A Bayesian Deep Learning Framework for RUL Prediction Incorporating Uncertainty Quantification and Calibration

Yan-Hui Lin, *Senior Member, IEEE*, and Gang-Hui Li

***Abstract*—In this article, deep learning (DL) has attracted increasing attention for remaining useful life (RUL) prediction. However, most DL-based prognostics methods only provide deterministic RUL values while ignoring the associated epistemic and aleatoric uncertainties. In practice, it is important to know the exact confidence in model predictions for decision making. In this article, a Bayesian deep learning (BDL) framework for RUL prediction incorporating uncertainty quantification and calibration is proposed. First, the epistemic and aleatoric uncertainties, which account for the ignorance about the model and the noise inherent in the observations, respectively, are characterized by integrating both types of uncertainties into a BDL framework. Second, to avoid under- and over-confident predictions, a novel iterative calibration method is proposed to jointly calibrate epistemic, aleatoric, and predictive uncertainties by combining isotonic regression with standard deviation scaling. The effectiveness of the proposed method is demonstrated by the case study of turbofan engines and lithium-ion batteries datasets.**

***Index Terms*—Bayesian deep learning (BDL), epistemic and aleatoric uncertainties, remaining useful life (RUL), uncertainty calibration.**

I. Introduction

Failures may lead to significant maintenance cost and safety hazards. To improve operational readiness and safety, it is essential to predict and manage future risks due to failures. Prognostics and health management (PHM) is an effective technology to achieve this goal. As the core technology of PHM, prognostics is the basis of subsequent decision making in health management, therefore, accurate prediction of the remaining useful life (RUL) is essential [1].

Various RUL prediction methods have been proposed, which can be mainly divided into model-based and data-driven approaches. Model-based methods establish analytical models of underlying degradation mechanisms [2]. However, the complex dynamics of degradation mechanisms are difficult to be accurately characterized. Data-driven methods do not need a full understanding of the degradation mechanisms. Instead, it builds the relevant behavior model based on historical data [3], and the model performance depends on the size and quality of data. Due to the widespread use of sensors, the monitoring techniques of health conditions have been widely adopted, which provides sufficient data for data-driven methods. Deep learning (DL) has attracted increasing attention in the field of RUL prediction because of its excellent ability of handling nonlinear features. Recently, various DL frameworks have been applied to RUL prediction [4], [5]. Xia *et al.* [6] estimated the RUL of bearings by a two-stage automated approach based on deep neural network (DNN), which utilized autoencoder-based DNN to classify the health stages and predicted RUL by a shallow neural network and a smoothing operator. Ren *et al.* [7] proposed an RUL prediction method for lithium-ion batteries based on improved convolution neural network (CNN) and long short-term memory (LSTM) network, which mined deeper information by deep CNN and LSTM and utilized an autoencoder to augment the dimensions of data for more effective training. Ma *et al.* [8] proposed a convolution-based LSTM (CLSTM) by conducting convolutional operation on the state transitions of the LSTM to predict the RUL of rotating machineries.

Although the DL-based RUL prediction methods have achieved remarkable performance, they can only provide point estimates through deterministic neural networks ignoring the related uncertainties. However, in practice, it is important to know if a model is uncertain about its prediction results, so that following decisions have to be made carefully, especially for safety-critical application domains. There are mainly two types of uncertainties in RUL prediction: aleatoric uncertainty and epistemic uncertainty [9]. The former captures noise inherent in the observations, reflecting the influence of unknown or missing information due to, e.g., measurement errors, which cannot be mitigated by collecting more data. The latter accounts for the uncertainty in the model due to lack of knowledge, and can be reduced given enough data [10]. For uncertainty quantification, Bayesian methods are often applied, which utilizes posterior reasoning combining prior information and new observations. As typical Bayesian methods, Kalman filter and particle filter [11] are widely used in RUL prediction. However, they are only

Manuscript received October 18, 2021; revised January 15, 2022 and February 13, 2022; accepted March 2, 2022. Date of publication March 7, 2022; date of current version July 11, 2022. This work was supported by the National Natural Science Foundation of China under Grant 51875016. Paper no. TII-21-4571. *(Corresponding author: Yan-Hui Lin.)*

The authors are with the School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China (e-mail: linyanhui@buaa.edu.cn; liganghui@buaa.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3156965.

Digital Object Identifier 10.1109/TII.2022.3156965

1551-3203 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

applicable to relatively simple systems. In recent years, Bayesian deep learning (BDL), which incorporates Bayesian methods into DL, has gained widespread attention. In this way, the powerful nonlinear processing capability of DL is extended for uncertainty quantification, however, the high computational and time cost also limits its practical application. Fortunately, Gal *et al.* [12] have shown that applying dropout [13], a regularization method, to a DNN could approximate it to a Bayesian deep neural network (BDNN), which is computationally efficient. Accordingly, this framework is applied to RUL prediction. Kim and Liu [14] leveraged BDNN to capture epistemic uncertainty and modeled aleatoric uncertainty as a function of RUL via feed forward NN, and then quantified the uncertainty through Monto Carlo (MC) dropout. Li *et al.* [15] proposed a framework for RUL prediction based on BDL, which captured epistemic uncertainty through conventional dropout and modeled the aleatoric uncertainty by placing a lifetime distribution on the output of BDNN, and quantified the two types of uncertainties separately with MC dropout based on uncertainty decomposition.

In practice, Bayesian uncertainty estimates usually fail to capture the uncertainty accurately due to model misspecification and approximate inference [16]. For instance, a 95% *posteriori* confidence interval usually cannot contain the true outcome 95% of the time. Therefore, calibration is essential for accurately quantifying uncertainty. To the best of our knowledge, none of the existing RUL prediction frameworks with uncertainty quantification has considered uncertainty calibration, which may result in under- and over-confidence in their predictions. The purpose of this article is to propose a complete and feasible RUL prediction framework with calibrated uncertainty quantification. In the proposed framework, predictive uncertainty is analytically decomposed into epistemic and aleatoric uncertainties, epistemic uncertainty is captured through Concrete dropout [17] instead of conventional dropout, and a Gaussian distribution is placed over the model output to capture aleatoric uncertainty. Moreover, the quantified uncertainties are further jointly calibrated. Specifically, an iterative calibration method is proposed by calibrating the predictive uncertainty through isotonic regression [16], and calibrating epistemic and aleatoric uncertainties using standard deviation (STD) scaling [18].

The main contributions of this article can be summarized as follows: first, a general and feasible BDL-based framework with uncertainty quantification is established, where Concrete dropout is applied to capture the epistemic uncertainty in RUL prediction. Compared with the conventional dropout which requires manual tuning of dropout probability, using Concrete dropout can achieve automatic optimization of dropout probability, which greatly relieves the computational burden and leads to more accurate results; and second, uncertainty calibration is integrated into the BDL-based RUL prediction framework, which is the first attempt to the best of our knowledge, besides, a novel iterative calibration method is also proposed to jointly calibrate epistemic, aleatoric and predictive uncertainties by combining isotonic regression with STD scaling.

The rest of this article is organized as follows. Section II introduces the BDL-based RUL prediction framework with uncertainty quantification. Section III describes the proposed methods for jointly calibrating uncertainties. Section IV evaluates the proposed framework on the degradation datasets of turbofan engines and lithium-ion batteries, and demonstrates its effectiveness from two aspects: prediction accuracy and uncertainty calibration quality. Finally, Section IV concludes this article.

II. Uncertainty Quantification

In real-world applications of PHM systems, the signals collected by sensors are inevitably affected by uncontrollable factors such as measurement errors. Besides, the reliabilities of the established RUL prediction models are also restricted by the size of available data. In order to avoid under- and over-confident RUL predictions, the models need to quantify the uncertainties associated with the prediction results instead of only provide deterministic values. This section analyzes the sources of uncertainties in RUL predictions, and the way to decompose these uncertainties. Finally, the RUL prediction framework with uncertainty quantification based on BDL is constructed.

A. Analysis of Uncertainty

There are mainly two types of uncertainties in prognostic tasks. One is aleatoric uncertainty, which captures the noise inherent in observations and cannot be mitigated by collecting more data. The other is epistemic uncertainty reflecting uncertainty in the model due to lack of knowledge, and can be reduced given enough data. Therefore, identifying the sources of uncertainties is crucial for model building and comprehensive understanding of the model outputs.

Let $f^{\omega}(\cdot)$ denote a general regression model parameterized by ω, and$X = \{x_n\}_{n=1}^{N}$, $Y = \{y_n\}_{n=1}^{N}$ denote the available degradation monitoring data and the related RUL labels, respectively. To consider the epistemic uncertainty that arises due to limited amount of data, the model parameters are treated as random variables, i.e., a distribution $p(\omega|X, Y)$ is placed on ω. Similarly, the distribution $p(y|x, \omega)$ is used to capture the aleatoric uncertainty given a new sample x and fixed ω, which directly affects the prediction outcome. By considering the related distributions under a Bayesian framework, the predictive distribution can be written as follows:

$$p\left(y|x, X, Y\right) = \int p\left(y|x, \omega\right) p\left(\omega|X, Y\right) d\omega. \tag{1}$$

As can be seen from (1), predictive uncertainty captured by $p(y|x, X, Y)$ is jointly affected by epistemic and aleatoric uncertainties. In this article, the uncertainty is measured by variance, so that the predictive uncertainty is quantified as predictive variance $\sigma^2_{y|x}(y|x)$, where X and Y are omitted for simplification. In order to obtain the explicit expressions of epistemic and aleatoric uncertainties, we decompose the predictive uncertainty as follows [19]:

$$\sigma^2_{y|x}\left(y|x\right) = \sigma^2_{\omega}\left[E_{y|x,\omega}\left(y|x, \omega\right)\right] + E_{\omega}\left[\sigma^2_{y|x,\omega}\left(y|x, \omega\right)\right] \tag{2}$$

where $\sigma^2_{\omega}[E_{y|x,\omega}(y|x, \omega)]$ and $E_{\omega}[\sigma^2_{y|x,\omega}(y|x, \omega)]$ represent epistemic and aleatoric uncertainties, respectively.

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

B. Uncertainty Inference in Bayesian Deep Learning

As explained earlier, epistemic and aleatoric uncertainties can be captured by the distributions of model parameters and that of the output by eliminating the effects of variability of model parameters, respectively. Instead of using deterministic values, we place probabilistic distributions over model parameters to model epistemic uncertainty. More specifically, we try to capture how much the model parameters vary given current data. Therefore, the posterior distribution $p(\omega|X,Y)$ has to be inferred as follows:

$$p\left(\omega|X,Y\right)=\frac{p\left(Y|X,\omega\right)p\left(\omega\right)}{p\left(Y|X\right)} \tag{3}$$

where $p(\omega)$ is the prior distribution.

However, a large amount of model parameters, nonlinear and nonconjugate prior in BDL bring great challenges to the posterior inference in (3). To cope with this difficulty, variational inference (VI) [20] can be applied to address this intractable distribution by defining an inference distribution $q_\theta(\omega)$ parameterized by θ which is computationally tractable, and minimizing the Kullback–Leibler (KL) divergence between $q_\theta(\omega)$ and $p(\omega|X,Y)$ defined as follows:

$$\begin{aligned} KL\left(q_\theta\left(\omega\right)||p\left(\omega|X,Y\right)\right) &= KL\left(q_\theta\left(\omega\right)||p\left(\omega\right)\right) \\ &\quad -\int q_\theta\left(\omega\right)\log\left(p\left(Y|X,\omega\right)\right)d\omega. \end{aligned} \tag{4}$$

The first term in the RHS of (4) is the KL divergence between $q_\theta(\omega)$ and $p(\omega)$, and the second term represents the expectation of the log-likelihood with respect to $q_\theta(\omega)$. Then, we can approximate posterior distribution by the inference distribution by deriving θ that minimizes the KL divergence between $p(\omega|X,Y)$ and $q_\theta(\omega)$.

However, it requires significant computational and memory cost to compute $KL(q_\theta(\omega)||p(\omega|X,Y))$. To avoid this problem, BDNN is constructed by applying dropout to DNN according to [12]. For a L-layer BDNN with random weight matrix $\omega=\{\boldsymbol{W}_l\}_{l=1}^{L}$ and variational parameters $\theta=\{\boldsymbol{M}_l,p_l\}_{l=1}^{L}$, where $\boldsymbol{M}_l$ is the mean weight matrix of dimensions K_{l+1} by K_l and scalar p_l is the dropout probability, and K_l denotes the number of units, the inference distribution can be expressed as follows:

$$q_\theta\left(\omega\right)=\prod_{l=1}^{L}q_{\theta_l}\left(\boldsymbol{W}_l\right) \tag{5}$$

with $q_{\theta_l}(\boldsymbol{W}_l)=\boldsymbol{M}_l\cdot diag[Bernoulli(1-p_l)^{K_l}]$. Furthermore, the above-mentioned distribution can be expressed as the discrete approximating distribution [21]

$$q_{\theta_l}\left(W_{l,ij}\right)=p_l\delta\left(W_{l,ij}-0\right)+\left(1-p_l\right)\delta\left(W_{l,ij}-M_{l,ij}\right) \tag{6}$$

where $W_{l,ij}$ and $M_{l,ij}$ are the elements at row i and column j of $\boldsymbol{W}_l$ and $\boldsymbol{M}_l$, respectively, and $\delta(\cdot)$ is delta distribution. To avoid manual tuning of dropout probability p_l, Concrete dropout [17] is applied by treating p_l as the model parameter rather than hyper-parameter, which can hence be automatically optimized. As a parameter of inference distribution, p_l may be affected by specific task and the size of dataset. Therefore, Concrete dropout is more rational than conventional dropout.

According to [21], the KL divergence between $p(\omega|X,Y)$ and $q_\theta(\omega)$ can be simplified by introducing a discrete prior $p(\omega)=\prod_{l=1}^{L}p(\boldsymbol{W}_l)$, where $p(W_{l,ij})\propto e^{-\frac{\alpha_l^2}{2}W_{l,ij}^2}$ with length scale α_l defined over a finite space, and the first term in the RHS of (4) can be transformed to

$$\begin{aligned} KL\left(q_\theta\left(\omega\right)||p\left(\omega\right)\right) &= \sum_{l=1}^{L}KL\left(q_{\theta_l}\left(\boldsymbol{W}_l\right)||p\left(\boldsymbol{W}_l\right)\right) \\ &\propto \sum_{l=1}^{L}\left[\frac{\alpha_l^2\left(1-p_l\right)}{2}\|\boldsymbol{M}_l\|^2-K_l\mathrm{H}\left(p_l\right)\right] \end{aligned} \tag{7}$$

with $\mathrm{H}(p_l)=-p_l\log p_l-(1-p_l)\log(1-p_l)$.

The second term in the RHS of (4), which is the expectation of the log-likelihood on the inference distribution, can be expressed as follows:

$$\begin{aligned} &\int q_\theta\left(\omega\right)\log\left(p\left(Y|X,\omega\right)\right)d\omega \\ &\quad=E_{q_\theta(\omega)}\left(\log\left(p\left(Y|X,\omega\right)\right)\right) \\ &\quad\approx\frac{1}{S}\sum_{i=1}^{S}\log\left(p\left(Y|X,\omega_i\right)\right),\omega_i\sim q_\theta\left(\omega\right) \\ &\quad=\frac{1}{S}\sum_{i=1}^{S}\sum_{j=1}^{N}\log\left(p\left(y_j|x_j,\omega_i\right)\right),\omega_i\sim q_\theta\left(\omega\right). \end{aligned} \tag{8}$$

Accordingly, the loss function based on Concrete dropout can be written as follows:

$$\begin{aligned} Loss = &-\frac{1}{N}\sum_{i=1}^{N}\log\left(p\left(y_i|x_i,\omega\right)\right) \\ &+\frac{1}{N}\sum_{l=1}^{L}\left[\frac{\alpha_l^2\left(1-p_l\right)}{2}\|\boldsymbol{M}_l\|^2-K_l\mathrm{H}\left(p_l\right)\right] \end{aligned} \tag{9}$$

with $\omega\sim q_\theta(\omega)$.

Notably, the reparameterization trick cannot be directly used to optimize p_l due to the discrete distribution $Bernoulli(1-p_l)$. Therefore, it is necessary to replace the discrete Bernoulli distribution with its continuous relaxation, which allows us to reparameterize the distribution as follows [17]:

$$z=sigmoid\left[\frac{1}{t}\left(\log\left(1-p_l\right)-\log p_l+\log u-\log\left(1-u\right)\right)\right] \tag{10}$$

with uniform $u\sim Unif(0,1)$, and t is the temperature parameter, which controls the mass on the boundaries of interval [0,1].

In the earlier, the epistemic uncertainty is captured by approximating $p(\omega|X,Y)$ with $q_\theta(\omega)$. To capture aleatoric uncertainty, we place a probabilistic distribution on the model output. In regression tasks, the model output can usually be modeled as $p(y|x,\omega)=N(\mu(x,\omega),\sigma^2)$ with observation noise σ^2. In

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: MC Dropout for Uncertainty Inference.

Input: test data x_* , model $f^{\omega}(\cdot)$, dropout probability $\{p_l\}_{l=1}^{L}$, iterations B
Output: predictive mean μ^*, uncertainties $\eta^*_{pred}, \eta^*_{alea}, \eta^*_{epis}$

1: **For** $b = 1$ to B, **do:**
2: **For** $l = 1$ to L, **do**:
3: Take a sample ω_b^l with dropout probability p_l;
4: **End for**
5: Obtain (μ_b, σ_b^2) from model $f^{\omega_b}(\cdot)$ with $\omega_b = \{\omega_b^l\}_{l=1}^{L}$;
6: **End for**
7: Compute predictive mean $\mu^* = \frac{1}{B}\sum_{b=1}^{B}\mu_b$;
8: Compute epistemic uncertainty $\eta^*_{epis} = \frac{1}{B}\sum_{b=1}^{B}(\mu_b - \mu^*)^2$;
9: Compute aleatoric uncertainty $\eta^*_{alea} = \frac{1}{B}\sum_{b=1}^{B}\sigma_b^2$;
10: Compute predictive uncertainty $\eta^*_{pred} = \eta^*_{alea} + \eta^*_{epis}$.

real-world prognostic systems, σ^2 is data-dependent rather than constant for each input, i.e., the aleatoric uncertainty considered in our work is heteroscedastic instead of homoscedastic. Note that the modeling of observation noise is also restricted by the size of data, which will also be affected by epistemic uncertainty. Therefore, the aleatoric uncertainty can be modeled as $\sigma^2 = \sigma^2(x, \omega)$. And (9) can be further expressed as follows:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2\sigma^2(x_i,\omega)}\|y_i - \mu(x_i,\omega)\|^2 + \frac{1}{2}\log\sigma^2(x_i,\omega)\right] + \frac{1}{N}\sum_{l=1}^{L}\left[\frac{\alpha_l^2(1-p_l)}{2}\|\boldsymbol{M}_l\|^2 - K_l\mathrm{H}(p_l)\right] \tag{11}$$

with $\omega \sim q_\theta(\omega)$.

In order to quantify the uncertainties of test data x_* given the trained model $f^{\omega}(\cdot)$, i.e., epistemic uncertainty, aleatoric uncertainty, and predictive uncertainty, the MC dropout and uncertainty decomposition in (2) are applied as shown in Algorithm 1.

III. Uncertainty Calibration

In practice, the BDL-based models may lead to inaccurate uncertainty estimates due to model misspecification and approximate inference. Therefore, the obtained uncertainties should be calibrated to achieve accurate uncertainty quantification. In this section, the epistemic, aleatoric, and predictive uncertainties are jointly calibrated.

A. Calibration of Predictive Uncertainty

For a regression task, given a calibration set $\{x_t, y_t\}_{t=1}^{T}$, the model outputs a cumulative probability distribution (CDF) F_t associated with x_t targeting y, and it is well-calibrated if the empirical CDF matches the predicted CDF when the dataset size goes to infinity as follows:

$$\lim_{T\to\infty}\sum_{t=1}^{T}\mathrm{I}\{F_t(y_t) \le p\}/T \to p \text{ for all } p \in [0,1] \tag{12}$$

where $\mathrm{I}\{\cdot\}$ is the indicator function.

To calibrate the regression model, an auxiliary model $R : [0,1] \to [0,1]$ is built to estimate the related empirical probability for each predicted probability. Then, the p quantile of F_t can be adjusted to the $R(p)$ quantile, such that $R \circ F_t$ is calibrated. Specifically, R is fitted on the calibration dataset $\{F_t(y_t), \hat{p}(F_t(y_t))\}_{t=1}^{T}$ with

$$\hat{p}(p) = \sum_{t=1}^{T}\mathrm{I}\{F_t(y_t) \le p\}/T\ . \tag{13}$$

However, the analytical expression of R is difficult to obtain. According to [16], for an increasing function associated with data x $F_x(y) : \mathrm{y} \to \Phi$ where $y \in \mathrm{y}$ and $\Phi \subseteq \mathbb{R}$ that defines a feature space that correlates with the confidence in model predictions, the model R can be approximated as $F_\Phi : \Phi \to [0,1]$, which is the CDF of the feature space Φ. In this article, the feature associated with data x_t is defined as $F_{x_t}(y_t) = (y_t - \mu^t)/\sqrt{\eta^t_{pred}}$, where μ^t is the predictive mean and η^t_{pred} is the predictive variance. By constructing another calibration dataset $S = \{F_{x_t}(y_t), \hat{p}(F_{x_t}(y_t))\}_{t=1}^{T}$ with

$$\hat{p}(\phi) = \sum_{t=1}^{T}\mathrm{I}\{F_{x_t}(y_t) \le \phi\}/T \tag{14}$$

where $\phi \in \Phi$. And F_Φ can be obtained by fitting it on S through isotonic regression.

Then, the CDF of the calibrated model output associated with x_*, F_*^c, can be expressed as follows:

$$F_*^c(y) = F_\Phi\left((y - \mu^*)\Big/\sqrt{\eta^*_{pred}}\right)\ . \tag{15}$$

As a result, the calibrated predictive uncertainty $\eta^{*,c}_{pred}$ can be obtained as follows:

$$\eta^{*,c}_{pred} = \lambda_p \eta^*_{pred} \tag{16}$$

where λ_p is the calibration coefficient of predictive uncertainty, which is the variance of F_Φ , and can be estimated as follows:

$$\lambda_p = 2\left(\int_0^{+\infty}\phi\cdot(1 - F_\Phi(\phi))\,d\phi - \int_{-\infty}^{0}\phi\cdot F_\Phi(\phi)\,d\phi\right) - \left(\int_0^{+\infty}(1 - F_\Phi(\phi))\,d\phi - \int_{-\infty}^{0}F_\Phi(\phi)\,d\phi\right)^2 \tag{17}$$

with numerical integration.

B. Joint Calibration of Epistemic, Aleatoric, and Predictive Uncertainties

Calibration of only predictive uncertainty cannot lead to calibrated epistemic and aleatoric uncertainties. Note that accurate quantification of epistemic and aleatoric uncertainties is crucial to measure the model reliability and prediction confidence, and plays a significant role in health management. In order to jointly

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

calibrate the two types of uncertainties, the probabilistic distributions related with all uncertainties that need to be calibrated, and the required values for calibration are analyzed as follows.

1) Predictive uncertainty $\sigma^2_{y|x}(y|x)$: $p(y|x)$ needs to be calibrated based on the value of y given data x.
2) Epistemic uncertainty $\sigma^2_{\omega}[E_{y|x,\omega}(y|x,\omega)]$: $p(E_{y|x,\omega}(y|x,\omega))$ needs to be calibrated based on the mean value of y under different model parameters ω given data x.
3) Aleatoric uncertainty $E_{\omega}[\sigma^2_{y|x,\omega}(y|x,\omega)]$: $p(y|x,\omega)$ needs to be calibrated based on the value of y under different model parameters ω given data x.

Except for the predictive uncertainty for which the required values for calibration are available, the epistemic and aleatoric uncertainties cannot be calibrated directly due to lack of required values. To solve this issue, we formulate the predictive distribution as follows:

$$\begin{aligned} p\left(y|x,X,Y\right) &= \int p\left(y|x,\omega\right)p\left(\omega|X,Y\right)d\omega \\ &= E_{\omega|X,Y}\left(p\left(y|\mu\left(x,\omega\right),\sigma\left(x,\omega\right)\right)\right) \\ &\approx \frac{1}{B}\sum_{b=1}^{B} p\left(y|\mu\left(x,\omega_b\right),\sigma\left(x,\omega_b\right)\right)\omega_b \sim q_\theta\left(\omega\right) \\ &= \frac{1}{B}\sum_{b=1}^{B}\frac{1}{\sqrt{2\pi}\sigma\left(x,\omega_b\right)} \\ &\quad \times \exp\left(-\frac{\left(y-\mu\left(x,\omega_b\right)\right)^2}{2\sigma^2\left(x,\omega_b\right)}\right). \end{aligned} \tag{18}$$

As can be seen in (18), the predictive distribution can be approximated as a Gaussian mixture distribution. According to Algorithm 1, $\frac{1}{B}\sum_{b=1}^{B}\left(\mu(x,\omega_b)-\frac{1}{B}\sum_{b=1}^{B}\mu(x,\omega_b)\right)^2$ and $\frac{1}{B}\sum_{b=1}^{B}\sigma^2(x,\omega_b)$ measure the epistemic and aleatoric uncertainties associated with data x, respectively. Therefore, epistemic and aleatoric uncertainties can be calibrated by optimizing $\mu(x,\omega)$ and expectation of $\sigma^2(x,\omega)$, respectively, which can be achieved through STD scaling [18]. To calibrate a probabilistic distribution, STD scaling directly multiplies its variance by a scaling factor to scale the represented uncertainty. In this way, $\mu(x,\omega)$ can be calibrated as follows:

$$\sigma^2_{\omega}\left(\mu^c\left(x,\omega\right)\right) = \lambda_e\sigma^2_{\omega}\left(\mu\left(x,\omega\right)\right) \tag{19}$$

where λ_e is the scaling factor associated with epistemic uncertainty. $\mu^c(x,\omega)$ has the same mean but different variance as $\mu(x,\omega)$. The expectation of $\sigma^2(x,\omega)$ can be calibrated by calibrating $p(y|x,\omega)=N(\mu(x,\omega),\sigma^2(x,\omega))$ in (18) as follows:

$$p^c\left(y|x,\omega\right) = N\left(\mu^c\left(x,\omega\right),\lambda_a\sigma^2\left(x,\omega\right)\right) \tag{20}$$

where λ_a is the scaling factor associated with aleatoric uncertainty. Therefore, the calibrated uncertainties based on STD scaling can be obtained as follows:

$$\eta^c_{pred} = \lambda_p\cdot\eta_{pred}\eta^c_{alea} = \lambda_a\cdot\eta_{alea}\eta^c_{epis} = \lambda_e\cdot\eta_{epis} \tag{21}$$

where λ_a and λ_e can be described as calibration coefficients of aleatoric uncertainty and epistemic uncertainty, respectively.

As $\eta^c_{pred}=\eta^c_{alea}+\eta^c_{epis}$, by obtaining λ_p and η^c_{pred} according to (17), we can first obtain η^c_{alea} by estimating λ_a , and directly obtain η^c_{epis} and$\lambda_e$based on uncertainty decomposition.

Besides, the predictive distribution in the LHS of (18) can be calibrated as follows:

$$\begin{aligned} p^c\left(y|x,X,Y\right) &= \frac{1}{B}\sum_{b=1}^{B}\frac{1}{\sqrt{2\pi\lambda_a}\sigma\left(x,\omega_b\right)} \\ &\quad \exp\times\left(-\frac{\left(y-\mu^c\left(x,\omega_b\right)\right)^2}{2\lambda_a\sigma^2\left(x,\omega_b\right)}\right) \end{aligned} \tag{22}$$

with $\omega_b\sim q_\theta(\omega)$and $\mu^c(x,\omega_b)\sim\mu^c(x,\omega)$. However,$\mu^c(x,\omega_b)$ cannot be directly sampled because the explicit expression of $\mu^c(x,\omega)$ is unavailable, and has to be inferred from $\mu(x,\omega_b)$ instead. By assuming that the calibration of $\mu(x,\omega)$ results in scaling the distance of its sample$\mu(x,\omega_b)$ from the predictive mean $\frac{1}{B}\sum_{b=1}^{B}\mu(x,\omega_b)$, the sample after calibration $\mu^c(x,\omega_b)$ in (22) can be obtained as follows:

$$\begin{aligned} \mu^c\left(x,\omega_b\right) &= \sqrt{\lambda_e}\left(\mu\left(x,\omega_b\right)-\frac{1}{B}\sum_{b=1}^{B}\mu\left(x,\omega_b\right)\right) \\ &\quad +\frac{1}{B}\sum_{b=1}^{B}\mu\left(x,\omega_b\right) \end{aligned} \tag{23}$$

where $\lambda_e=(\lambda_p\cdot\eta_{pred}-\lambda_a\cdot\eta_{alea})/\eta_{epis}$.

Once λ_a is estimated, the previous calibrated distributions and uncertainties can be obtained, of which the maximum likelihood estimate (MLE) on calibration set $\{x_t,y_t\}_{t=1}^{T}$ can be calculated with log-likelihood as follows:

$$\begin{aligned} LL\left(\lambda_a\right) &= \sum_{t=1}^{T}\log\left(\frac{1}{B}\sum_{b=1}^{B}\frac{1}{\sqrt{2\pi\lambda_a}\sigma\left(x_t,\omega_b\right)}\right. \\ &\quad \left.\times\exp\left(-\frac{\left(y-\mu^c\left(x_t,\omega_b\right)\right)^2}{2\lambda_a\sigma^2\left(x_t,\omega_b\right)}\right)\right) \end{aligned} \tag{24}$$

with $\omega_b\sim q_\theta(\omega)$.

To further improve the uncertainty calibration quality, we propose to fine-tune the model parameters in the last layer related to $\mu(x,\omega)$ on original training set by updating the loss function based on the calibrated aleatoric uncertainty as follows:

$$\begin{aligned} Loss^c &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2\lambda_a\sigma^2\left(x_i,\omega\right)}\left\|y_i-\mu\left(x_i,\omega\right)\right\|^2\right. \\ &\quad \left.+\frac{1}{2}\log\lambda_a\sigma^2\left(x_i,\omega\right)\right] \\ &\quad +\frac{1}{N}\sum_{l=1}^{L}\left[\frac{\alpha_l^2\left(1-p_l\right)}{2}\left\|\boldsymbol{M}_l\right\|^2-K_l\mathrm{H}\left(p_l\right)\right]. \end{aligned} \tag{25}$$

Then, the new predictive mean, epistemic, aleatoric, and predictive uncertainties of the fine-tuned model can be obtained with the same procedures shown in Section II, which can be further recalibrated as previously described. The fine-tuning

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

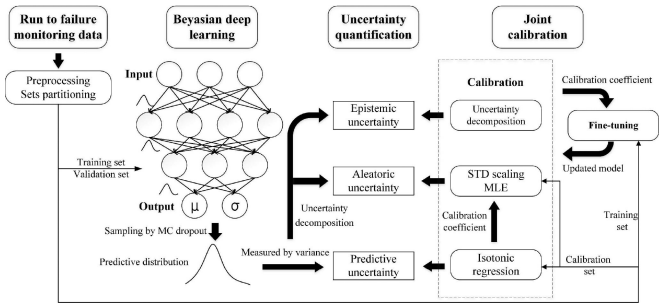

Fig. 1. Flowchart of model training and uncertainty calibration.

Algorithm 2: Joint Calibration of Uncertainties.

Input: test data x_*, calibration set, training set, model $f^{\omega}(\cdot)$, criterion ε

Output: new predictive mean μ^*, calibrated uncertainties $\eta^{*,c}_{pred}, \eta^{*,c}_{alea}, \eta^{*,c}_{epis}$

1: Obtain λ_p according to Eq. (18) on calibration set with $f^{\omega}(\cdot)$;
2: Obtainλ_a by MLE with λ_p on calibration set with $f^{\omega}(\cdot)$;
3: Fine-tune the model $f^{\omega}(\cdot)$ with λ_a on training set;
4: Recalibrate and update λ_p and λ_a with the same methods in step 1 and step 2;
5: Save the fine-tuned model $f^{\omega}(\cdot)$ and calibration coefficients λ_p and λ_a if the change of λ_a is smaller than the criterion ε, otherwise return to step 3;
6: Obtain new predictive mean μ^* , predictive uncertainty η^*_{pred} and aleatoric uncertainty η^*_{alea} with the fine-tuned model $f^{\omega}(\cdot)$ given x_* by Algorithm 1;
7: Compute calibrated predictive uncertainty $\eta^{*,c}_{pred} = \lambda_p \eta^*_{pred}$;
8: Compute calibrated aleatoric uncertainty $\eta^{*,c}_{alea} = \lambda_a \eta^*_{alea}$;
9: Compute calibrated epistemic uncertainty $\eta^{*,c}_{epis} = \eta^{*,c}_{pred} - \eta^{*,c}_{alea}$.

and recalibration steps can be iterated until the change of λ_a between two successive iterations is smaller than the predefined criterion ε. In the end, the epistemic, aleatoric, and predictive uncertainties are jointly calibrated since the model parameters and the calibration coefficients remain consistent. The detailed algorithm for joint calibration of the three types of uncertainties is shown in Algorithm 2. And the flowchart of model training and uncertainty calibration is shown in Fig. 1.

IV. Case Study

The proposed method is applied to two degradation datasets of turbofan engines and lithium-ion batteries, to demonstrate its effectiveness Section IV-A provides an overview of the dataset. Section IV-B presents the model setup and training information. Section IV-C shows the prediction results and evaluates the performance from two aspects: prediction accuracy and uncertainty calibration quality.

TABLE I
Overview of Turbofan Engines Dataset

Dataset description	Sub-dataset			
	FD001	FD002	FD003	FD004
Training engine number	100	260	100	249
Testing engine number	100	259	100	248
Operational condition	1	6	1	6
Fault mode	1	1	2	2

A. Degradation Dataset Description

1) Turbofan Engines Dataset: Turbofan engines dataset is generated by Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) program [22], which is widely applied for RUL prediction, and consists of four subdatasets: FD001, FD002, FD003, and FD004. Each subdataset contains a training set of run-to-failure multivariate time series and a test set of truncated multivariate time series, the overview of them is summarized in Table I. Each time series corresponds to one turbofan engine, which is composed of 21 sensor signals and 3 operational condition measurements. The degradation data can be considered as collected from a fleet of engines of the same type, where each engine is with different degrees of initial wear and subjects to manufacturing variation, which is unknown to users.

Among all of 21 sensor signals, some remain constant during the entire monitoring cycles, therefore, these sensor signals are excluded. Besides, K-means clustering is applied to reduce the dimension of operational condition measurements from 3 to 1 for simplification. As a result, a 16-dimensional input is obtained which contains 14 sensor signals, one operational condition measurement, and the cycle time.

In addition, the normalization technique is utilized to convert the input data into normalized scale

$$x^{i,j}_{norm} = \frac{x^{i,j} - x^{i,j}_{\min}}{x^{i,j}_{\max} - x^{i,j}_{\min}} \tag{26}$$

where $x^{i,j}_{norm}$ is the normalized data of the *j*th sensor signal under the *i*th operational condition. And $x^{i,j}_{\max}, x^{i,j}_{\min}$ are the maximum and minimum values of the *j*th sensor signal under the *i*th operational condition among all training engines, respectively.

2) Lithium-Ion Batteries Dataset: At present, the research of renewable energy sources represented by batteries has received extensive attention [23], [24], and the proposed method is applied on the commercial lithium-ion batteries dataset to demonstrate its effectiveness. This dataset consists of degradation data collected from 124 lithium-ion phosphate (LFP)/graphite cells rather than simulation system. These batteries, manufactured by A123 Systems (APR18650M1A), were cycled under fast-charging conditions in horizontal cylindrical fixtures on a 48-channel Arbin LBT potentiostat in a forced convection temperature chamber set to 30°C. And these batteries in this dataset are charged with a one-step or two-step fast-charging policy with a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V [25].

In this article, degradation data of 43 batteries in batch 2 are considered, and 20% of them are randomly chosen as test

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

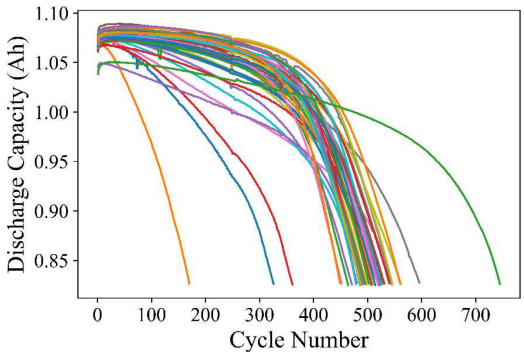

Fig. 2. Discharge capacity by removing the outliers.

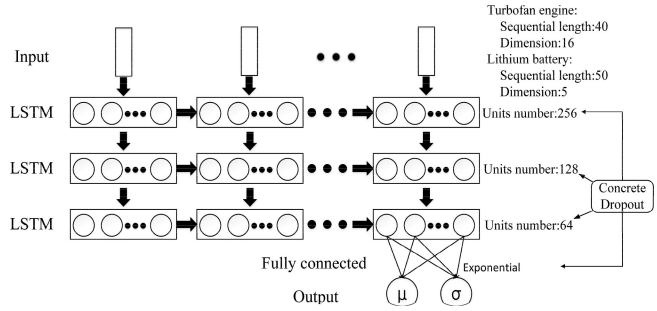

Fig. 3. Structure of BDL for RUL prediction.

set. Each battery is cycled under fast charging until failure. The degradation data of each battery is 8-D containing charging time, cycle time, internal resistance, discharge capacity, charge capacity, average temperature, maximum temperature, and minimum temperature. The discharge capacity by removing the outliers can effectively reflect the degradation trends of lithium-ion batteries, as shown in Fig. 2. And the battery cycle life is defined as the number of cycles when 80% of maximum discharge capacity is reached. By excluding the redundant information, the selected data include internal resistance, discharge capacity, average temperature, charging time and cycle time. Similar to turbofan engines, normalization technique is applied to map the raw signals to be within the range of [0, 1].

B. Model Setup and Training

LSTM architecture, one widely applied variant of recurrent neural network, is adopted for RUL prediction to capture the long-term temporal dependencies in the degradation data. The prediction model is constructed by stacking multiple LSTM layers to obtain stronger learning ability. The latest output of the last LSTM layer is combined with a fully connected layer, which contains two nodes that output the predictive mean and predictive STD, respectively. The exponential activation function is chosen for the node outputting predictive STD. According to the experiments, 3 LSTM layers are selected for the degradation dataset. To apply approximate Bayesian inference to recurrent layers, Variational dropout [26] is utilized when implement Concrete dropout, i.e., the dropout mask at each time step is consistent for each recurrent layer. The structure of whole model is shown in Fig. 3.

Adam algorithm is adopted to train the constructed model. The hyper-parameters, i.e., learning rate, batch size, epoch, etc., are determined through a trail-and-error strategy, for which the combinations of various hyper-parameters are tried through grid search, and the optimal combination is chosen via training and validation, as shown in Table II.

C. Results and Discussions

The performance of the proposed method is evaluated from prediction accuracy and uncertainty calibration quality.

TABLE II
HYPER-PARAMETERS FOR PROPOSED MODEL

Hyper-parameter	Learning rate	Epoch	Batch size	Length scale	Initial dropout probability
Engines	0.001	50	300	0.1	0.2
Batteries	0.0005	60	200	0.1	0.2

1) Prediction Accuracy: To evaluate the prediction accuracy, we employ the commonly used metrics: mean absolute error (MAE) and root mean square error (RMSE). Besides, to assess whether prediction results can offer good decision support for health management, two other metrics: mean absolute percentage error (MAPE) and Score, are also applied. The former emphasizes, the prediction accuracy when the failure is impending, while the latter penalizes late predictions more than early predictions. Given a test set $\{x_q, y_q\}_{q=1}^{Q}$, MAPE is defined as follows [27]:

$$MAPE = \frac{1}{Q}\sum_{q=1}^{Q}\left|\frac{100\Delta_q}{y_q}\right| \tag{27}$$

where $\Delta_q = \mu^q - y_q$ with predictive mean μ^q associated with x_q. Note that MAPE differentiates prediction errors in different stages of system life and puts forward a higher requirement for prediction accuracy when approaching system failure. As late predictions lead to the delay of maintenance activities, which may lead to tremendous economic loss and safety hazards due to failures, especially for safety-critical domains, they are more serious than early predictions. The Score is defined as follows [14]:

$$Score = \sum_{q=1}^{Q} s\left(q\right), s\left(q\right) = \begin{cases} e^{-\frac{\Delta_q}{13}} - 1, \Delta_q < 0 \\ e^{\frac{\Delta_q}{10}} - 1, \Delta_q \geq 0 \end{cases}. \tag{28}$$

The proposed method is compared with the following benchmark methods in prediction accuracy.

1) Conventional LSTM, which directly outputs predicted RUL.
2) Heteroscedastic Bayesian LSTM, which incorporates conventional dropout to capture epistemic uncertainty, and places a Gaussian distribution over the output to capture aleatoric uncertainty, and the uncertainties are quantified by MC dropout [15].

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PREDICTION ACCURACY OF DIFFERENT METHODS FOR RUL PREDICTION

Dataset		Metric	Proposed method	LSTM	Heteroscedastic Bayesian LSTM	Homoscedastic Bayesian LSTM	Bayesian VAE-LSTM
Turbofan engines	FD001	RMSE	**18.6**	20.9	20.8	19.1	20.4
		MAE	12.8	14.1	13.9	**12.6**	13.8
		MAPE	**15.4**	19.5	18.3	16.6	20.7
		Score	**2774.1**	5225.7	3800.2	2832.1	4379.5
	FD002	RMSE	**22.9**	24.3	23.0	23.7	25.6
		MAE	16.1	17.3	**15.1**	17.2	17.7
		MAPE	22.0	25.8	**20.1**	25.6	26.6
		Score	7734.7	**5882.2**	5926.7	8131.4	14711.8
	FD003	RMSE	**27.9**	35.9	30.0	28.2	37.2
		MAE	**17.2**	23.8	19.9	19.2	24.2
		MAPE	**19.6**	34.5	23.8	26.6	38.2
		Score	**19990.6**	986483.7	24889.3	20784.3	959587.1
	FD004	RMSE	28.1	33.7	**27.7**	30.9	34.1
		MAE	**19.5**	23.1	19.8	23.1	25.6
		MAPE	**20.8**	35.3	27.1	42.5	42.9
		Score	53295.6	283019.8	28583.9	**26043.3**	64199.9
Lithium-ion batteries		RMSE	**15.2**	15.8	20.7	22.4	22.4
		MAE	11.9	**11.4**	17.4	17.5	17.5
		MAPE	**13.2**	18.8	35.2	47.8	47.8
		Score	**6336.1**	7757.6	14521.4	22856.1	22856.1

3) Homoscedastic Bayesian LSTM, which utilizes Bayesian LSTM similar to the method (2) to capture epistemic uncertainty, and assumes constant observation noise for each input [28].
4) Bayesian VAE-LSTM, which combines variational autoencoder (VAE) with LSTM, just as the framework in [29], and applies conventional dropout to both of them to capture epistemic uncertainty. In the first phase, Bayesian VAE is used to extract the features of degradation data, and the epistemic and aleatoric uncertainties associated with features can also be obtained. In the second phase, the features subject to uncertainties are input into Bayesian LSTM to predict the RUL and quantify the related epistemic and aleatoric uncertainties.

To demonstrate the effectiveness of the proposed method, the comparison with method 1) demonstrates the benefits of uncertainty quantification in RUL prediction; comparison with method 2) illustrates the improvement in prediction accuracy by applying Concrete dropout rather than conventional dropout to capture epistemic uncertainty; comparison with method 3) shows the effectiveness of incorporating heteroscedastic aleatoric uncertainty into prediction model; comparison with method 4) shows the superiority of the proposed method for RUL prediction with uncertainty quantification.

Table III summarizes the prediction results of different methods. For the turbofan engines with single failure mode (FD001, FD002), method 1) shows comparable performance with the proposed method. However, it behaves poorly on the datasets with multiple failure modes (FD003, FD004). This shows that uncertainty quantification can improve the prediction accuracy when run-to-failure series show different behaviors, which can also be observed for the lithium-ion batteries dataset, of which the major degradation data share the same failure mode.

For method 2), of which the dropout probability needs to be manually tuned for each subdataset, it performs well on FD002 and FD004 but poorly on other subdatasets, which demonstrates that inappropriate dropout probability has a significant influence on prediction accuracy. On the contrary, the proposed method can obtain satisfactory results for all subdatasets, because Concrete dropout optimizes the dropout probability automatically and adaptively for each subdataset, and therefore, is more effective and can achieve better results in general. In addition, method 3) shows relatively poor performance in certain datasets compared to the proposed method and method 2), which indicates the observation noise may not be constant and incorporating heteroscedastic aleatoric uncertainty into prediction model is more effective.

Finally, the proposed method outperforms method 4) because the latter neglects the RUL labels when extracting the features and the associated uncertainties of degradation data by Bayesian VAE in the first phase. Moreover, it inputs features subject to epistemic and aleatoric uncertainties into Bayesian LSTM for RUL prediction. In this way, the predictive uncertainty of RUL cannot be decomposed into epistemic and aleatoric uncertainties, because it is influenced jointly by the epistemic and aleatoric uncertainties of extracted features, which may lead to inaccurate uncertainty quantification and predictions. Therefore, accurately quantifying the epistemic and aleatoric uncertainty is the precondition for improving prediction accuracy.

2) Uncertainty Calibration Quality: The calibration plot for regression [16] is adopted to qualitatively evaluate whether the uncertainty is well-calibrated, and this plot is drawn with $\{p_m, \hat{p}_m\}_{m=1}^{M}$ where $p_m \in [0, 1]$ is the confidence level and $\hat{p}_m$ is the corresponding empirical frequency as follows:

$$\hat{p}_m = \sum_{q=1}^{Q} \mathrm{I}\left\{F_q^c\left(y_q\right) \le p_m\right\}/Q \tag{29}$$

where F_q^c is the CDF of calibrated model output associated with x_q. For well-calibrated uncertainty, the confidence levels can match the probabilities that the labels fall into the confidence

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

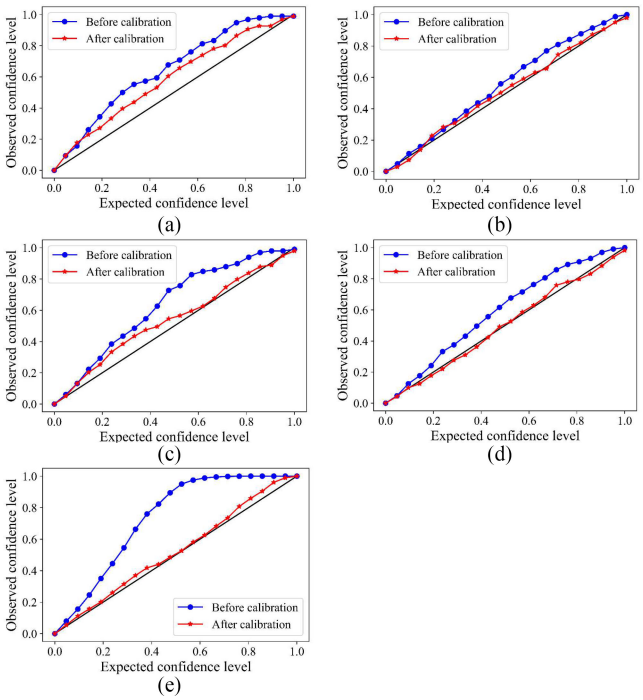

Fig. 4. Calibration plots before and after calibration. (a) FD003, (b) FD004 and (c) Lithium-ion batteries dataset.

intervals, i.e., the empirical frequency is equal to the corresponding confidence level. In the calibration plot, the closer the line segment is to the straight line with a slope of 1, the better the calibration is. Besides, to quantitatively evaluate the calibration plot, expected calibration error (ECE) [30] is adopted as follows:

$$ECE = \sum_{m=1}^{M} (\hat{p}_m - p_m)^2 \ . \tag{30}$$

To evaluate the heteroscedasticity of the calibrated predictive uncertainty, STDs coefficient of variation (Cv) and Sharpness [18] are adopted as follows:

$$Cv = \frac{\sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q}\left(\sqrt{\eta_{pred}^{q}} - \frac{1}{Q}\sum_{q=1}^{Q}\sqrt{\eta_{pred}^{q}}\right)^2}}{\frac{1}{Q}\sum_{q=1}^{Q}\sqrt{\eta_{pred}^{q}}} \tag{31}$$

$$Sharpness = \frac{1}{Q}\sum_{q=1}^{Q}\eta_{pred}^{q} \tag{32}$$

where η_{pred}^{q} is predictive uncertainty associated with x_q. Cv measures the dispersion of predictive uncertainty, and Sharpness reflects the concentration of the predictive distribution. If the predictive uncertainty is well-heteroscedastic, its Cv will be high and Sharpness will be low. Note that the above-mentioned two metrics are meaningful on the basis of the well-calibrated uncertainty. Therefore, we propose using the ECE as the primary calibration evaluation metric and Cv and Sharpness as the secondary metrics.

TABLE IV
QUALITY OF UNCERTAINTY QUANTIFICATION BEFORE CALIBRATION (IN PARENTHESES) AND AFTER CALIBRATION

Dataset		ECE	Cv	Sharpness
Turbofan engines	FD001	0.20(0.45)	0.74(0.67)	600.56(720.94)
	FD002	0.017(0.116)	0.81(0.64)	592.82(958.78)
	FD003	0.09(0.47)	1.79(1.09)	691.80(1344.02)
	FD004	0.004(0.23)	0.80(0.54)	692.80(1520.53)
Lithium-ion batteries		0.02(1.51)	0.56(0.27)	728.63(2917.27)

Fig. 4 shows the calibration plots of RUL predictions of the proposed model before and after calibration, which demonstrates the effectiveness of calibration. Table IV shows the uncertainty quantification quality before and after calibration of the proposed method on each subdataset. As can be seen, the calibrated uncertainty possesses lower ECE and Sharpness, and higher Cv, which indicates that calibration can effectively improve the accuracy of quantified uncertainty. Besides, according to the Sharpness in Table IV, the predictive uncertainty without calibration may lead to under-confident RUL predictions.

Fig. 5 shows the prediction results of the proposed method with 0.9 confidence level (left column), together with the calibrated epistemic and aleatoric uncertainties (right column) on FD003, FD004, and Lithium-ion batteries dataset. It can be observed that the predictive uncertainty generally increases with the increase of RUL, and there is a positive correlation between aleatoric uncertainty and RUL. This is because the effects of noise inherent in observations and the operational and environmental factors will accumulate over functioning, so that longer RUL associates with higher aleatoric uncertainty. On the contrary, the epistemic uncertainty generally does not correlate with RUL, because it captures the uncertainty in the model and depends on the number of training samples. It can be seen that the epistemic uncertainties with smaller RUL are relatively lower because the associated training samples are more in whole subdataset, and the epistemic uncertainties can be reduced with enough training samples as expected.

In the end, the epistemic and aleatoric uncertainties of in-distribution and out-of-distribution (OOD) test samples compared with training samples are analyzed. The OOD test samples are from different distributions than the training samples, and can be the degradation data with new failure modes or those that the model has not been trained with. For the turbofan engines dataset, the degradation data related with high-pressure compressor failure in FD001 are selected as training and in-distribution test samples, and those related with Fan failure in FD003 are selected as OOD test samples. For the lithium-ion batteries dataset, the degradation data with lifetime between 400 and 600 cycles are selected as training and in-distribution test samples, and the others are selected as OOD test samples. The histograms of epistemic and aleatoric uncertainties related to different types of samples are shown in Fig. 6. It can be observed that epistemic and aleatoric of training samples and those of in-distribution test samples are approximately the same as these two types of samples are from the same distributions.

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

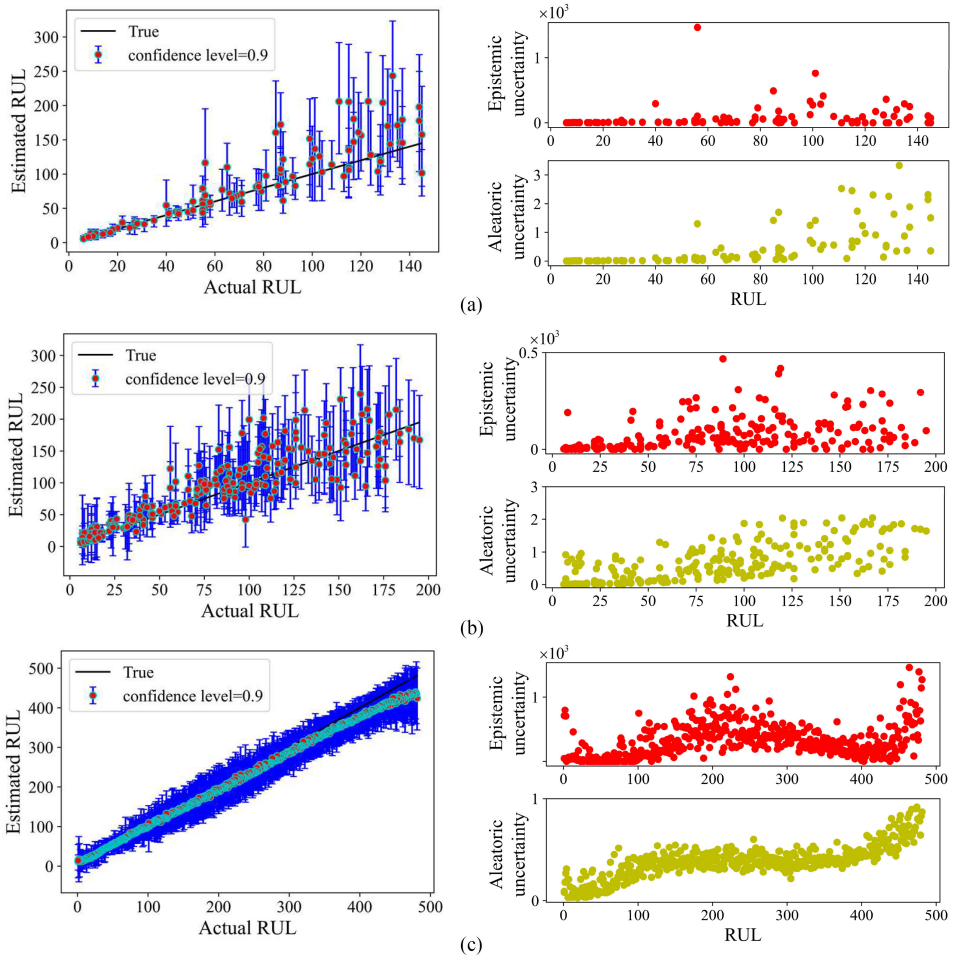

Fig. 5. Uncertainty quantification after calibration on (a) FD003, (b) FD004, and (c) Lithium-ion batteries dataset.

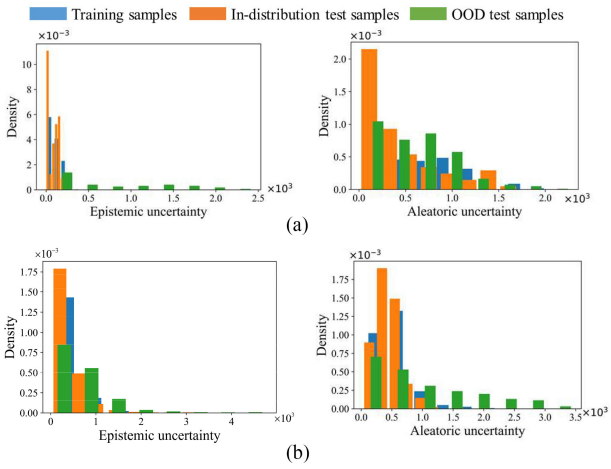

Fig. 6. Histogram of the frequency distribution of uncertainties with different types of samples. (a) Turbofan engines. (b) Lithium-ion batteries.

For the OOD test samples, their epistemic uncertainties increase significantly, but their aleatoric uncertainties are similar to those of in-distribution test samples. Therefore, the epistemic uncertainty can serve as measure to detect whether the input data are OOD, and the corresponding prediction results are undependable.

V. Conclusion

In this article, we proposed a novel prognostics framework to achieve RUL prediction in consideration of both uncertainty quantification and calibration. Concrete dropout was applied to capture the epistemic uncertainty, and aleatoric uncertainty was modeled by placing a Gaussian distribution on the model output. On the other hand, a fine-tuning and recalibration algorithm was proposed to jointly calibrate epistemic, aleatoric, and predictive uncertainties by iterative optimizations. The effectiveness of the proposed framework was validated on turbofan engines and lithium-ion batteries degradation datasets, the results indicated the excellent performance of the proposed method. According to the experimental results, the proposed method can achieve accurate and calibrated prediction results, which were effective for subsequent decision making.

We plan to investigate how to incorporate spatio-temporal dependencies in the proposed framework to construct the Bayesian neural networks for addressing the tightly coupled problems and incorporate uncertainty quantification into physical and causal models to extend the applicability of the proposed framework.

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] K. T. Huynh, I. T. Castro, A. Barros, and C. Bérenguer, "On the use of mean residual life as a condition index for condition-based maintenance decision-making," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 44, no. 7, pp. 877–893, Dec. 2014.

[2] L. Placca and R. Kouta, "Fault tree analysis for PEM fuel cell degradation process modelling," *Int. J. Hydrogen Energy*, vol. 36, no. 19, pp. 12393–12405, 2011.

[3] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Prognostics of PEM fuel cell in a particle filtering framework," *Int. J. Hydrogen Energy*, vol. 39, no. 1, pp. 481–494, 2014.

[4] R.-N. Liu, B.-Y. Yang, and A. G. Hauptmann, "Simultaneous bearing fault recognition and remaining useful life prediction using joint-loss convolutional neural network," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 87–96, May 2020.

[5] P. Ding *et al.*, "Useful life prediction based on wavelet packet decomposition and two-dimensional convolutional neural network for lithium-ion batteries," *Renewable Sustain. Energy Rev.*, vol. 148, Sep. 2021, Art. no. 111287.

[6] M. Xia, T. Li, T.-X. Shu, J.-F. Wan, C. W. D. Silva, and Z.-R. Wang, "A two-stage approach for the remaining useful life prediction of bearings using deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3703–3711, Sep. 2019.

[7] L. Ren, J.-B. Dong, X.-K. Wang, Z.-H. Meng, L. Zhao, and M. J. Deen, "A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3478–3487, Jul. 2021.

[8] M. Ma and Z. Mao, "Deep-convolution-based LSTM network for remaining useful life prediction," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1658–1667, May 2021.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[10] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, Mar. 2021.

[11] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Particle filter-based prognostics: Review, discussion and perspectives," *Mech. Syst. Signal Process.*, vol. 72/73, pp. 2–31, 2016.

[12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[13] 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. 15, no. 6, pp. 1929–1958, 2014.

[14] M. Kim and K.-B. Liu, "A Bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics," *IISE Trans.*, vol. 53, no. 3, pp. 326–340, 2020.

[15] G.-Y. Li, L. Yang, C.-G. Lee, X.-H. Wang, and M.-Z. Rong, "A Bayesian deep learning RUL framework integrating epistemic and aleatoric uncertainties," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8829–8841, Jul. 2021.

[16] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2796–2804.

[17] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 3581–3590.

[18] D. Levi, L. Gispan, N. Giladi, and E. Fetaya, "Evaluating and calibrating uncertainty prediction in regression tasks," 2019, *arXiv:1905.11659v3*.

[19] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1184–1193.

[20] 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.

[21] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Dept. Eng., Cambridge Univ., Cambridge, U.K., 2016.

[22] 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.*, 2008, pp. 1–9.

[23] M. Bagheri *et al.*, "Renewable energy sources and battery forecasting effects in smart power system performance," *Energies*, vol. 12, no. 3, pp. 373–390, 2019.

[24] M. Bagheri, V. Nurmanova, O. Abedinia, M. S. Naderi, N. Ghadimi, and M. S. Naderi, "Impacts of renewable energy sources by battery forecasting on smart power systems," in *Proc. IEEE Int. Conf. Environ. Elect. Eng.*, 2018, pp. 1–6.

[25] K. A. Severson *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[26] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.

[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] 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, Apr. 2020.

[29] L. Ren, Y.-Q. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *J. Manuf. Syst.*, vol. 48, pp. 71–77, 2018.

[30] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

Yan-Hui Lin (Senior Member, IEEE) received the Ph.D. degree in industrial science and technology from the Université Paris-Saclay, Paris, France, in 2016.

He was with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, as a Senior Research Associate. He is currently an Associate Professor of System Engineering with the School of Reliability and Systems Engineering, Beihang University, Beijing, China. His current research interests include degradation modeling, reliability assessment and prognostic, and health management.

Gang-Hui Li received the bachelor's degree in quality and reliability engineering in 2020 from Beihang University, Beijing, China, where he is currently working toward the master's degree with the School of Reliability and Systems Engineering.

His research interests include deep learning for remaining useful life prediction.

Authorized licensed use limited to: Chongqing Jiaotong University. Downloaded on November 02,2022 at 11:55:43 UTC from IEEE Xplore. Restrictions apply.