Uncertainty in Aircraft Turbofan Engine Prognostics on the C-MAPSS Dataset

Mariana Salinas-Camus[1] and Nick Eleftheroglou[2]

[1, 2] *Intelligent Sustainable Prognostics Group, Aerospace Structures and Materials Department,*
Faculty of Aerospace Engineering, Delft University of Technology
m.salinascamus@tudelft.nl
n.eleftheroglou@tudelft.nl

Abstract

Prognostics and Health Management (PHM) plays a crucial role in maximizing operational efficiency, minimizing maintenance costs, and enhancing system reliability. Predicting Remaining Useful Life (RUL) is a key aspect of PHM, inherently incorporating uncertainty. This paper focuses on uncertainty quantification (UQ) within Data-Driven Models (DDMs), particularly Machine Learning (ML), such as Long Short-Term Memory (LSTMs), and stochastic models namely Hidden Markov Models (HMMs). While ML models emphasize accuracy, stochastic models offer a different paradigm for prognostics, directly addressing uncertainty. Traditional categorizations of uncertainty as aleatory and epistemic face challenges in practical implementation. This paper explores how, in prognostics, HMMs primarily tackle aleatory uncertainty, whereas LSTMs predominantly address epistemic uncertainty. It also discusses the complexities of uncertainty management in prognostics and analyzes further an already proposed alternative approach to categorize uncertainties. Despite theoretical advancements, practical implementation remains challenging, especially for DL models due to their limited interpretability. This study sheds light on UQ challenges and offers insights for future research directions in prognostics.

1. Introduction

Prognostics and Health Management (PHM) is a field that provides users with a thorough analysis of both the current and future health condition of a system. PHM has gained attention during the last years due to the potential that it has to maximize the operational availability, reduce maintenance costs, and improve the system reliability.

Prognostics, as part of the PHM field, aim at predicting the Remaining Useful Life (RUL) of a given engineering system while it is in operation. By definition, the prediction of RUL incorporates uncertainty. Therefore, it is imperative to model RUL as a random variable rather than a deterministic one to account for the inherent uncertainties in prognostics. The prediction of RUL is then used by a decision-making module, which will make health management decisions to fulfill PHM goals.

Nonetheless, uncertainty quantification (UQ) is a challenge within prognostics. In particular, when prognostics are performed with Data-Driven Models (DDMs), which only rely on historical sensor data, UQ can become a greater challenge depending on which type of prognostic DDM is used. Hence, this paper will solely focus on UQ for DDMs, given their extensive use in prognostics and their sensitivity to uncertainty sources related to the data.

As previously mentioned, DDMs use historical sensor data to predict the RUL of the engineering system, and there are different types of DDMs. For the purposes of this paper, we would consider Machine Learning (ML) and stochastic models as the two main categories of DDMs. ML models, which include decision trees, Support Vector Regressor (SVR), and Deep Learning (DL) models, among others, have gained attention in prognostics because of the high accuracy of the RUL predictions.

In contrast to ML, stochastic models offer a different paradigm for data-driven prognostics. Stochastic models, such as Hidden Markov Models (HMMs) and Wiener processes, model the degradation process of the engineering system based on the sensor data, i.e. unsupervised learning, unlike ML models that can find complex relationships between the sensor data and RUL, i.e. supervised learning. Thus, ML models find patterns in the data which allows them to have a good performance when trained with large and labeled datasets but struggle with outliers.

Another important difference between ML models and stochastic models is UQ. In stochastic models, UQ is direct since the output of the model is a probability density function (pdf) of the RUL prediction. For ML models, UQ is a challenge given that ML models are deterministic by nature, i.e. they provide a single-point prediction for RUL. There are techniques to perform UQ, but these might not be suitable for a prognostic application, as will be discussed in Section 3.

It is important to know that UQ is not the ultimate goal, but it is a key step towards uncertainty management. Uncertainty management is defined as the identification of sources of uncertainty and the reduction of uncertainty by leveraging data to characterize better the inherent prognostic uncertainties. Hence, reducing their impact on RUL predictions, which is necessary for the decision-making process (Sankararaman, 2015).

The question is then, which are the sources of uncertainty? The classical categorization considers two sources of uncertainty, aleatory and epistemic. Aleatory refers to the uncertainties that are intrinsic randomness of a phenomenon. Epistemic uncertainty is caused by a lack of knowledge, thus, it is the uncertainty that comes from the model itself (Der Kiureghian & Ditlevsen, 2009). Another way to look at them is that aleatory uncertainty is irreducible, since there is no control over the randomness of the phenomenon, and epistemic uncertainty is reducible given that the model can be changed. Consequently, to perform uncertainty management epistemic uncertainty needs to be addressed.

Nevertheless, even if we manage to identify epistemic uncertainty effectively, how can this information be used to perform uncertainty management in prognostics? Aside from the variability of the data, i.e. aleatory uncertainty, there is uncertainty in the identification of the current state of the system's health or the future loading operation that the system will be subjected to. By considering all these different sources as part of "epistemic uncertainty", it is unclear what actions need to be taken to reduce the RUL uncertainty.

For that reason, it has been claimed that the aleatory and epistemic categorization is not suitable for prognostics (Sankararaman & Goebel, 2013) and a more suitable categorization has been proposed, which will be further explained in Section 5. Although this categorization has been presented in different publications, it has not been applied, to the best of the author's knowledge, to a real-life scenario. Until now, the few prognostics publications that identify sources of uncertainty continue to use the classical categorization.

This paper presents both an stochastic model and a DL model under the same case study. To understand the use of stochastic models, an HMM presented. With the HMM, a new expression for RUL prediction is introduced in this study and is compared with the state-of-the-art RUL expression in terms of UQ. For DL models, a Long-Short Term Memory (LSTM) is used, given that it has been argued as the one with the best performance in terms of accuracy for several engineering applications. The LSTM is analyzed by using different parameters for UQ.

Therefore, by the use of these models this paper aims to provide an understanding of uncertainty in prognostics, and how different types of DDMs deal with UQ. As well as to offer a discussion in terms of future perspectives to address the UQ challenge, ultimately aiming towards the goal of uncertainty management.

The paper is organized as follows, Section 2 offers the theoretical background of HMMs and the new prognostic expression and Section 3 details the UQ methods for DL models, as well as DL model approaches in prognostics. The case study, including the data prepossessing and results, is presented in Section 4. Section 5 offers a discussion about the future perspective on UQ for prognostics. Finally, the paper is concluded in Section 6.

2. Hidden Markov Models

HMMs are a widely used stochastic model for different engineering applications. In the context of prognosis, it has been used for composites (Eleftheroglou, 2020; Eleftheroglou et al., 2024), lithium-polymer batteries (Eleftheroglou et al., 2019), turbofan engines (Giantomassi et al., 2011), and simulated fatigue crack growth (Le et al., 2014). In each one of these publications, different variants of HMM are used. A multi-branch HMM is used in (Le et al., 2014) to take into account the multiple degradation modes that can occur. A more complex version of HMM is used in (Eleftheroglou, 2020; Eleftheroglou et al., 2019), called the Non-Homogeneous Hidden Semi Markov Model (NHHSMM). When applied to composites the author used an adaptive approach of the NHHSMM that allowed the model to predict the RUL for testing data that had gone through unexpected phenomena.

Thus, HMMs have demonstrated their applicability through the use of different variants of it. In this paper, the classical HMM will be used, and a new definition for prognostic is presented.

HMMs can model a sequence of observations, which in this case is the data coming from the sensors. It is used in processes in which the state of the engineering system cannot be directly observed, hence, they are hidden. The engineering system is modeled as a Markov process, meaning that the probability of transitioning from one state to another depends only on the current state. The sojourn time of each hidden state is defined by an exponential distribution (continuous case) or a geometrical distribution (discrete case). Each state emits an observation with a certain probability distribution. Below, the parameters that describe an HMM are detailed (Rabiner, 1989).

- N: number of states. Individual states are denoted as $S = \{S_1, S_2, ..., S_N\}$, and the state at time t as q_t.
- M: number of distinct observation symbols per state. Individual observations are denoted as $V = \{v_1, v_2, ..., v_M\}$.
- State transition: the state transition probability distribution is denoted as $A = \{a_{ij}\}$, where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$. This expression is the probability that the state at time $t+1$ is equal to the hidden state S_j given that the current state q_t is equal to the hidden state S_i.
- Observation distribution: the observation symbol probability distribution in state j, $B = b_j(k)$, where $b_j = P[v_k \text{ at t} | q_t = S_j]$, with $1 \leq j \leq N$ and $1 \leq k \leq M$.
- Initial state: the initial state distribution $\pi = \{\pi_i\}$ where $\pi_i = P[q_1 = S_i]$ with $1 \leq i \leq N$.

The complete parameter set of the model is denoted as $\lambda = (A, B, \pi)$. To train an HMM it is necessary then to adjust the model parameters λ to maximize $P(O|\lambda)$, meaning that the parameters are optimized to best describe the observation sequences, which in the case of prognostics are the degradation histories. Since there is no possible way of analytically calculating $P(O|\lambda)$, the iterative algorithm Baum-Welch can locally maximize it.

In the particular case of prognostics and this paper, some assumptions are made. First, the last state is not hidden but observable and it represents failure. Second, in the failure state, only one observation value is emitted. Third, only left-to-right transitions are allowed, meaning that while in hidden state i, it is only possible to remain in state i or to transit to state $i+1$. This last assumption is valid only when modeling a degradation process independently from maintenance actions.

Once the model parameters λ are estimated and we have an observation sequence $O = O_1 O_2 ... O_T$, two questions arise. First, what is the probability of the observation sequence given the model $P(O|\lambda)$? The second question is, which is the most likely sequence of hidden states $Q = q_1 q_2 ... q_T$?

The answer to the first question, the Forward-Backward algorithm is used. The forward part calculates the likelihood of being in a hidden state at a certain time point given the available observations. The result of the forward part is then $P(O|\lambda)$, which answers the first question. The backward part is then used to answer the second question since calculates the likelihood of observing the remaining data, given the current hidden state. The result of the complete Forward-Backward algorithm is the posterior distribution which is the probability of being in each state at each time, given the entire sequence of observed data.

However, to answer fully to the second question it is necessary to find the single best state sequence that maximizes $P(Q|O, \lambda)$ that is equivalent to maximizing $P(Q, O|\lambda)$. The maximization is done via the Viterbi algorithm. After the Viterbi has estimated the most likely sequence of hidden states, it is possible to calculate the RUL. In the state-of-the-art, a time-invariant (TI) (Dong & He, 2007a) prognostic measure used is defined in (1). The variables $a_{i,i}$ and $a_{i,i+1}$ represent the probability of remaining in the current hidden state or transitioning to the next hidden state, respectively. The variable $D_i(d)$ represents the pdf (or pmf for the discrete case) evaluated in the probability of transition to the same state i, i.e. a_{ii}.

$$RUL_i^t = a_{i,i}(D_i(d) + RUL_{i+1}) + a_{i,i+1}(RUL_{i+1}) \quad (1)$$

In this paper, a new time-dependent (TD) prognostic measure is introduced in (2). This TD prognostic measure is expected to improve accuracy of the RUL prediction and to reduce the spread of the confidence intervals, which can be calculated by the weighted spread of uncertainty (WSU) presented in Appendix A.

$$\begin{aligned} RUL_i^t = d_{i,i}^T &\left(D_i(d-\tau) + \sum_{i=k+1}^{N-1} D_k(d) + \mathcal{N}(1, \epsilon) \right) \\ &+ d_{i,i+1}^T \left(\sum_{k=i+1}^{N-1} D_k(d) + \mathcal{N}(1, \epsilon) \right) \quad (2) \end{aligned}$$

The notation for this expression is as follows. RUL_i^t, is the RUL in the state i and time step t. Once again, $D_i(d)$ represents the pdf (or pmf for the discrete case) evaluated in the probability of transition to the same state i, i.e. a_{ii}. The variable τ is the time spent in the current state i. Therefore, the term $D_i(d-\tau)$ represents a shift in the pdf making this RUL expression time-dependent. The variables $d_{i,i+1}^T$ and $d_{i,i}^T$ are derived from the transition matrix and are defined as shown in (3) and (4), respectively.

$$d_{i,i+1}^T = P(d \leq \tau | S_t = i) \quad (3)$$

$$d_{i,i}^T = 1 - d_{i,i+1}^T \quad (4)$$

The result of the expression 2 is the pdf of RUL per time step. Therefore, the confidence intervals can easily be obtained by calculating the cumulative density function (CDF) and, later, choosing the confidence level, usually 95%.

However, even if the HMM has a closed form for the posterior distribution, the distribution captures aleatory uncertainty, including uncertainty propagation and quantification via the prognostic measure. In state-of-the-art prognostics, including the publications mentioned above, the HMMs presented usually address only aleatory uncertainty. Yet, epistemic uncertainty can be included in HMMs through a time-consuming sensitivity analysis, which traditionally has been used for accounting for epistemic uncertainty in stochastic models. In (Xie et al., 2016) a Generalized Hidden Markov Model (GHMM) is introduced that can identify both epistemic and aleatory uncertainties by using imprecise probabilities. The results show that the GHMM can make more robust decisions because the uncertainties can be differentiated. Yet, it is a computationally expensive model.

3. Deep Learning Models

For DL models, as well as for any ML model, uncertainty quantification is a challenge since they are by nature deterministic, i.e. a single-point value for RUL prediction. Bayesian Neural Networks (BNNs), which are an extension of Neural Networks (NNs), overcome this by providing a pdf as a result. However, BNNs still have a problem when quantifying uncertainty since they offer an approximation of the posterior distribution (Abdar et al., 2021). The posterior distribution cannot be directly calculated because it is intractable to calculate the marginal distribution. Therefore, there is no close-form expression for the posterior distribution.

BNNs can provide a pdf as an output because they have distributions over the weights parameters and not deterministic values as in the case of NNs. These distributions in the weights parameters are learned over Bayesian inference, which uses the Bayes rules as shown in equation (5). In this expression, $P(w|X, Y)$ is the posterior distribution, $P(w)$ is the prior distribution, $P(X, Y|w)$ is the likelihood and $P(X, Y)$ is the marginal distribution.

$$P(w|X, Y) = \frac{P(X, Y|w)P(w)}{P(X, Y)} \tag{5}$$

Once again, it is computationally intractable to calculate $P(X, Y)$. Thus, these models offer an approximation of $P(w|X, Y)$ by using Variational Inference (VI). VI approximates the posterior distribution by using a variational parameter $q_\theta(w)$. The distribution q_θ is approximated by minimizing θ with the Kullback-Leibler (KL) divergence.

$$KL(q_\theta(w)||P(w|X, Y)) = \int q_\theta(w) \log \frac{q_\theta(w)}{P(w|X, Y)} dw \tag{6}$$

However, KL minimization is still intractable because it needs the posterior distribution that it was impossible to obtain in the first place. By rearranging KL into the evidence lower bound (ELBO), the need to have the posterior is avoided.

$$\mathcal{L}_{VI}(\theta) = \int q_\theta \log P(Y|X, w)dw - KL(q_\theta(w)||P(w)) \tag{7}$$

However, even though VI offers a good approximation of the posterior it is still challenging to implement given their computational cost (Nastos, Komninos, & Zarouchas, 2023). As a result, other techniques have arisen, such as Monte Carlo (MC) dropout, Deep Gaussian Processes, and Markov Chain Monte Carlo (Abdar et al., 2021).

MC Dropout has been introduced as a technique to quantify epistemic uncertainty and is the most used one due to its simple implementation (Gal & Ghahramani, 2016). This technique approximates the posterior by randomly switching off neurons, given a dropout probability. The same architecture is run multiple times and each dropout configuration corresponds to a different sample from the approximate posterior distribution.

However, MC dropout struggles to approximate complex posterior distributions, which may lead to good approximations only in certain regions of the posterior distribution but poor approximations in others (Fort, Hu, & Lakshminarayanan, 2019). Even more, it has even been questioned the fact that MC dropout is Bayesian since it fails sanity checks and is a design artifact since the posterior distribution converges to different values depending on the dropout probability assigned by a user (Folgoc et al., 2021). Hence, these techniques although easy to implement, do not always provide a good approximation of the desired distribution. The latter leads to uncertainty about the posterior distribution approximation that is already quantifying RUL uncertainty, adding up to uncertainty propagation of the entire prognostic model.

In the context of prognostics, these types of models have been applied with Bayesian LSTMs since LSTM, in general, provides the best results in terms of accuracy metrics. However, in (Peng, Ye, & Chen, 2019) and in (Xiahou, Wang, Liu, & Zhang, 2023) a point estimation of the final RUL value is made, instead of a prediction of RUL through the operation time. Nevertheless, (Xiahou et al., 2023) includes a RUL prediction during the operation time by including a credible interval. The results are promising, yet the main drawback of this approach is the complexity of the model and its optimization, as the authors have claimed to be “extremely intractable and time-consuming”.

Other Bayesian approaches such as (Caceres, Gonzalez, Zhou, & Droguett, 2021) perform UQ including both aleatory and epistemic uncertainty. However, it is not reported in the results how much each source contributes to the confidence intervals, which are also quite volatile. Epistemic uncertainty is quantified with MC dropout with a probability dropout value of 0.25, which is considered lower than the standard value of 0.5.

In (Pei et al., 2022) a Bayesian RNN is used with the dropout technique, however, they use a value of dropout between 0.05 to 0.2, which once again is considered low given that the standard dropout value. Low dropout values lead to narrow confidence intervals, meaning less estimated uncertainty in the RUL predictions. Thus, the choice of low dropout values can cause an underestimation of uncertainty that can be prejudicial for decision-making.

When it comes to aleatory uncertainty in DL models, it is split into two categories: homoscedastic and heteroscedastic. Homoscedastic uncertainty corresponds to the noise in the data and it remains constant through the whole data set, while heteroscedastic uncertainty corresponds to the noise that varies with the input (Nemani et al., 2023). The few DL models that include aleatory uncertainty, include only one part of it. For example in (Li, Yang, Lee, Wang, & Rong, 2020) a Bayesian DL framework is developed that takes into account heteroscedastic aleatory uncertainty. In the already mentioned work of (Caceres et al., 2021), only heteroscedastic aleatory uncertainty is address and it is also assumed to follow a Gaussian distribution.

4. Case Study

To perform a comparison between a stochastic and a DL model, the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is used (Frederick, DeCastro, & Litt, 2007). The C-MAPSS dataset is used as a benchmark within the prognostics community. This dataset is composed of four sub-datasets of simulated run-to-failure degradation histories from turbofan engines, with information from 21 sensors. Each sub-dataset considers a variety of operational conditions and injects different fault modes. For this paper only the sub-dataset FD001 is used, which consists of 100 training degradation histories. This dataset is divided into two in a random manner to have a training set of 80 degradation histories for training and 20 for testing. Additionally, sensors 1, 5, 6, 10, 16, 18, and 19 were eliminated from all the analyses since the values were fixed for every time measurement.

4.1. Pre-processing and training phase

For the HMM, only one feature can be used given the capabilities of the library used in Matlab. Therefore, sensor 11 is chosen since it is the sensor with the highest correlation to RUL. The sensor data is then discretized into 20 clusters using K-Means. The number of clusters was chosen based on the monotonicity index (MI), which allows to identification of the optimal number of clusters that can reasonably represent the degradation process. Once the data has been pre-processed, the optimal number of states is identified as 10, via the Bayesian Information Criterion (BIC). The expressions and results of both the MI and the BIC are shown in Appendix A, along with the estimated transition and emission matrices.

For the LSTM, first, an analysis of the importance of the sensors with respect to RUL was done. The sensors were selected based on their absolute Pearson Correlation Coefficient (PCC) with respect to RUL. Table 1 shows the results for all the sensors under analysis. The sensors selected were the ones with an absolute PCC higher than 0.6. Thus, sensors 2, 4, 7, 11, 12, 15, 17, 20, and 21 were used to train the LSTM. It is important to keep in mind that the LSTM is being trained with more data than the HMM, which only uses data coming from one sensor.

LSTMs need to receive sequences that have the same length, thus, the degradation histories were modified to fulfill this requirement. A sequence length of 362 was selected and values zeros were added in the RUL column, while for the sensors the last measurement was repeated. Thus, the shape of the training set tensor is (80, 362, 9).

The architecture of the LSTM is displayed in Figure 1. The last layer, which corresponds to a Dense layer, uses a linear function as activation. The model was trained using Adam as an optimizer, with a Mean Squared Error (MSE) loss function for 30 epochs.

Table 1. Sensor correlation to RUL values for dataset FD001 in C-MAPSS.

Sensor	PCC
2	-0.61
3	-0.58
4	-0.68
7	0.66
8	-0.56
9	-0.39
11	-0.7
12	0.67
13	-0.56
14	-0.31
15	-0.64
17	-0.61
20	0.63
21	0.64

4.2. Results and Discussion

The results are examined individually because the uncertainty captured by the HMM pertains to aleatory uncertainty, while that captured by the LSTM corresponds to epistemic uncertainty. While it is feasible to incorporate epistemic uncertainty for HMM, most publications employing this model overlook it. Therefore, this paper focuses on analyzing the impact of the prognostic measure on aleatory UQ.

Similarly, for LSTM, MC Dropout is often employed as a

Proceedings of the 8th European Conference of the Prognostics and Health Management Society 2024 - ISBN – 978-1-936263-40-0

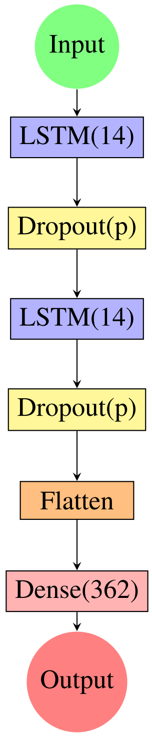

Figure 1. LSTM architecture for prognostics.

methodology to address uncertainty, hence, only epistemic uncertainty is considered. Though it is plausible to include aleatory uncertainty for LSTM, the few publications that do so, only address heteroscedastic aleatory uncertainty and disregard homoscedastic aleatory uncertainty. Consequently, this paper exclusively analyzes epistemic uncertainty via MC Dropout, a common methodology for UQ in LSTMs for prognostics.

4.3. HMM

The results for HMM both with the TI and TD prognostic measure are shown in Table 2. The results correspond to the average RMSE error and the average spread of uncertainty measured by the metric WSU, for the testing set.

Table 2. Average values of the test dataset for the prognostic performance metrics considering the TI and TD expressions of RUL for the HMM.

RUL Expression	RMSE	WSU
TI	45.00	3328839.60
TD	43.10	2978334.12

To visualize confidence intervals, engine #13 is utilized as an example. Figure 2 shows the RUL prediction alongside uncertainty quantification when employing an HMM with TI and TD prognostic measures. It is evident from the visualization that the TD approach provides results with reduced uncertainty and higher accuracy. Thus, the choice of prognostic measure significantly influences how aleatory uncertainty is quantified, as it propagates the aleatory uncertainty captured inherently by the HMM. Even with a simple model, as the HMM is, an improvement can be achieved merely by adopting a different prognostic measure. Therefore, for HMMs in prognostics, one course of action for managing uncertainty could be the development of new prognostic measures that mitigate the tendency to over-propagate inherent aleatory uncertainty captured by the HMM.

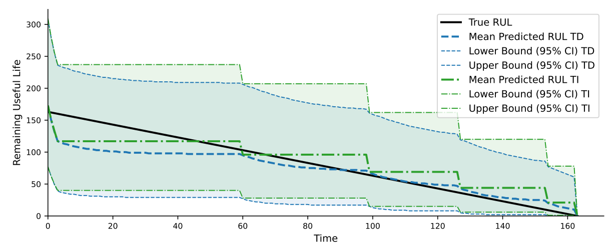

Figure 2. HMM RUL prediction for testing engine #13.

While higher performance is expected with more complex variants of HMMs, such as with a Hidden Semi Markov Model (HSMM) (Dong & He, 2007b) or the NHHSMM previously mentioned in Section 2. However, this paper offers a new time-dependent prognostic measure for the classical HMM that can be extended to other variants in future work. Furthermore, the goal of this paper is not to analyze RUL prediction accuracy but to discuss the challenges and potentials of different DDMs in terms of uncertainty.

4.4. LSTM

The results for LSTM with MC Dropout are summarized in Table 3 for dropout values 0.3, 0.6, and 0.9. The results show high accuracy in terms of RMSE for all three dropout values used, with a slightly better performance for lower dropout values. In terms of epistemic UQ, the value of WSU is higher for higher dropout values as expected.

Table 3. Average values of the test dataset for the prognostic performance metrics considering different dropout values for LSTM.

Dropout value	RMSE	WSU
0.3	1.19	150187.81
0.6	1.57	247935.42
0.9	1.63	645949.41

Figure 3 shows the RUL predictions and confidence intervals for engine #13 (the same engine used for visualization for the HMM). For clarity, only the RUL predictions with dropout values 0.3 and 0.9 are presented. The confidence intervals of the RUL predictions with the LSTM remain approximately the same throughout the degradation history since only the epistemic uncertainty is considered. Additionally, as explained in section 3, it has been claimed that MC Dropout is not even Bayesian and the posterior distribution converges to different values depending on the dropout probability chosen by the user. In these results, it can be seen that according to the dropout values different model uncertainties are calculated. The question arises then on which is the best value to converge to the right posterior distribution of the model, meaning that there is an uncertainty on how to calculate the epistemic uncertainty.

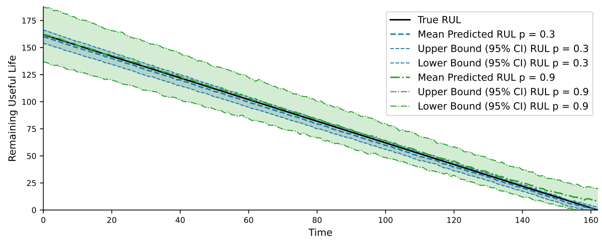

Figure 3. LSTM RUL prediction for testing engine #13 with dropout probability value of 0.3 and 0.9.

5. Future Directions in Uncertainty Quantification for Prognostics

The case study analyzed how UQ is commonly performed in prognostics for HMMs and LSTMs. While for HMMs in most cases only aleatory uncertainty is taken into account, in LSTMs only epistemic uncertainty is addressed via MC Dropout due to its simple implementation. However, even if both models could consider both aleatory and epistemic uncertainty, despite the concerns rising for both approaches, can uncertainty management be performed? Let us remember that epistemic uncertainty is reducible because it comes from the lack of knowledge. Nonetheless, it has been stated that aleatory and epistemic uncertainties often coexist, which makes it difficult to separate them (Nemani et al., 2023). By consequence, uncertainty management would not be feasible.

Hence, a different categorization of uncertainties is needed to allow differentiation. The categorization must be based on the variable of time, inherent in prognostics. This categorization should be subjective and focus on characterizing uncertainties specific to the studied system rather than uncertainties in the population. The need of a different categorization of the sources of uncertainty has been already mentioned in (Sankararaman, 2015), where the author identifies four sources of uncertainties: present, future, model and prognostic measure. The categorization was further extended in (Eleftheroglou, 2020) where a fifth source of uncertainty was included, past uncertainty.

To further explain, the five sources of uncertainty proposed are the following: first, past uncertainties are the ones that come from the manufacturing or assembly process and material quality. Second, present uncertainty refers to the lack of knowledge of the true state of health of an engineering system. Third, future uncertainty is the most difficult and important one to deal with. The future is unknown, and it is not possible to foresee the environmental conditions, loading profile, etc. Another source of uncertainty is the one from the model and it compromises several parts such as model parameters, biases, etc. The last source is the prediction method uncertainty, which is related to the uncertainty coming from the prognostic measure. In the case of supervised techniques, i.e. ML models, the model uncertainty and the prediction method uncertainty become one source.

A remark here is done for past uncertainties since they are not an uncertainty in the present, once uncertainty management is performed. For example, if sufficient data is gathered about the manufacturing process, it can be possible to manage past uncertainties and take them into account when predicting the RUL.

To the best of the author's knowledge, this categorization has not been applied to a real case study and it has only been introduced theoretically. However, an attempt to provide a better understanding on how this categorization can be implemented for HMMs is offered briefly in this section.

For HMMs, past uncertainties can be addressed by the initial parameters distributions π. Present uncertainty can be reflected by the hidden state with the highest probability at the current time step by using the forward probabilities. Future uncertainty, as already mentioned, is the most challenging one. Based on training data, loading profiles can be identified and the probability of changing from one loading profile to another one can be calculated. To account for unexpected phenomena a loading profile can be included that considers an extreme degradation rate, to give an example. Model uncertainty can be addressed by imprecise probabilities or by a sensitivity analysis, as mentioned in Section 2. Finally, the prediction method uncertainty is already considered by the prognostic measure.

DL models remain more challenging to implement under the alternative UQ categorization for prognostics due to the lack of interpretability that has already been mentioned in other publications such as (Fink et al., 2020). Given their black-box nature, the parameters of DL models do not hold a physical meaning making it intractable to connect them to each one of the five sources of uncertainties.

6. Conclusions

This paper explores uncertainty in prognostics, focusing on two main models: HMMs and LSTM networks. It finds that while HMMs primarily deal with aleatory uncertainty (inherent randomness), LSTMs predominantly address epistemic uncertainty (uncertainty from lack of knowledge).

For HMMs, results show the importance of understanding how different prognostic measures affect UQ by broadening the confidence intervals by introducing a new prognostic measurement that is time-dependant. Similarly, for LSTMs, when using the MC Dropout technique, the results show the importance of the parameter selection of the dropout probability value. Even more, from the theoretical background it has been claimed by other authors how a different dropout probability can lead to not converging to the posterior distribution needed to calculate epistemic uncertainty.

However, this paper also opens the discussion about how UQ can be used for uncertainty management in prognostics. Despite attempts to categorize uncertainty, such as distinguishing between epistemic and aleatory uncertainty, challenges persist, particularly in effectively reducing uncertainty. Future directions advocate for a different approach that considers five sources of uncertainty, such as past, present, and future uncertainties, model uncertainties, and prediction method uncertainties.

This alternative approach aims to offer a more comprehensive understanding. However, the prevalence of epistemic uncertainty poses challenges in disentangling from each one of the sources of uncertainty. Even when attempting to quantify past or model uncertainties, the presence of epistemic uncertainty persists due to data limitations and knowledge gaps. While theoretical discussions on implementing alternative categorizations for HMMs exist, practical implementation is constrained. Managing uncertainties in HMMs requires addressing multiple dimensions, encompassing past, present, and future uncertainties, as well as model and prediction method uncertainties. Conversely, implementing this alternative approach on UQ in DL models remains challenging due to their limited interpretability, raising questions about their efficacy in real-world prognostic applications.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., … Acharya, U. R. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, *76*, 243–297.

Caceres, J., Gonzalez, D., Zhou, T., & Droguett, E. L. (2021). A probabilistic bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties. *Structural Control and Health Monitoring*, *28*(10), e2811.

Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? does it matter? *Structural safety*, *31*(2), 105–112.

Dong, M., & He, D. (2007a). Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis. *European Journal of Operational Research*, *178*(3), 858–878.

Dong, M., & He, D. (2007b). A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology. *Mechanical systems and signal processing*, *21*(5), 2248–2266.

Eleftheroglou, N. (2020). Adaptive prognostics for remaining useful life of composite structures.

Eleftheroglou, N., Galanopoulos, G., & Loutas, T. (2024). Similarity learning hidden semi-markov model for adaptive prognostics of composite structures. *Reliability Engineering & System Safety*, *243*, 109808.

Eleftheroglou, N., Mansouri, S. S., Loutas, T., Karvelis, P., Georgoulas, G., Nikolakopoulos, G., & Zarouchas, D. (2019). Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the lithium-polymer batteries of unmanned aerial vehicles with uncertainty quantification. *Applied Energy*, *254*, 113677.

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, *92*, 103678.

Folgoc, L. L., Baltatzis, V., Desai, S., Devaraj, A., Ellis, S., Manzanera, O. E. M., … Glocker, B. (2021). Is mc dropout bayesian? *arXiv preprint arXiv:2110.04286*.

Fort, S., Hu, H., & Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. arxiv 2019. *arXiv preprint arXiv:1912.02757*.

Frederick, D. K., DeCastro, J. A., & Litt, J. S. (2007). *User's guide for the commercial modular aero-propulsion system simulation (c-mapss)* (Tech. Rep.).

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).

Giantomassi, A., Ferracuti, F., Benini, A., Ippoliti, G., Longhi, S., & Petrucci, A. (2011). Hidden markov model for health estimation and prognosis of turbofan engines. In *International design engineering technical conferences and computers and information in engineering conference* (Vol. 54808, pp. 681–689).

Le, T. T., Chatelain, F., & Bérenguer, C. (2014). Hidden markov models for diagnostics and prognostics of systems under multiple deterioration modes. In *Proceedings of the in european safety and reliability conference-esrel* (pp. 1197–1204).

Li, G., Yang, L., Lee, C.-G., Wang, X., & Rong, M. (2020). A bayesian deep learning rul framework integrating epistemic and aleatoric uncertainties. *IEEE Transactions on Industrial Electronics*, *68*(9), 8829–8841.

Nastos, C., Komninos, P., & Zarouchas, D. (2023). Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods. *Composite Structures*, *311*, 116815.

Nemani, V., Biggio, L., Huan, X., Hu, Z., Fink, O., Tran, A.,

... Hu, C. (2023). Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, *205*, 110796.

Pei, H., Si, X.-S., Hu, C., Li, T., He, C., & Pang, Z. (2022). Bayesian deep-learning-based prognostic model for equipment without label data related to lifetime. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(1), 504–517.

Peng, W., Ye, Z.-S., & Chen, N. (2019). Bayesian deep-learning-based health prognostics toward prognostics uncertainty. *IEEE Transactions on Industrial Electronics*, *67*(3), 2283–2293.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Sankararaman, S. (2015). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, *52*, 228–247.

Sankararaman, S., & Goebel, K. (2013). Why is the remaining useful life prediction uncertain? In *Annual conference of the phm society* (Vol. 5).

Xiahou, T., Wang, F., Liu, Y., & Zhang, Q. (2023). Bayesian dual-input-channel lstm-based prognostics: Toward uncertainty quantification under varying future operations. *IEEE Transactions on Reliability*.

Xie, F.-Y., Hu, Y.-M., Wu, B., & Wang, Y. (2016). A generalized hidden markov model and its applications in recognition of cutting states. *International Journal of Precision Engineering and Manufacturing*, *17*, 1471–1482.

APPENDIX A

Weighted Spread of Uncertainty (WSU)

The weighted spread of uncertainty (WSU) metric is shown in 8. It calculates the area between the confidence intervals while penalizing wider confidence intervals at the end of the lifetime. The penalization is considered because the longer time that has passed, the more information is available. Variable t_i is the time unit, RUL^{upper} is the RUL value of the upper confidence interval and RUL^{lower} is the value of the lower confidence interval.

$$WSU = \sum_{i=1}^{T-1}(t_{i+1} - t_1)\left(\left(\frac{RUL_{i+1}^{upper} + RUL_{i}^{upper}}{2}\right) - \left(\frac{RUL_{i+1}^{lower} + RUL_{i}^{lower}}{2}\right)\right) \tag{8}$$

Monotonicity

The equation for the MI is provided in 9 with $y(t_i)$ as the feature value at time measurement t_i and D as the number of measurements. The results in Figure 4 show that after 20 clusters the monotonicity index converges and remains stable.

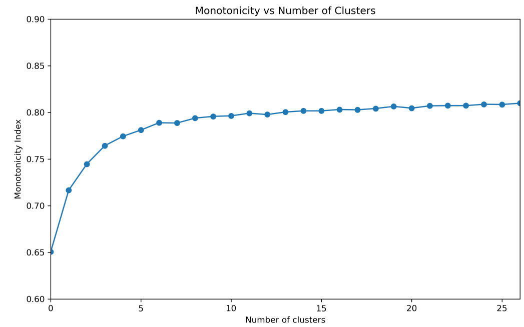

Figure 4. Monotonicity index versus the number of clusters for the sensor 11.

$$\mathrm{MI} = \frac{\sum_i^D \sum_{j=1,j>i}^D (t_j - t_i) sgn(y(t_j) - y(t_i))}{\sum_i^D \sum_{j=1,j>i}^D (t_j - t_i)} \tag{9}$$

Bayesian Inference Criterion

In equation 10 M_i is the candidate model, $y^{(k)}$ is the sensor data from K degradation histories, $Q^{(k)}$ the state sequence for the kth degradation history, H is the number of estimated parameters of model M_i, and n the number of all the samples from the K training sessions. Figure 5 shows the results of the BIC, from which 10 states are proven to be the optimal number.

$$BIC(M_i) = \sum_{k=1}^{K} \log(P(y^{(k)}, Q^{(k)}|M_i)) - w\frac{H_i}{2}\log(n) \tag{10}$$

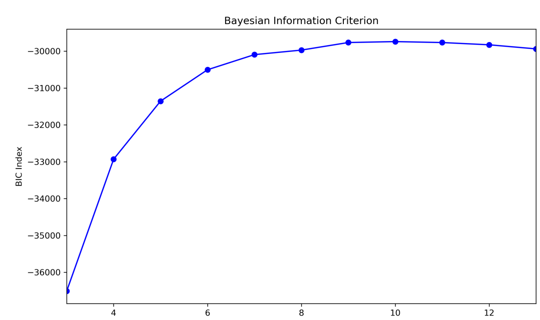

Figure 5. BIC to select the number of optimal states.

After training the HMM, the transition matrix A and the emission matrix B are estimated. The values of the elements of the matrices have been approximated

$$A = \begin{bmatrix}
0.9646 & 0.0354 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.9648 & 0.0352 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.9709 & 0.0291 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.9735 & 0.0265 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.9591 & 0.0409 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9417 & 0.0583 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.9323 & 0.0677 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9281 & 0.0719 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8941 & 0.1059 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

$$B = \begin{bmatrix} 0.01 & 0.07 & 0.16 & 0.26 & 0.24 & 0.16 & 0.05 & 0.01 & 0.001 & 0.039 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0008 & 0.006 & 0.02 & 0.09 & 0.19 & 0.31 & 0.23 & 0.09 & 0.01 & 0.01 & 1.7e{-}12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.07e{-}06 & 0.003 & 0.01 & 0.05 & 0.17 & 0.29 & 0.30 & 0.11 & 0.02 & 0.003 & 0.044 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0002 & 0.001 & 0.006 & 0.04 & 0.14 & 0.30 & 0.28 & 0.17 & 0.03 & 0.005 & 0.0278 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.19e{-}09 & 0.003 & 0.03 & 0.16 & 0.27 & 0.31 & 0.15 & 0.05 & 0.009 & 0.018 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.004 & 0.02 & 0.10 & 0.29 & 0.31 & 0.16 & 0.06 & 0.02 & 0.36 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.002 & 0.02 & 0.07 & 0.21 & 0.27 & 0.28 & 0.09 & 0.03 & 0.028 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.23e{-}07 & 0 & 0.005 & 0.01 & 0.10 & 0.27 & 0.30 & 0.21 & 0.07 & 0.017 & 0.018 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7.16e{-}05 & 0.005 & 0.03 & 0.12 & 0.26 & 0.28 & 0.19 & 0.07 & 0.03 & 0.012 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$