Probabilistic deep learning methodology for uncertainty quantification of remaining useful lifetime of multi-component systems

Khanh T.P. Nguyen, Kamal Medjaher, Christian Gogu

Version of Record: https://www.sciencedirect.com/science/article/pii/S0951832022000606
Manuscript_300cce587cb4a3c77dfd7571af0b8328

Probabilistic deep learning methodology for uncertainty quantification of remaining useful lifetime of multi-component systems

Khanh T.P. Nguyen[a,*], Kamal Medjaher[a], Christian Gogu[b]

[a] *Laboratoire Génie de Production, LGP, Université de Toulouse, INP-ENIT, 47 Av. d'Azereix, 65016, Tarbes, France.*

[b] *Institut Clément Ader (ICA), Université de Toulouse, UPS, CNRS, INSA, Mines Albi, ISAE, 3 rue Caroline Aigle, 31400 Toulouse, France*

Abstract

For dealing with uncertainty in Remaining Useful Life (RUL) predictions, numerous studies in literature use stochastic models to characterize the degradation process and predict the RUL distribution. However, in practice, it is difficult to derive stochastic models to capture degradation mechanisms of complex physical systems. Besides, the outstanding achievements in sensing technologies have facilitated the development of data-driven methods. Among them, deep learning methods become one of the most popular trends in recent studies; but they usually provide point predictions without quantifying the output uncertainties. In this paper, we present a new probabilistic deep leaning methodology for uncertainty quantification of multi-component systems' RUL. It is a combination of a probabilistic model and a deep recurrent neural network to predict the components' RUL distributions. Then, using the information about the system's architecture, the formulas to quantify system reliability or system-level-RUL uncertainty are derived. The performance of the proposed methodology is investigated through the benchmark data provided by NASA. The obtained results highlight the point prediction accuracy and the uncertainty management capacity of the proposed methodology. In addition, thanks to the explicit RUL distributions of components, the system reliability for different structures is obtained with high accuracy, especially for series structures.

Keywords: Prognostics, uncertainty management, remaining useful life time, system reliability, LSTM, lognormal distribution, multi-component systems.

1. Introduction

Prognostics and health management (PHM) of complex industrial systems is increasingly a key challenge for guaranteeing system reliability and reducing lifetime operational cost. Accurate predictions of remaining useful life time

* Corresponding author.
Email address: tnguyen@enit.fr (Khanh T.P. Nguyen)

(RUL) of equipment provide valuable information for maintenance organization, thus avoiding systems breakdowns and improving their overall performance. However, as prognostics deals with prediction of future system behavior, several sources of uncertainties exist in RUL predictions. The main types of uncertainty that affect prognostics results are usually: 1) variability of process behavior due to different operating and environmental conditions, 2) inaccuracy of prediction models, 3) measurement noise from sensors, and 4) imperfect information of current system state [1]. Therefore, uncertainty management in prognostics is vital to ensure accurate RUL predictions for industrial systems.

In the literature, model-based prognostic approaches use explicit mathematical models or stochastic processes to characterize the degradation mechanisms, predict their future evolution and estimate the RUL of the system and its uncertainty [2, 3]. However, it is challenging for complex systems to obtain models of their degradation. Alternatively, data-driven methods, which mainly rely on historical monitoring data to learn degradation trends and to discover the system's behavior, allow overcoming this drawback of model-based methods but require sufficient statistical data. Given the large amounts of data that are increasingly available in industry, data-driven approaches currently hold a lot of promise for efficiently exploiting the available data and leading to accurate prognostics models even for complex systems.

Among the large variety of data-driven approaches, deep learning (DL) based methods have attracted a lot of attention in recent years, particularly in prognostics context [4]. However, most of DL based methods only provide point-wise estimates of the RUL while the multiple uncertainties in prognostics makes it difficult to provide absolutely accurate values. Therefore, decision-making based on point-wise estimations can be error-prone, even put the system, its operators, and the environment at risk. Consequently, handling prognostics uncertainty is crucial in RUL prediction. Recent studies on DL based methods attempted to quantify prognostics uncertainty. For illustration, the authors in [5] proposed a hybrid approach, using a long short-term memory (LSTM) network as an expressive black-box predictor and the Wiener process as a surrogate to model the propagation of prediction uncertainty. Besides, combining DL based methods with particle filtering (PF) techniques can provide empirical probability distribution functions (pdf) of RUL. Particularly, PF uses a set of weighted particles to represent uncertainty propagation in time, based on an underlying deterioration model, which could be a DL model [6]. Given noisy and/or partial observations, the particles' weights are updated using Bayes rule to sequentially approximate the posterior pdf of system states [7]. In [8], the authors proposed a hybrid approach, combining a deep belief network, which is used to extract hidden degradation features from monitoring data, with a particle filter approach for quantifying the RUL uncertainty. The above studies allow addressing aleatoric uncertainty due to measurement errors or operating conditions variability but require a statistically sufficient number of particles to obtain a reasonable empirical RUL distribution.

In addition to aleatoric uncertainty, epistemic uncertainty caused by model capacity was addressed by re-sampling ensemble methods. For instance, Vishnu *et al.* [9] used the ensembling technique with deep ordinal regression

models to calculate the predictive uncertainty. Echo state Gaussian processes were used in [10], while Liao *et al.* [11] obtained the prediction uncertainty based on a classical bootstrapping approach with a long short term memory network (LSTM). The uncertainty treatment capability of bootstrapped ensemble techniques have been compared to a particle filtering method in [12]. Using the re-sampling ensemble method, the entire training and prediction process must be iterated many times to obtain enough samples for the construction of confidence intervals. This requires large computational resources and thus it is not suitable for real-time applications and high-dimensional datasets.

Besides, various Bayesian deep learning (BDL) frameworks were proposed in [13, 14] to determine the epistemic uncertainty in the RUL prediction. Recently, Biggio *et al.* [15] compared various BDL formulations for uncertainty aware RUL predictors. In BDL frameworks, network's parameters are treated as random variables presented by a probability distribution. Their prior distributions are updated with Bayesian inference given the observed data. Although BDL is robust for epistemic uncertainty quantification, its computational cost is heavier, including the training phase, i.e. updating the posterior distribution of all parameters, as well as the inference phase, i.e. sampling sufficient observations of the parameters from their posterior distribution to get the proper output distribution.

Unlike the previous mentioned studies, an explicit predicted RUL distribution at component level is directly obtained through recurrent neural networks and convolutional neural networks in papers [16, 17]. This can open a promising perspective to facilitate the uncertainty inference when evaluating RUL distribution at system level. However, the methods presented in [16, 17] are based on the assumption that the equipment's RUL follows a Gaussian distribution which can get negative values while RUL is always positive. In addition, the works presented in [16, 17] only consider the uncertainty of RUL prediction at component level while the study of prognostic uncertainty inference and propagation at system level is still a relatively unexplored area.

Considering studies on system reliability, there are many publications that investigate the uncertainty of system reliability taking into account the dependence among different components [18, 19]. However, based on a comprehensive review on prognostic approaches [20], we find that the prognostics has often been approached from a component view without considering interactions between components. A few rare studies, which consider the dependencies between components when estimating system RUL, are based on the assumption about probabilistic and stochastic models of the component degradation and lack the evidence to show that these assumptions are indeed consistent with real monitoring data [21]. Hence, the practical application of these methods in industry is still limited. Besides, the analysis and uncertainty quantification of system RUL as well as system reliability, based on data-driven methods, is still an unexplored research area.

In summary, the above survey of prognostics uncertainty management studies in the literature allows us to identify the following two challenges that we will seek to address:

1. most existing probabilistic deep learning based prognostics approaches require high computational resources

and are not suitable for real-time applications. Furthermore, they only provide the component's empirical RUL distribution, that is difficult or inefficient to propagate for quantifying RUL of multi-component systems.

2. most existing data-driven approaches only consider the prognostics of individual components but not that of the entire system. Yet, predicting the SRUL and its uncertainty is also essential for developing the most appropriate maintenance strategy.

Accordingly, this paper aims to simultaneously address these two challenges by developing a new methodology for quantifying the prognostics uncertainty of multi-independent-component systems. In terms of the first challenge, we seek to develop an efficient approach able to provide the pdf of the RUL and outperform existing methods for both point-wise and probabilistic RUL predictions with a reasonable computational cost. The proposed approach is a combination between a probabilistic model, i.e. lognormal distribution, and a recurrent neural network, i.e. LSTM model, to directly predict component's RUL distribution from condition monitoring data. The flexibility of the lognormal distribution allows adapting different characteristics of modeling data that can be roughly symmetric or skewed to the right, while satisfying the physical constraint of RUL positivity. In terms of the second challenge, we derive analytical formulations to evaluate the system reliability and SRUL probability according to its structure. To our humble knowledge, this is the first data-driven model that allows handling uncertainty propagation of RUL predictions from the component level to the system level, taking into account different system architectures. Our proposed methodology could be seen as a bridge between two fields of research: traditional system reliability and data-driven prognostics. Although the proposed methodology can be still further improved (e.g., regarding the assumption of independence between components), we believe that it is an essential premise for the development of further studies in the fields of system reliability as well as prognostics and health management.

The rest of the paper is organized as follows. Section 2 presents the proposed methodology. First the proposed model, called Lognorm-LSTM, for predicting the component's RUL distribution is described, followed by the models for calculating the system RUL and its uncertainty. Section 3 presents numerical experiments on a NASA turbofan engine degradation dataset. On a component level, both the point-wise estimates and the RUL distribution predictions are benchmarked against existing approaches using multiple evaluation metrics. On a systems level, the system's RUL and its uncertainty are estimated and their quality are assessed using two different evaluation metrics. Finally, section 4 provides concluding remarks and outlines further possible works.

2. Proposed methodology

Figure 1 presents an overview of the proposed methodology that consists of online and offline phases. The offline phase is dedicated to the training of a Lognorm-LSTM model, using run-to-failure data, allowing to map a RUL distribution to corresponding historical condition monitoring (CM) data. Without loss of generality, let us assume N components of the same type are monitored during their operation by m sensors for each component. Then,

the monitoring data acquired for each component i, $i = 1, 2, ..., N$, during its life time T_i can be expressed in a matrix form: $X_i = [x^1, x^2, ..., x^{T_i}]$, $X_i \in R^{m \times T_i}$, where $x^t = [x_1^t, x_2^t, \ldots, x_m^t]$ is a vector of sensor measurements at time t. During the training stage, the proposed Lognorm-LSTM model takes the sensor measurement sequences X_i, $i = 1, 2, ..., N$, to learn the parameter vector Θ_i, where $\Theta_i = [\theta_i^1, \theta_i^2, ..., \theta_i^{T_i}]$ characterizes the RUL distribution of component i during its life time T_i. If the RUL distribution follows a lognormal distribution, then the vector of parameters θ_i^t, characterizing RUL distribution of component i at time t, includes 2 values of (μ_i^t, σ_i^t). Note that μ_i^t and σ_i^t are the mean value and the standard deviation of the natural logarithm of the RUL at time t of component i. For an efficient training, the CM data require an appropriate preprocessing step, while the RUL labels need to be verified and rectified. The details of this step are described in subsection 2.1. The construction of the proposed Lognorm-LSTM model for prediction of component's RUL distribution is then presented in subsection 2.2. Next, during the online phase, at time t, the trained Lognorm-LSTM model will take N vectors of sensor measurements x^t as input and then output N vectors of Θ_i. These results serve to infer the system reliability and SRUL probability according to different system structures. The details of the latter step are described in subsection 2.3.

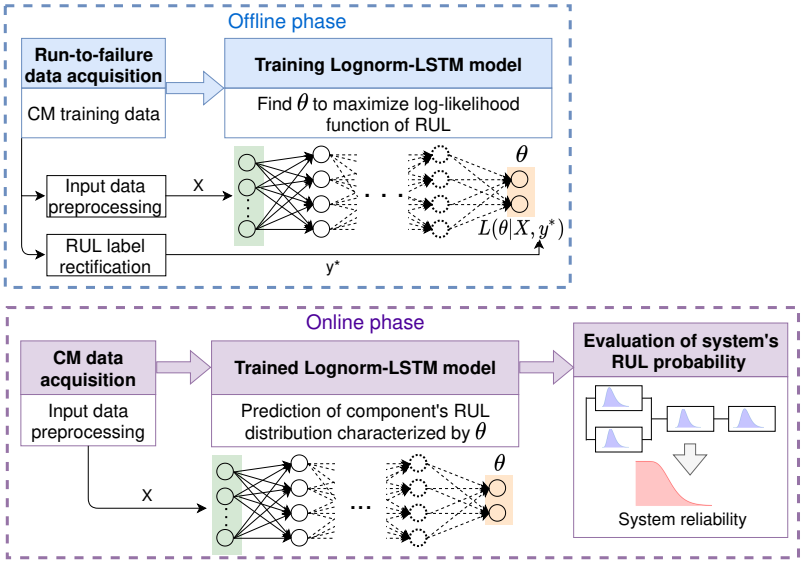


Figure 1: Overview of the proposed methodology

2.1. Data preprocessing

Feature selection (FS): It aims to identify a subset of features which allows identifying significant effects from irrelevant ones, and therefore providing good prediction results. Based on the evaluation criteria, the FS techniques can be classified into three groups: (1) filter methods, (2) wrapper methods, and (3) embedded methods [22]. The filter approaches are suitable for unlabeled data or when there is no correlation between features and labeled data. Otherwise, the embedded approaches are the best candidates for the cases that require high accuracy and non-

expensive computation. However, these approaches are not suitable for high dimensional data case which can be overcome by using wrapper methods based on heuristic search algorithms. In a recent study [23], the authors proposed an automated health indicator construction methodology that allows automatically choosing the relevant measurements among various sources and also handling raw data from high-frequency sensors to extract the useful low-level features. The results obtained by the feature engineering phase of that work are inherited and exploited in this manuscript.

Normalization: It serves to transform the data within a specific scale to enhance its consistency. Among the most popular normalization techniques, the Z-transform technique is chosen in this paper. Given x_k^l, where k is k-th observation of the l-th sensor, μ^l and σ^l are the mean and variance of all observation values from the l-th sensor, then the normalized k-th observation value from the l-th sensor is given by:

$$x_{k_{norm}}^l = \frac{x_k^l - \mu^l}{\sigma^l} \tag{1}$$

Right padding: In this paper, a component's life cycle is considered as a sample. As individual components have different life cycles, samples lengths are not equal. However, the input data for a LSTM model must be a 3D tensor having a fixed dimension (*samples (n_s), time steps (n_t), and features (m)*). Hence, it is necessary to pad samples that are shorter than the longest item with some placeholder values. In this paper, we perform right padding by adding zeros value at the end of shorter sequences.

RUL label rectification: RUL labels in the training set have a significant impact on the model performance. Without rectification, the system RUL is represented by a linear decreasing function over time: it equals to the maximum life time at the beginning and decreases linearly to reach zero at the end. However, in practice, when the system is in a healthy state at the early operating stage, RUL is usually considered as a constant value. After the first anomaly signs, the system's health state gradually deteriorates and its RUL decreases overtime. To better simulate this RUL change process, the piecewise linear function is widely used in prognostics studies [17, 24].

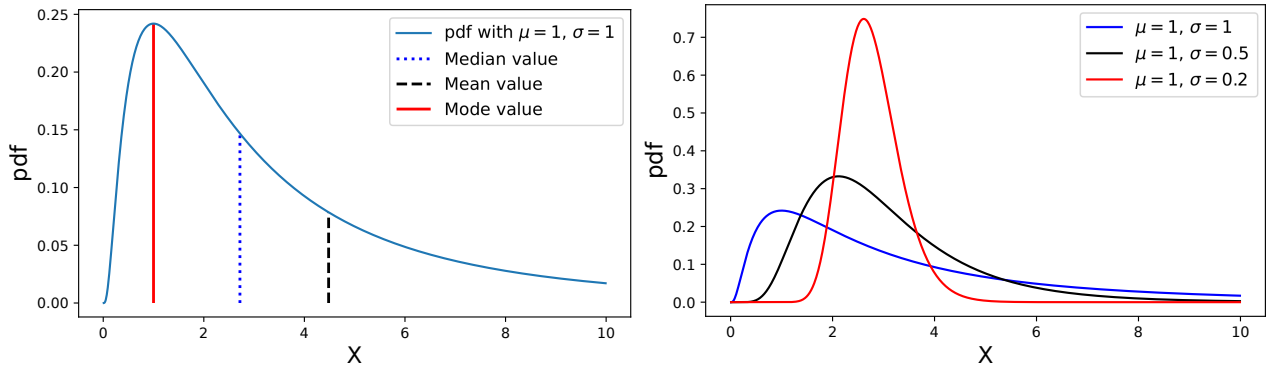
2.2. Prediction of component's RUL distribution

In this subsection, we introduce the Lognorm-LSTM model to predict the lognormal distribution parameters (μ_i^t, σ_i^t) that characterize the RUL distribution at time t of component i. Particularly, instead of predicting a target RUL value, y^*, the Lognorm-LSTM will provide a couple of parameters (μ_i^t, σ_i^t) that maximizes the probability when $RUL_i^t = y^*$. To do this, we firstly recall the properties of lognormal distribution, and then describe how to construct the proposed Lognorm-LSTM model.

2.2.1. *Properties of lognormal distribution*

In this manuscript, the lognormal distribution is chosen to manage the uncertainty in component's RUL prediction for the following reasons:

1. It is one of the most common continuous probability distributions used to model the lifetimes of units;
2. It can only take positive values, which is consistent with the physical constraint that RUL can only be postivie;
3. It is based on the multiplicative growth model and is therefore suitable for diverse components that fail primarily due to fatigue-stress nature, e.g., semiconductor failure or time to fracture in metallic structures;
4. It is a flexible distribution, particularly useful for modeling data that are roughly symmetric or skewed to the right. Depending on its scale parameter, this distribution can have different shapes as illustrated in Figure 2.



(a) Statistical values of lognormal distribution

(b) Different shapes of lognormal distribution

Figure 2: Illustration for lognormal distribution

Given a positive random variable X following a lognormal distribution with two parameters μ and σ, its statistical characteristics are evaluated by the formulas in Table 1.

Table 1: Statistical characteristics of lognormal distribution

Probability distribution function	Mean value	Median value	Mode value
$\frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right)$	$\exp\left(\mu + \frac{\sigma^2}{2}\right)$	$\exp(\mu)$	$\exp\left(\mu - \sigma^2\right)$

2.2.2. *Lognorm-LSTM model*

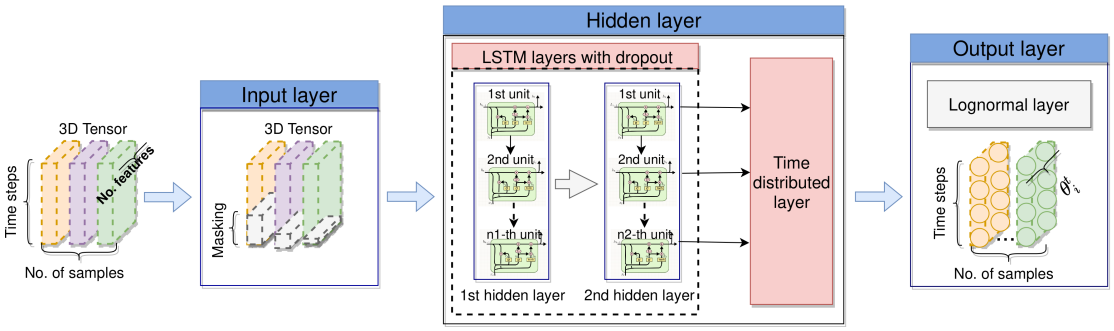


Figure 3: Architecture of Lognorm-LSTM model for prediction of components' RUL distributions.

Recall that θ_i^t is the output vector characterizing RUL distribution of component i at time t, including 2 values of lorgnormal distribution's parameters, μ_i^t and σ_i^t.

Figure 3 presents the architecture of the Lognorm-LSTM model for the prediction of the component's RUL distribution. It consists of the following layers:

1. *Input layer:* It serves as a prototype bringing the data, which are formalized as a 3D tensor with the shape *number of samples* (n_s), *time steps* (n_t), *and number of features* (m) into the network for further processing. To avoid bias errors caused by the input data's right padding part, a **masking layer** is used for skipping this informal part when training the model.
2. *Hidden layer:* It is the principal part of the network, seeking to construct the relationship between the input and the output. In this paper, two LSTM layers and one time distributed layer are sequentially stacked into the hidden layer.

   LSTM layer is a particular architecture of recurrent neural networks (RNN), firstly proposed in [25] to solve the vanishing gradient problem of RNN when learning long-term dependencies. Each LSTM unit has three gates (forget, input and output gates) which provide them the ability to selectively learn, retain important information or throw away unnecessary one, see Figure 4. The LSTM core is the cell state, considered as "memory" of the network, that transfers relative information throughout the sequence's processing. An LSTM network is well suited to time series problems but it can easily over-fit training data. Therefore, the "Dropout" regularization technique is added to every LSTM layer for improving the model's performance [26]. It involves randomly removing some hidden units in a neural network during training by a defined probability. Therefore, each hidden unit in the network trained with a dropout can adapt to random cooperations and therefore becomes more robust. This should drive the network towards more generalization and avoid the over-fitting issue.

   Time distributed layer is a wrapper layer used to design a many-to-many LSTM for the prediction of the sequence output. It applies a same fully-connected operation (i.e. dense operation with same weights) to every time step of the LSTM outputs and generates an output vector per time step input. The dimension

of this output vector depends on the number of RUL distribution's parameters in the output layer. For example, assuming that the RUL follows a lognormal distribution. Then, the time distributed layer will be defined by a dense operation with 2 units (representing the 2 parameters of a lognormal distribution, μ and σ). If this time distributed layer gets data of shape $(None, n_t, n_2)$ from the output of the last LSTM layer, then the time distributed layer's output takes the shape of $(None, n_t, 2)$. Note that *None* corresponds to samples/batch size while n_t and n_2 are respectively time steps and number of the last LSTM hidden unit.

3. *Output layer:* A new layer, namely **lognormal layer**, is defined in this manuscript to take into account particular characteristics of RUL distribution parameters when training the model. It serves as a prototype between the network and the output to provide the proper parameters representing the RUL distribution instead of a point-wise RUL prediction. Particularly, it includes specific activation functions that are suitable to the characteristics of the two lognormal parameters. For illustration, an efficient management of the prognostic uncertainty requires a small enough positive value of standard deviation σ, while a normal dense layer cannot satisfy this requirement. Hence, it is necessary to define a specific function in the lognormal layer to capture these characteristics of σ as follows:

$$\sigma = \begin{cases} e^x, & \text{if } x \leq 0 \\ x + 1, & \text{if } 0 < x < b \\ b + 1, & \text{if } x > b \end{cases} \tag{2}$$

where x is one of two values in the time distributed layer output vector. The result of this operation is illustrated in Figure 5.

Eq.(2) is used as a bounded constraint function for the standard deviation value. It is inspired from the idea of ELU (Exponential Linear Unit) activation function, and is defined as:

$$ELU(x) = \begin{cases} e^x - 1, & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

As the standard deviation, σ, is not negative, Eq.(2) added one to the ELU function, $ELU(x) + 1$, to ensure positive values are always outputs. Inheriting the advantage of ELU function, Eq.(2) smoothly tends towards zero when $x < 0$. In addition, we used an extra constant b to overcome one of the drawbacks of the ELU function, that is for $x > 0$, it can blow up the activation with the output range of $[0, \infty]$. Particularly, b is the bounded constraint value of σ, which is expected to be small enough.

For μ value, we directly use the remaining value in the output vector of the time distributed layer. However, it is easy to limit the range of μ value, if necessary, by defining an appropriate piece-wise linear function. For example, given RUL_{max} be the RUL maximum value, a bounded value of μ can be approximately estimated as: $\mu_{max} = \ln(RUL_{max})$.

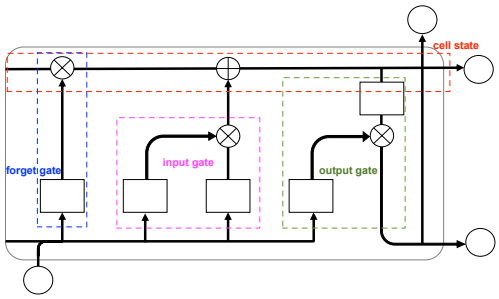


Figure 4: Architecture of LSTM unit

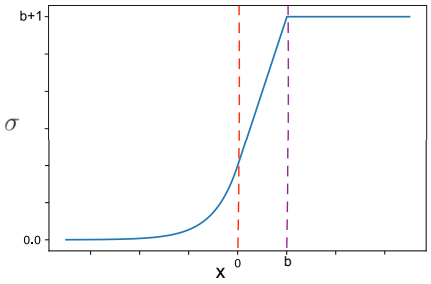


Figure 5: Illustration of the result given by Eq.2

Definition of loss function. The Lognorm-LSTM model has now been constructed to predict sets of parameters (μ_i^t, σ_i^t), where (μ_i^t, σ_i^t) describe the RUL probability distribution for component i at time t. Note that $i \in \{1, 2, ..., n_s\}$ and $t \in \{1, 2, ..., n_t\}$. When training, the weights and bias of the Lognorm-LSTM model will be tuned to get the sets of (μ_i^t, σ_i^t) values that maximize the likelihood function, i.e. probability across the values of the RUL of component i from the beginning to the time until which the observations from the component i were recorded, given as follows:

$$\prod_{i=1}^{n_s}\prod_{t=1}^{n_t} L(\mu_i^t, \sigma_i^t | RUL_i^{*(0:t)}) = \prod_{i=1}^{n_s}\prod_{t=1}^{n_t}\prod_{j=0}^{t} \frac{1}{RUL_i^{*j} \cdot \sigma_i^t \sqrt{2\pi}} \exp\left(-\frac{(\ln RUL_i^{*j} - \mu_i^t)^2}{2(\sigma_i^t)^2}\right) \tag{3}$$

Note that $RUL_i^{*(0:t)} = [RUL_i^{*0}, RUL_i^{*1}, ..., RUL_i^{*t}]$.

However, at the beginning of the training process when the parameters are far from their optimal values, the likelihood function is relatively unchanging, and the updates to the network will be small. Hence, the logarithm likelihood function is used to accelerate the model converge rate to the optimal values. In fact, after taking the logarithm, the rate of change is small nearby the optimal value and large far away from it. Besides, the logarithm of the likelihood function gives a sum of individual and statistically independent observation factors, which facilitates the derivative operation to find the optimal values. Next, as a traditional training process usually minimizes a loss function (rather than maximizing an objective function), so we define here the **loss function** as the negative logarithm likelihood function, given by the following equation:

$$NLL = \sum_{i=1}^{n_s}\sum_{t=1}^{n_t} -\log L(\mu_i^t, \sigma_i^t | RUL_i^{*(0:t)}) \tag{4}$$

Finally, the adaptive moment estimation algorithm (ADAM) is used for the optimization of the Lognorm-LSTM's weights. Note that the initial values of network weights are randomly drawn from a uniform distribution within $[-l, l]$, where $l = \sqrt{6/(n_{in} + n_{out})}$ with n_{in} and n_{out} being the number of input units and the number of output units, respectively.

2.3. Analysis and uncertainty quantification of system RUL

The component's RUL distribution obtained in Section 2.2, i.e. the proposed LogNorm-LSTM's output, can be used to evaluate the reliability and the remaining useful life of systems of multi-independent-components under different structures. In practice, there are diversities of structures. However, in the scope of this paper, we only consider some common structures, such as series, parallel, combinations between them, and bridge-type (see Figure 6 for illustration). Furthermore, we discuss the applicability of the proposed methodology for large complex systems.

2.3.1. Ground truth SRUL analysis

Without loss of generality, we assume that at time t_p, the system of N survival independent components still works. The ground truth RUL of these components are denoted by RUL_i, with $i \in \{1, 2, ..., N\}$. Then, the ground truth SRUL is evaluated according to the system structure as follows,

1. Series system (see the first image of Figure 6 for illustration):

$$SRUL = \min(RUL_i); \quad i \in \{1, 2, ..., N\} \tag{5}$$

2. Parallel system (see the last image of Figure 6 for illustration):

$$SRUL = \max(RUL_i); \quad i \in \{1, 2, ..., N\} \tag{6}$$

Let us assume that the components' RULs are given as illustrated in Figure 7, the SRUL in series and parallel cases are equal to the RUL of component 3, RUL_3, and the RUL of component 2, RUL_2, respectively.

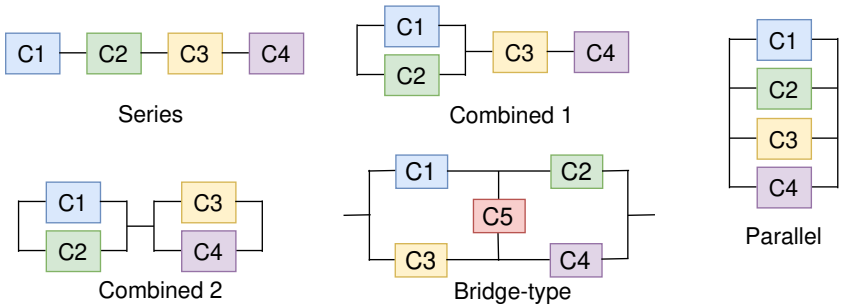


Figure 6: Illustration of SRUL evaluation for multi-composant systems with different structures

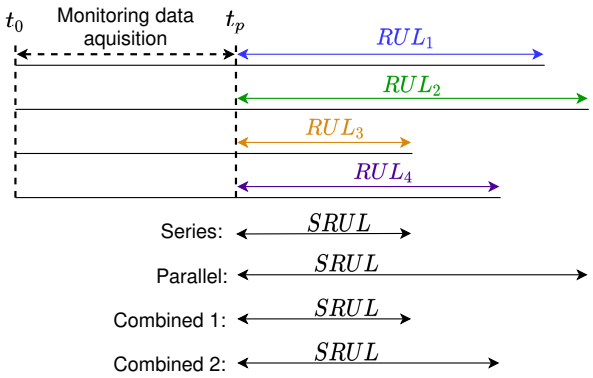


Figure 7: Illustration for multi-composant systems with different structures

In the case of a combined system with simple topology, we firstly decompose its entire structure by combinations of serial or parallel blocks of elementary components. Then, the RUL of each block is evaluated according to the equation corresponding to its architecture (Eq.5 for serial blocks and Eq.6 for parallel blocks). Finally, the SRUL is evaluated according to the connexion type between these blocks. For example, the combined system (combined 1) represented by the second image of Figure 6, is composed of a parallel block (of component 1 and 2) connected in series with components 3 and 4. Then, its SRUL is given by:

$$SRUL = \min(\max(RUL_1, RUL_2), RUL_3, RUL_4) \tag{7}$$

Given the components RULs as shown in Figure 7, the SRUL in this case is equal to the RUL of component 3, RUL_3.

Besides, the SRUL of the combined-2 structure is evaluated by:

$$SRUL = \min(\max(RUL_1, RUL_2), \max(RUL_3, RUL_4)) \tag{8}$$

With the components RULs assumed as shown in Figure 7, the SRUL in this case is equal to the RUL of component 4, RUL_4.

However, the above method is only suitable for systems whose structure is presented by simple topology networks (series-parallel ones). Systems with complex topology can be handled by other decomposition methods, such as minimal cut sets. Particularly, through the analysis of the system's behavior, we identify the minimal cut sets such that the system will fail if at least a single cut set is present [27]. The details of minimal-cut-set analysis method can be consulted in [28]. For illustration, considering the bride-type structure shown in Figure 6, the minimal cut sets are: ($\overline{C1}$, $\overline{C3}$), ($\overline{C2}$, $\overline{C4}$), ($\overline{C1}$, $\overline{C5}$, $\overline{C4}$), ($\overline{C3}$, $\overline{C5}$, $\overline{C2}$). Then, its SRUL is given by:

$$SRUL = \min(\max(RUL_1, RUL_3), \max(RUL_2, RUL_4), \max(RUL_1, RUL_4, RUL_5), \max(RUL_2, RUL_3, RUL_5)) \tag{9}$$

2.3.2. Uncertainty quantification for the predicted SRUL

As the system of N survival independent components still works at time t_p (Figure 7), the monitoring data of component i, with $i \in \{1, 2, ..., N\}$, from the initial moment t_0 until t_p are used as the input of the proposed Lognorm-LSTM model to predict its RUL distributions. The Lognorm-LSTM outputs a set of parameters $(\mu_i^{t_p}, \sigma_i^{t_p})$ characterizing the RUL distribution of component i from the prediction time t_p. To simplify our analysis, without loss of generality, let's set t_p be the current time origin, the reliability of component i at time t (from the time origin t_p) is given by:

$$R_i(t) = P(RUL_i > t) = 1 - P(RUL_i \leq t) = 1 - CDF(t, \mu_i^{t_p}, \sigma_i^{t_p}) \tag{10}$$

where $CDF(t, \mu_i^{t_p}, \sigma_i^{t_p})$ is the cumulative log-normal distribution function at t given two estimated parameters $(\mu_i^{t_p}, \sigma_i^{t_p})$ characterizing the predicted RUL distribution of component i from the origin time t_p. We have:

$$CDF(t, \mu_i^{t_p}, \sigma_i^{t_p}) = \Phi\left(\frac{\ln t - \mu_i^{t_p}}{\sigma_i^{t_p}}\right) \tag{11}$$

where Φ is the cumulative distribution function of the standard normal distribution.

Series systems

After predicting the RUL distributions and deriving the reliabilities of N components using their monitoring data, the reliability of series system (which consists of N independent components) at time t is given by:

$$R_s(t) = \prod_{i=1}^{N} R_i(t) = \prod_{i=1}^{N} \left(1 - CDF(t, \mu_i^{t_p}, \sigma_i^{t_p})\right) \tag{12}$$

Parallel systems

The reliability of parallel system (which consists of N independent components) at time t is given by:

$$R_s(t) = 1 - \prod_{i=1}^{N} CDF(t, \mu_i^{t_p}, \sigma_i^{t_p}) \tag{13}$$

Combined architecture systems

To evaluate the reliability of combined systems, their structure must be decomposed into serial or parallel connections of blocks of elementary components. Next, the reliability of each elementary component's block is evaluated by the equation corresponding to its architecture (Eq.12 for serial blocks and Eq.13 for parallel blocks). Finally, the reliability is evaluated according to the connection type between these blocks. For example, considering the combined system (combined-1) shown in Figure 6, it includes a parallel block (of component 1 and 2) connected in series with components 3 and 4. Then, its reliability is given by:

$$\begin{aligned} R_s(t) &= \left(1 - \prod_{i=1}^{2} R_i(t)\right) \prod_{j=3}^{4} R_j(t) \\ &= \left(1 - \prod_{i=1}^{2} CDF(t, \mu_i^{t_p}, \sigma_i^{t_p})\right) \prod_{j=3}^{4} \left(1 - CDF(t, \mu_j^{t_p}, \sigma_j^{t_p})\right) \end{aligned} \tag{14}$$

Similarly, the reliability of the system with structure combined-2, see Figure 6, is given by:

$$\begin{aligned} R_s(t) &= \left(1-\prod_{i=1}^{2} R_i(t)\right)\left(1-\prod_{j=3}^{4} R_i(t)\right) \\ &= \left(1-\prod_{i=1}^{2} CDF(t,\mu_i^{t_p},\sigma_i^{t_p})\right)\left(1-\prod_{j=3}^{4} CDF(t,\mu_j^{t_p},\sigma_j^{t_p})\right) \end{aligned} \tag{15}$$

Complex topology systems

The reliability of systems with complex topology can be evaluated through the union of all minimal cut sets in the reliability network [27]. For illustration, considering the bridge-type structure, shown in Figure 6, the system reliability is given by: $R_s(t) = 1 - P(F)$, where $P(F)$ is the probability of system failure.

$$\begin{aligned} P(F) &= \overline{C1}\cdot\overline{C3} + \overline{C2}\cdot\overline{C4} + \overline{C1}\cdot\overline{C5}\cdot\overline{C4} + \overline{C3}\cdot\overline{C5}\cdot\overline{C2} - \overline{C1}\cdot\overline{C3}\cdot\overline{C2}\cdot\overline{C4} - \overline{C1}\cdot\overline{C3}\cdot\overline{C5}\cdot\overline{C4} \\ &\quad -\overline{C1}\cdot\overline{C3}\cdot\overline{C5}\cdot\overline{C2} - \overline{C1}\cdot\overline{C2}\cdot\overline{C5}\cdot\overline{C4} - \overline{C2}\cdot\overline{C3}\cdot\overline{C5}\cdot\overline{C4} \\ &\quad +2\overline{C1}\cdot\overline{C2}\cdot\overline{C3}\cdot\overline{C4}\cdot\overline{C5} \end{aligned} \tag{16}$$

where $\overline{Ci}$ is the failure probability of component i, $i \in \{1, 2, ..., 5\}$ given by $CDF(t, \mu_i^{t_p}, \sigma_i^{t_p})$.

However, the number of minimal cut sets exponentially increases with the size of the system. This combinatorial explosion could lead to the numerous difficulties in manipulation of a huge number of cut sets as well as storage problems. Hence, an efficient algorithm based on adjacency arrays could be deployed for determining the reliability of complex systems including a large number of components [27].

Probability of the predicted SRUL

After evaluating the system reliability at time t_1 and t_2 according to their structures, the probability that the SRUL falls within the interval $[t_1, t_2]$ can be easily derived as follows:

$$\begin{aligned} P(t_1 \leq SRUL \leq t_2) &= P(SRUL \leq t_2) - P(SRUL \leq t_1) \\ &= (1 - R_s(t_2)) - (1 - R_s(t_1)) = R_s(t_1) - R_s(t_2) \end{aligned} \tag{17}$$

Discussion on the practical applicability of the proposed methodology

The proposed methodology not only provides the pdf of the RUL of components but also offers other prognostics information that can meet different requirements of operation and maintenance planners. For instance, one can cite its obtained useful results as follows:

1. point-wise prediction of components' RUL as well as system's RUL (RUL_i and $SRUL$) by using the mode, mean or median value of the estimated RUL distribution and Eqs. 5-9. This information is widely used for plan-

ing maintenance activities at tactical level, e.g integrating maintenance actions into the problem of production scheduling [29].

2. component reliability ($R_i(t)$) by calculating the integral of components' RUL pdfs as well as system reliability ($R_s(t)$) by using Eqs. 12-16. This information is needed to optimize maintenance activities for high-reliability systems such as nuclear power plants and transportation units [30].
3. probability that the system will fail into different time intervals ($P(t_1 \leq SRUL \leq t_2)$). These time intervals can be defined according to the requirements of the operation planner for better preparing and performing maintenance activities, e.g. joint optimization of maintenance and inventory management [31].

In addition, the proposed methodology could be extended as a generic framework for handling the uncertainties in RUL prediction by integrating the relevant RUL's probability distribution according to different characteristics of degradation mechanisms. For illustration, the exponential distribution can be used to model the RUL distribution of a component, whose failure rate is constant over time, while the Weibull distribution can adapt a variety of life behaviors (increasing, decreasing or constant failure rate). To integrate the exponential distribution in our proposed methodology, the output layer should be redesigned to generate only one parameter (the failure rate, λ), such that $\lambda > 0$ and $1/\lambda$ is bounded by the RUL_{min} and RUL_{max} of the component. For the Weibull distribution, the output layer must provide 3 parameters: the shape parameter (β), the scale parameter (η) and the location parameter (γ), such that: $\gamma, \eta > 0, 0 < \beta < 1$ for decreasing failure rates, $\beta = 1$ for constant failure rates, and $\beta > 1$ for increasing failure rates. Accordingly, from a technical aspect, it is not difficult to integrate the exponential distribution in the proposed methodology but its practicability is limited to components whose failure rate is independent of its working time. Furthermore, the Weibull distribution can be widely used for most situations but from a technical aspect, it is not easy to take into account the mutual impact of three parameters when training the model (e.g., to reduce the RUL prediction's uncertainty, the value of η should be decreasing while keeping β and γ constant, but this will push the RUL pdf in towards the left, hence accelerate the failure probability in early time). To trade-off the practicability and the technical difficulty, this work has considered the lognormal distribution. However, the proposed methodology is only suitable for systems of independent components whose states can be monitored through sensing technologies.

3. Numerical experiments

This section aims to evaluate the proposed methodology's performance on a well known engeineering benchmark problem. For this purpose, we firstly describe the considered benchmarking dataset in subsection 3.1. Then, the performance evaluation metrics for point-wise prediction and uncertainty quantification are proposed in subsection 3.2. Subsection 3.3 is dedicated to discussing the prognostics results for individual components. Finally, uncertainty management for prognostics of systems of multi-independent-components is investigated in subsection 3.4.

3.1. Case study description

In this section, the benchmarking data set, Turbofan Engine Degradation Simulation provided by NASA Ames Prognostics Data Repository, is used to evaluate the proposed methodology's performance. Among 4 subsets: FD001, FD002, FD003 and FD004, the first case FD001 is the most used in PHM field [1, 11, 16, 32, 33]. It is generated by the C-MAPSS tool that simulates various degradation scenarios of a fleet of engines of the same type, subject to a single fault mode under same operating conditions. The details of this dataset can be consulted in paper [34].

Among 21 sensor measurements, some sensors provides useless information that are constant throughout the engine's life time. The use of these measurements can reduce the efficiency of the prediction model. Therefore, it is necessary to correctly chose the relevant measurements according to the feature performance criteria [23]. As a result, 14 features corresponding to the outputs of 14 sensors with indexes 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21, and one feature corresponding to the life cycle are selected. The selected features are then normalized, padded and masked as described in section 2.1. For RUL label rectification, a piece-wise linear function with the maximum RUL value of 130 is used [24].

Next, the Lognorm-LSTM model presented in subsection 2.2 is constructed using the Keras and TensorFlow probability libraries of Python. Its architecture is summarized in Table 2. The parameters' values presented in Table 2 are manually optimized through an analysis of multiple numerical experiments feedback. Although the maximum number of epochs is set to 500, an early stopping mechanism is implemented to end the training process if the loss function is not improved after 50 consecutive epochs. During this training process, we use *ModelCheckPoint* to monitor the loss function and then output the model weights each time an improvement is recognized.

Table 2: Configuration parameters of the proposed Lognorm-LSTM

1^{st} LSTM units	2^{nd} LSTM units	Dropout	b	Learning rate	Epochs
100	50	0.2	0.5	0.001	500

3.2. Performance evaluation metrics

This section aims to presents the metrics for performance evaluation of the proposed Lognorm-LSTM on both aspects: point-wise prediction and uncertainty management.

3.2.1. Point prediction accuracy metrics

Given M the total number of prediction points and d_k be the difference between the k-th estimated and actual RUL values, the performance of prognostic models can be evaluated by the following point prediction accuracy metrics.

Mean squared error (MSE): MSE is a widely used metric to evaluate the point prediction accuracy [32, 35, 36, 37]. It equally penalizes both late and early predictions by the following formulation:

$$MSE = \frac{1}{M}\sum_{k=1}^{M} d_k^2 \tag{18}$$

Scoring function (SF): SF, which is a popular metric used to evaluate the performance of prognostics algorithms on the C-MAPSS dataset [1, 32, 33, 35, 36], allows punishing late prediction more heavily than an early prediction.

$$SF = \sum_{k=1}^{M} s_k; \qquad s_k = \begin{cases} e^{-\frac{d_k}{13}-1}, & \text{if } d_k < 0 \\ e^{-\frac{d_k}{10}-1}, & \text{if } d_k \geq 0 \end{cases} \tag{19}$$

Accuracy (A): Accuracy is a metric to measure the percentage of prediction errors that fall within a tolerance interval. For the C-MAPSS dataset, the tolerance interval is defined as [-13, 10] [33, 37, 38, 39], then it is given by:

$$A = \frac{100}{M}\sum_{k=1}^{M} I(d_k); \qquad I(d_k) = \begin{cases} 1, & \text{if } d_k \in [-13, 10] \\ 0 & \text{if } d_k \notin [-13, 10] \end{cases} \tag{20}$$

3.2.2. Uncertain prediction evaluation metrics

Uncertain evaluation metrics for prognostics at component level

In addition to measuring the accuracy of point-wise estimates, it is necessary to evaluate the uncertainty management performance of the proposed methodology. Particularly, the prediction interval coverage probability (PICP) and the normalized mean prediction interval width (NMPIW) metrics, that can properly resume the prediction uncertainty, are widely used in literature [11, 1, 16].

Prediction interval coverage percentage (PICP): represents the probability that the true targets (RUL_k^*) fall within the lower and upper bounds ($[L_\alpha(RUL_k), U_{\alpha(RUL_k)}]$) of predictions, RUL_k, with a prescribed confidence level $(1-\alpha)$. It is given by:

$$PICP = \frac{1}{M}\sum_{k=1}^{M} I(RUL_k^*); \qquad I(RUL_k^*) = \begin{cases} 1, & \text{if } RUL_k^* \in [L_\alpha(RUL_k), U_\alpha(RUL_k)] \\ 0 & \text{if } RUL_k^* \notin [L_\alpha(RUL_k), U_\alpha(RUL_k)] \end{cases} \tag{21}$$

Normalized mean prediction interval width (NMPIW): is used to evaluate the width of prediction intervals and is defined as:

$$NMPIW = \frac{1}{M.(RUL_{max} - RUL_{min})}\sum_{k=1}^{M}(U_\alpha(RUL_k) - L_\alpha(RUL_k)) \tag{22}$$

where RUL_{min} and RUL_{max} are the minimum and maximum values of the target RUL respectively.

Uncertain evaluation metrics for prognostics at system level

Besides, given M_s the total prediction points at system level, we propose to use the two following metrics to measure the uncertainty management capacity for system prognostics:

Mean accuracy probability (MAP). MAP is used to evaluate the proper level of the system RUL (SRUL) probability function. It represents the probability that the estimated SRULs falls within a tolerance interval around the target values. For the C-MAPSS dataset, the tolerance interval is defined as $[SRUL_k^* - 13, SRUL_k^* + 10]$, with $k \in 1, 2, ..., M_s$.

$$MAP = \frac{1}{M_s} \sum_{k=1}^{M_s} P\left(SRUL_k \in [SRUL_k^* - 13, SRUL_k^* + 10]\right) \tag{23}$$

Mean normalized prognostics horizon (MNPH). Prognostic horizon (PH) is defined by the difference between the system's end-of-life time (t_{SEOL}) and the prognostic moment (t_p), i.e. the current time starting RUL prediction, see Figure 7. MNPH is then used to measure the PH width compared to the system life time.

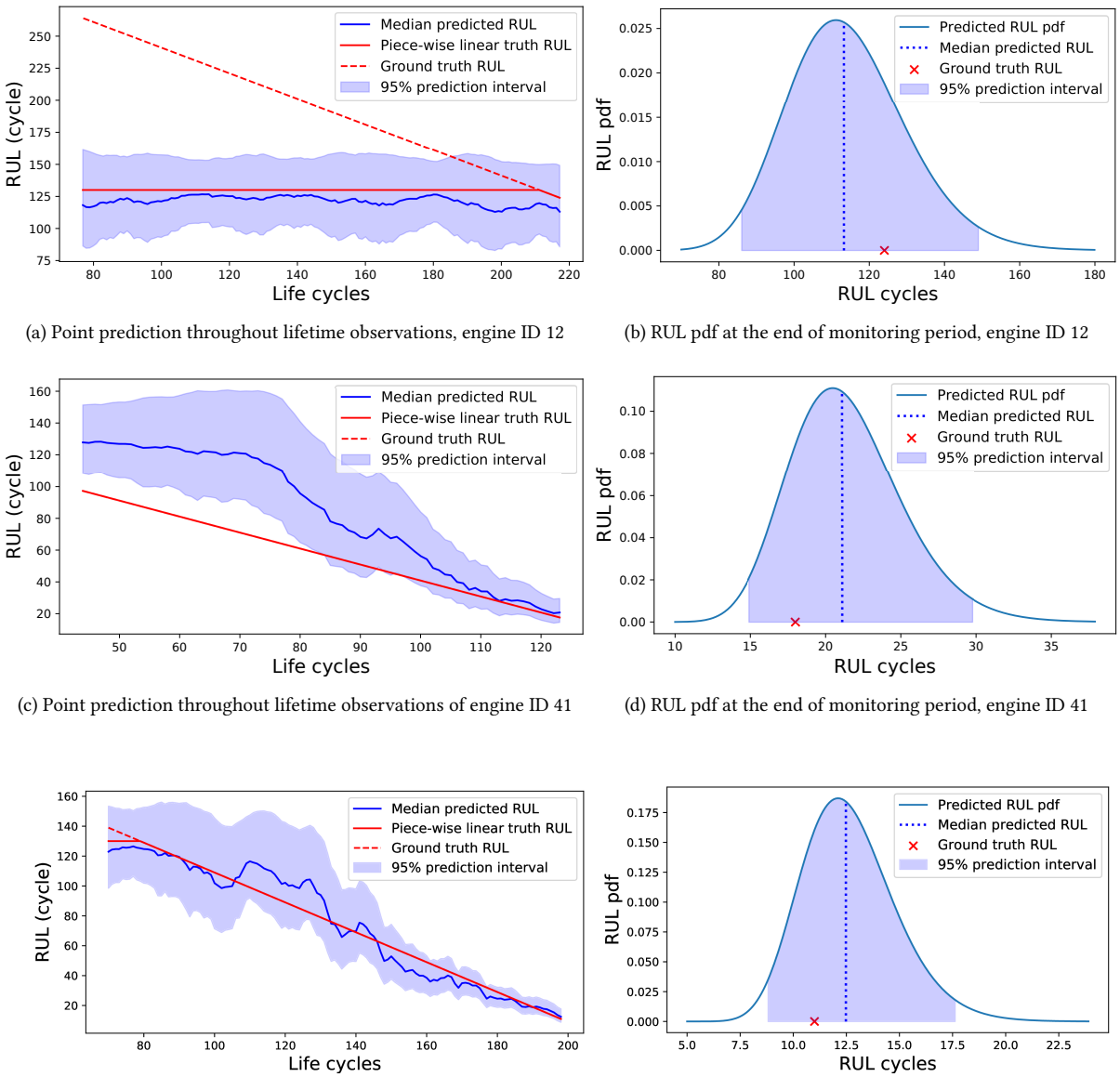
$$MNPH = \frac{1}{M_s} \sum_{k=1}^{M_s} \frac{t_{SEOL_k} - t_{p_k}}{t_{SEOL_k}} \tag{24}$$

3.3. Discussion of prognostics results for components

Using the metrics presented in subsection 3.2, the performance of the proposed Lognorm-LSTM model is investigated in terms of both point prediction accuracy and uncertainty management capacity. Since the Lognormal-LSTM output provides 2 parameters characterizing components RUL distributions instead of a single predicted RUL value, the mean, median or mode values of these RUL distributions will be calculated by equations given in Table 1 to assess the model's point prediction accuracy.

Figure 8 illustrates the prognostics results throughout lifetime observations of some engines in the test set FD001: the longest lifetime engine (ID 12), the shortest lifetime engine (ID 41) and a random engine (ID 35). One can see that for engine ID 12, although its monitoring data are collected until 217 life cycles, this engine is only in the first deterioration stage and its true RUL is still large, i.e. 124 cycles. Therefore, the median value of the predicted RUL (characterized by the continuous blue line in Figure 8.a) is close to the maximum value of the piecewise linear RUL function (presented by the continuous red line in Figure 8.a). In addition, its 95% RUL prediction interval at cycle 217 (the end of the observation period) is quite large, from 86 until 148 (Figure 8.b).

For the shortest life time engine (ID 41), as its degradation process is particularly fast compared to other units, the predicted RUL distribution at the early stage of the observation period is not good: the ground truth RUL is outside of the 95% RUL prediction interval (Figure 8.c). However, as shown in Figure 8.c, the prediction results are improved



(a) Point prediction throughout lifetime observations, engine ID 12

(b) RUL pdf at the end of monitoring period, engine ID 12

(c) Point prediction throughout lifetime observations of engine ID 41

(d) RUL pdf at the end of monitoring period, engine ID 41

(e) Point prediction throughout lifetime observations of engine ID 35

(f) RUL pdf at the end of monitoring period, engine ID 35

Figure 8: Prediction results for engines in test set FD001

along the engine's life time when collecting more monitoring data: the predicted RUL's median value is getting closer and closer to the ground truth RUL. And at the end of the observation period (cycle 123), its true RUL (18 cycles) falls within a narrow enough 95%-RUL-prediction interval [14.9, 29.8] (Figure 8.d).

The convergence of the predicted RUL distribution towards the ground truth RUL along the engine's monitoring life time is further illustrated when considering engine ID 35 (Figure 8.e), which leads to a particularly narrow prediction interval. In fact, at the end of the observation period (cycle 198), its truth RUL (11 cycles) falls within a quite narrow

interval [8.8, 17.6] representing the 95% RUL prediction confidence interval (Figure 8.f).

Next, the Lognorm-LSTM model's performance is comprehensively evaluated on the test set FD001. Table 3 presents the point prediction accuracy at the end of the observation period for all engines. Considering the proposed model's results, due to the asymmetric property of the predicted RUL distribution, one can see that there exist slight differences when using its mode, mean or median values for the evaluation of the point prediction accuracy metrics. Among them, the use of mean value provides the best results for MSE and Accuracy metrics while the best score function (SF) is attained with the mode value.

Compared to other state-of-the-art prognostics algorithms in the literature, although the proposed Lognorm-LSTM model does not achieve the highest SF value (only 4% less than the best score provided by RULCLIPPER [32]), it is the best one according to the MSE (149.5) and Accuracy metrics (72%) shown in Table 3. Note that the MSE and Accuracy metric values given by RULCLIPPER are respectively 17.7% and 6.9%, which are worse than the ones obtained by our model. Furthermore, RULCLIPPER only provides a single predicted RUL value instead of a RUL distribution that allows quantification of prognostics uncertainty.

Table 3: Accuracy comparison on point prediction of other state-of-the-art methods

	MSE	**SF**	**Accuracy**
RULCLIPPER [32]	176.0	**216.0**	67%
MODBNE [35]	226.2	334.2	_
Embed-LR1 [40]	155.0	219.0	59%
LSTMBS [11]	209.7	481.1	_
IESGP [1]	216.7	331.9	_
DBNBP-IPF [38]	_	543.0	51%
DBN-IPF [38]	_	314.0	63%
BiLSTM-ED [39]	217.3	273.0	57%
M3-1 [17]	155.7	242.3	_
TSCG [36]	304.1	468.5	_
SBI-EN [37]	184.5	228	67%
MCLSTM [33]	188	315	_
Lognorm-LSTM (Mode)	161.3	225.5	69%
Lognorm-LSTM (Mean)	**149.5**	243.8	**72%**
Lognorm-LSTM (Median)	151.6	234.2	69%

Figure 9 presents the ability to capture the prognostics uncertainty of our Lognorm-LSTM model. One can see that for the engines (in the test set FD001) whose truth RUL is lower than 40 cycles, the 95% RUL prediction interval is narrow. In addition, the RUL distribution's median value is close to the truth RUL. Compared to other existing algorithms (Table 4), the proposed model provides the tightest width of 95% RUL prediction interval (NMPIW = 0.316) with a good enough score of the prediction interval coverage percentage (PICP = 0.95). Note that although

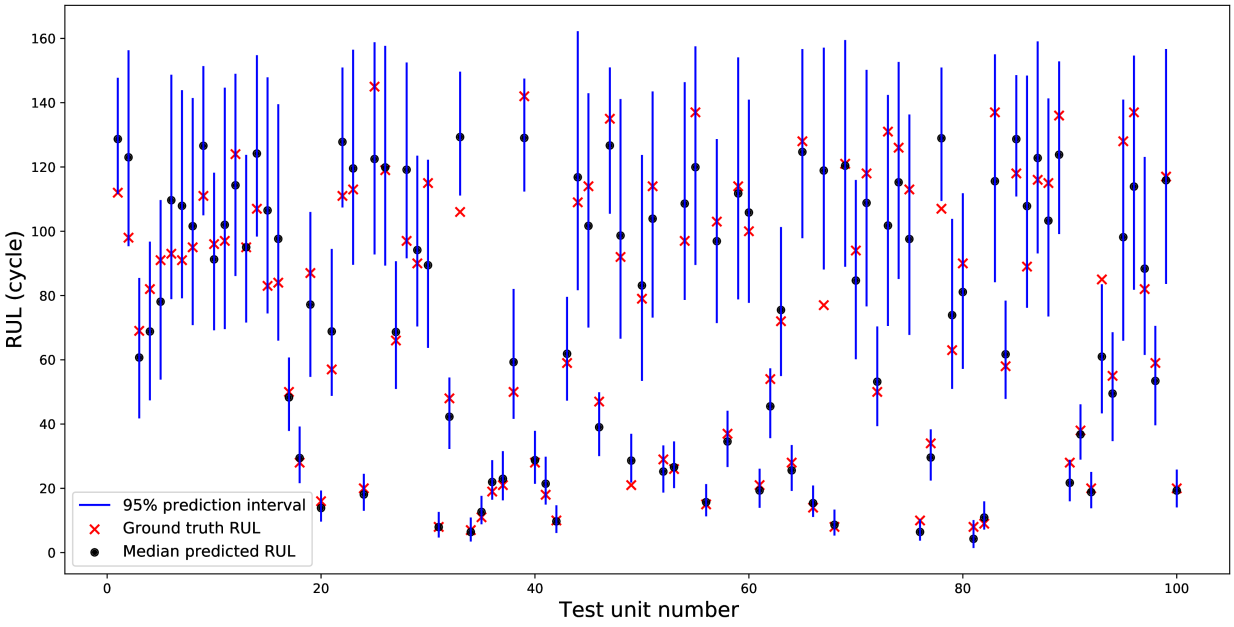


Figure 9: Prognostics results of Lognorm-LSTM model on test set FD001 with PICP = 0.95, NMPIW = 0.316

the IESGP model [1] gets the best result according to the PICP metric, its mean prediction interval width is more than a half of the target RUL's range (NMPIW = 0.540). This large prediction interval width presents a high uncertainty level in prognostics results and is problematic when needing to take a decision. Besides, the LSTMBS method [11] has the same uncertainty management ability as our model (its PICP and NMPIW scores are close to the ones of our model) but its point prediction accuracy metrics (MSE and SF) are much worse than ours, see Table 3.

Table 4: Performance comparison of uncertainty quantification of other state-of-the-art methods

	LSTMBS [11]	IESGP [1]	RNP [16]	Lognorm-LSTM
PICP	0.960	**0.995**	0.870	0.950
NMPIW	0.377	0.540	_	**0.316**

3.4. Discussion of uncertainty management for prognostics of multi-component systems

In the previous section, the performance of the proposed model for prognostics of turbofan engines (i.e. prognostics at component level) was highlighted when compared to other advanced prognostics algorithms in the literature. Now, based on the predicted RUL distributions (at component level), the prognostics uncertainty at system level will be quantified. To illustrate the wide applicability of the proposed methodology, four common structures: 1) series, 2) combined-1, 3) combined-2, and 4) parallel (cf. Figure 6) considered in this section.

In addition, to investigate the impact of monitoring data quantity on system prognostics uncertainty management,

complete life cycles of engines are necessary. Then, different amount of data collected during these complete life time are used to evaluate the SRUL probability. To do this, the FD001 training set will be divided into two parts. The first part, including 70 random engines, is used for training the Lognorm-LSTM to predict engine's RUL distribution while the second part, consisting of 30 remaining engines, called the validation set is used for quantification of the prognostics uncertainty at system level.

Considering an aircraft having 4 turbofan engines, we assume that these engines are connected in 1) series, 2) combined-1, 3) combined-2, and 4) parallel structures (cf. Figure 6) to ensure the aircraft operation. These cases can be typical of various failure modes affecting differently the failure of the system (i.e. the entire aircraft). Each engine is subject to its own degradation process and independently monitored. To investigate the ability of uncertainty management for prognostics of aircraft failure due to turbofan engines, we randomly select 4 engines among 30 engines in the validation set. The monitoring data of each engine are used as the proposed model's input to predict its RUL lognormal distribution. Based on these predicted RUL distributions, the aircraft reliability or the SRUL probability can by derived corresponding to the type of system using Eqs. 12, 13, 14, 16, or 17. This procedure is repeated 1000 times to simulate 1000 scenarios of aircraft failures due to engine deteriorations. Given the truth SRUL calculated by Eqs. 5, 6, 7 and 8 according to system structures, we evaluate system prognostics uncertainty management metrics, i.e. mean accuracy probability (MAP) and mean normalized prognostics horizon (MNPH), at 70%, 80% and 90% of lifetime ($pc = 0.7, 0.8$, and 0.9) of the first failed engine in such aircraft, $t_p = pc \cdot \min_{\forall i}(t_{EOL_i})$. These results are presented in Table 5.

Table 5: Uncertainty management for prognostics of multi-component systems with different structures, $t_p = pc \cdot \min_{\forall i}(t_{EOL_i})$

		Series	Combined 1	Combined 2	Parallel
$pc = 0.7$	**MAP**	0.821	0.739	0.526	0.543
	MNPH	30.0%	33.4%	44.1%	51.0%
$pc = 0.8$	**MAP**	0.907	0.845	0.628	0.527
	MNPH	20.0%	23.8%	36.4%	46.2%
$pc = 0.9$	**MAP**	0.991	0.943	0.733	0.580
	MNPH	10.0%	14.3%	29.2%	41.1%

Table 6: Uncertainty management for prognostics of multi-component systems with different structures, $t_p = pc \cdot t_{SEOL}$

		Series	Combined 1	Combined 2	Parallel
$pc = 0.7$	**MAP**	0.822	0.782	0.675	0.591
	$\overline{SRUL^*}$ (cycles)	50.1	53.2	64.6	82.6
$pc = 0.8$	**MAP**	0.904	0.900	0.886	0.794
	$\overline{SRUL^*}$ (cycles)	32.4	35.7	45.3	53.0
$pc = 0.9$	**MAP**	0.989	0.988	0.974	0.942
	$\overline{SRUL^*}$ (cycles)	16.1	17.2	20.9	26.9

Considering Table 5, we can note that the proposed method's performance strictly depends on the monitoring data quantity collected along the system's lifetime. For series system, as the aircraft's lifetime is also the one of the first failed engine, when $pc = 0.7, 0.8$, or 0.9, the MNPH for aircraft are respectively 30%, 20% and 10%. In these cases, the probability that the predicted aircraft RUL (SRUL) falls within the accuracy interval $[SRUL^* - 13, SRUL^* + 10]$ is great enough, e.g. MAP = 0.991 when $pc = 0.9$. Contrarily, the MAP in the case of parallel systems is not good enough, e.g. MAP = 0.58 when $pc = 0.9$. In fact, the prognostic horizon in this case is quite large (MNPH = 41.1%) and there is not enough monitoring data to reduce the prognostics uncertainty. Similar to the case of the combined-1 and 2 structures, the method's performance is decreasing when the prognostics horizon is increasing, e.g. for combined-1 structure, MAP reduces from 0.943 to 0.739 when MNPH rises from 14.3% to 33.4%.

The significant impact of the prognostics horizon on the uncertainty management of SRUL predictions is highlighted one more time when considering Table 6, for which the prediction time t_p is defined by a percentage of the system life time, $t_p = pc \cdot t_{SEOL}$. Although the mean normalized prognostics horizon's value is the same when considering different system structures (MNPH is equal to 30%, 20% and 10% for $pc = 0.7, 0.8$, and 0.9 respectively), the mean truth value of SRUL ($\overline{SRUL^*}$) decreases according to the system architecture as follows: parallel, combined-2, combined-1, and series. Besides, one can see that in Figure 9, the greater value of RUL is, the wider 95% prediction interval is. Therefore, the MAP increases according to the system structures: parallel, combined-2, combined-1, and series, for different values of pc. However, when $pc = 0.9$, the MAP of the four systems are reliable enough for all system configurations, including for the parallel system (MAP = 0.942), which is the toughest among the four cases.

4. Conclusion

In this paper, a new methodology to quantify prognostics uncertainty of a multi-independent-component system was presented. It uses heterogeneous monitoring data of components as the Lognorm-LSTM's input to predict the RUL distribution. Then, the component's RUL distribution is considered to derive the system reliability based on mathematical formulas corresponding to the system's architecture. It also provides the probabilities that the system RUL will fall within different time intervals, and therefore shows promising abilities for industry applications. In fact, as the previous mentioned time intervals can be defined according to the requirements of the operation planner, the proposed methodology's result can be easily adapted to practical demands.

The performance and the efficiency of the proposed Lognorm-LSTM model were highlighted when compared to existing prognostics algorithms available in the literature for the benchmark dataset: the turbofan engines dataset provided by the NASA Ames Prognostics Center of Excellence. Considering the point prediction accuracy aspect, the proposed model offers the best results among the compared models, according to *Mean squared error* and *Accuracy* metrics. In terms of uncertainty management aspect, it gives narrow RUL distributions with reasonable scores

of prediction interval coverage percentage. Moreover, thanks to the component's explicit RUL distributions, the useful information of system reliability according to different structures can be easily evaluated. Through numerical experiments, one can see that the prognostics uncertainty management ability of the proposed method strictly depends on the prognostic horizon. Its mean accuracy probability is decreasing when the prediction time is far from the system end of life. Hence, it would be interesting to investigate the impact of different quality and quantity levels of monitoring data on the prognostic performance to optimize the inspection policy or the decision time.

Finally, the proposed methodology works under the assumption that the component's RUL follows a lognormal distribution. Further works could consider different probability distributions and allow to automatically select the best suitable distribution according to the experimental data characteristics. Another relevant direction for future work could be the extension of the methodology to take into account the statistical dependencies between components.

References

[1] Chongdang Liu, Linxuan Zhang, Yuan Liao, Cheng Wu, and Gongzhuang Peng. Multiple sensors based prognostics with prediction interval optimization via echo state gaussian process. IEEE Access, 7:112397–112409, 2019.

[2] Naipeng Li, Nagi Gebraeel, Yaguo Lei, Xiaolei Fang, Xiao Cai, and Tao Yan. Remaining useful life prediction based on a multi-sensor data fusion model. Reliability Engineering & System Safety, 208:107249, 2021.

[3] Sen-Ju Zhang, Rui Kang, and Yan-Hui Lin. Remaining useful life prediction for degradation with recovery phenomenon based on uncertain process. Reliability Engineering & System Safety, 208:107440, April 2021.

[4] Behnoush Rezaeianjouybari and Yi Shang. Deep learning for prognostics and health management: State of the art, challenges, and opportunities. Measurement, 163:107929, 2020.

[5] Yingjun Deng, Alessandro Di Bucchianico, and Mykola Pechenizkiy. Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model. Reliability Engineering & System Safety, 196:106727, 2020.

[6] Ferhat Tamssaouet, Nguyen Khanh, Kamal Medjaher, and Marcos Orchard. Combination of Long Short-Term Memory and Particle Filtering for Future Uncertainty Characterization in Failure Prognostic. In Proceedings of the 31st European Safety and Reliability Conference, pages 275–281, 2021.

[7] Jiachen Guo, Jing Cai, Heng Jiang, and Xin Li. Remaining useful life prediction for auxiliary power unit based on particle filter. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 234(15):2211–2217, 2020.

[8] Ruihua Jiao, Kaixiang Peng, Jie Dong, and Chuanfang Zhang. Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics. Reliability Engineering & System Safety, 203:107028, 2020-11-01.

[9] TV Vishnu, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, et al. Data-driven prognostics with predictive uncertainty estimation using ensemble of deep ordinal regression models. International Journal of Prognostics and Health Management, 10(4), 2019.

[10] Marco Rigamonti, Piero Baraldi, Enrico Zio, Indranil Roychoudhury, Kai Goebel, and Scott Poll. Ensemble of optimized echo state networks for remaining useful life prediction. Neurocomputing, 281:121–138, 2018.

[11] Yuan Liao, Linxuan Zhang, and Chongdang Liu. Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method. In 2018 ieee international conference on prognostics and health management (icphm), pages 1–8. IEEE, 2018.

[12] Piero Baraldi, Francesca Mangili, and Enrico Zio. Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data. Reliability Engineering & System Safety, 112:94–108, April 2013.

[13] Weiwen Peng, Zhi-Sheng Ye, and Nan Chen. Bayesian deep-learning-based health prognostics toward prognostics uncertainty. IEEE Transactions on Industrial Electronics, 67(3):2283–2293, 2019.

[14] Minhee Kim and Kaibo Liu. A bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics. IISE Transactions, 53(3):326–340, 2020.

[15] Luca Biggio, Alexander Wieland, Manuel Arias Chao, Iason Kastanis, and Olga Fink. Uncertainty-aware remaining useful life predictor. arXiv preprint arXiv:2104.03613, 2021.

[16] Guozhen Gao, Zijun Que, and Zhengguo Xu. Predicting remaining useful life with uncertainty using recurrent neural process. In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pages 291–296. IEEE, 2020.

[17] Zhibin Zhao, Jingyao Wu, David Wong, Chuang Sun, and Ruqiang Yan. Probabilistic remaining useful life prediction based on deep convolutional neural network. Proc. of TESConf 2020 - 9th International Conference on Through-life Engineering Services, Available at SSRN 3717738, 2020.

[18] Ning-Cong Xiao, Kai Yuan, and Hongyou Zhan. System reliability analysis based on dependent Kriging predictions and parallel learning strategy. Reliability Engineering & System Safety, 218:108083, 2022.

[19] Yingshi Hu, Zhenzhou Lu, Xia Jiang, Ning Wei, and Changcong Zhou. Time-dependent structural system reliability analysis model and its efficiency solution. Reliability Engineering & System Safety, 216:108029, 2021.

[20] Ferhat Tamssaouet. Towards system-level prognostics : Modeling, uncertainty propagation and system remaining useful life prediction. PhD thesis, National Polytechnic Institute of Toulouse, 2020.

[21] Ferhat Tamssaouet, Khanh TP Nguyen, and Kamal Medjaher. System-level prognostics under mission profile effects using inoperability input-output model. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019.

[22] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. Computers & Electrical Engineering, 40(1):16–28, 2014.

[23] Khanh T. P. Nguyen and Kamal Medjaher. An automated health indicator construction methodology for prognostics based on multi-criteria optimization. ISA Transactions, pages 81 – 96, July 2021.

[24] Dengshan Huang, Rui Bai, Shuai Zhao, Pengfei Wen, Jiawei He, Shengyue Wang, and Shaowei Chen. A hybrid bayesian deep learning model for remaining useful life prognostics and uncertainty quantification. In 2021 IEEE International Conference on Prognostics and Health Management (ICPHM), pages 1–8. IEEE, 2021.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[26] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, pages 1–18, 2012.

[27] M. T. Todinov. 3 - Methods for analysis of complex reliability networks. In M. T. Todinov, editor, Risk-Based Reliability Analysis and Generic Principles for Risk Reduction, pages 31–58. Elsevier, 2007.

[28] M. Rausand and A Hoyland. System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Edition. 2004.

[29] K. Benaggoune, S. Meraghni, J. Ma, L. H. Mouss, and N. Zerhouni. Post Prognostic Decision for Predictive Maintenance Planning with Remaining Useful Life Uncertainty. In 2020 Prognostics and Health Management Conference (PHM-Besançon), pages 194–199, May 2020.

[30] Tianyi Wu, Li Yang, Xiaobing Ma, Zihan Zhang, and Yu Zhao. Dynamic maintenance strategy with iteratively updated group information. Reliability Engineering & System Safety, 197:106820, May 2020.

[31] Khanh TP Nguyen and Kamal Medjaher. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliability Engineering & System Safety, 188:251–262, 2019.

[32] Emmanuel Ramasso. Investigating Computational Geometry for Failure Prognostics in Presence of Imprecise Health Indicator: Results and Comparisons on C-MAPSS Datasets. In 2nd Europen Confernce of the Prognostics and Health Management Society., volume 5, pages 1–13, July 2014.

[33] Sheng Xiang, Yi Qin, Jun Luo, Huayan Pu, and Baoping Tang. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. Reliability Engineering & System Safety, 216:107927, 2021.

[34] Abhinav Saxena and Kai Goebel. Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository, pages 878–887, 2008.

[35] Chong Zhang, Pin Lim, A. K. Qin, and Kay Chen Tan. Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation

in Prognostics. IEEE Transactions on Neural Networks and Learning Systems, 28(10):2306–2318, October 2017.

[36] Huyang Xu, Nasser Fard, and Yuanchen Fang. Time series chain graph for modeling reliability covariates in degradation process. Reliability Engineering & System Safety, 204:107207, 2020.

[37] Wennian Yu, Il Yong Kim, and Chris Mechefske. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. Reliability Engineering & System Safety, 199:1–12, 2020.

[38] Kaixiang Peng, Ruihua Jiao, Jie Dong, and Yanting Pi. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. Neurocomputing, 361:19–28, 2019.

[39] Wennian Yu, Il Yong Kim, and Chris Mechefske. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mechanical Systems and Signal Processing, 129:764–780, August 2019.

[40] Narendhar Gugulothu, Vishnu Tv, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks. International Journal of Prognostics and Health Management, 9(1), 2018.