}\sum_{l=1}^{L_{ijk}} \left(c^p_{ijkl} + c^v_r \cdot t^p_{ijkl}\right) \\ & \cdot S_{ijk} \cdot x_{ijkrl} \\ & + \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}}\sum_{r=1}^{Q}\sum_{l=1}^{L_{ijk}} \left(c^c_{ijkl} + c^v_r \cdot t^c_{ijkl}\right) \\ & \cdot \left(1 - S_{ijk}\right) \cdot x_{ijkrl} + \sum_{r=1}^{Q} c^f_r \cdot z_r \end{aligned} \tag{24}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}}\sum_{l=1}^{L_{ijk}} \left(t^p_{ijkl} \cdot S_{ijk} + t^c_{ijkl} \cdot \left(1 - S_{ijk}\right)\right) \\ & \cdot x_{ijkrl} \leq D \cdot z_r, \quad \forall r \end{aligned} \tag{25}$$

$$\begin{aligned} \mathcal{R}_i(U) = & \prod_{j=1}^{N_i} \left(\sum_{e_{k_{ij}}=1}^{M_{ij}} \sum_{e_{k_{ij}-1}=1}^{e_{k_{ij}}-1} \cdots \sum_{e_1=1}^{e_2-1} \right. \\ & \times \left(\prod_{v \in \{e_1,\dots,e_{k_{ij}}\}} \sum_{r=1}^{Q}\sum_{l=1}^{L_{ijv}} R_{ijvl}(U) \cdot x_{ijvrl} \right) \Bigg) \cdot \\ & \times \left(\prod_{\substack{u=1 \\ u \notin \{e_1,\dots,e_{k_{ij}}\}}}^{e_{k_{ij}}} (1 - \sum_{r=1}^{Q}\sum_{l=1}^{L_{iju}} R_{ijul}(U) \cdot x_{ijurl}) \right), \\ & \forall i \end{aligned} \tag{26}$$

$$\mathcal{R}_i(U) \geq \mathcal{R}_0, \quad \forall i \tag{27}$$

$$\sum_{r=1}^{Q}\sum_{l=1}^{L_{ijk}} x_{ijkrl} = 1, \quad \forall i, j, k \tag{28}$$

$$x_{ijkrl} \in \{0, 1\}, \quad \forall i, j, k, r, l \tag{29}$$

$$z_r \in \{0, 1\}, \quad \forall k \tag{30}$$

In the above optimisation model, the objective function seeks to minimise the expected predicted grand total cost. The first term represents the expected penalty cost of component failure. The second and third terms represent the cost of performing PM and CM maintenance respectively. These costs include the variable repair crew cost and the fixed maintenance cost. The fourth term represents the fixed cost for hiring repairpersons. Constraints (25) ensure that the time to perform maintenance for each repairperson does not exceed the break duration. Constraints (26) and (27) ensure that the target reliability level is met for each system. Constraints (28) ensure that each component is replaced only once if it is selected to undergo maintenance. Finally Constraints (29) and (30) are binary variable restrictions.

The optimisation model presented above is complex and requires the development of an efficient algorithm to provide optimal solutions. To solve the complex problem, an adaptation of the two-phase approach initiated by Diallo et al. (2018) and used in Khatab et al. (2022) is developed. The first phase of the approach consists in generating all feasible maintenance patterns for each subsystem, and the second phase consists in solving a Mixed Integer Linear Program (MILP). A maintenance pattern $\boldsymbol{p}$ for subsystem j in system i is defined as a vector of length M_{ij} where the k^{th} element p_k^h takes a value of 1 if component k is selected for maintenance and 0 otherwise. For a small two-component parallel subsystem, the four possible maintenance patterns would be: $\boldsymbol{p}^1 = [0,0]^T, \boldsymbol{p}^2 = [1,0]^T, \boldsymbol{p}^3 = [0,1]^T, \boldsymbol{p}^4 = [1,1]^T$ where $[.]^T$ is the transpose operator. The first pattern implies no maintenance is selected, pattern two implies only the first component is maintained, and so on. For each pattern generated, the vector $\boldsymbol{t}_{ij}^h$ of length M_{ij} containing the time t_{ijk}^h to maintain each component E_{ijk} $(k = 1, \dots, M_{ij})$ is stored, as well as the subsystem reliability $\mathcal{R}_{ij}^h(U)$ achieved and the total cost $\mathcal{C}_{ij}^h$ excluding repair crew variable and fixed cost. The output of the first phase is a dataset containing the following parameters: i, j, $\boldsymbol{p}^h$, $\boldsymbol{t}_{ij}^h$, $\mathcal{R}_{ij}^h(U)$, $\mathcal{C}_{ij}^h$, $\mathcal{H}_{ij}$ for each pattern h $(h = 1 \cdots \mathcal{H}_{ij})$ of each subsystem j in each system i. Algorithm (1) outlines how all feasible patterns are generated.

The aim of the second phase of the approach is to select a maintenance pattern for each subsystem and assign a repairperson to each maintenance task such that the total cost is minimised. The following new decision variables λ_{ij}^h, w_{ijkr}, and τ_{ijkr} are introduced.

$$\lambda_{ij}^h = \begin{cases} 1 & \text{if pattern } h \text{ is selected for subsystem } j \text{ in system } i, \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

$$w_{ijkr} = \begin{cases} 1 & \text{if repairperson } r \text{ performs maintenance on } E_{ijk}, \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

τ_{ijkr} is the total time spent performing maintenance on component E_{ijk} by repairperson r. With the above notation, the MILP formulation of the FSMO problem is as follows:

$$\min\ \mathcal{Z} = \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{h=1}^{\mathcal{H}_{ij}} \mathcal{C}_{ij}^h \cdot \lambda_{ij}^h + \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}}\sum_{r=1}^{Q} c_r^v \cdot \tau_{ijkr} + \sum_{r=1}^{Q} c_r^f \cdot z_r \tag{33}$$

$$\text{s.t.} \quad \sum_{h=1}^{\mathcal{H}_{ij}} \lambda_{ij}^h = 1, \quad \forall i, j \tag{34}$$

$$\sum_{j=1}^{N_i}\sum_{h=1}^{\mathcal{H}_{ij}} \ln\left(\mathcal{R}_{ij}^h(U)\right) \cdot \lambda_{ij}^h \geq \ln(\mathcal{R}_0), \quad \forall i \tag{35}$$

$$\sum_{r=1}^{Q} w_{ijkr} = \sum_{h=1}^{\mathcal{H}_{ij}} \lambda_{ij}^h \cdot p_k^h, \quad \forall i, j, k \tag{36}$$

$$\tau_{ijkr} \leq \sum_{h=1}^{\mathcal{H}_{ij}} \lambda_{ij}^h \cdot t_{ijk}^h, \quad \forall i, j, k, r \tag{37}$$

$$\tau_{ijkr} \geq \sum_{h=1}^{\mathcal{H}_{ij}} \lambda_{ij}^h \cdot t_{ijk}^h - \left(1 - w_{ijkr}\right) \cdot D, \quad \forall i, j, k, r \tag{38}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{k=1}^{M_{ij}} \tau_{ijkr} \leq D \cdot z_r, \quad \forall r \tag{39}$$

$$\tau_{ijkr} \leq D \cdot w_{ijkr}, \quad \forall i, j, k, r \tag{40}$$

$$\lambda_{ij}^h \in \{0, 1\}, \quad \forall i, j, h \tag{41}$$

$$w_{ijkr} \in \{0, 1\}, \quad \forall i, j, k, r \tag{42}$$

$$z_r \in \{0, 1\}, \quad \forall r \tag{43}$$

$$\tau_{ijkr} \geq 0, \quad \forall i, j, k, r \tag{44}$$

In the above optimisation model, the objective function (33) seeks to minimise the grand total cost. Constraints (34) ensure that exactly one maintenance pattern is selected for each subsystem. Constraint (35) warrants that the target reliability level is met for each system. Constraints (36) ensure that a repairperson is assigned to each maintenance task in the selected pattern. Constraints (37) and (38) are used to track the maintenance time for each repairperson. Constraints (39) ensure that repairperson is hired (fixed cost paid) before they can perform any maintenance action while constraints (40)

Algorithm 1 Compute $\mathcal{R}_{ij}^{h}(U)$, $\mathcal{C}_{ij}^{h}$, $\boldsymbol{t}_{ij}^{h}$, p^h for all valid patterns for subsystem j in system i

Input data: I, N_i, M_{ij}, t_{ijkl}, R_{ijk1}, $\mathcal{R}_0$, D, L_{ijk}
Initialize: $i = 1$
while $i \leq I$ **do**
  Initialize: $j = 1$
  while $j \leq N_i$ **do**
    – Generate all valid maintenance patterns for subsystem j in system i and store them in the set H_{ij}.
    – Find the cardinality $\mathcal{H}_{ij}$ of the set H_{ij}: $\mathcal{H}_{ij} = |H_{ij}|$.
    Initialize: $h = 1$
    while $h \leq \mathcal{H}_{ij}$ **do**
      – Compute $\mathcal{C}_{ij}^{h}$ the total cost of subsystem j excluding repair crew fixed and variable cost.
      – Compute $\mathcal{R}_{ij}^{h}(U)$ the reliability of subsystem j using the algorithm proposed by Kuo and Zuo (2003).
      if $\mathcal{R}_{ij}^{h}(U) \geq \mathcal{R}_0$ **then**
        – Store i, j, h, $\mathcal{R}_{ij}^{h}(U)$, $\mathcal{C}_{ij}^{h}$, $\boldsymbol{t}_{ij}^{h}$, $\boldsymbol{p}^{h}$.
      else
        – Remove current pattern h from the list $\mathcal{H}_{ij}$. (all patterns above h get shifted down by one position after the removal of h).
        – Update $h = h - 1$ (to account for the removed pattern).
        – Update $\mathcal{H}_{ij} = |H_{ij}|$.
      end if
      $h = h + 1$.
    end while
    $j = j + 1$.
  end while
  $i = i + 1$.
end while

guarantee that the time to perform maintenance for each repairperson does not exceed the break duration. Finally, constraints (41)–(44) are binary variables restrictions and non negativity constraints.

5. Numerical experiments

In this section, multiple numerical experiments are conducted to demonstrate the efficiency and the added value of the proposed predictive selective maintenance approach and its ability to deal with maintenance decisions for fleets of multi-component systems. All experiments are run on an Intel$^{\text{TM}}$ i5 2.9 GHz desktop computer with 12 GB of RAM running Windows 10$^{\text{TM}}$. All algorithms were coded in Python 3.8. The optimisation runs were carried out by Gurobi 9.1 using gurobipy. All deep learning algorithms were coded in Python 3.8 using TensorFlow. The DL algorithms were implemented in the Google Collaboratory environment using the T4 GPU. The dataset used, preprocessing steps, and evaluation metrics will first be presented. Then after, three sets of experiments are provided and the numerical results obtained are fully discussed.

5.1. Dataset, preprocessing, and evaluation metrics

The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is used to train the FCM algorithm, Bi-LSTM and to validate the predictive selective maintenance strategy. This dataset provides degradation trajectories under 4 different operational conditions and fault modes for turbofan engines. Because these different sets consider different operational conditions and fault modes they cannot be jointly studied. In this paper, the first set called FD001 which considers 1 fault mode is selected. This set provides run to failure information for 100 engines and includes 3 operational settings and 21 sensor values for each operating cycle. An operating cycle refers to a single flight of approximately 90 min which includes both ascent to cruise at 35,000 ft and descent back to sea level. A detailed description of the 21 sensor variables can be found in Saxena et al. (2008). In total, 20631 cycles are available for training. Each engine starts with different degrees of initial wear and manufacturing variation. The operational settings and sensor values are provided for every cycle up until engine failure. A second test set is also provided that includes truncated time series of various lengths prior to failure for 100 engines and the actual RULs.

Data pre-processing of the NASA C-MAPSS FD001 dataset was carried out according to the following steps:

- **Basic data visualisation**: The first step taken was to visualise the sensor readings over the lifetime of the engines. This reveals that some sensor readings remain constant over time and add no value as inputs to the ML/DL algorithms. In particular, data collected from sensors 1, 5, 6, 10, 16, 18, and 19 were not considered when training the FCM clustering algorithm or Bi-LSTM.
- **Data labelling**: A column was added to both the training and testing sets that specified the RUL after completion of each cycle. This column provides the labels that will be used to train and test the models.
- **Input data scaling**: It is widely acknowledged that scaling input data for neural networks leads to more efficient training. The min-max normalisation approach was used to scale the input of the DL algorithm to be in the range $[0, 1]$. The min-max normalisation of data is performed according to the following formula:

$$\check{x}_j^i = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}}, \tag{45}$$

where x_j^i denotes the i^{th} data measure of the j^{th} sensor, $\check{x}_j^i$ is the scaled value of x_j^i, x_j^{max} and $x_j^{\min}$ are the respective maximum and minimum raw measurements from the j^{th} sensor.
- **Data reformatting**: In preparation for fitting the Bi-LSTM, the training and testing data was reformatted into 3-dimensional tensors of the form (N_s, N_{ts}, N_f). Recall that N_s refers to the number of samples, N_{ts} denotes the time sequence dimension, and N_f refers to the number of selected features.
- **Noise reduction**: To prepare the sensor data for the unsupervised clustering, denoising is carried out using the moving average technique. Similar to Wang et al. (2019), we use a window size of 20. Denoising is a crucial aspect of sensor data processing as it seeks to minimise the background noise in the acquired sensor measurements.

The following three commonly used metrics are used to evaluate the proposed Bi-LSTM performance for point prediction: root mean squared error (RMSE), scoring function (SC), and accuracy (AC). Smaller values of RMSE and SC, along with higher values of AC are preferred. Assuming a total of $\mathcal{N}$ test examples with $\mathcal{M}$ predictions for each test example $n \in \{1, \ldots \mathcal{N}\}$, we define $d_n = (\frac{1}{\mathcal{M}} \sum_{m=1}^{\mathcal{M}} \widehat{y}_{nm}) - y_n$ as the difference between the average prediction and the target. The mathematical expressions for the three evaluation metrics are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{\mathcal{N}} (d_n)^2}{\mathcal{N}}}. \tag{46}$$

$$\mathrm{SC} = \sum_{n=1}^{\mathcal{N}} s_n, \quad \text{where}$$

$$s_n = \begin{cases} \exp\left(\frac{-d_n}{13}\right) & \text{if } d_n \leq 0, \\ \exp\left(\frac{d_n}{10}\right) & \text{if } d_n \geq 0. \end{cases} \tag{47}$$

$$\mathrm{AC} = \frac{100}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} a_n, \quad \text{where}$$

$$a_n = \begin{cases} 1 & \text{if } d_n \in [-13, 10], \\ 0 & \text{if } d_n \notin [-13, 10]. \end{cases} \tag{48}$$

The above metrics are valuable in assessing point prediction performance. However, as pointed out earlier, it is also important to assess the performance and quality of the predicted distribution. For this, the α-Coverage (α-C) and α-Mean Width (α-MW) metrics are used (de Pater and Mitici 2022). The α-Coverage metric evaluates the reliability of the predicted distribution and is defined as follows:

$$\alpha\text{−C} = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \mathcal{I}(\alpha)_n, \quad \text{where}$$

$$\mathcal{I}(\alpha)_n = \begin{cases} 1 & \text{if } y_n \in [\widehat{y}_n^{0.5-0.5\alpha}, \widehat{y}_n^{0.5+0.5\alpha}], \\ 0 & \text{otherwise.} \end{cases} \tag{49}$$

Here, for predicted values corresponding to test instance n, $\widehat{y}_n^k$ denotes the k^{th} percentile and $[\widehat{y}_n^{0.5-0.5\alpha}, \widehat{y}_i^{0.5+0.5\alpha}]$ represents the α percent confidence interval around the median. Ideally, α-C is very close to α, as this indicates more reliable predictions. The second metric α-MW defines the mean width of the α confidence interval. Obviously, tighter intervals are preferred as this implies sharper predicted distributions. The α-MW metric is defined as follows:

$$\alpha\text{−MW} = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \left(\widehat{y}_n^{0.5+0.5\alpha} - \widehat{y}_n^{0.5-0.5\alpha}\right) \tag{50}$$

5.2. Results and analysis of FCM clustering and Bi-LSTM

The FCM clustering algorithm presented in Section 4.1 is used to partition the training examples from the C-MAPSS dataset into discrete health states. Similar to

(Chen, Lu et al. 2021), the number of clusters C is set to 4, and the degree of fuzziness m is set to 2.00. The clustering results for four of the training engines is provided in Figure 2. What can be seen is that the FCM algorithm has been able to identify four very clear health states for the training engines ranging from normal, mild degradation, moderate degradation, and severe degradation or near to failure. Three of the four training engines begin in the first health state and progressively degrade and deteriorate over operational time until failure. One may observe that training engine 81 begins in the second identified health state and progressively degrades into states 3 and 4. This may be an indication that both health states 1 and 2 are similar in terms of performance and degradation level. After training the FCM algorithm, the cluster centroids can be used to identify the health states of previously unseen observations. Figure 2 also displays the results of the clustering algorithm on four of the testing engines that have not been used in training. These results seem to indicate that the algorithm has been able to generalise well to the testing set as the four health states have been preserved. Test engine 27 is quite healthy as it has only reached health state 2, where test engine 37 is more degraded as it has entered the third health state. The RUL of test engines 27 and 37 are 66 and 21 cycles respectively. Test engines 76 and 81 are clearly much more degraded as they have entered the fourth health state and are nearing failure. The actual RULs of these two engines are 10 and 8 cycles respectively.

In terms of the DL model used for RUL prediction, initial hyperparameters were informed by reviewing similar models proposed in the literature (Zhuang, Xu, and Wang 2023). The final network structure is defined by the following hyperparameter values:

- 2 stacked Bi-LSTM layers each consisting of 20 hidden units,
- 100 neurons for the fully connected dense layer,
- dropout rate set to 0.5,
- learning rate value of 0.0005,
- batch size of 128,
- number of training epochs set to 100,
- parameter μ controlling the strength of regularisation set to $1e{-}3$, and
- the number of Monte Carlo simulations $\mathcal{M}$ set to 500.

To ensure efficiency while preventing overfitting, an early stopping procedure is implemented. Furthermore, a learning rate decay strategy is used to improve the convergence to a local minima and reduce oscillation. If no improvement to the validation loss is observed in 10 epochs, the learning rate is reduced by a factor of 0.5. Consistent with previous works (Li, Ding, and Sun 2018; Mitici et al. 2023; Xia et al. 2021), a piecewise linear target function is adopted, where the maximum Remaining Useful Life (RUL) for all training and test engines is set to 125. A sliding window of size of $N_{ts} = 50$ is also used. To deal with test instances with no more than 50 historical flight cycles, the missing flight cycles are padded with the first sensor recordings.

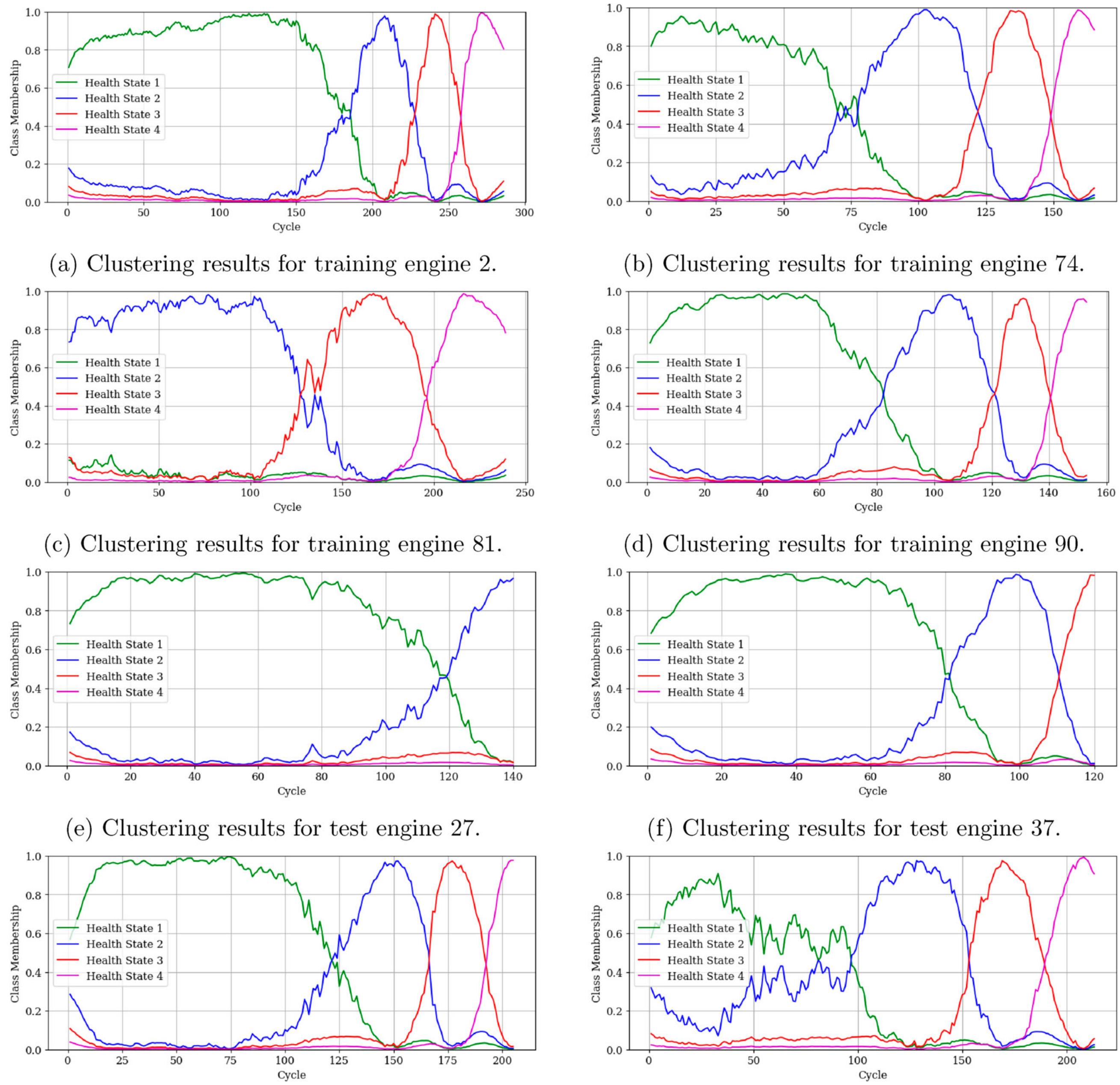
Table 1. Comparison of proposed method for RUL point prediction with the SOTA.

	RMSE	SC	AC
BiLSTM-ED (Yu, Kim, and Mechefske 2019)	14.74	273.00	57%
DBN-IPF (Peng et al. 2019)	13.11	314.00	63%
SBI (Yu, Kim, and Mechefske 2020)	13.58	228.00	67%
1-FCLCNN-LSTM (Peng et al. 2021)	**11.17**	**204.00**	–
PGRU (Zeng and Liang 2023)	12.39	–	–
C-Transformer (Zhou et al. 2023)	13.79	475.46	–
Bi-LSTM (Zhuang, Xu, and Wang 2023)	12.70	234.90	70%
Multi-channel CNN (Lee and Mitici 2023)	11.81	–	–
CNN (Mitici et al. 2023)	12.42	–	–
BGT (Xiang et al. 2024)	12.09	262.67	–
Bi-LSTM (this paper)	**11.58**	230.00	**73%**
Bi-LSTM-MCD (this paper)	11.64	**214.85**	**74%**

Table 1 displays the results obtained from the proposed Bi-LSTM models across the three RUL point prediction metrics considered. This table also reports the results of various SOTA models. From the overall results obtained, one may indeed observe that the proposed predictive RUL model with MCD is very competitive and even outperforms some SOTA models. The proposed predictive RUL model shows high accuracy, and is among the best for RMSE and the scoring function.

For selected test engines, Figures 3, 4, and 5 display the predicted RUL, the empirical RUL probability density function (*pdf*), and the failure probability, respectively. The latter is computed by assessing the proportion of predictions that are less than the specified mission length. The following key observations can be made.

- From Figure 3, it appears that the Bi-LSTM is capable of making accurate RUL predictions. One can note that the accuracy of the model seemingly increases as the true RUL approaches 0. This means that the model makes more accurate predictions as more data becomes available. For test engines 24, 56, and 100, it can be seen that when the true RUL is at most 50 cycles, the model can almost perfectly predict the true remaining life. The information presented in Table 2 further supports this observation. This table shows the RMSE and AC for varying monitoring points along with the α-C and α-MW metrics for two values of α. The metrics are computed for test engines that have RULs of 75, 50, and 25 cycles. The results obtained show that as the true RUL decreases, RMSE decreases while the AC increases. This further demonstrates the



(a) Clustering results for training engine 2.
(b) Clustering results for training engine 74.
(c) Clustering results for training engine 81.
(d) Clustering results for training engine 90.
(e) Clustering results for test engine 27.
(f) Clustering results for test engine 37.
(g) Clustering results for test engine 76.
(h) Clustering results for test engine 81.

Figure 2. Clustering results for multiple training and testing engines.

enhanced ability of the model to accurately predict RUL as more cycles are completed and more data becomes available. This enhanced predictive ability may be a result of clear degradation trends emerging. It should be noted that the number of samples available for the three different monitoring points are 43, 33, and 19 respectively.

- Still dealing with Figure 3, one may observe that as the true RUL decreases, the size of the 95% confidence interval reduces. Looking at the RUL *pdf* curves displayed in Figure 4, the same trend emerges, and the standard deviation of the predicted RUL distribution increases as the true RUL value increases. Referring to the results reported in Table 2, one may observe that the uncertainty is overestimated for the test engines with 50 and 25 cycles remaining, and a drastic underestimation in the uncertainty occurs when dealing with test engines with 75 cycles remaining. Results also show that α-MW decreases as the true RUL decreases, emphasising the sharper predictions achieved for engines with smaller remaining lifetimes.

- Figure 4 demonstrates the importance of considering the RUL distribution rather than the RUL point prediction when dealing with maintenance decisions. To illustrate, let us consider the example of test engine 36, and assume that the upcoming mission length $U = 20$ cycles. In this case, as the mean predicted RUL of 23.2 is greater than 20, a maintenance decision based solely on the RUL point prediction would result in no maintenance being performed. This decision would lead to engine failure as the true RUL is only 19 cycles. However, the RUL distribution shows that the probability of mission success is approximately only 50%. In this case, if a moderately high reliability target was set, the engine would be selected for maintenance, ultimately preventing failure. This highlights the need to incorporate the RUL distribution into the maintenance decision process.
- The failure curves in Figure 5 show that as the true RUL approaches the mission length U, the predicted failure probability increases significantly. In most cases, the probability is close to 1 when the RUL value is approaching the mission length U. This indicates that the proposed approach for failure prediction is accurate and behaving as expected. Ideally, if the model could perfectly predict RUL, the failure curves would exactly match the ideal failure curves. However, imperfection is inherent in every prediction model used, which introduces some level of non-smoothness in the failure curves and inaccuracy in the failure prediction.
- Dealing with the prediction of the RUL distribution, the quality of the Bi-LSTM model is discussed by comparing it to the CNN model used in Mitici et al. (2023). This comparison is based on the α-C and α-MW metrics as reported in Table 3 for $\alpha = \{0.5, 0.9, 0.95\}$. The α-C values are very close to the α values. When $\alpha = 0.5$, the proposed Bi-LSTM model slightly overestimates the uncertainty. However, it slightly underestimating uncertainty in the case where $\alpha = 0.95$. For all values of α, the results obtained also show that the α-MW values obtained with the proposed model are smaller (*i.e.* better) than those obtained for the CNN model in Mitici et al. (2023).

Table 2. Comparison of prediction metrics at different monitoring points.

		Monitoring point with 75 cycles left	Monitoring point with 50 cycles left	Monitoring point with 25 cycles left
	RMSE	17.33	6.71	5.07
	AC	56%	94%	100%
$\alpha = 0.90$	α-C	0.70	0.91	0.95
	α-MW	33.50	24.26	15.06
$\alpha = 0.95$	α-C	0.74	0.97	1.00
	α-MW	39.89	28.65	17.92

Table 3. Comparison of α-Coverage (α-C) and α-Mean Width (α-MW).

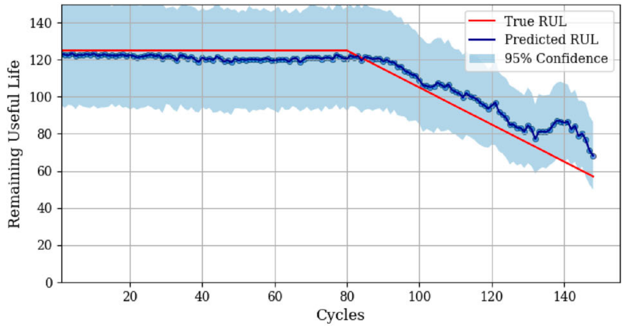
		Bi-LSTM-MCD (this paper)	CNN (Mitici et al. 2023)
$\alpha = 0.50$	α-C	0.52	0.54
	α-MW	12.72	16.30
$\alpha = 0.90$	α-C	0.90	0.91
	α-MW	30.99	39.20
$\alpha = 0.95$	α-C	0.92	0.95
	α-MW	36.86	46.4

5.3. Experiment #1: comparison of reliability prediction methods

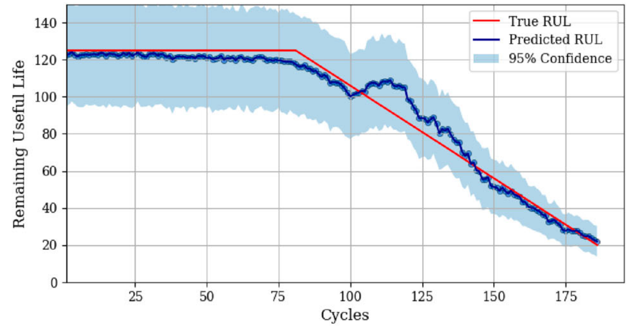
This set of numerical experiments intends to compare the proposed approach for reliability assessment with traditional statistical approaches and other data-driven methods from the literature. The following methods and models are compared with the perfect RUL prognostics model as a benchmark.

- Bi-LSTM with Monte Carlo dropout (Bi-LSTM-MCD), the proposed approach,
- An approach that leverages the RUL point prediction (P.RUL) of the Bi-LSTM,
- The best fitting statistical distribution,
- A probabilistic classifier for reliability prediction (Bi-LSTM-PC), and
- A perfect/true RUL (T.RUL) prognostics model (Benchmark).

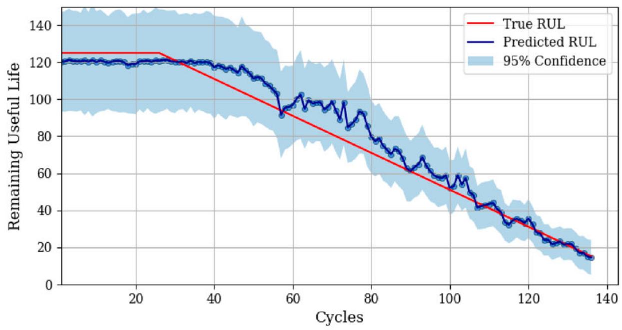
This first set of experiments considers a single system comprised of one turbo fan engine. To analyse the performance of the above listed methods for reliability assessment over multiple instances, simulations are run for the entire life cycle of all 100 training engines in the FD001 dataset. Each engine is put into operation at time 0 and performs a mission followed by a maintenance break. The proposed optimisation model is used to determine if maintenance is required during the break duration. If no maintenance is selected, the engine is sent off for another mission, and the process is repeated until the engine is replaced or fails during a mission. This procedure is performed for all 100 engines, which are divided into 5 folds, each composed of 80 training engines and 20 testing engines. It is worth noting that each testing set is totally unique, meaning that the 5 testing sets comprise all 100 engines. To ensure a fair comparative study, for each training set, a different DL model is trained and used



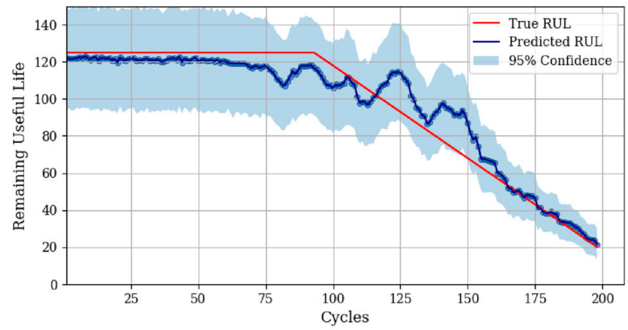
(a) RUL interval estimate for test engine 21.



(b) RUL interval estimate for test engine 24.

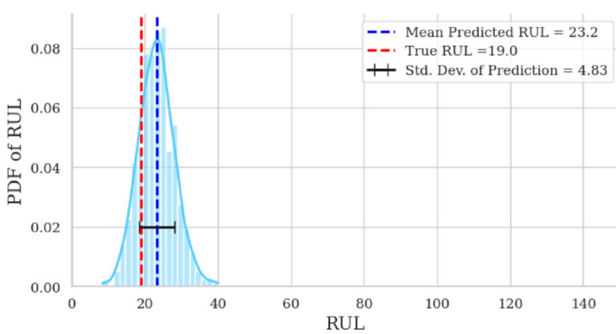


(c) RUL interval estimate for test engine 56.

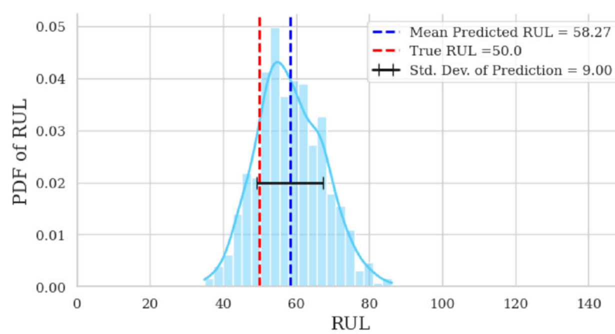


(d) RUL interval estimate for test engine 100.

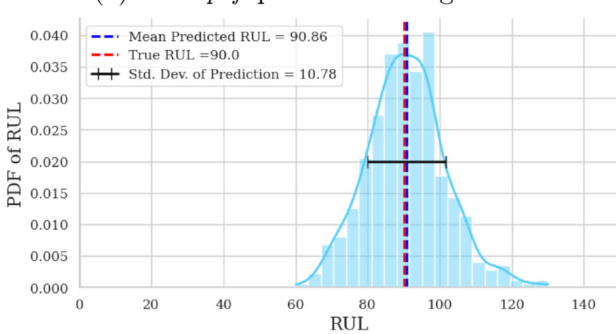
Figure 3. RUL interval estimates for multiple test engines.



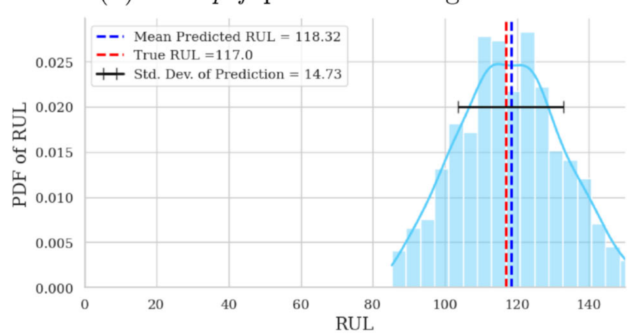
(a) RUL *pdf* plot for test engine 36.



(b) RUL *pdf* plot for test engine 72.

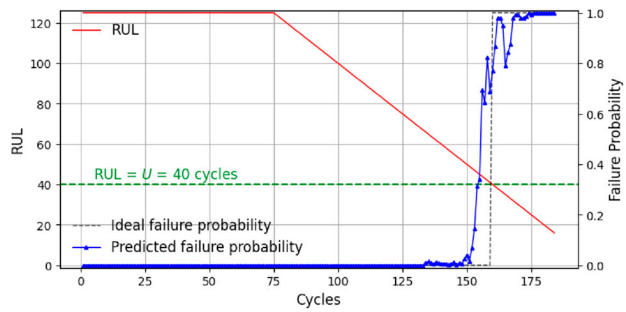


(c) RUL *pdf* plot for test engine 80.

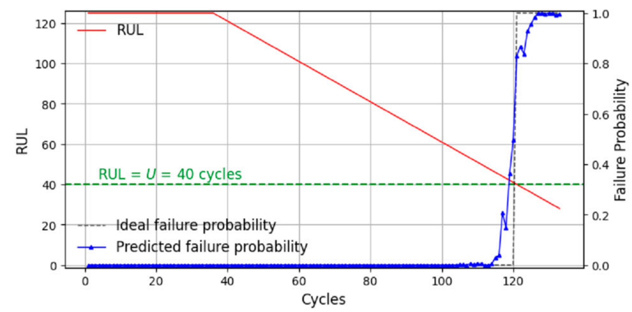


(d) RUL *pdf* plot for test engine 99.

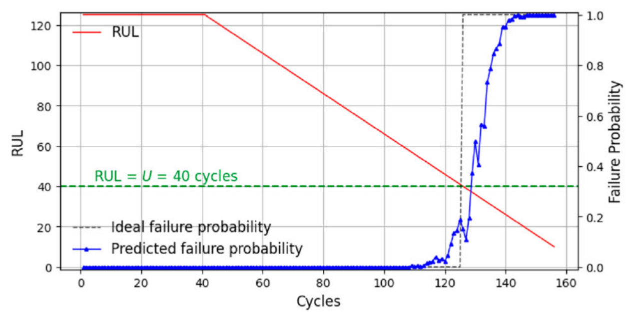
Figure 4. RUL *pdf* plots for multiple test engines.



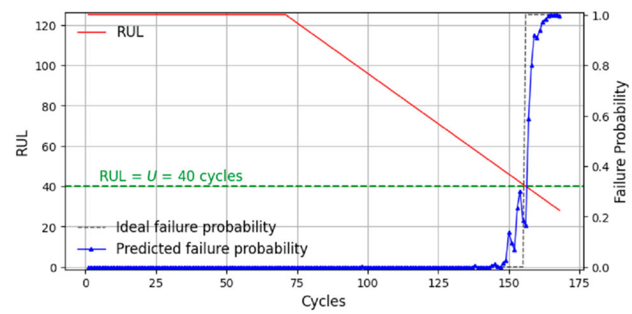
(a) Failure curves for test engine 20.



(b) Failure curves for test engine 40.



(c) Failure curves for test engine 42.



(d) Failure curves for test engine 64.

Figure 5. Failure probability curves when $U = 40$ cycles for multiple test engines.

to predict the RUL distribution of the test engines in the corresponding test set.

The best fitting statistical distribution is found to be the Weibull distribution with a shape and scale parameter of $\beta = 4.39$ and $\eta = 224$ respectively, similar to the parameters found in previous papers (Hesabi, Nourelfath, and Hajji 2022; Shoorkand, Nourelfath, and Hajji 2024). For the probabilistic classifier, an architecture similar to the one discussed in Section 5.2 is used, with the exception of the dropout rate reduced to 0.2, and the initial learning rate set to 0.001. The final layer of the classifier is comprised of two neurons, where the *softmax* activation function is employed to determine the probability that the component belongs to one of the two following classes:

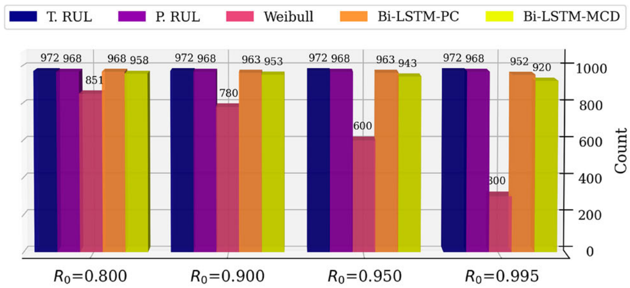
- **Class 0:** RUL of the component is greater than the mission length U.
- **Class 1:** RUL of the component is less than or equal to the mission length U.

The probability that a component belongs to Class 0 can be interpreted as its reliability. Here, the categorical cross entropy loss function is minimised during the training of the probabilistic classifier.

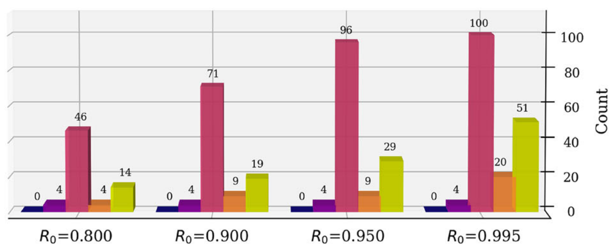
To compare the methods listed above under similar conditions, all costs excluding maintenance cost are set to 0, while the reliability target $\mathcal{R}_0$ is varied. Furthermore, the maintenance cost is set to a very small value to perform maintenance only when necessary, and sufficient time is available to perform maintenance actions. The comparison uses three indicators: the number of early repairs, the number of system failures, and the number of missions completed. The number of early repairs refers to the number of times that a repair was performed when it was not necessary. An early repair is recorded when a replacement is selected despite the true RUL of a given component being greater than the upcoming mission length U. The decision to perform maintenance depends on both the predicted reliability and the target reliability threshold. The experiments are carried out for $U = 20$ cycles.

The first immediate observation that can be made from the results in Figure 6 is the significant improvement achieved by the DL approaches. Another observation is that as the target reliability increases, both the number of system failures and the number of completed missions decrease, while the number of early repairs increases. This is an expected outcome since an increase in the target reliability implies a heightened emphasis on failure prevention.

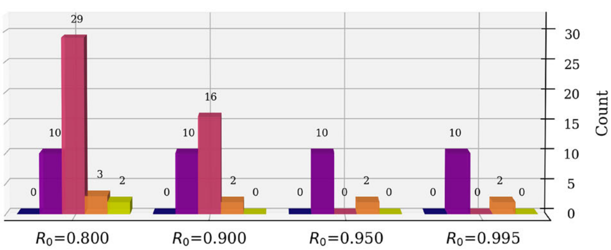
Utilising a statistical approach (Weibull distribution) results in significantly more early replacements, and far fewer completed missions in most cases. These results highlight the advantage of data-driven methods over distribution-based approaches for reliability prediction.



(a) Number of completed missions.



(b) Number of early repairs.



(c) Number of system failures.

Figure 6. Comparing methods for reliability assessment for $U = 20$ cycles and varying values of $\mathcal{R}_0$. Case of experiment #1.

The data-driven approaches are indeed more accurate as they base their predictions on the current health-condition of an engine.

Another important observation to point out, is the seemingly better prediction made by the probabilistic classifier compared to the regression model implementing Monte Carlo dropout. In fact, for many of the instances solved, the probabilistic classifier for reliability prediction results in fewer early repairs, and higher number of successful missions. In other words, the probabilistic classifier seems to better utilise the full operational life of the engines. However, what must also be emphasised is the difference in the number of system failures observed when the reliability target is set to a high value (*i.e.* greater than or equal to 95%). Even with extremely high reliability targets of 95.0% and 99.5%, the probabilistic classifier still allows several system failures to occur. This means that the model is predicting an extraordinarily high likelihood of mission success, when, in fact, the system will fail during the upcoming mission. In the context of mission-critical systems, whose failure could be catastrophic, the Bi-LSTM-MCD should be preferred to the Bi-LSTM-PC. Conversely, the Bi-LSTM-PC should be preferred if failures are not costly.

The results displayed in Figure 6 also highlight the benefit of using the RUL distribution rather than the point prediction. The RUL point prediction yields fewer early repairs and more missions are completed. However, this approach leads to many more system failures.

5.4. Experiment #2: application to fleets of large multi-component systems

This set of numerical experiments solves the fleet selective maintenance problem (FSMP) for a fleet of 5 large multi-component systems. Each system is comprised of 3 subsystems consisting of 5 turbofan engines, where at least 3 of the 5 engines must function in order for the subsystem to function. Thus, the reliability structure of each subsystem is a 3-out-of-5:G. To study the long-run performance of the proposed fleet selective maintenance (FSM) approach, several breaks and missions are replicated: an engine from the FD001 training is randomly assigned to each component within each system. At the start of the first maintenance break, all engines are considered to be brand new. The fleet performs a mission of length $U = 30$ cycles, after which the proposed maintenance optimisation model is used to determine what maintenance actions to perform, the number of repairpersons to hire and utilise, in addition to the assignment of these repairpersons to maintenance tasks. After the maintenance break, the system is put into operation to perform another mission of the same length $U = 30$ cycles. This process is repeated for 100 mission and break cycles. During a break, if an engine is selected for maintenance, it is replaced by an engine which is randomly drawn from the training set. The entire process is repeated 5 times and the mean and standard deviations are reported for multiple metrics in Figure 7. These metrics are reported for the different methods used for reliability prediction.

For this experiment, 10 repairpersons are available with fixed and variables costs of $c_r^f = 5$ monetary units (m.u.) and $c_r^v = 1$m.u./hr, respectively. The time and cost of preventive maintenance for all health states is assumed to be the same and set to $c_{ijk2}^p = 2$ m.u. and $t_{ijk2}^p = 1$ hour. Additionally, the time and cost of corrective maintenance actions are set to $c_{ijk2}^c = 5$ m.u. and $t_{ijk2}^c = 2$ hours. Finally, the break duration is set to 10 hours. The penalty cost Π_{ijk} is the same for all components and the target reliability threshold $\mathcal{R}_0 = 0.95$.

The results obtained are summarised in Figure 7 which displays the average number of system failures, the average total cost per break, the average component

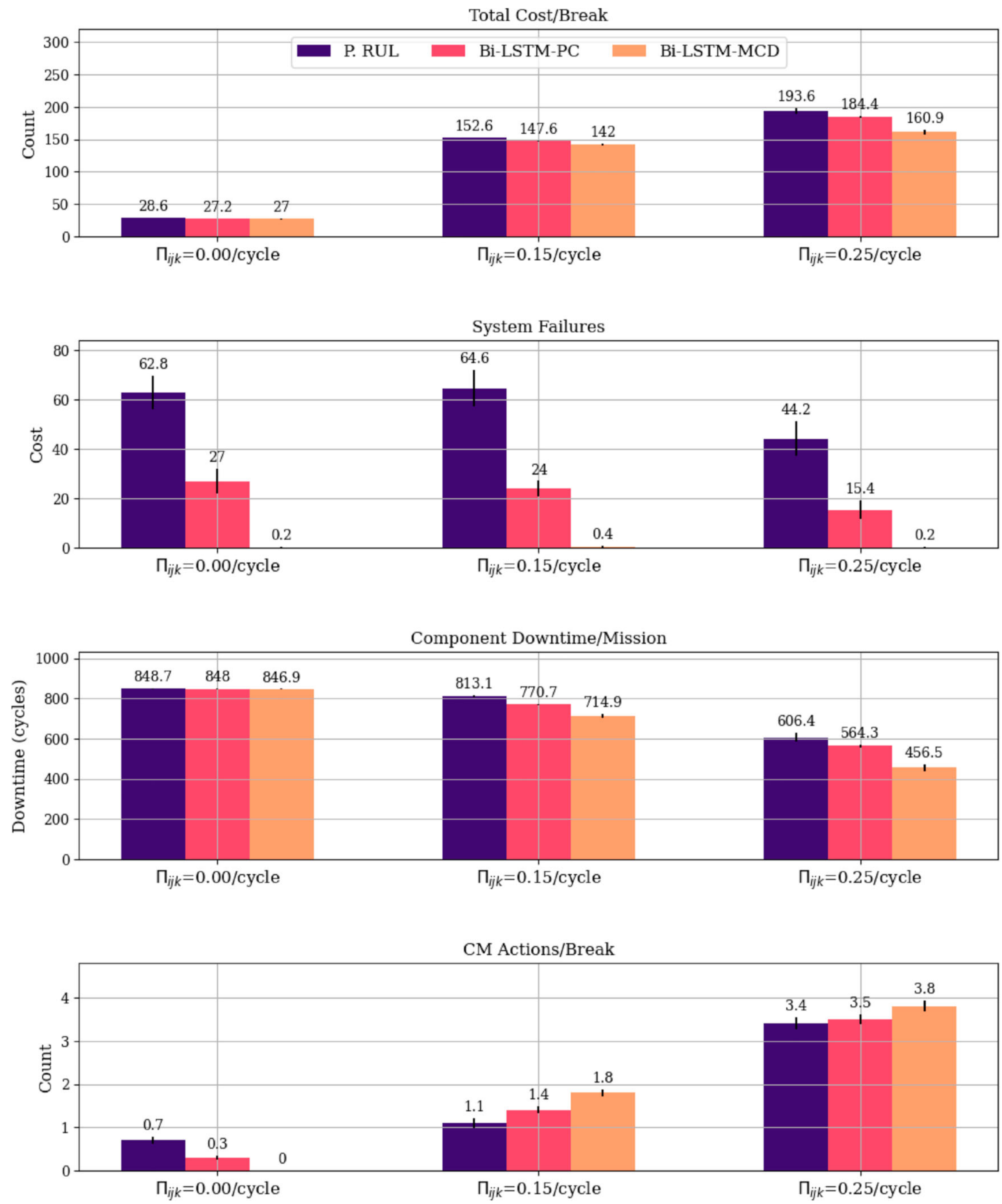


Figure 7. Comparing methods for reliability assessment for $\mathcal{R}_0 = 0.95$ and $D = 10$ hours for varying values of Π_{ijk}. Case of experiment #2.

downtime per mission, and the average number of CM actions performed per maintenance break. An advantage of using the Bi-LSTM-MCD can be seen immediately from the histograms corresponding to the average number of system failures, and the resulting grand total cost. Compared to the probabilistic classifier or the method using the RUL point prediction, the Bi-LSTM-MCD method is very competitive as it provides cost effective decisions while reducing system failures. For all reliability prediction methods considered, the number of system failures decreases as the component downtime penalty cost increases. This is due to the fact that when the penalty cost increases, components are more likely to be selected for maintenance, which then leads to an increase in their reliability and therefore a reduction in their downtime. The component penalty cost 'takes over' the objective function and the primary goal is to reduce component downtime.

Dealing with the particular case where no penalty cost is incurred for component failure during missions, a very small number of CM are performed resulting in high downtime. This is due to the fact that CM is typically costlier than PM, thus it is unlikely that the model will ever elect to perform a CM action. However, in a context where individual components contribute to the overall efficiency and global performance of the system, it is undesirable and impractical to allow a component to remain in the failed state for an extended period of time. Thus, it is required to explicitly account for component downtime in the maintenance optimisation problems in general, and in the SMP in particular.

5.5. Illustrative case study: multi-engine aircraft

This set of experiments illustrates how the proposed predictive SM approach can deal with joint maintenance and repairpersons assignment decisions for a fleet composed of several multi-engine aircraft. The fleet investigated is composed of 6 aircraft ($i = 1, \ldots, 6$) each equipped with 4 turbofan engines ($k = 1, \ldots, 4$). An aircraft will successfully operate the upcoming mission if at least two of the four engines survive the mission duration. Thus, the reliability structure of an aircraft's engine system is a 2-out-of-4:G structure where each component E_{i1k} is assigned the sensor values of an engine from the test set of the NASA-CMAPSS FD001 dataset. The duration of the next mission is set at $U = 40$ cycles. To perform maintenance activities, $Q = 15$ repairpersons are available with fixed cost and variable cost rate of $c_r^f = 50$ m.u. and $c_r^v = 5$m.u./hr respectively. Table 4 provides the engine data for each aircraft. This data includes the number of cycles that each engine has completed, health state membership, its corresponding true and predicted remaining useful life, predicted reliability $\mathcal{R}_{i1k1}$ and predicted downtime Δ_{i1k1}. Aircraft 3, 4, and 5 will fail during the next mission as at least 3 of the engines equipped have true remaining lifetimes that are less than the 40 cycle mission length. The average maintenance time corresponding to health states HS1 (normal), HS2 (mild degradation), HS3 (moderate degradation), and HS4 (near failure) are 4, 5, 8, and 10 hours respectively and the maintenance cost $c_{ijk2}^p = 30$ m.u.

5.5.1. Illustrative case study: exploring the impact of $\mathcal{R}_0$ and Π_{ijk}

To show how the required minimum reliability level impacts the maintenance and repairpersons assignment decisions within a FSMP, this experiment is conducted in the case where the break duration $D = 15$ hours while the required minimum reliability $\mathcal{R}_0$ and component penalty cost Π_{ijk} are varied. The latter is set to be the same for all engines/components for simplicity. The results obtained are reported in Table 5. This table gives the optimal predicted objective function value $\mathcal{Z}^*$, the true grand total cost $\mathcal{Z}'$, the total number NC^* of components maintained, the number Q^* of repairpersons hired and utilised, the total maintenance time (MT), in addition to the total component downtime (DT). The total maintenance time is computed by summing the individual repairperson maintenance times. It is worth noting here that because components downtimes are computed using the predicted RUL distribution opposed to the true RUL, it results in a difference between the predicted optimal objective function and true grand total cost.

When the downtime penalty cost is 0, it can be seen that as the required minimum reliability is increased, the total maintenance time and total number of maintenance actions increase. An increase in the number of repairpersons utilised when $\mathcal{R}_0$ increases from 0.950 to 0.995 can also be observed. The case where $\Pi_{ijk} = 4$ m.u./cycle shows that increasing the reliability target results in an increase in the total maintenance time, number of maintenance actions, and number of repairpersons utilised. Both increases and decreases in the total component downtime can be seen as the reliability target is increased, highlighting the complex interaction between penalty, maintenance, and crewing costs. When Π_{ijk} is set to 8 m.u./cycle and 16 m.u./cycle, we see the respective maintenance plans remain fixed for all reliability targets. This indicates that the reliability constraint is not tight, *i.e.* the maintenance plan is not constrained by the target reliability threshold.

The penalty cost per cycle Π_{ijk} clearly influences maintenance decisions related to the number of repair crews to hire/utilise and the components to maintain. When $\mathcal{R}_0 = 0.95$, Figure 8 displays both the total cost as well as the total component downtime for increasing values of Π_{ijk}. It is evident that as Π_{ijk} increases, there is an emphasis on ensuring that no components fail during the subsequent mission, effectively minimising downtime. However, achieving minimal downtime requires increases in maintenance cost, fixed and variable crew costs, ultimately leading to an increase in grand total cost.

When $\mathcal{R}_0 = 0.75$ and $\Pi_{ijk} = 0$ m.u./cycle, the following components are selected for maintenance: $E_{3,1,2}$, $E_{3,1,3}$, $E_{4,1,1}$, $E_{5,1,2}$, and $E_{5,1,3}$ resulting in a grand total cost of 560. It is interesting to note that when the fleet reliability target is set to a relatively low value of 0.75, all aircraft that would have failed during the mission are selected to be maintained. As the reliability target is increased, additional aircraft equipped with engines having RULs nearing 40 cycles are also selected for maintenance. For fleets comprised of critical assets where the failure of

Table 4. Engine data used for case study.

Components	Engine assignment	Health state membership	True RUL (cycles)	Predicted RUL (cycles)	$\mathcal{R}_{i1k1}$	Δ_{i1k1}
E_{111}	17	[0.05, 0.57, 0.35, 0.03]	50	54	0.98	0.07
E_{112}	18	[0.01, 0.06, 0.91, 0.02]	28	33	0.11	7.59
E_{113}	20	[0.00, 0.01, 0.02, 0.96]	16	13	0.00	26.85
E_{114}	21	[0.29, 0.63, 0.06, 0.01]	57	68	1.00	0.01
E_{211}	24	[0.02, 0.09, 0.86, 0.03]	20	22	0.00	18.07
E_{212}	27	[0.02, 0.96, 0.02, 0.00]	66	66	1.00	0.00
E_{213}	31	[0.00, 0.01, 0.02, 0.96]	8	11	0.00	28.95
E_{214}	32	[0.03, 0.93, 0.03, 0.01]	48	45	0.76	0.92
E_{311}	34	[0.01, 0.01, 0.03, 0.95]	7	8	0.00	31.28
E_{312}	35	[0.01, 0.02, 0.04, 0.93]	11	12	0.00	28.26
E_{313}	46	[0.01, 0.04, 0.92, 0.03]	47	38	0.38	3.28
E_{314}	49	[0.00, 0.01, 0.03, 0.96]	21	25	0.00	14.7
E_{411}	52	[0.04, 0.38, 0.55, 0.03]	29	30	0.00	10.02
E_{412}	56	[0.00, 0.01, 0.02, 0.97]	15	14	0.00	25.82
E_{413}	58	[0.02, 0.07, 0.82, 0.09]	37	37	0.29	3.44
E_{414}	62	[0.05, 0.44, 0.47, 0.03]	54	46	0.85	0.49
E_{511}	68	[0.00, 0.00, 0.00, 0.99]	8	12	0.00	28.27
E_{512}	77	[0.02, 0.07, 0.80, 0.11]	34	29	0.01	10.00
E_{513}	81	[0.01, 0.02, 0.06, 0.91]	8	9	0.00	30.84
E_{514}	84	[0.00, 0.01, 0.02, 0.97]	38	34	0.14	7.06
E_{611}	91	[0.01, 0.05, 0.90, 0.04]	58	62	1.00	0.00
E_{612}	94	[0.02, 0.07, 0.83, 0.09]	55	52	0.97	0.13
E_{613}	98	[0.03, 0.24, 0.69, 0.03]	59	80	1.00	0.00
E_{614}	100	[0.02, 0.08, 0.86, 0.04]	20	21	0.00	18.3

Table 5. Results for varying values of $\mathcal{R}_0$ and Π_{ijk} with $D = 15$ hours.

Π(m.u./cycle)	$\mathcal{R}_0$	$\mathcal{Z}^*$(m.u.)	$\mathcal{Z}'$(m.u.)	$\frac{\mathcal{Z}^* - \mathcal{Z}'}{\mathcal{Z}'} \cdot 100$ (%)	NC^*	Q^*	MT (hours)	DT (cycles)
0	0.750	560	560	0.00	5	4	42.0	222
	0.850	616	616	0.00	6	4	47.1	222
	0.950	678	678	0.00	7	4	53.6	222
	0.995	789	789	0.00	8	5	59.8	222
4	0.750	1245	1242	0.24	5	5	49.2	149
	0.850	1259	1243	1.29	6	6	59.1	117
	0.950	1315	1300	1.15	7	6	63.6	118
	0.995	1337	1334	0.22	8	7	73.5	94
8	0.750	1587	1542	2.92	10	9	91.2	42
	0.850	1587	1542	2.92	10	9	91.2	42
	0.950	1587	1542	2.92	10	9	91.2	42
	0.995	1587	1542	2.92	10	9	91.2	42
16	0.750	1820	1654	10.04	13	12	116.7	5
	0.850	1820	1654	10.04	13	12	116.7	5
	0.950	1820	1654	10.04	13	12	116.7	5
	0.995	1820	1654	10.04	13	12	116.7	5

one system would have catastrophic consequences, the decision maker may set a high reliability target. This will indeed reduce the risk of mission failure, but at the expense of maintenance cost.

5.5.2. Illustrative case study: exploring the impact of mission and break durations

The purpose of the present experiment is to demonstrate the impact of the scheduled break duration D on the maintenance plans. The FSMO problem is solved for each break duration $D \in \{10, 15, 20, 25\}$, while the required minimum reliability level $\mathcal{R}_0 = 0.95$. For each problem instance, the optimal grand total cost $\mathcal{Z}'$, the total number NC^* of components maintained, the number Q^* of repairpersons hired and utilised, and the total downtime (DT) are provided in Table 6. This table also displays the results for the perfect RUL predictive model (T.RUL), which serves as the ideal scenario.

As expected, when $\Pi_{ijk} = 0$ m.u./cycle, increasing the maintenance break duration results in fewer repairpersons being hired to complete the maintenance duties. Table 6 also depicts the gap between the optimal grand total cost values resulting from the two reliability prediction methods. The Bi-LSTM-MCD model suggests performing 7 maintenance actions, while the T.RUL method recommends only 4 maintenance actions to meet the required minimum reliability $\mathcal{R}_0 = 0.95$. This once again highlights the tendency of the Bi-LSTM-MCD model to select more maintenance actions for high reliability thresholds to protect against failures.

Results from Table 6 show that the gap in grand total cost decreases as the downtime penalty cost rate Π_{ijk}

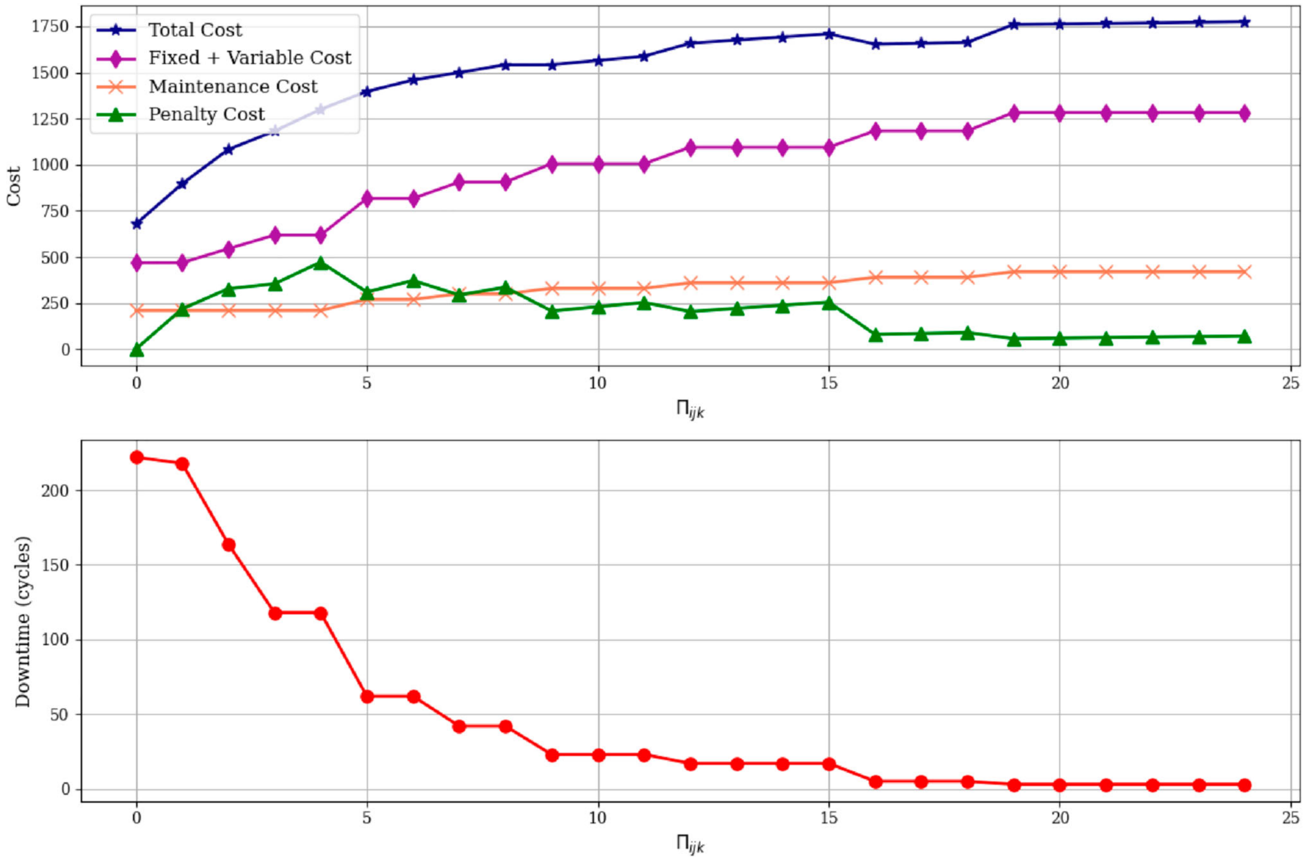


Figure 8. Component downtime and total cost for varying values of Π_{ijk}.

Table 6. Impact of the break and mission duration on maintenance plans.

		Bi-LSTM-MCD				T.RUL				
Π (m.u./cycle)	D (hours)	$\mathcal{Z}'$(m.u.)	NC^*	Q^*	DT (cycles)	$\mathcal{Z}'$(m.u.)	NC^*	Q^*	DT (cycles)	Gap(%)
0	10	828	7	7	222	491	4	4	222	68.84
	15	678	7	4	222	441	4	3	222	53.74
	20	628	7	3	222	391	4	2	222	60.61
	25	628	7	3	222	391	4	2	222	60.61
4	10	1313	7	7	106	1229	4	4	178	6.83
	15	1300	7	6	118	1229	4	4	178	5.78
	20	1146	8	4	82	1093	6	3	117	4.85
	25	1135	9	4	62	1093	6	3	117	3.84
8	10	1592	10	10	42	1543	10	10	34	3.18
	15	1542	10	9	42	1519	11	10	23	1.51
	20	1342	10	5	42	1292	12	6	11	3.87
	25	1269	11	5	23	1266	11	5	22	0.24
16	10	1704	13	13	5	1680	12	12	11	1.43
	15	1654	13	12	5	1630	12	11	11	1.47
	20	1451	14	7	3	1380	12	6	11	5.14
	25	1401	14	6	3	1354	13	6	5	3.47

increases. The same trend is observed for the optimal number NC^* of components to be maintained, and the number Q^* of the repair crews hired and utilised.

One trend emerging for larger values of Π_{ijk} is the decreasing downtime as the break duration is increased. This is an expected result as increasing the break duration allows for more maintenance tasks to be performed, ultimately leading to a reduction in total component downtime.

Finally, Figure 9 depicts the relationship between total cost, break duration and mission length. It shows that the total cost increases as the length of the mission is increased, and as the break duration is decreased. The first observation is due to the fact that when the mission length is increased, more components need to be maintained to reduce components downtime which in turn increase the probability of mission success. Regarding the second observation, it is due to the fact that when

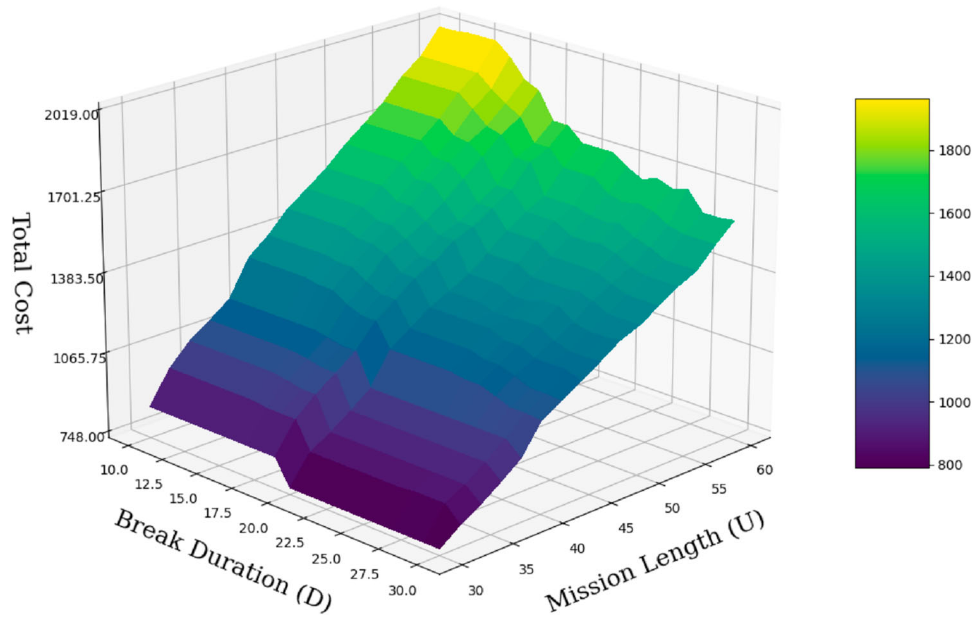


Figure 9. Total cost for varying values of U and D with $\Pi_{ijk} = 4$/cycle and $\mathcal{R}_0 = 0.950$.

the break duration is shortened, more repairpersons are required to perform the maintenance duties.

The results obtained and discussed above show that the proposed integrated DL and optimisation method yields reasonable and valid maintenance and repairperson assignment decisions.

6. Conclusion

In this paper, a fleet selective maintenance strategy is proposed that utilises recent advancements in artificial intelligence (AI) for reliability assessment. An integrated deep learning (DL) and optimisation approach is used to identify optimal maintenance decisions. This paper improved upon the original data-driven selective maintenance strategy proposed by Hesabi, Nourelfath, and Hajji (2022), specifically by considering fleets of large multicomponent systems and by developing and efficient solution method. This paper considered the impact that degradation has on maintenance time and also dealt with the assignment of multiple repairpersons to maintenance tasks. A Bidirectional Long Short-Term Memory (Bi-LSTM) network implementing Monte Carlo dropout was used to predict a remaining useful life (RUL) distribution from which component and system reliability could be derived. A comparison of several state-of-the-art DL algorithms for the purpose of reliability assessment was also provided. The results from the comparative study indicated that the proposed approach was preferable compared to methods leveraging remaining useful life (RUL) point predictions and probabilistic classifiers. The Bi-LSTM network was able to achieve a very high RMSE and accuracy indicating its ability to accurately predict RUL. The Bi-LSTM also showed excellent results in metrics used for assessing the quality of the predicted RUL distribution such as α-Coverage and α-Mean Width.

In this work we consider perfect repair of components as the only maintenance action. A way of extending and improving upon this work would be to consider several imperfect maintenance levels, where a component is brought back to a state that is between perfect replacement and do-nothing. Modelling imperfect maintenance and its impact when using data-driven approaches for reliability assessment is something that needs to be studied. Another very important extension of the proposed framework is the development of a maintenance optimisation model that takes into consideration multiple future missions. Currently, our approach only considers the upcoming mission, however, this does not necessarily produce cheap maintenance plans in the long run. Thus, it would be of great benefit to extend the proposed framework to the multi-mission context.

Due to time restrictions, the number of iterations or combinations of hyperparameters used in the random search was limited to 25. By trying more combinations of hyperparameters, we would have stronger evidence that the selected hyperparameters are optimal for model performance. Areas of future work would include implementing data reduction methods such as principle component analysis (PCA), autoencoders (AE), and variational autoencoders (VAEs). AEs and VAEs have the ability to learn compressed representations of the data and extract meaningful features. It would also be interesting to study more complex DL architecture for the task of predicting RUL, such as transformers and hybrid deep learning approaches. Hybrid approaches that utilise multiple DL algorithms have shown promise in RUL prediction. Another area of future research would be studying deep clustering algorithms for the purpose of health state identification. Finally, what has yet to be studied in the context of predictive selective maintenance is the development of multimodal learning models for the task of reliability prediction. The development of a multimodal learning model that combines not only time-series sensor data, but also maintenance reports, thermal imaging, and machine video to make reliability predictions has the potential to significantly improve prediction accuracy.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

Notes on contributors

Ryan O'Neil is currently pursuing a Ph.D. degree from the Department of Industrial Engineering at Dalhousie University, Halifax, Nova Scotia, Canada. He holds a Master of Applied Science degree and a bachelor's degree in Industrial Engineering from Dalhousie University. His current research interests include machine learning, large-scale and robust optimisation of intelligent maintenance operations.

Abdelhakim Khatab Ph.D., is Professor at Université de Lorraine (France). After graduating from the High Normal School (ENS-Mohammedia, Morocco), He received a DEA (Msc degree) and a PhD in Industrial automation engineering from National Institute of Applied Science (INSA) of Lyon (France). Since 2018, Dr. Khatab is adjunct Professor at the Department of Industrial Engineering-Dalhousie University (Canada). He is member of IFAC TC 5.2 where he is chair of a working group on Smart, Reliable and Sustainable Manufacturing-Distribution Systems. He also served as a scientific expert member of the Natural Sciences and Engineering Research Council of Canada (NSERC). Dr. Khatab research interests include reliability theory, optimisation and decision support for system's production and intelligent maintenance management, design and optimisation of reverse supply chains, sustainability, and remanufacturing.

Claver Diallo Ph.D., P.Eng., is Professor in the Department of Industrial Engineering at Dalhousie University in Halifax, Nova Scotia. He has taught at Dalhousie University since October 2007. He holds a Ph.D. and a Master of Applied Science degree in Industrial Engineering, and a bachelor's degree in mechanical engineering from Laval University, Quebec, Canada. He is a member of the Institute of Industrial and Systems Engineering (IISE), the Canadian Operational Research Society (CORS) and Engineers Nova Scotia (ENS). He is a member of IFAC TC 5.2. His current research is focussed on reliability engineering & predictive maintenance, production and distribution systems design within the Industry 4.0/5.0 context which includes hyperconnected logistics networks, smart production planning and control, and sustainable supply chain management.

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article.

ORCID

A. Khatab http://orcid.org/0000-0003-1166-3806
C. Diallo http://orcid.org/0000-0002-7381-2187

References

Abdar, M., F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, et al. 2021. "A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges." *Information Fusion* 76: 243–297. https://doi.org/10.1016/j.inffus.2021.05.008.

Achamrah, F. E., and A. Attajer. 2023. "Multi-Objective Reinforcement Learning-Based Framework for Solving Selective Maintenance Problems in Reconfigurable Cyber-Physical Manufacturing Systems." *International Journal of Production Research* 62 (10): 3460–3482. https://doi.org/10.1080/00207543.2023.2240433.

Al-Jabouri, H., A. Saif, C. Diallo, and A. Khatab. 2023. "Branch-and-Price Algorithms for Large-Scale Mission-Oriented Maintenance Planning Problems." *Computers & Operations Research* 153: 106191. https://doi.org/10.1016/j.cor.2023.106191.

Al-Jabouri, H., A. Saif, C. Diallo, and A. Khatab. 2024. "Distributionally-Robust Chance-Constrained Optimization of Selective Maintenance Under Uncertain Repair Duration." *Expert Systems with Applications* 239: 122303. https://doi.org/10.1016/j.eswa.2023.122303.

Al-Jabouri, H., A. Saif, A. Khatab, C. Diallo, and U. Venkatadri. 2022. "Selective Maintenance Optimization: A Condensed Critical Review and Future Research Directions." *IFAC-PapersOnLine* 55 (10): 1213–1218. https://doi.org/10.1016/j.ifacol.2022.09.555.

Al-Jabouri, H., A. Saif, A. Khatab, C. Diallo, and U. Venkatadri. 2023. "A Critical Review of Selective Maintenance for Mission-Oriented Systems: Challenges and a Roadmap for Novel Contributions." *International Journal of Production Research*: 1–36. https://doi.org/10.1080/00207543.2023.2270689.

An, D., J.-H. Choi, and N. H. Kim. 2018. "Prediction of Remaining Useful Life Under Different Conditions Using Accelerated Life Testing Data." *Journal of Mechanical Science and Technology* 32 (6): 2497–2507. https://doi.org/10.1007/s12206-018-0507-z.

Arulmozhi, G. 2002. "Exact Equation and an Algorithm for Reliability Evaluation of K-out-of-n: G System." *Reliability Engineering & System Safety* 78 (2): 87–91. https://doi.org/10.1016/S0951-8320(02)00046-7.

Benaichouche, A. N., H. Oulhadj, and P. Siarry. 2013. "Improved Spatial Fuzzy C-Means Clustering for Image Segmentation Using PSO Initialization, Mahalanobis Distance and Post-Segmentation Correction." *Digital Signal Processing* 23 (5): 1390–1400. https://doi.org/10.1016/j.dsp.2013.07.005.

Bora, D. J., D. Gupta, and A. Kumar. 2014. "A Comparative Study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm." arXiv preprint arXiv:1404.6059.

Cassady, C. R., E. A. Pohl, and W. P. Murdock. 2001. "Selective Maintenance Modeling for Industrial Systems." *Journal of Quality in Maintenance Engineering* 7 (2): 104–117. https://doi.org/10.1108/13552510110397412.

Chaabane, K., A. Khatab, C. Diallo, E.-H. Aghezzaf, and U. Venkatadri. 2020. "Integrated Imperfect Multimission Selective Maintenance and Repairpersons Assignment Problem." *Reliability Engineering & System Safety* 199: 106895. https://doi.org/10.1016/j.ress.2020.106895.

Chen, C., N. Lu, B. Jiang, Y. Xing, and Z. H. Zhu. 2021. "Prediction Interval Estimation of Aeroengine Remaining Useful Life Based on Bidirectional Long Short-Term Memory Network." *IEEE Transactions on Instrumentation and Measurement* 70: 1–13.

Chen, C., J. Shi, N. Lu, Z. H. Zhu, and B. Jiang. 2022. "Data-Driven Predictive Maintenance Strategy Considering the Uncertainty in Remaining Useful Life Prediction." *Neurocomputing* 494: 79–88. https://doi.org/10.1016/j.neucom.2022.04.055.

Chen, C., Z. H. Zhu, J. Shi, N. Lu, and B. Jiang. 2021. "Dynamic Predictive Maintenance Scheduling Using Deep Learning Ensemble for System Health Prognostics." *IEEE Sensors Journal* 21 (23): 26878–26891. https://doi.org/10.1109/JSEN.2021.3119553.

Dao, C. D., and M. J. Zuo. 2017. "Optimal Selective Maintenance for Multi-State Systems in Variable Loading Conditions." *Reliability Engineering & System Safety* 166: 171–180. https://doi.org/10.1016/j.ress.2016.11.006.

Dao, C. D., M. J. Zuo, and M. Pandey. 2014. "Selective Maintenance for Multi-State Series–Parallel Systems Under Economic Dependence." *Reliability Engineering & System Safety* 121: 240–249. https://doi.org/10.1016/j.ress.2013.09.003.

Dehghan Shoorkand, H., M. Nourelfath, and A. Hajji. 2023. "A Deep Learning Approach for Integrated Production Planning and Predictive Maintenance." *International Journal of Production Research* 61 (23): 7972–7991. https://doi.org/10.1080/00207543.2022.2162618.

de Pater, I., and M. Mitici. 2022. ""Novel Metrics to Evaluate Probabilistic Remaining Useful Life Prognostics with Applications to Turbofan Engines." *PHM Society European Conference* 7: 96–109.

de Pater, I., A. Reijns, and M. Mitici. 2022. "Alarm-Based Predictive Maintenance Scheduling for Aircraft Engines with Imperfect Remaining Useful Life Prognostics." *Reliability Engineering & System Safety* 221: 108341. https://doi.org/10.1016/j.ress.2022.108341.

Diallo, C., U. Venkatadri, A. Khatab, and Z. Liu. 2018. "Optimal Selective Maintenance Decisions for Large Serial K-out-of-n: G Systems Under Imperfect Maintenance." *Reliability Engineering & System Safety* 175: 234–245. https://doi.org/10.1016/j.ress.2018.03.023.

Diallo, C., U. Venkatadri, A. Khatab, Z. Liu, and E.-H. Aghezzaf. 2019. "Optimal Joint Selective Imperfect Maintenance and Multiple Repairpersons Assignment Strategy for Complex Multicomponent Systems." *International Journal of Production Research* 57 (13): 4098–4117. https://doi.org/10.1080/00207543.2018.1505060.

Duan, C., C. Deng, A. Gharaei, J. Wu, and B. Wang. 2018. "Selective Maintenance Scheduling Under Stochastic Maintenance Quality with Multiple Maintenance Actions." *International Journal of Production Research* 56 (23): 7160–7178. https://doi.org/10.1080/00207543.2018.1436789.

Fu, S., L. Lin, Y. Wang, F. Guo, M. Zhao, B. Zhong, and S. Zhong. 2024. "MCA-DTCN: A Novel Dual-Task Temporal Convolutional Network with Multi-Channel Attention for First Prediction Time Detection and Remaining Useful Life Prediction." *Reliability Engineering & System Safety* 241: 109696. https://doi.org/10.1016/j.ress.2023.109696.

Gal, Y., and Z. Ghahramani. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." In *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48, edited by Maria Florina Balcan and Kilian Weinberger, 1050–1059. New York, NY: PMLR.

Hesabi, H., M. Nourelfath, and A. Hajji. 2022. "A Deep Learning Predictive Model for Selective Maintenance Optimization." *Reliability Engineering & System Safety* 219: 108191. https://doi.org/10.1016/j.ress.2021.108191.

Jardine, A. K., D. Lin, and D. Banjevic. 2006. "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance." *Mechanical Systems and Signal Processing* 20 (7): 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012.

Javed, K., R. Gouriveau, and N. Zerhouni. 2015. "A New Multivariate Approach for Prognostics Based on Extreme Learning Machine and Fuzzy Clustering." *IEEE Transactions on Cybernetics* 45 (12): 2626–2639. https://doi.org/10.1109/TCYB.2014.2378056.

Khatab, A., and E.-H. Aghezzaf. 2016. "Selective Maintenance Optimization when Quality of Imperfect Maintenance Actions are Stochastic." *Reliability Engineering & System Safety* 150: 182–189. https://doi.org/10.1016/j.ress.2016.01.026.

Khatab, A., E. H. Aghezzaf, C. Diallo, and I. Djelloul. 2017. "Selective Maintenance Optimisation for Series-Parallel Systems Alternating Missions and Scheduled Breaks with Stochastic Durations." *International Journal of Production Research* 55 (10): 3008–3024. https://doi.org/10.1080/00207543.2017.1290295.

Khatab, A., E.-H. Aghezzaf, I. Djelloul, and Z. Sari. 2017. "Selective Maintenance Optimization for Systems Operating Missions and Scheduled Breaks with Stochastic Durations." *Journal of Manufacturing Systems* 43: 168–177. https://doi.org/10.1016/j.jmsy.2017.03.005.

Khatab, A., C. Diallo, E.-H. Aghezzaf, and U. Venkatadri. 2018. "Condition-Based Selective Maintenance for Stochastically Degrading Multi-Component Systems Under Periodic Inspection and Imperfect Maintenance." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 232 (4): 447–463.

Khatab, A., C. Diallo, E.-H. Aghezzaf, and U. Venkatadri. 2022. "Optimization of the Integrated Fleet-Level Imperfect Selective Maintenance and Repairpersons Assignment Problem." *Journal of Intelligent Manufacturing* 33: 703–718.

Khatab, A., C. Diallo, U. Venkatadri, Z. Liu, and E.-H. Aghezzaf. 2018. "Optimization of the Joint Selective Maintenance and Repairperson Assignment Problem Under Imperfect Maintenance." *Computers & Industrial Engineering* 125: 413–422. https://doi.org/10.1016/j.cie.2018.09.012.

Kuo, W., and M. J. Zuo. 2003. *Optimal Reliability Modeling: Principles and Applications*. Hoboken, NJ: John Wiley & Sons.

Lee, J., and M. Mitici. 2023. "Deep Reinforcement Learning for Predictive Aircraft Maintenance Using Probabilistic Remaining-Useful-Life Prognostics." *Reliability Engineering & System Safety* 230: 108908. https://doi.org/10.1016/j.ress.2022.108908.

Li, X., Q. Ding, and J.-Q. Sun. 2018. "Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks." *Reliability Engineering & System Safety* 172: 1–11. https://doi.org/10.1016/j.ress.2017.11.021.

Li, J., X. Li, and D. He. 2019. "A Directed Acyclic Graph Network Combined with CNN and LSTM for Remaining Useful Life Prediction." *IEEE Access* 7: 75464–75475. https://doi.org/10.1109/Access.6287639.

Li, H., W. Zhao, Y. Zhang, and E. Zio. 2020. "Remaining Useful Life Prediction Using Multi-Scale Deep Convolutional Neural Network." *Applied Soft Computing* 89: 106113. https://doi.org/10.1016/j.asoc.2020.106113.

Liu, Y., Y. Chen, and T. Jiang. 2018. "On Sequence Planning for Selective Maintenance of Multi-State Systems Under Stochastic Maintenance Durations." *European Journal of Operational Research* 268 (1): 113–127. https://doi.org/10.1016/j.ejor.2017.12.036.

Liu, Y., Y. Chen, and T. Jiang. 2020. "Dynamic Selective Maintenance Optimization for Multi-State Systems Over a Finite Horizon: A Deep Reinforcement Learning Approach." *European Journal of Operational Research* 283 (1): 166–181. https://doi.org/10.1016/j.ejor.2019.10.049.

Liu, Y., and H.-Z. Huang. 2010. "Optimal Selective Maintenance Strategy for Multi-State Systems Under Imperfect Maintenance." *IEEE Transactions on Reliability* 59 (2): 356–367. https://doi.org/10.1109/TR.2010.2046798.

Liu, Y., and X. Qian. 2022. "Selective Maintenance Optimization with Stochastic Break Duration Based on Reinforcement Learning." *Eksploatacja I Niezawodność-Maintenance and Reliability* 24 (4): 771–784.

Maaroufi, G., A. Chelbi, and N. Rezg. 2013. "Optimal Selective Renewal Policy for Systems Subject to Propagated Failures with Global Effect and Failure Isolation Phenomena." *Reliability Engineering & System Safety* 114: 61–70. https://doi.org/10.1016/j.ress.2012.12.019.

Mitici, M., I. de Pater, A. Barros, and Z. Zeng. 2023. "Dynamic Predictive Maintenance for Multiple Components Using Data-Driven Probabilistic RUL Prognostics: The Case of Turbofan Engines." *Reliability Engineering & System Safety* 234: 109199. https://doi.org/10.1016/j.ress.2023.109199.

Namuduri, S., B. N. Narayanan, V. S. P. Davuluru, L. Burton, and S. Bhansali. 2020. "Deep Learning Methods for Sensor Based Predictive Maintenance and Future Perspectives for Electrochemical Sensors." *Journal of The Electrochemical Society* 167 (3): 037552. https://doi.org/10.1149/1945-7111/ab67a8.

Nguyen, K. T., and K. Medjaher. 2019. "A New Dynamic Predictive Maintenance Framework Using Deep Learning for Failure Prognostics." *Reliability Engineering & System Safety* 188: 251–262. https://doi.org/10.1016/j.ress.2019.03.018.

Niu, J., C. Liu, L. Zhang, and Y. Liao. 2019. "Remaining Useful Life Prediction of Machining Tools by 1D-CNN LSTM Network." In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1056–1063. Xiamen: IEEE.

O'Neil, R., C. Diallo, and A. Khatab. 2022. "A Novel Predictive Selective Maintenance Strategy Using Deep Learning and Mathematical Programming." *IFAC-PapersOnLine* 55 (10): 1207–1212. https://doi.org/10.1016/j.ifacol.2022.09.554.

O'Neil, R., C. Diallo, A. Khatab, and E.-H. Aghezzaf. 2023. "A Hybrid Column-Generation and Genetic Algorithm Approach for Solving Large-Scale Multimission Selective Maintenance Problems in Serial K-out-Of-n: G Systems." *International Journal of Production Research* 61 (9): 3071–3087. https://doi.org/10.1080/00207543.2022.2077670.

O'Neil, R., A. Khatab, C. Diallo, and U. Venkatadri. 2023. "Optimal Joint Maintenance and Orienteering Strategy for Complex Mission-Oriented Systems: A Case Study in Offshore Wind Energy." *Computers & Operations Research* 149: 106020. https://doi.org/10.1016/j.cor.2022.106020.

Pandey, M., M. J. Zuo, and R. Moghaddass. 2013. "Selective Maintenance Modeling for a Multistate System with Multistate Components Under Imperfect Maintenance." *IIE Transactions* 45 (11): 1221–1234. https://doi.org/10.1080/0740817X.2012.761371.

Pandey, M., M. J. Zuo, R. Moghaddass, and M. Tiwari. 2013. "Selective Maintenance for Binary Systems Under Imperfect Repair." *Reliability Engineering & System Safety* 113: 42–51. https://doi.org/10.1016/j.ress.2012.12.009.

Peng, C., Y. Chen, Q. Chen, Z. Tang, L. Li, and W. Gui. 2021. "A Remaining Useful Life Prognosis of Turbofan Engine Using Temporal and Spatial Feature Fusion." *Sensors* 21 (2): 418. https://doi.org/10.3390/s21020418.

Peng, K., R. Jiao, J. Dong, and Y. Pi. 2019. "A Deep Belief Network Based Health Indicator Construction and Remaining Useful Life Prediction Using Improved Particle Filter." *Neurocomputing* 361: 19–28. https://doi.org/10.1016/j.neucom.2019.07.075.

Rajagopalan, R., and C. R. Cassady. 2006. "An Improved Selective Maintenance Solution Approach." *Journal of Quality in Maintenance Engineering* 12 (2): 172–185. https://doi.org/10.1108/13552510610667183.

Ren, L., J. Dong, X. Wang, Z. Meng, L. Zhao, and M. J. Deen. 2020. "A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life." *IEEE Transactions on Industrial Informatics* 17 (5): 3478–3487. https://doi.org/10.1109/TII.9424.

Rice, W., C. Cassady, and J. Nachlas. 1998. "Optimal Maintenance Plans Under Limited Maintenance Time." In *Proceedings of the Seventh Industrial Engineering Research Conference*, 1–3. Banff, AB: IIE.

Ruan, D., Y. Wu, and J. Yan. 2021. "Remaining Useful Life Prediction for Aero-Engine Based on LSTM and CNN." In *2021 33rd Chinese Control and Decision Conference (CCDC)*, 6706–6712. Kunming: IEEE.

Saxena, A., K. Goebel, D. Simon, and N. Eklund. 2008. "Damage Propagation Modeling for Aircraft Engine Run-To-Failure Simulation." In *2008 International Conference on Prognostics and Health Management*, 1–9. Denver, CO: IEEE.

Schneider, K., and C. R. Cassady. 2015. "Evaluation and Comparison of Alternative Fleet-Level Selective Maintenance Models." *Reliability Engineering & System Safety* 134: 178–187. https://doi.org/10.1016/j.ress.2014.10.017.

Shoorkand, H. D., M. Nourelfath, and A. Hajji. 2024. "A Hybrid CNN-LSTM Model for Joint Optimization of Production and Imperfect Predictive Maintenance Planning." *Reliability Engineering & System Safety* 241: 109707. https://doi.org/10.1016/j.ress.2023.109707.

Song, T., C. Liu, R. Wu, Y. Jin, and D. Jiang. 2022. "A Hierarchical Scheme for Remaining Useful Life Prediction with Long Short-Term Memory Networks." *Neurocomputing* 487: 22–33. https://doi.org/10.1016/j.neucom.2022.02.032.

Su, C., L. Li, and Z. Wen. 2020. "Remaining Useful Life Prediction Via a Variational Autoencoder and a Time-Window-Based Sequence Neural Network." *Quality and Reliability Engineering International* 36 (5): 1639–1656. https://doi.org/10.1002/qre.v36.5.

Tai, A. H., and L.-Y. Chan. 2010. "Maintenance Models for a Continuously Degrading System." *Computers & Industrial Engineering* 58 (4): 578–583. https://doi.org/10.1016/j.cie.2009.12.005.

Usuga-Cadavid, J. P., S. Lamouri, B. Grabot, and A. Fortin. 2022. "Using Deep Learning to Value Free-Form Text Data for Predictive Maintenance." *International Journal of Production Research* 60 (14): 4548–4575. https://doi.org/10.1080/00207543.2021.1951868.

Wang, C., N. Lu, Y. Cheng, and B. Jiang. 2019. "A Data-Driven Aero-Engine Degradation Prognostic Strategy." *IEEE Transactions on Cybernetics* 51 (3): 1531–1541. https://doi.org/10.1109/TCYB.6221036.

Xia, J., Y. Feng, C. Lu, C. Fei, and X. Xue. 2021. "LSTM-Based Multi-Layer Self-Attention Method for Remaining Useful Life Estimation of Mechanical Systems." *Engineering Failure Analysis* 125: 105385. https://doi.org/10.1016/j.engfailanal.2021.105385.

Xiang, F., Y. Zhang, S. Zhang, Z. Wang, L. Qiu, and J.-H. Choi. 2024. "Bayesian Gated-Transformer Model for Risk-Aware Prediction of Aero-Engine Remaining Useful Life." *Expert Systems with Applications* 238: 121859. https://doi.org/10.1016/j.eswa.2023.121859.

Xu, Q.-Z., L.-M. Guo, H.-P. Shi, and N. Wang. 2016. "Selective Maintenance Problem for Series–Parallel System Under Economic Dependence." *Defence Technology* 12 (5): 388–400. https://doi.org/10.1016/j.dt.2016.04.004.

Yang, B., R. Liu, and E. Zio. 2019. "Remaining Useful Life Prediction Based on a Double-Convolutional Neural Network Architecture." *IEEE Transactions on Industrial Electronics* 66 (12): 9521–9530. https://doi.org/10.1109/TIE.41.

Yu, W., I. Y. Kim, and C. Mechefske. 2019. "Remaining Useful Life Estimation Using a Bidirectional Recurrent Neural Network Based Autoencoder Scheme." *Mechanical Systems and Signal Processing* 129: 764–780. https://doi.org/10.1016/j.ymssp.2019.05.005.

Yu, W., I. Y. Kim, and C. Mechefske. 2020. "An Improved Similarity-Based Prognostic Algorithm for RUL Estimation Using an RNN Autoencoder Scheme." *Reliability Engineering & System Safety* 199: 106926. https://doi.org/10.1016/j.ress.2020.106926.

Zeng, J., and Z. Liang. 2023. "A Dynamic Predictive Maintenance Approach Using Probabilistic Deep Learning for a Fleet of Multi-Component Systems." *Reliability Engineering & System Safety* 238: 109456. https://doi.org/10.1016/j.ress.2023.109456.

Zhang, Q., Q. Liu, and Q. Ye. 2024. "An Attention-Based Temporal Convolutional Network Method for Predicting Remaining Useful Life of Aero-Engine." *Engineering Applications of Artificial Intelligence* 127: 107241. https://doi.org/10.1016/j.engappai.2023.107241.

Zhang, Y., R. Xiong, H. He, and M. G. Pecht. 2018. "Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries." *IEEE Transactions on Vehicular Technology* 67 (7): 5695–5705. https://doi.org/10.1109/TVT.25.

Zhang, L., and J. Zhang. 2020. "A Data-Driven Maintenance Framework Under Imperfect Inspections for Deteriorating Systems Using Multitask Learning-Based Status Prognostics." *IEEE Access* 9: 3616–3629. https://doi.org/10.1109/Access.6287639.

Zhao, R., R. Yan, J. Wang, and K. Mao. 2017. "Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks." *Sensors* 17 (2): 273. https://doi.org/10.3390/s17020273.

Zhou, K.-L., D.-J. Cheng, H.-B. Zhang, Z.-t. Hu, and C.-Y. Zhang. 2023. "Deep Learning-Based Intelligent Multilevel Predictive Maintenance Framework Considering Comprehensive Cost." *Reliability Engineering & System Safety* 237: 109357. https://doi.org/10.1016/j.ress.2023.109357.

Zhuang, L., A. Xu, and X.-L. Wang. 2023. "A Prognostic Driven Predictive Maintenance Framework Based on Bayesian Deep Learning." *Reliability Engineering & System Safety* 234: 109181. https://doi.org/10.1016/j.ress.2023.109181.