IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 1, 2021, accepted August 19, 2021, date of publication September 6, 2021, date of current version September 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110049

Uncertainty-Aware Prognosis via Deep Gaussian Process

LUCA BIGGIO[1,2], ALEXANDER WIELAND[1], MANUEL ARIAS CHAO[1], IASON KASTANIS[2], AND OLGA FINK[1], (Member, IEEE)

[1]Eidgenössische Technische Hochschule Zürich (ETH Zürich), 8093 Zürich, Switzerland
[2]Centre Suisse d'Électronique et de Microtechnique (CSEM), 2002 Neuchâtel, Switzerland

Corresponding author: Luca Biggio (luca.biggio@inf.ethz.ch)

ABSTRACT The task of predicting how long a certain industrial asset will be able to operate within its nominal specifications is called Remaining Useful Life (RUL) estimation. Efficient methods of performing this task promise to drastically transform the world of industrial maintenance, paving the way for the so-called Industry 4.0 revolution. Given the abundance of data resulting from the advent of the digitalization era, Machine Learning (ML) models are the ideal candidates for tackling the RUL estimation problem in a fully data-driven fashion. However, given the safety-critical nature of maintenance operations on industrial assets, it's crucial that such ML-based methods be designed such that their levels of transparency and reliability are maximized. Modern ML algorithms, however, are often employed as black-box methods, which do not provide any clue regarding the confidence level associated with their output. In this paper, we address this limitation by investigating the performance of a recently proposed class of algorithms, Deep Gaussian Processes, which provide uncertainty estimates associated with their RUL prediction, yet retain the expressive power of modern ML techniques. Contrary to standard approaches to uncertainty quantification, such methods scale favourably with the size of the available datasets, allowing their usage in the ''big data'' setting. We perform a thorough evaluation and comparison of several variants of DGPs applied to RUL predictions. The performance of the algorithms is evaluated on the NASA N-CMAPSS (New Commercial Modular Aero-Propulsion System Simulation) dataset for aircraft engines. The results show that the proposed methods are able to yield very accurate RUL predictions along with sensible uncertainty estimates, providing more reliable solutions for (safety-critical) real-life industrial applications.

INDEX TERMS Deep Gaussian processes, remaining useful life estimation, uncertainty quantification.

I. INTRODUCTION

Recently, Predictive Maintenance (PM) methods have been gaining popularity for many different industrial applications. PM aims at predicting the need for maintenance actions based on the information extracted from condition monitoring data describing the health state of the system. Efficient Remaining Useful Life (RUL) estimation is a key enabler of PM and the application of Machine Learning (ML) and Deep Learning (DL) techniques to RUL prediction tasks has been an active research area over the last several years [1]–[5].

While the majority of model-based prognostics approaches quantify the associated uncertainty, only a few research studies on data-driven RUL prediction have tackled the challenge of quantifying the level of uncertainty associated with the predictions of the proposed techniques. Nevertheless, Uncertainty Quantification (UQ) is crucial in the context of PM because RUL models are used for critical decision-making and, therefore, need to be transparent regarding the level of uncertainty in their predictions. As a result, the deployment of ML techniques in real-world engineering scenarios cannot prescind from the design of reliable algorithms capable of providing a probability density function over RUL predictions instead of simple point estimates.

While Deep Neural Networks (DNN) have delivered their most prominent achievements in the fields of Computer Vision and Natural Language Processing, recent research works have also shown their effectiveness with regard to prognostics [6], [7]. DNNs owe a great part of their success to their substantial representational power and to their capacity for learning sets of hierarchical features across their multilayer architectures directly from raw data. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Sheng Si.

one of the limitations of standard DNN models is that they do not provide an explicit quantification of the uncertainty associated with the predicted RUL. Their effective extension within a Bayesian framework, enabling them to perform UQ without sacrificing their state-of-the-art performance, has recently become an active research area in the ML community [8]–[10], [13]. However, a very limited number of solutions have been proposed for prognostics. Previous works on UQ solutions for purely data-driven prognostics have been based mainly on Relevance Vector Machines [11] and Gaussian Process (GP) regression [12]. GPs, in particular, exhibit good adaptability and the capacity to handle nonlinear, relatively complex regression problems. In addition, compared to standard neural networks, they are based on a well-understood probabilistic formulation. Their flexibility and the availability of open-source software implementations [14] have led to a number of interesting applications in prognostics of engineered systems [15]–[19].

Despite their desirable properties in terms of UQ and their elegant theoretical formulation, GPs are affected by two main limitations hindering their application to real-world datasets. First and foremost, they suffer from cubic complexity to the data size. Specifically, given a dataset of N input-output pairs $(\mathbf{X}, \mathbf{y}) = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, the exact calculation of the marginal likelihood involves the computation of the inverse of the $N \times N$ kernel matrix $\mathbf{K}_{NN} = k(\mathbf{X}, \mathbf{X})$, which results in a computationally prohibitive $\mathcal{O}\left(N^3\right)$ cost. Second, their hypothesis space, i.e. the function space they are able to model, is completely determined by the choice of the kernel function, which might not be complex enough to describe certain types of data. However, over the last decades, a number of efficient solutions to address the aforementioned limitations and refine standard GP models have been proposed.

The contribution of this work is a thorough evaluation of three different enhancements of standard GP models in the context of prognosis: namely, Stochastic Variational Gaussian Processes (SVGPs) [20], [21], Deep Gaussian Processes (DGPs) [22], [23], and Deep Sigma Point Processes (DSPPs)[1] [24], [25]. These methods are also compared to a standard feed-forward neural network (FFNN) and the Monte Carlo Dropout technique (MCD). While both methods are valid baselines to compare the RUL prediction accuracy, only the second one provides the UQ in addition to the actual prediction. The quality of the resulting uncertainty estimates is assessed via two different metrics, namely the α-λ [26] and the probabilistic α-λ [27], both well established in prognosis applications. To the best of our knowledge, these approaches have not yet been applied for RUL prediction and, moreover, the three approaches have not yet been evaluated on a common task. Such an evaluation will provide guidance to decision-makers and a better understanding of both the advantages and limitations of each method. The evaluation is performed on a case study for RUL prediction on aircraft engines using the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) dataset from NASA. Our evaluation results highlight that, while retaining high prediction accuracy, the proposed models are able to successfully perform UQ. The comparisons with the FFNN and MCD baselines show that DGP models yield competitive performance in terms of both accuracy and UQ.

The remainder of the paper is organized as follows. Section II outlines related work on UQ in data-driven prognostics. In Section III, the applied methods are described. In Section IV, the case study is introduced and the experiments are explained. In section VI, the results are presented. Finally, a summary of the work and an outlook are given in Section VII.

II. RELATED WORK

A. DEEP LEARNING TECHIQUES IN PROGOSTICS

Over the last several years, different types of DNNs have been developed for RUL prediction, ranging from relatively complex, fully-connected networks to Convolutional Neural Networks [2], [28], [29] and Recurrent Neural Networks [4], [30], [31]. DL models have shown promising performance in estimating the RUL from sensor data on prognostics benchmark datasets [32], [33] using several different network architectures (see [7] for an extensive review). More sophisticated extensions to the aforementioned standard architectures have also recently been applied to prognosis, including attention mechanisms [34] and capsule neural networks [35]. As opposed to classical ML techniques, DL methods can extract relevant information directly from raw data, with very little need for pre-processing and feature extraction. However, a common drawback shared by the majority of the DL models proposed in the literature is that they do not provide uncertainty estimates associated with their predictions, thus severely limiting the usefulness of their deployment in real-life applications.

B. BAYESIAN DNNs

Equipping DL predictions with meaningful uncertainty estimates is a very active research area in the ML community. In the context of Bayesian DL, the central idea is to replace overconfident DL models with Bayesian neural networks, whereby the weights are treated as random variables. A predictive distribution is then obtained through weight marginalization in such a way that uncertainty in weight space is transferred into probabilistic predictions rather than simple point estimates. A large portion of current research is focused on approximating such predictive distributions, whose exact calculation is typically intractable. Popular methods that follow this approach are, for example, Hamiltonian Monte Carlo [13], Laplace approximation [36], expectation-propagation [37], and variational inference [8], [9], [38]. Among these, MC Dropout [9] has found a broad range of applications due to its simple yet effective rationale: by applying the dropout technique at inference time and

[1] With a slight abuse of notation, we will refer to both deep Gaussian process model variants (DGP and DSPP) as "DGP models."

forward-propagating the input data through the network several times, one can approximate the first two moments of the predictive distribution. More details concerning MC Dropout are reported in an apposite paragraph in Section III. An alternative class of methods for UQ is Deep Ensembles [10], a non-Bayesian technique for estimating uncertainty in DNNs based on training multiple models independently and then aggregating their outputs. These methods provide competitive results but are very computationally expensive.

C. UQ IN DATA-DRIVEN PROGNOSTICS

In light of their flexible, probabilistic, non-parametric framework, GPs have found several applications in prognosis, e.g. nuclear component degradation [15], lithium-ion batteries [16], [18], [19], and bearings [17].

On the other hand, despite the increasing efforts to integrate DNNs with effective UQ techniques, very few of the methods mentioned in the previous subsection have been successfully transferred to prognosis tasks. For instance, ensemble approaches were applied for UQ in prognostics in [39] where, rather than simply training independent models, Bayesian model-averaging was also applied to each model in order to obtain multiple predictions for the elements in the ensemble. UQ based on Bayesian neural networks and variational inference were only recently investigated in [40], [41], with relatively good results in terms of UQ.

The goal of this work is to introduce a new class of methods, DGP models, in an attempt to integrate the benefits of DNNs into the well-understood Bayesian framework of GP regression. We elaborate more on these approaches in the following sections.

III. METHODS

As mentioned above, the main idea of this work is to apply DGP models to the problem of RUL estimation. Our main motivation is that DGP models combine the benefits of DL, via their expressive hierarchical representation, and GPs, in light of their ability to perform UQ. In this section, we provide some details concerning how such a combination can be realized by going through the main mathematical features characterizing each of the investigated methods. A comprehensive analysis of the GP-based techniques used here can be found, for instance, in [24], by which the discussion below is largely inspired. In order to compare our GP-based methods with a strong Bayesian DL baseline, we additionally implement Monte Carlo Dropout (MCD) [9] and apply it to the same RUL benchmark dataset. Our choice of MCD is also motivated by the interpretation provided in [9], which establishes a connection between MCD and the probabilistic GP introduced in [22]. A description of the main principles underlying MCD can be found at the end of this section.

A. STOCHASTIC VARIATIONAL GAUSSIAN PROCESSES - SVGPs

SVGP is a popular inducing point method [42] based on variational inference [43] that enables the application of the GP framework to big datasets. SVGPs introduce a multivariate Normal variational distribution, $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$, over the inducing variables $\mathbf{u}$, where $\mathbf{m}$ and $\mathbf{S}$ are the mean and the covariance, respectively. These variables are obtained from a set of inducing points $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{M}$, lying within the same space as $\mathbf{X}$, through the data generating function f, i.e. $\mathbf{u} = f(\mathbf{Z})$. The parameters of such a distribution can be estimated through the optimization of the ELBO (evidence lower bound), which can be compactly written as follows:

$$\begin{aligned}\mathcal{L}_{\text{svgp}} = \sum_{i=1}^{N} &\left\{ \log \mathcal{N}\left(y_i \mid \mu_{\text{f}}(\mathbf{x}_i), \sigma_{\text{obs}}^2\right) - \frac{\sigma_{\text{f}}(\mathbf{x}_i)^2}{2\sigma_{\text{obs}}^2} \right\} \\ &- \text{KL}(q(\mathbf{u}) \mid p(\mathbf{u}))\end{aligned} \tag{1}$$

where σ_{obs}^2 is the variance of the Normal likelihood $p(\mathbf{y} \mid f)$, KL denotes the Kullback-Leibler divergence, and the two terms μ_{f} and σ_{f} indicate the predictive mean and the latent function variance, respectively, which have the following form:

$$\begin{aligned}\mu_{\text{f}}(\mathbf{x}_i) &= \mathbf{k}_i^T \mathbf{K}_{MM}^{-1} \boldsymbol{m} \\ \sigma_f(\mathbf{x}_i)^2 &= \tilde{\mathbf{K}}_{ii} + \mathbf{k}_i^T \mathbf{K}_{MM}^{-1} \mathbf{S} \mathbf{K}_{MM}^{-1} \mathbf{k}_i\end{aligned} \tag{2}$$

where $\tilde{\mathbf{K}}_{NN} = \mathbf{K}_{NN} - \mathbf{K}_{NM} \mathbf{K}_{MM}^{-1} \mathbf{K}_{MN}$, $\mathbf{k}_i = k(\mathbf{x}_i, \mathbf{Z})$, $\mathbf{K}_{MM} = k(\mathbf{Z}, \mathbf{Z})$ and $\text{K}_{NM} = \mathbf{K}_{MN}^{\text{T}} = k(\mathbf{X}, \mathbf{Z})$.

Given a new test datum $\mathbf{x}_*$ (newly recorded sensor readings), SVGPs yield the following Normal predictive distribution over the corresponding test output y_* (corresponding RUL estimate):

$$p(y_* \mid \mathbf{x}_*) = \mathcal{N}\left(y_* \mid \mu_f(\mathbf{x}_*), \sigma_f(\mathbf{x}_*)^2 + \sigma_{\text{obs}}^2\right) \tag{3}$$

Eq. 3 computes a probability distribution over the algorithm's predictions. This is in stark contrast to standard DL methods applied to prognosis, whose output is limited to a simple point estimate of the RUL.

SVGPs offer two main advantages over standard GPs: first, their formulation involves, at most, the calculation of $\mathbf{K}_{MM}^{-1}$, which results in a significant computational advantage if $M \ll N$. Second, the objective in Eq. 1 is written as a sum over single data points and naturally lends itself to mini-batch training.

B. DEEP GAUSSIAN PROCESSES - DGPs

The classes of functions modelled by standard GP models, including SVGPs, are limited by the expressiveness of the chosen kernel. One possible way to tackle this shortcoming is to use a DNN to automatically learn the kernel from data [44]. However, these approaches often require problem-specific architectures and are prone to overfitting.

Analogous to the transition from shallow to deep networks, DGPs consist of hierarchical compositions of GPs and offer a powerful alternative means of increasing the representational power of "single-layer" GPs. They retain many of the advantages of shallow GPs and introduce a relatively small number of parameters to optimize compared to standard neural network models.

In this work, we apply a variant of DGPs, recently proposed in [23], to overcome some of the drawbacks of the original DGP formulation [22]. This improved model enjoys the same advantages as SVGPs, i.e. it reduces computational complexity by introducing inducing variables for each GP in the hierarchy and supports mini-batch training. More specifically, similarly to SVGPs, the following ELBO[2] is optimized:

$$\mathcal{L}_{\text{dsvi}} = \mathbb{E}_Q \left[\log p \left(\mathbf{y} \mid \mathbf{f}, \sigma^2_{\text{obs}} \right) \right] - \sum \text{KL} \tag{4}$$

where $Q = Q\left(\mathbf{f}, \mathbf{u}_f, \ldots, \mathbf{g}_W, \mathbf{u}_{gW}\right)$ is a variational distribution depending on each of the GP's latent function values and the corresponding inducing variables. Hidden GP latent functions values are referred to as $\mathbf{g}_w$, with $w = 1, .., W$ where W is the number of GPs in the first hidden layer. The KL term in Eq. 4 is of the same form as in the SVGP objective and it is summed over all the inducing variables in the DGP architecture, i.e. $\left\{\mathbf{u}_f, \ldots, \mathbf{u}_{gW}\right\}$. The first term in Eq. 4 can be written as a sum over data points since sampling from Q can be reduced to sampling from $\{q(f_i), \ldots, q(g_{iw})\}$ where the index i ranges over the number of data points N. The resulting method is based on a doubly-stochastic variational inference pipeline since the sampling procedure involves the use of the re-parametrization trick [45] and the minimization of the factorized objective can be performed with mini-batch training. An illustration of a 2-layer DGP architecture implementing the technique introduced in [23] is provided in Fig. 1.

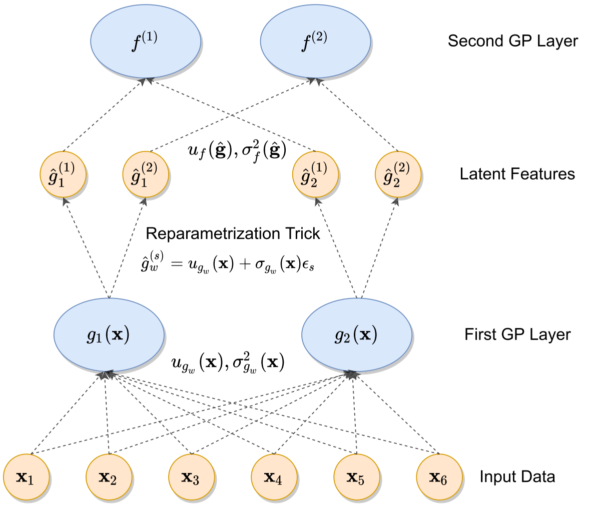

FIGURE 1. 2-layer DGP architecture. The first hidden layer consists of $W = 2$ GPs, taking as input the data x to calculate $\mu_{gw}(\mathbf{x})$ and $\sigma_{gw}(\mathbf{x})$ as prescribed by Eq. 2. The re-parametrization trick is then used to sample from $\mathcal{N}(\mu_{gw}(\mathbf{x}), \sigma_{gw}(\mathbf{x}))$ and obtain the features to be fed into the next GP layer.

The final predictive distribution can be written as a continuous mixture of Normal distributions:

$$p(y_* \mid \mathbf{x}_*) = \mathbb{E}_{\prod_{w=1}^{W} q(g_{*w}|\mathbf{x}_*)} \left[\mathcal{N}\left(y_* \mid \mu_f(\mathbf{g}_*), \sigma_f(\mathbf{g}_*)^2 + \sigma^2_{\text{obs}} \right) \right] \tag{5}$$

where the expectation is analytically intractable but can be approximated via Monte Carlo samples, resulting in a *finite* mixture of Gaussians. As in Eq. 3, Eq. 5 computes a distribution over RUL values y_*, given new sensor readings $\mathbf{x}_*$, thus allowing us to quantify the uncertainty of the model.

C. DEEP SIGMA POINT PROCESSES - DSPPs

Despite their many practical successes, variational inference methods often tend to provide overly confident uncertainty estimates [46].

A recent series of works [24], [25] aimed to address this limitation by reformulating the variational inference scheme at the basis of SVGPs and DGPs. In particular, the authors note an inconsistency between the ELBO (the objective function to be optimized shown in Eq. 1) and the predictive distribution to be used at test time (Eq. 3). More specifically, both quantities are written as functions of two variance terms, one input-dependent, $\sigma_f(x)^2$, and one input-independent, σ^2_{obs}. However, these two contributions appear asymmetrically in Eq. 1. By opportunely modifying the ELBO to fix the aforementioned asymmetry between the objective and the predictive posterior, the authors introduce a new loss function whereby the two variance terms, $\sigma_f(x)^2$ and σ^2_{obs}, are treated consistently. The new objective is given by:

$$\begin{aligned}
\mathcal{L}_{\text{ppgpr}} &= \sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i) - \beta_{\text{reg}} \text{KL}(q(\mathbf{u}) \mid p(\mathbf{u})) \\
&= \sum_{i=1}^{N} \log \mathcal{N}\left(y_i \mid \mu_f(\mathbf{x}_i), \sigma_f(\mathbf{x}_i)^2 + \sigma^2_{\text{obs}} \right) \\
&\quad - \beta_{\text{reg}} \, \text{KL}(q(\mathbf{u}) \mid p(\mathbf{u}))
\end{aligned} \tag{6}$$

where β_{reg} acts as a regularization hyperparameter.

In [25], the authors show that equipping SVGPs with this new objective results in a significant improvement in terms of UQ.

In [24], the DGP framework proposed in [23] is combined with the new loss function introduced in [25].

DSPPs arise from the necessity of overcoming one last theoretical obstacle: the direct application of the objective introduced in [25] to the DGP predictive distribution in Eq. 5 would result in the computation of the logarithm of a continuous mixture of Normal distributions. The approximation of such expectation via Monte Carlo sampling would yield a biased estimator.

To cope with this issue, the authors propose replacing the continuous mixture of Gaussians with a parametric (finite) mixture of Gaussians. This procedure is practically implemented by applying an opportune quadrature rule (e.g. the Gauss-Hermite quadrature rule). To better understand this point, let's rewrite Eq. 5 as:

$$\begin{aligned}
& p(y_i \mid \mathbf{x}_i) \\
&= \int d\mathbf{g}_i \mathcal{N}\left(y_i \mid \mu_f(\mathbf{g}_i), \sigma_f(\mathbf{g}_i)^2 + \sigma^2_{\text{obs}} \right) \prod_{w=1}^{2} q(g_{iw} \mid \mathbf{x}_i)
\end{aligned} \tag{7}$$

[2] Here we consider the case of a 2-layer DGP, for the sake of clarity.

and approximate each distribution inside the product over w as an S-component mixture of delta distributions, i.e.:

$$\prod_{w=1}^{W} q\left(g_{iw} \mid \mathbf{x}_i\right) \to \sum_{s=1}^{S} \omega^{(s)} \prod_{w=1}^{W} \delta\left(g_{iw} - \left(\mu_{g_w}\left(\mathbf{x}_i\right) + \xi_w^{(s)} \sigma_{g_w}\left(\mathbf{x}_i\right)\right)\right) \tag{8}$$

where $\left\{\omega^{(s)}\right\}_{s=1}^{S}$ and $\left\{\xi_w^{(s)}\right\}_{s=1}^{S}$ are sets of learnable parameters. This transformation allows us to replace the continuous mixture of Gaussians in Eq. 7 with a (parametric) finite mixture by exploiting the properties of the Dirac delta function.

D. MONTE CARLO DROPOUT - MCD

The Dropout technique [47] is based on randomly dropping units and the corresponding weights from a neural network at training time. As pointed out in the original paper, this procedure results in sampling an exponential number of different "thinned" networks during training. Predictions are then made by using the entire network, including all units and connections. The resulting technique is straightforward to implement and provides an effective strategy to counter overfitting in DNNs.

MCD provides a Bayesian interpretation of the classic Dropout framework and shows that, by enabling dropout at test time, an approximation of a Bayesian neural network can be obtained and standard point predictions can be paired to meaningful uncertainty estimates. Furthermore, it can be shown that a standard neural network model with dropout applied before every weight layer represents an approximation of the probabilistic Gaussian Process introduced in [22]. This observation motivates the analysis of MCD in the context of our work in light of its close relation to GP-based models.

As already mentioned earlier, in Bayesian inference, given a model with parameters W (in the case of MCD, these will be the weights and biases of the neural network), the final goal is to calculate a predictive posterior distribution $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$ for a new data point $(\mathbf{x}_*, \mathbf{y}_*)$ as

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}_*|\mathbf{x}_*, W) p(W|\mathbf{X}, \mathbf{Y}) \mathrm{d}W. \tag{9}$$

However, the likelihood distribution $p(\mathbf{y}|\mathbf{x}, W)$ is typically a very complicated function of the weights, due to the complex nonlinear mapping implemented by the neural network. This aspect effectively prevents the analytical calculation of the weight posterior and the predictive posterior.

The framework of variational inference aims at tackling this problem by introducing an approximation, $q_\theta(W)$, of the true posterior, $p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$, such that:

$$KL\left(q_\theta(W) \mid (p(W|\mathbf{X}, \mathbf{Y}))\right) = \int q_\theta(W) \log \frac{q_\theta(W)}{p(W|\mathbf{X}, \mathbf{Y})} dW. \tag{10}$$

is minimized for some optimal variational parameters θ^*. It can be easily shown that this optimization problem is equivalent to the maximization of the so-called evidence lower bound (ELBO), $\mathcal{L}_{VI}(\theta)$:

$$\mathcal{L}_{VI}(\theta) = \int q_\theta(W) \log p(\mathbf{Y}|\mathbf{X}, W) \mathrm{d}W - \mathrm{KL}(q_\theta(W)|p(W)). \tag{11}$$

Now, the main novelty introduced in [9] is a specific form of the approximate posterior $q_\theta(W)$ and a resulting unbiased estimator of Eq. 11. More specifically, we consider the following from of $q_\theta(W)$:

$$\begin{aligned} W_i &= M_i \cdot diag([Z_{i,j}]_{j=1}^{K_i}) \\ Z_{i,j} &\sim \mathrm{Bernoulli}(p_i) \quad \forall\, i = 1, \ldots, L;\ j = 1, \ldots, K_{i-1}, \end{aligned}$$

where M_i and p are the variational parameters, L is the number of layers in the network, and K_i is the number of nodes in the i-th layer. The parameter p represents the probability of keeping the input and can be interpreted as the opposite of the classical dropout rate. This choice allows us to obtain the following unbiased estimator of the ELBO:

$$\mathcal{L}_{MCD} = \frac{1}{N} \sum_{i=1}^{N} E\left(y_i, \hat{y}_i\right) + \lambda \sum_{i=1}^{L} \|W_i\|_2^2 \tag{12}$$

where $E(y_i, \hat{y}_i)$ refers to arbitrary loss function (e.g. Mean Squared Error for regression, Softmax for classification). We can now obtain the mean and the variance of an approximation $q_\theta(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$ of the true predicting posterior defined in Eq. 9 as follows:

$$\begin{aligned} \mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})}\left(\mathbf{y}_\star\right) &\approx \frac{1}{T} \sum_{t=1}^{T} f^W\left(\mathbf{x}_\star\right) \\ \mathrm{Var}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})}\left(\mathbf{y}_\star\right) &\approx \tau^{-1}\mathbf{I}_D + \frac{1}{T} \sum_{t=1}^{T} \left(f^W\left(\mathbf{x}_\star\right)\right)^T f^W\left(\mathbf{x}_\star\right) \\ &\quad - \left(\mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})}\left(\mathbf{y}_\star\right)\right)^T \\ &\quad \times \mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})}\left(\mathbf{y}_\star\right) \end{aligned} \tag{13}$$

where f^W is our neural network model parametrized by its weights W, T is the number of samples used for averaging, and τ is the model precision. In practice, the equation above tells us that, in order to determine the mean and the variance of the approximate predictive posterior, it is sufficient to forward-propagate the input through the trained network T times. Since dropout is enabled at testing time, each iteration will result in a different network model.

Note that the variance calculated, as in Eq. 13, contains two terms: the first term models the intrinsic uncertainty, whereas the second term captures the epistemic uncertainty. Since in Eq. 13 the first term does not depend on $\mathbf{x}$, what we are ultimately modelling is homoscedastic noise. In order to make the intrinsic uncertainty term more expressive, we allow τ to depend on $\mathbf{x}$ and we model it by adding an additional output

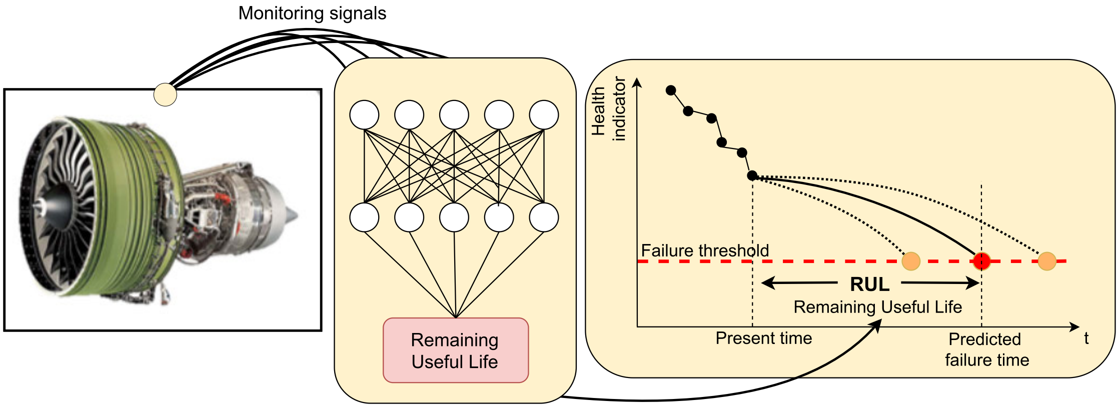

FIGURE 2. Schematic of the RUL estimation task. Given some monitoring signals provided by a set of sensors measuring the health state of a machine (left), a data-driven model (middle) outputs a prediction of the RUL of the machine. Such an estimate (right) represents the number of cycles left for the industrial component to perform until a failure occurs. If the algorithm is designed to perform UQ, it will also output the confidence interval (region between the two orange dots) associated with its mean prediction (red dot).

to the network. This consideration results in the following modified version of the variance expression:

$$\begin{aligned}&\mathrm{Var}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y})}(\mathbf{y}_\star)\\ &\approx \frac{1}{T}\sum_{t=1}^{T}\tau^{-1}(\mathbf{x}_\star) + \left(f^W(\mathbf{x}_\star)\right)^T f^W(\mathbf{x}_\star)\\ &\quad - \left(\mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y})}(\mathbf{y}_\star)\right)^T \mathbb{E}_{q_\theta(\mathbf{y}_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y})}(\mathbf{y}_\star) \qquad (14)\end{aligned}$$

The expression of the variance reported above accounts for the eventual heteroscedasticity of the data noise.

IV. CASE STUDY

The main focus of this work is on the problem of RUL estimation of industrial assets, providing not only the point estimate but also the associated uncertainty of the prediction. We perform the evaluation on the case study of nine turbofan engines that are operated under different conditions. In this framework, the goal is to map a set of high-dimensional time series input data, **x**, describing the health state of the system (e.g. sensor readings), onto the target value, y, representing the remaining number of cycles the system will be able to operate without incurring any failure. A simple schematic representation of the RUL task is illustrated in Fig. 2. By the nature of the problem, predictions of the RUL when the level of degradation of the machine is very low are much more challenging and are subject to a high degree of indeterminacy.

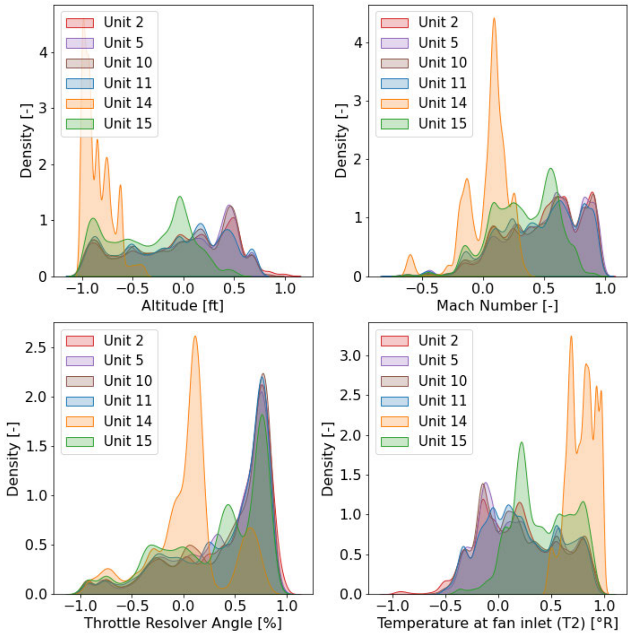

FIGURE 3. Approximate density distributions of the flight envelopes given by recordings of altitude, flight Mach number, throttle resolver angle, and total temperature at the fan inlet for complete trajectories (from start until a failure occurs) of three training units (2, 5, 10) and three test units (11, 14, 15).

A. CASE STUDY OF PREDICTING THE RUL OF TURBOFAN ENGINES

We evaluate and compare the UQ capabilities of the selected ML techniques on a fleet of nine large turbofan engines under real flight conditions (the data is part of the new C-MAPSS dataset) [48]. Concretely, the flight data cover climb, cruise, and descent flight conditions corresponding to different commercial flight routes. Full degradation trajectories of the turbofan engines are available. The degradation trajectories are given in the form of multivariate time-series of sensor readings. Overall, we split the full dataset into six training units (2,5,10,16,18,20) and three test units (11,14,15). Figure 3 shows the distribution of the flight envelopes for a subset of three[3] out of six training units and all three test units. It is worth noting that test unit 14 has an operation distribution that is *significantly* different from the training units. Concretely, it operates at shorter and lower altitude flights compared

[3] The distributions of the remaining three training units are very close to those shown in the figure and have not been shown for the sake of clarity.

to other units. The training dataset contains, thereby, flight profiles that are not fully representative of the test conditions of this unit. Such a discrepancy between training and testing distributions was left on purpose when selecting a subset of the C-MAPSS dataset since a desirable property of any UQ algorithm is to provide lower confidence levels associated with data significantly different from those seen during training. We assess the extent to which this property is satisfied by the proposed methods later in the paper.

Two distinctive failure modes are present in the development dataset: a high pressure turbine (HPT) efficiency degradation and a more complex failure mode that affects the low pressure turbine (LPT) efficiency and flow in combination with the high pressure turbine (HPT) efficiency degradation. Test units (i.e., units 11, 14, and 15) are subjected to the latter complex failure mode. The sampling rate of the data is 0.1 Hz, resulting in a total dataset size of 0.53M samples for model development and 0.12M samples for testing. More details about the generation process can be found in [48].

B. PROBLEM FORMULATION

Given are multivariate time series of condition monitoring sensor readings and physics-inferred process features $X_i = [x_i^{(1)}, \dots, x_i^{(n_i)}]^T \in R^m$ and their corresponding RUL, i.e. $Y_i = [y_i^{(1)}, \dots, y_i^{(n_i)})]^T$, from a fleet of six units (i.e. $N_{train} = 6$). The length of the input feature vector for the *i-th* unit is given by n_i, which differs from unit to unit. The total combined length of the available dataset is $N = \sum_{i=1}^{N_{train}} n_i$ and the dimension of the input features is 41 (i.e. $m = 41$). More compactly, we denote the available dataset as $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N_{train}}$. Given this set-up, the task is to obtain a predictive model that provides a reliable RUL estimate $(\hat{\mathbf{Y}})$ with UQ on a test dataset of $M = 3$ units $\mathcal{D}_{T*} = \{X_{s_j*}\}_{j=1}^{M}$.

V. MODEL ARCHITECTURES AND EVALUATION METRICS

A. MODEL ARCHITECTURES

We compare the performance of our GP-based methods with two baselines, namely MCD and a standard feed-forward neural network (FFNN). Below, we detail all the design choices made in the implementation of each technique.

1) GP MODELS

For the SVGP model, we performed a hyperparameter grid search over the number of inducing points $I \in \{200, 400, 800\}$. We consider the NLL on the validation set as our model-selection metric. The lowest validation NLL is reached with $I = 800$.

The DGP model uses a single hidden layer with a skip-connection enhancing the input, and $W = 4$ hidden GPs. We perform a hyperparameter grid-search over the number of inducing points $I \in \{50, 100, 200\}$. The lowest NLL on the validation set is achieved for $I = 100$.

For the DSPP, we perform a hyperparameter grid search over the number of inducing points $I \in \{50, 100, 200\}$, the width of the hidden layer $W \in \{2, 3\}$, and the number of quadrature sites $Q \in \{5, 8, 10, 15, 20\}$. The lowest NLL on the validation set is reached with $I = 100$, $W = 2$, and $Q = 15$.

2) MCD MODEL

For the MCD approach, we perform a grid search over the hyperparameter space characterized by $L \in \{2, 3, 4, 5\}$ hidden linear layers with $H_f \in \{50, 65, 80, 100, 150, 200\}$ hidden units each. The dropout rate is searched over a log range of 12 possible values within the interval [0.01, 1]. A ReLU function is used after each hidden layer. The lowest NLL on the validation set is achieved for $L = 5$, $H_f = 200$ and $p = 0.46$.

3) FFNN MODEL

For the FFNN model, we perform a grid search over the hyperparameter space characterized by $L \in \{2, 3, 4, 5\}$ hidden layers with $H_f \in \{50, 65, 80, 100, 150, 200\}$ hidden units each. A ReLU function is used after each hidden layer and a constant dropout rate of $p = 0.15$. In this case, we consider the RMSE on the validation set as our evaluation metric. The lowest RMSE value is reached with $L = 5$ and $H_f = 65$.

The batch size is set to 2000 samples using the Adam optimizer with a learning rate of 10^{-3} for all the above models.

B. EVALUATION METRICS

We evaluate the prediction accuracy and uncertainties obtained by the proposed techniques in terms of the standard Root-Mean-Square Error (RMSE) and negative log-likelihood (NLL). In addition, we also incorporate the $\alpha - \lambda$ metric, which is commonly used in prognostics analysis [26] and is defined as:

$$\alpha - \lambda = \begin{cases} 1, & \text{if } (1-\alpha)\lambda^* \leq \lambda_p \leq (1+\alpha)\lambda^* \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where λ^* is the ground truth and λ_p the prediction. Therefore, the $\alpha - \lambda$ metric measures whether the prediction accuracy of an RUL model is within an α % error at specific time instances during the relative lifetime λ of the system.

For an arbitrary chosen accuracy α, the metric can be evaluated and averaged over the whole trajectory with N samples:

$$\overline{\alpha - \lambda} = \frac{1}{N} \sum_{n=0}^{N} (\alpha - \lambda)_n. \tag{16}$$

However, this evaluation metric takes only single-value predictions into account, neglecting the uncertainty associated with them. In order to account for predictive uncertainty, a probabilistic version of the standard $\alpha - \lambda$ is used [27]. Given the variance obtained from the model, we can fit a Gaussian distribution to each output and calculate the probability for a given prediction of being inside the boundary α. For a generic Gaussian distribution $\mathcal{N}(\mu, \sigma)$, we define the

TABLE 1. Comparison of SVGP, DGP, DSPP, MCD, and FFNN in terms of negative log-likelihood (NLL), RMSE, $\alpha-\lambda$, and $P_{\alpha-\lambda}$ on the test data.

Gaussian Processes				
Models	**NLL**	**RMSE**	$\alpha-\lambda$	$\mathbf{P}_{\alpha-\lambda}$
SVGP	3.50	8.70	0.43	0.36
DGP	3.24	7.37	0.49	0.46
DSPP	**3.10**	7.38	**0.56**	**0.53**
Deep Neural Networks				
Models	**NLL**	**RMSE**	$\alpha-\lambda$	$\mathbf{P}_{\alpha-\lambda}$
MCD	4.26	**7.31**	**0.56**	0.48
FFNN	-	7.71	0.55	-

cumulative distribution function as

$$F(x,\mu,\sigma)=\Phi\left(\frac{x-\mu}{\sigma}\right)=\frac{1}{2}\left[1+\operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]. \tag{17}$$

This allows us to define the probabilistic $\alpha-\lambda$ for a single prediction $\mathcal{N}(\mu,\sigma)$ as

$$\mathbf{P}_{\alpha-\lambda}=F((1+\alpha)\lambda^*,\mu,\sigma)-F((1-\alpha)\lambda^*,\mu,\sigma). \tag{18}$$

Again, for an arbitrarily chosen accuracy α, the metric can be evaluated and averaged over the whole trajectory with N samples:

$$\overline{\mathbf{P}_{\alpha-\lambda}}=\frac{1}{N}\sum_{n=0}^{N}P_{(\alpha-\lambda)_n}. \tag{19}$$

For both cases we set the α value equal to 20%, as is commonly done in the literature.

VI. RESULTS

In this section, we apply the methods described above to the N-CMAPPS dataset in order to predict the RUL of the three test units (i.e., units 11, 14, and 15) and quantify the uncertainty of the predictions. We report the results provided by all the methods listed above. All the GP-based methods are equipped with the new objective introduced in [25][4] since we found empirically that, in accord with the results of [25], using the ELBO has a negative impact on the UQ quality for both SVGP and DGPs. All the considered algorithms are implemented using PyTorch [49]. For the GP-based models, we used the open-source library GPyTorch [50].

A. PERFORMANCE ANALYSIS

In this section, we compare the prediction accuracy of the considered models in terms of the probabilistic negative log-likelihood (NLL), RMSE, $\alpha-\lambda$ and $\mathbf{P}_{\alpha-\lambda}$. The results are shown in Tab. 1. The table shows that DSPPs provide the best NLL results, whereas MCD only slightly outperforms DGPs and DSSPs in terms of RMSE. While the performances of DSPPs and MCD agree in terms of $\alpha-\lambda$, DSSPs outperform all the other methods on the $\mathbf{P}_{\alpha-\lambda}$ metric. All the results reported in Tab. 1 are obtained on the hold-out test dataset.

B. UQ ANALYSIS

1) VISUALIZATIONS

In this paragraph, we provide some visualizations to demonstrate the UQ performance of the proposed methods. Since all our GP-based models provide very similar confidence bounds, we report only the results obtained by the DSPP model.

We start our analysis by showing the test prediction error of the considered FFNN model. The results are shown in Fig. 4.

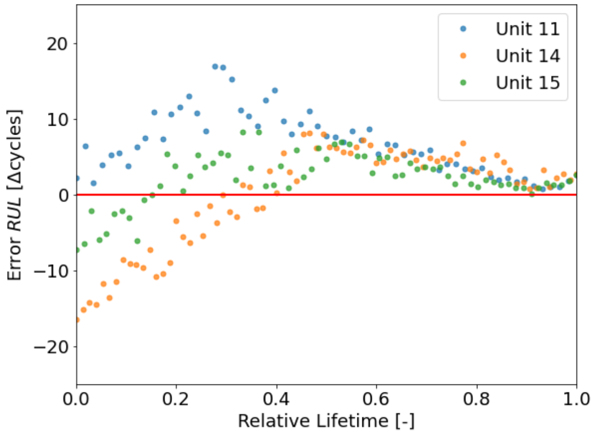

FIGURE 4. Prediction error of the FFNN models as a function of the relative lifetime.

As expected, the predictions tend to align to the ground truth towards the end of the units' lifetime. However, the network is overly confident in its RUL estimates, even when they significantly diverge from the ground truth (first predictions are far removed from the ground truth).

In contrast, as shown in Fig. 5, the predictions provided by MCD (left) and DSPP (right) are supported by meaningful uncertainty estimates. In both cases, the confidence bounds show an important desirable characteristic for RUL models. The values of the predictive variance decrease over time. This is physically meaningful since predictions are much more uncertain when the system is far from the end of its life. As a result, the confidence bounds associated with early operating times are significantly larger than those corresponding to the machine's end of life. Such a property has very important practical implications since it enables the design of risk-aware maintenance strategies.

2) ROBUSTNESS TO A SHIFT BETWEEN TRAINING AND TESTING DISTRIBUTION

As discussed in Section IV, test unit 14 operates at shorter and lower altitude flights compared to the training units. This aspect results in a distributional shift between training and testing distributions, thus challenging the generalization capabilities of the proposed method. In this section, we evaluate whether the obtained UQ models are robust under the aforementioned distributional shift. In particular, we are interested in assessing whether the uncertainty estimates exhibit

[4] In particular, in the case of DGPs, we compute a biased estimator of the continuous mixture of Gaussians obtained by applying Monte Carlo sampling.

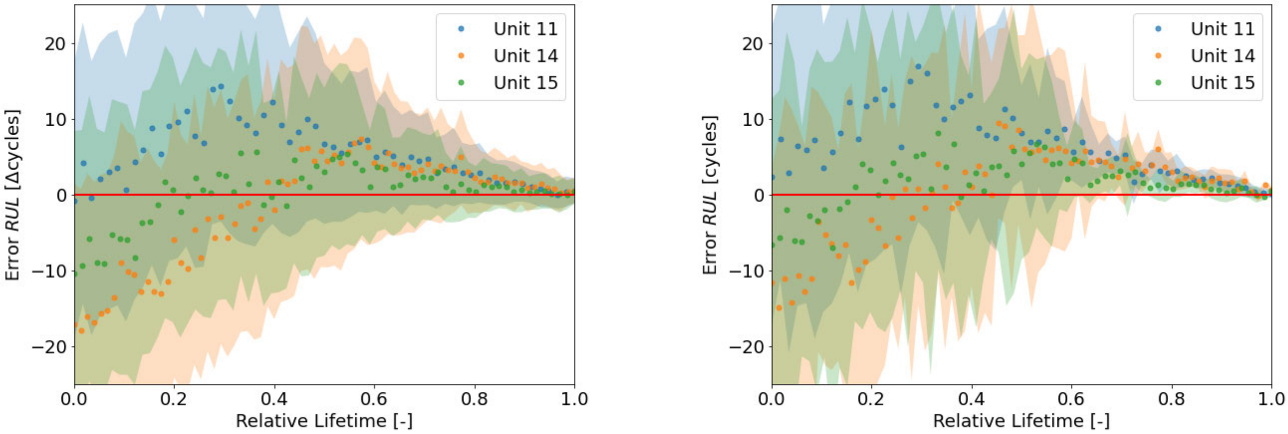

FIGURE 5. Evolution of predictive uncertainty (i.e. ±2σ) provided by MCD (left) and DSPP (right) over the relative time for each test unit, i.e. Unit 11 (blue), Unit 14 (orange), and Unit 15 (green).

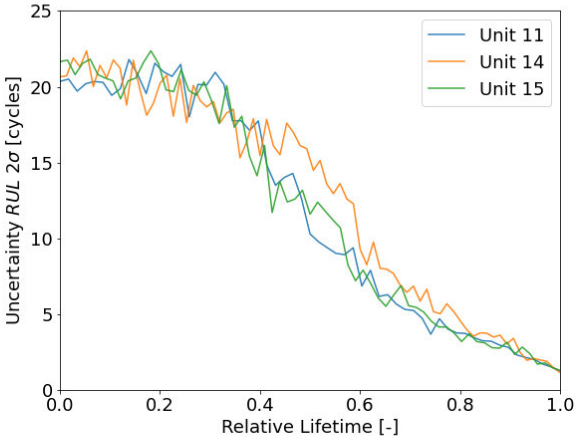

FIGURE 6. Evolution of predictive uncertainty (i.e. ±2σ) provided by the DSPP model over the relative time for each test unit, i.e. Unit 11 (blue), Unit 14 (orange), and Unit 15 (green).

higher uncertainty on inputs that are far away from the training data distribution.

Figure 6 shows the evolution of the predictive uncertainty (i.e. ±2σ) provided by the DSSP model over time for each unit. While at the very first cycles, the level of uncertainty is quite high for all the units (due to the inherent indeterminacy of estimating the RUL when the machines are operating under nominal conditions), the RUL predictions for Unit 14 exhibit greater uncertainty compared to the test units 11 and 15 at later cycles closer to the end of life, when the signs of a fault are increasingly apparent. Therefore, the confidence bounds of the proposed methods reflect another important and desirable characteristic for RUL models: the values of the predictive variance exhibit greater uncertainty on inputs that are far away from training data.

VII. CONCLUSION

In this work, we analyzed a number of methods capable of modelling the uncertainty associated with their predictions. In particular, we focused on the problem of RUL estimation, i.e. predicting the remaining useful lifetime of an industrial asset of interest. In light of the safety-critical nature of this task, UQ is vital in order to enable the deployment of reliable and transparent machine learning algorithms in such real-life industrial applications. The considered methods combine the strengths of neural networks and GPs, merging the expressive power and scalability of neural networks with the probabilistic nature of GPs.

Overall, our results demonstrate that the best performing models are DSPP and MCD: DSSP achieves the highest NLL score while the MCD achieves the best RMSE score. Both of them outperform all other techniques in terms of $\alpha - \lambda$ and $P_{\alpha-\lambda}$. Furthermore, our visualizations show that the confidence bounds provided by the considered models are meaningful: uncertainty decreases as the system approaches the end of life and it is higher for units whose operating conditions differ significantly from those of the training units. These aspects are in strong contrast to the behaviour of a standard deep neural network model, which does not take uncertainty into account in its predictions and solely returns point estimates. Last but not least, contrary to the standard GP models, all the proposed methods are characterized by favourable scaling properties and can be applied to large training datasets.

In the future, we will focus on two main aspects. First, we would like to extend the proposed methods so that they can better capture the temporal correlations present in time-series data. We expect such a modification to have a positive impact on the final performance. However, although this is relatively straightforward for MCD (it amounts to replacing the current fully-connected architecture with a one-dimensional Convolutional Neural Network), it is not a trivial matter with regard to the GP-based models. Second, we would like to investigate more recent Bayesian DNNs and compare them to the methods proposed in this work. We leave these potential research directions to future work.

REFERENCES

[1] B. Saha, K. Goebel, and J. Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," *Trans. Inst. Meas. Control*, vol. 31, nos. 3–4, pp. 293–308, Jun. 2009.

[2] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018.

[3] H.-Z. Huang, H.-K. Wang, Y.-F. Li, L. Zhang, and Z. Liu, "Support vector machine based estimation of remaining useful life: Current research status and future trends," *J. Mech. Sci. Technol.*, vol. 29, no. 1, pp. 151–163, Jan. 2015.

[4] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.

[5] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 11–20, Jan. 2018.

[6] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103678.

[7] L. Biggio and I. Kastanis, "Prognostics and health management of industrial assets: Current progress and road ahead," *Frontiers Artif. Intell.*, vol. 3, pp. 1–24, Nov. 2020.

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 1613–1622.

[9] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd ICML*, vol. 48, 2016, pp. 1050–1059.

[10] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6405–6416.

[11] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001.

[12] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.

[13] R. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. L. Jones, and X. L. Meng, Eds. London, U.K.: Chapman & Hall, 2011, pp. 116–162.

[14] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, Dec. 2010.

[15] P. Baraldi, F. Mangili, and E. Zio, "A prognostics approach to nuclear component degradation modeling based on Gaussian process regression," *Prog. Nucl. Energy*, vol. 78, pp. 141–154, Jan. 2015.

[16] D. Liu, J. Pang, J. Zhou, Y. Peng, and M. Pecht, "Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression," *Microelectron. Rel.*, vol. 53, no. 6, pp. 832–839, 2013.

[17] S. Hong, Z. Zhou, C. Lu, B. Wang, and T. Zhao, "Bearing remaining life prediction using Gaussian process regression with composite kernel functions," *J. Vibroeng.*, vol. 17, no. 2, pp. 695–704, 2015.

[18] L. Li, P. Wang, K.-H. Chao, Y. Zhou, and Y. Xie, "Remaining useful life prediction for lithium-ion batteries based on Gaussian processes mixture," *PLoS ONE*, vol. 11, no. 9, Sep. 2016, Art. no. e0163004.

[19] D. Liu, J. Pang, J. Zhou, and Y. Peng, "Data-driven prognostics for lithium-ion battery based on Gaussian process regression," in *Proc. IEEE Prognostics Syst. Health Manage. Conf. (PHM- Beijing)*, Beijing, China, May 2012, pp. 1–5, doi: 10.1109/PHM.2012.6228848.

[20] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *Proc. Mach. Learn. Res.*, vol. 38, pp. 351–360, Apr. 2015.

[21] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Proc. 29th Conf. Uncertainty Artif. Intell.*, 2013, pp. 282–290.

[22] A. C. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proc. 16th Int. Conf. Artif. Intell. Statist.*, vol. 31, JMLR, 2013.

[23] H. Salimbeni and M. P. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4591–4602.

[24] M. Jankowiak, G. Pleiss, and J. R. Gardner, "Deep sigma point processes," 2020, *arXiv:2002.09112*. [Online]. Available: http://arxiv.org/abs/2002.09112

[25] M. Jankowiak, G. Pleiss, and J. R. Gardner, "Parametric Gaussian process regressors," 2019, *arXiv:1910.07123*. [Online]. Available: http://arxiv.org/abs/1910.07123

[26] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," in *Proc. Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–17.

[27] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *Int. J. Prognostics Health Manage.*, vol. 1, no. 1, pp. 4–23, 2010.

[28] L. Wen, Y. Dong, and L. Gao, "A new ensemble residual convolutional neural network for remaining useful life estimation," *Math. Biosci. Eng.*, vol. 16, no. 2, pp. 862–880, 2019.

[29] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of bearing remaining useful life with deep convolution neural network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018.

[30] J. Chen, H. Jing, Y. Chang, and Q. Liu, "Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process," *Rel. Eng. Syst. Saf.*, vol. 185, pp. 372–382, May 2019.

[31] J. Wu, K. Hu, Y. Cheng, H. Zhu, X. Shao, and Y. Wang, "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network," *ISA Trans.*, vol. 97, pp. 241–250, Feb. 2020, doi: 10.1016/j.isatra.2019.07.004.

[32] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, IEEE, 2008, pp. 1–9.

[33] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, "PRONOSTIA: An experimental platform for bearings accelerated degradation tests," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Jun. 2012, pp. 1–8.

[34] P. R. D. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Rel. Eng. Syst. Saf.*, vol. 195, Mar. 2020, Art. no. 106682.

[35] A. R.-T. Palazuelos, E. L. Droguett, and R. Pascual, "A novel deep capsule neural network for remaining useful life estimation," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 234, no. 1, pp. 151–167, 2020.

[36] H. Ritter, A. Botev, and D. Barber, "A scalable laplace approximation for neural networks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, vol. 6, 2018, pp. 1–15.

[37] J. M. Hernández-Lobato, J. Miguel, and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.

[38] M. Teye, H. Azizpour, and K. Smith, "Bayesian uncertainty estimation for batch normalized deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4907–4916.

[39] Y. Deng, A. D. Bucchianico, and M. Pechenizkiy, "Controlling the accuracy and uncertainty trade-off in RUL prediction with a surrogate Wiener propagation model," *Rel. Eng. Syst. Saf.*, vol. 196, Apr. 2020, Art. no. 106727.

[40] W. Peng, Z.-S. Ye, and N. Chen, "Bayesian deep-learning-based health prognostics toward prognostics uncertainty," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, Mar. 2020.

[41] M. Benker, L. Furtner, T. Semm, and M. F. Zaeh, "Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo," *J. Manuf. Syst.*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520301928, doi: 10.1016/j.jmsy.2020.11.005.

[42] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1–8.

[43] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[44] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 51, A. Gretton and C. C. Robert, Eds. Cadiz, Spain: PMLR, May 2016, pp. 370–378. [Online]. Available: http://proceedings.mlr.press/v51/wilson16.pdf and https://proceedings.mlr.press/v51/wilson16.html

[45] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *Stat*, vol. 1050, p. 1, Apr. 2014.

[46] R. E. Turner and M. Sahani, "Two problems with variational expectation maximisation for time-series models," in *Bayesian Time Series Models*, D. Barber, T. Cemgil, and S. Chiappa, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2011, ch. 5, pp. 109–130.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 14, no. 56, pp. 1929–1958, 2014.

[48] M. A. Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics," *Data*, vol. 6, no. 1, p. 5 2021.

[49] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*. [Online]. Available: http://arxiv.org/abs/1912.01703

[50] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7587–7597.

LUCA BIGGIO received the B.Sc. and M.Sc. degrees in physics and theoretical physics from the University of Genoa, Italy, in 2016 and 2018, respectively, and the M.Phil. degree in machine learning and machine intelligence from the University of Cambridge, U.K., in 2019. He is currently pursuing the Ph.D. degree in computer science with ETH Zürich. His research interests include machine learning, including deep learning for time series analysis and computer vision, reinforcement learning, and Bayesian deep learning.

ALEXANDER WIELAND received the B.Sc. and M.Sc. degrees in chemical and process engineering from ETH Zürich, and the master's degree in machine learning and technical system finishing with a thesis on "Uncertainty Quantification in Remaining Useful Lifetime Estimation with Deep Learning Models" from the Chair of Intelligent Maintenance Systems, ETH Zürich. His research interests include uncertainty-aware diagnostic and prognostic techniques for complex engineered systems.

MANUEL ARIAS CHAO received the M.Sc. degree in thermal power from Cranfield University, U.K., in 2008. He is currently pursuing the Ph.D. degree with the Chair of Intelligent Maintenance Systems, ETH Zürich, Switzerland. He has an industrial experience as a Lead Engineer in system engineering and thermodynamics for gas turbines. He was a Research Associate with the Institute of Data Analysis and Process Design (IDP), Zurich University of Applied Sciences. His research interests include deep learning techniques for diagnostics and prognostics of complex engineered systems.

IASON KASTANIS received the Ph.D. degree in computer science from UCL. He studied mathematics and computer science and specializes in advanced vision and signal processing. He is currently employed at CSEM as an Expert in computer vision and machine learning, where he is leading various projects in the area of industrial quality control and predictive maintenance systems. He is also supervising Ph.D. students in the aforementioned topics in collaboration with ETHZ and actively involved in the implementation of the latest technological advancements in the industry. His research interests include research and development of the latest methods concerning the topic of predictive maintenance systems, and the problems encountered in real-world applications where data is scarce and not curated.

OLGA FINK (Member, IEEE) received the Diploma degree in industrial engineering from Hamburg University of Technology and the Ph.D. degree from ETH Zürich. Before joining the ETH Faculty, she was heading the research group "Smart Maintenance" at Zurich University of Applied Sciences (ZHAW). She is currently a Swiss National Science Foundation (SNSF) Professor of intelligent maintenance systems at ETH Zürich. She is also a Researcher affiliate with Massachusetts Institute of Technology. She has gained valuable industrial experience as a Reliability Engineer for railway rolling stock and as a Reliability and Maintenance Expert for railway systems. Her research interests include intelligent maintenance systems, data-driven, condition-based, predictive maintenance, hybrid approaches fusing physical performance models, deep learning algorithms, deep learning and decision support algorithms for fault detection, and diagnostics of complex industrial assets.