Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2025) 16-30

ECCOMAS
Proceedia

UNCECOMP 2025
6th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15-18 June 2025

AN EMPIRICAL EVALUATION OF INTERVAL NEURAL NETWORKS FOR UNCERTAINTY QUANTIFICATION ON REGRESSION TASKS

Kaizheng Wang[1,3,4]* **Ghifari Adam Faza[2,3,4]***
Keivan Shariatmadar[2,3,4] **David Moens[2,3]** **Hans Hallez[1,4]**

[1]DistriNet, Department of Computer Science, KU Leuven, Belgium
[2]LMSD, Department of Mechanical Engineering, KU Leuven, Belgium
[3]Flanders Make@KU Leuven, Belgium [4]M-Group, Leuven, Belgium
{kaizheng.wang, adam.faza, keivan.shariatmadar, david.moens, hans.hallez}@kuleuven.be

Abstract. *Uncertainty quantification in deep neural networks has become increasingly important, as effective uncertainty quantification can significantly enhance the robustness and reliability of learning systems, particularly in safety-critical applications. Probabilistic frameworks, such as Bayesian neural networks, are widely regarded as effective approaches. However, despite their popularity and demonstrated success in various applications, these methods have some key limitations: they require the specification of a prior distribution, which forces practitioners to make assumptions about its form. An alternative body of research has increasingly focused on interval neural networks (INNs) in regression tasks, which employ deterministic intervals to model and estimate the uncertainty of the predictions or the weights of the neural networks. Compared to traditional probabilistic methods, INNs require fewer probabilistic assumptions while offering theoretical guarantees of reliability and robustness. INNs also provide a more intuitive interpretation by representing uncertainty through the range of possible values, from minimum to maximum. Moreover, certain INN models featuring interval-formed weights can address interval-based input data, making them especially useful in specific applications. Nevertheless, current evaluations of INNs are generally restricted to synthetic or simple regression datasets. In this research, we comprehensively assess distinct INN models for continuous value prediction (i.e., regression) in real-world engineering applications. Our evaluation mainly encompasses three test cases: a standard regression, a realistic remaining useful lifetime (RUL) prediction task, and the RUL task in cases of interval-valued data.*

Keywords: uncertainty quantification interval neural networks, evaluation on regression

* Equal contribution.

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2025.
doi: 10.7712/120225.12337.21076

1 INTRODUCTION

In recent years, uncertainty quantificatio (UQ) for neural network predictions has garnered growing interest. Effective UQ in regression tasks enables engineers to make more informed and safer decisions when utilizing engineering surrogate models [1–3], particularly in high-stakes and safety-critical applications.

Concerning regression tasks, Nix et al. [4] were among the firs to introduce a neural network architecture with two output nodes to quantify uncertainty, one to generate predictions and the other to estimate data noise. However, such neural networks do not account for parameter uncertainty, as they maintain point estimates for these parameters. Consequently, they fail to fully quantify the uncertainty associated with network predictions [5]. To address this limitation, Bayesian neural networks (BNNs), which model parameters as probability distributions, have emerged as a preferred approach [6–9]. Although BNNs are actively applied to different tasks [10, 11], there are still challenges, e.g., the huge computational complexity during training and inference [12]; also the user- or case-dependency to design reasonable priors [13]. An alternative to BNNs, interval neural networks (INNs) have gained increasing attention due to their ability to UQ without relying heavily on probabilistic and distributional assumptions [14, 15]. Research on INNs can be categorized into two main approaches. The firs type of approach, named light interval neural networks (light-INNs) in our work, involves INNs to generate deterministic interval predictions while maintaining fi ed-point estimates for network weights and biases. In addition, ensemble techniques are also employed to account for parameter uncertainty [14, 16–18]. The second type of approach, called full interval neural networks (full-INNs) in this study, involves representing network parameters as intervals and predicting intervals through interval arithmetic during the propagation process [19–22]. Despite their simple probabilistic assumptions and considerable potential, INNs have yet to undergo rigorous evaluation on modern deep neural architectures in complex real-world scenarios, particularly in comparison to Bayesian techniques.

In addition to conventional regression tasks, the engineering domain has increasingly emphasized the need for robust UQ methods in regression with interval-valued data (IVD), of which features are represented as intervals and embed measurement uncertainty and variability [23–25]. For example, in scenarios where sensor measurements or experimental observations are subject to inherent noise or imprecision, UQ frameworks designed for IVD can significantl enhance the reliability of downstream analyzes by explicitly propagating uncertainty through models [26]. Among existing neural network-based UQ methodologies, full-INNs distinguish themselves due to their unique compatibility with interval arithmetic. However, current research on full-INNs for regression tasks [15, 18, 27] remains restricted to shallow architectures, primarily limited to multilayer perceptrons (MLP) with few hidden layers. Although a recent approach, termed interval batch normalization (IBN), has demonstrated the feasibility of integrating full-INNs with ResNet50 architecture [28] for classificatio tasks, the potential of modern, deep full-INN frameworks for real-world engineering regression remains to be investigated.

To address the aforementioned gaps in evaluating INNs for regression tasks, we systematically assessed the UQ capabilities of INNs and benchmarked them against BNNs. This evaluation spans three distinct case studies: a one-dimensional analytical function, a realistic remaining useful lifetime (RUL) prediction task, and the RUL task in cases of IVD.

The remainder of this paper is organized as follows: Section 2 introduces the preliminary neural network approach for uncertainty quantificatio under our evaluation. Section 3 provides a detailed description of our empirical evaluation experiments. Finally, Section 4 summarizes

our finding and outlines directions for future work.

2 NEURAL NETWORKS FOR UNCERTAINTY QUANTIFICATION

In this section, we introduce three distinct approaches for uncertainty quantificatio in regression tasks: Bayesian neural networks in Section 2.1, light interval neural networks that feature fix- alued weights and generate prediction intervals in Section 2.2, and full interval neural networks that apply interval-valued weights in Section 2.3.

2.1 Bayesian neural networks

Bayesian neural networks (BNNs) have emerged as a prominentframework for uncertainty quantification where neural network parameters are modeled as probability distributions. After assigning prior distributions, BNNs leverage Bayesian inference to derive posterior distributions from the training data. To mitigate the significan computational complexity associated with BNNs, extensive research has been conducted, focusing on various approximation techniques [6, 12, 29–31]. Among these, Monte Carlo dropout (MC Dropout) [6] and deep ensembles [8] have widely served as Bayesian baselines for several benchmarks [32–35], due to their robust uncertainty quantificatio capabilities, adaptability to diverse datasets and network architectures, and relatively low computational overhead.

- MC Dropout [6] can be implemented by training a standard neural network (SNN) with dropout, approximating inference through stochastic forward passes during inference.
- Deep ensembles [8] can be constructed by training a set of SNNs under random initialization. Deep ensembles marginalize over multiple network models to derive a predictive distribution usually treated as a BNN inference approximation.

Given a training dataset $\mathbb{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ for a univariate regression task, both MC Dropout and deep ensembles can be trained by minimizing the Gaussian negative log-likelihood (NLL) loss, as follows:

$$\sum\nolimits_{n=1}^{N} \frac{\log \sigma_i^2(\boldsymbol{x}_n)}{2} + \frac{\big(\mu_i(\boldsymbol{x}_n)) - y_n\big)^2}{2\sigma_i^2(\boldsymbol{x}_n)}, \tag{1}$$

where $\mu_i(\boldsymbol{x}_n)$ and $\sigma_i^2(\boldsymbol{x}_n)$ are the mean and variance of the predicted single Gaussian distribution $\mathcal{N}(\mu_i(\boldsymbol{x}_n), \sigma_i^2(\boldsymbol{x}_n))$ for the data point $(\boldsymbol{x}_n, y_n)$. $\mathcal{N}(\mu_i(\boldsymbol{x}_n), \sigma_i^2(\boldsymbol{x}_n))$ is obtained from the i-th stochastic forward pass in MC Dropout or the i-th SNN ensemble member in deep ensembles.

When presenting a new sample $\boldsymbol{x}^*$, these models can predict a Gaussian mixture distribution $\mathcal{N}(\mu_*(\boldsymbol{x}^*), \sigma_*^2(\boldsymbol{x}^*))$ from a set of M sampled single Gaussian distributions, as follows:

$$\mu_*(\boldsymbol{x}^*) = M^{-1} \sum\nolimits_{m=1}^{M} \mu_m(\boldsymbol{x}^*) \quad \sigma_*^2(\boldsymbol{x}^*) = M^{-1} \sum\nolimits_{m=1}^{M} (\sigma_m^2\big(\boldsymbol{x}^*\big) + \mu_m^2(\boldsymbol{x}^*)\big) - \mu_*^2(\boldsymbol{x}^*). \tag{2}$$

Here, M represents the number of inference samples in MC Dropout or the number of SNNs in deep ensembles. The variance can serve as the measure of prediction uncertainty.

2.2 Light interval neural networks

As an alternative to traditional probabilistic techniques that rely heavily on probabilistic and distributional assumptions [14, 15], interval neural networks (INNs) have also garnered increasing attention for their application in uncertainty quantificatio for regression tasks.

A significan body of research has focused on INNs that produce deterministic interval predictions while maintaining fi ed point estimates for network weights and biases [14, 16–18], referred to as light-INNs in this work. The widely accepted principle for training light-INNs is to generate a high-quality prediction interval that remains as narrow as possible while ensuring a specifie coverage proportion of the training data. Specificall , the primary objective is to predict a prediction interval (PI), namely $[\hat{y}^L\ \hat{y}^U]$, which bounds the target y and satisfies

$$\text{PICP} := P_r(\hat{y}^L \leq y \leq \hat{y}^U) \geq P_c. \tag{3}$$

Here, P_c denotes the predefine confidenc level, usually set as $P_c = 0.95$. P_r is the prediction interval coverage probability (PICP). Following this principle, a recent advancement [18] has introduced a loss function that is less dependent on hyperparameters, as follows:

$$\underbrace{\sum_{n=1}^{N} \frac{(y_n - \hat{y}_n)^2}{2 \cdot \text{MPIW}^2} + \frac{\log(\text{MPIW}^2)}{2}}_{\mathcal{L}_{\text{UE}}} + \lambda \underbrace{\max\{0, P_c - \text{PICP}\}^2}_{\mathcal{L}_{\text{PI}}}. \tag{4}$$

Here, $\mathcal{L}_{\text{UE}}$ approximates the Gaussian NLL loss (equation (1)) and consists of a residual regression term and an uncertainty regularization component [18]. $\hat{y}_n$ is the predicted midpoint of the prediction interval $[\hat{y}_n^L\ \hat{y}_n^U]$ for the n-th data pair and MPIW represents mean prediction interval width [14, 17, 18] obtained from $\text{MPIW} = N^{-1}\sum_{n=1}^{N}(\hat{y}_n^U - \hat{y}_n^L)$. $\mathcal{L}_{\text{PI}}$ spontaneously measures the quality of the constructed PIs and PICP can be obtained from $\text{PICP} = \sum_{n=1}^{N} c_n$ where $c_n = 1$ for $y_n \in [\hat{y}_n^L\ \hat{y}_n^U]$ and $c_n = 0$ otherwise. λ serves as the training hyperparameter to balance the importance of two loss components [18]. A larger λ empirically demonstrates greater values of the PICP and MPIW.

The ensemble concept is widely applied to light-INNs [14, 17, 18] to account for the uncertainty associated with neural network parameters and improve uncertainty quantificatio performance. Specificall , given a collection of M independently trained light-INNs, the ensemble PIs, $[\overline{\hat{y}}^L\ \overline{\hat{y}}^U]$ can be computed as follows [18]:

$$\overline{\hat{y}}^L = \left(M^{-1}\sum_{m=1}^{M}\hat{y}_m^L\right) - \sigma_{\hat{y}^L} \quad \overline{\hat{y}}^U = \left(M^{-1}\sum_{m=1}^{M}\hat{y}_m^U\right) + \sigma_{\hat{y}^U}, \tag{5}$$

where $[\hat{y}_m^L\ \hat{y}_m^U]$ represent the PI from m-th ensemble light-INN and

$$\sigma_{\hat{y}^L}{}^2 = \frac{1}{M-1}\sum_{m}^{M}\left(\hat{y}_m^L - \frac{1}{M}\sum_{m}^{M}\hat{y}_m^L\right)^2 \quad \sigma_{\hat{y}^U}{}^2 = \frac{1}{M-1}\sum_{m}^{M}\left(\hat{y}_m^U - \frac{1}{M}\sum_{m}^{M}\hat{y}_m^U\right)^2. \tag{6}$$

Consequently, the prediction uncertainty can be quantifie as the range of $[\overline{\hat{y}}^L\ \overline{\hat{y}}^U]$.

2.3 Full interval neural networks

Unlike light-INNs and BNNs, the alternative approach to INNs is to utilize intervals over parameters (referred to as full-INNs in this work). The full-INN also incorporates deterministic interval inputs and outputs for each node and therefore conducts the interval forward propagation as follows [21, 22]:

$$[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l = f^l\Big([\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]^l \odot [\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1} \oplus [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l\Big), \tag{7}$$

where $\oplus$, $\ominus$, and $\odot$ represent interval addition, subtraction, and multiplication, respectively [36]. The quantities $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^l$, $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$, $[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}]^l$ and $[\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}]^l$ are the interval outputs of the l-th, the

interval outcomes of the preceding $(l-1)$-th layer, the interval weights and interval biases of the l-th layer, accordingly. $f^l(\cdot)$ corresponds to the activation function of the l-th layer, which is constrained to be monotonically increasing. When $[\underline{\boldsymbol{a}}, \overline{\boldsymbol{a}}]^{l-1}$ is non-negative, equation (7) can be simplifie as follows [21, 22]:

$$\begin{aligned} \underline{\boldsymbol{a}}^l &= f^l\left(\min\{\underline{\boldsymbol{W}}^l, \mathbf{0}\}\overline{\boldsymbol{a}}^{l-1} + \max\{\underline{\boldsymbol{W}}^l, \mathbf{0}\}\underline{\boldsymbol{a}}^{l-1} + \underline{\boldsymbol{b}}^l\right) \\ \overline{\boldsymbol{a}}^l &= f^l\left(\max\{\overline{\boldsymbol{W}}^l, \mathbf{0}\}\overline{\boldsymbol{a}}^{l-1} + \min\{\overline{\boldsymbol{W}}^l, \mathbf{0}\}\underline{\boldsymbol{a}}^{l-1} + \overline{\boldsymbol{b}}^l\right) \end{aligned}. \tag{8}$$

In our implementation of full-INNs, we adopt the methodology outlined in [22] to effectively ensure the validity of weight and bias intervals ($\underline{\boldsymbol{W}} \leq \overline{\boldsymbol{W}}$ and $\underline{\boldsymbol{b}} \leq \overline{\boldsymbol{b}}$) throughout the propagation process, as detailed below:

$$[\underline{\boldsymbol{W}}, \overline{\boldsymbol{W}}] = [\boldsymbol{W}_c - \boldsymbol{W}_r, \boldsymbol{W}_c + \boldsymbol{W}_r] \quad [\underline{\boldsymbol{b}}, \overline{\boldsymbol{b}}] = [\boldsymbol{b}_c - \boldsymbol{b}_r, \boldsymbol{b}_c + \boldsymbol{b}_r]. \tag{9}$$

Here, $\boldsymbol{W}_c$, $\boldsymbol{b}_c$ and $\boldsymbol{W}_r \geq \mathbf{0}$, $\boldsymbol{b}_r \geq \mathbf{0}$ are the centers (midpoints) and radii (half of ranges) of the weight and bias intervals, accordingly [22]. In this context, full-INNs can be initialized and trained using standard techniques [21, 22]. Additionally, interval batch normalization (IBN) [22], an interval-based extension of the conventional batch normalization technique, has been employed to improve the stability and adaptability of deep architectures. Figure 1 presents a simple MLP-based full-INN architecture for a univariate regression task.

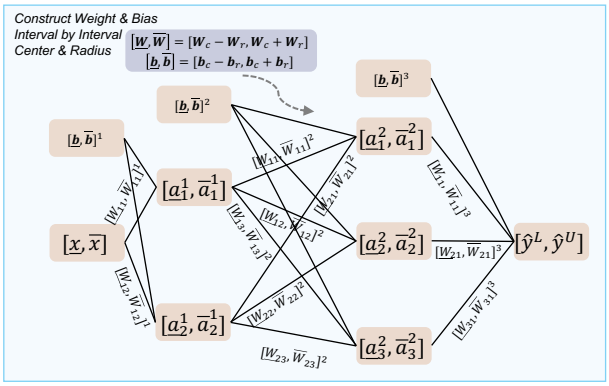


Figure 1: A simple MLP-based full-INN architecture for a univariate regression task. [22].

Given a standard training dataset $\mathbb{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the full-INN can be trained by setting $\underline{\boldsymbol{x}}_n = \overline{\boldsymbol{x}}_n = \boldsymbol{x}_n$ for the input and minimizing the following loss [21]:

$$\sum_{n=1}^{N}\left(\max\{y_n - \hat{y}_n^U, 0\}^2 + \max\{\hat{y}_n^L - y_n, 0\}^2 + \beta\cdot(\hat{y}_n^U - \hat{y}_n^L)\right), \tag{10}$$

in which $[\hat{y}_n^L \ \hat{y}_n^U]$ is the full-INN prediction for the n-th data point, and the hyperparameter parameter $\beta > 0$ determines how outlier-sensitive the intervals are trained [21]. The prediction interval range subsequently functions as a quantitative measure of prediction uncertainty.

In addition to standard datasets, full-INNs also encode the ability to handle uncertain interval-valued data. Specificall , when training data incorporate interval target values, i.e., $\{\boldsymbol{x}_n, [y_n^L \ y_n^U]\}_{n=1}^N$, one of widely applied training strategies for full-INN is to minimize the following loss function [37]

$$\sum_{n=1}^{N}\frac{1}{2}\left((\hat{y}_n^L - y_n^L)^2 + (\hat{y}_n^U - y_n^L)^2\right). \tag{11}$$

3 EXPERIMENTAL EVALUATION

In this section, we conduct three case evaluation studies: a one-dimensional analytical function in Section 3.1, a real-world remaining useful lifetime (RUL) prediction task in Section 3.2.1, and the RUL task involving interval-valued data (IVD) in Section 3.2.2.

3.1 One-dimensional analytical function

In our evaluation, we consider a one-dimensional analytical function, as follows:

$$f(x) = x\sin(x) + \epsilon_1 x + \epsilon_2, \tag{12}$$

where $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, 0.3)$. This function incorporates both homoscedastic (ϵ_2) and heteroscedastic (ϵ_1) aleatoric uncertainty, requiring the model to estimate both components. We generate 1,000 training samples with $x \in [0, 10]$, while out-of-training-set samples are constructed using 200 samples for $x \in [10, 15]$.

For Bayesian baselines, we train five SNNs with different random seed initializations to form deep ensembles and apply MC Dropout, both utilizing the Gaussian NLL loss in equation (1). Similarly, for INNs, we train five light-INNs with different random seed initializations to construct ensembles of light-INNs, employing the loss function in equation (4) with $\lambda = 5$. However, we are unable to obtain a well-trained full-INN model using the training strategy in equation (10), as severe underfittin is observed across various values of the training hyperparameter β. Specificall , all models follow the same MLP architecture and training settings as outlined in [9].

Figure 2 illustrates the comparison of the three approaches for prediction uncertainty. Compared to Bayesia n baselines, the ensemble of light-INNs performs best in the out-of-training-set region due to its higher estimated prediction uncertainty. However, in the in-training-set region, the smoothness of the predicted lower boundary—particularly for $x \in [6, 10]$ is inferior to that of deep ensembles. This is primarily due to the incorporation of the training data's coverage proportion in the training procedure of light-INNs, and the training noise is largely increased. Additionally, we do not present results for MC Dropout with larger sampling numbers M because increasing M from 5 to 100 did not significantl affect its poor performance.

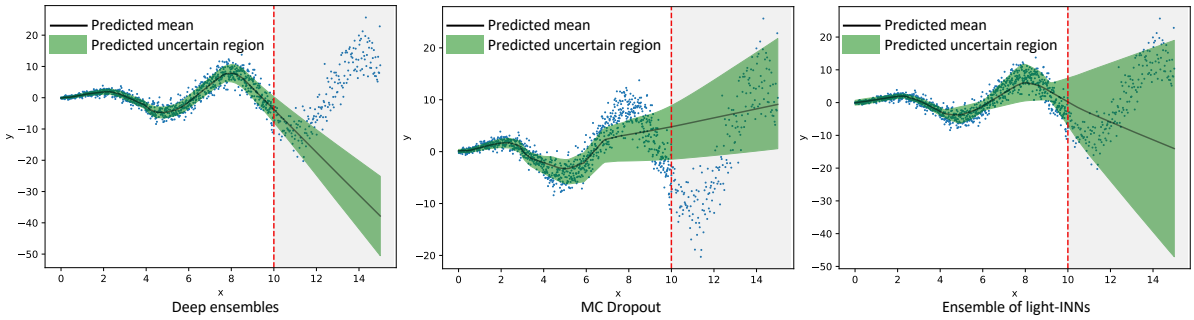


Figure 2: Comparison of three approaches for prediction uncertainty quantificatio in a one-dimensional analytical function regression. The predicted uncertainty regions for deep ensembles and MC Dropout are computed using $\mu_* \pm \sigma_*$ whereas for the ensemble of light-INNs, it is determined by $\hat{y}^U - \hat{y}^L$. The out-of-training-set region is marked in light gray.

3.2 Jet engine remaining useful lifetime prediction

Remaining useful lifetime (RUL) prediction plays a pivotal role in prognostics and health management (PHM), enhancing system reliability while ultimately reducing operational and maintenance costs for mechanical systems. Given its predictive nature, accuracy has been the primary focus of research efforts. To improve prediction performance, various models have been proposed, including Kalman filte -ensemble multi-layer perception (MLP) [38], recurrent neural networks (RNN) [39], and deep convolutional neural networks (DCNN) [40–42]. However, the presence of noise in the data as well as assumptions being used to generate the data, have heightened the importance of uncertainty estimation. In this test case, we explore how INNs could serve as a viable alternative to the existing Bayesian methods.

In this evaluation, we use the NASA turbofan engine degradation simulation dataset generated by C-MAPSS platform [43]. The dataset consists of four sub-datasets simulated under different two working conditions and two fault modes. A total of 21 sensor measurements are collected in each sub-dataset. However, some of the sensor measurements were dropped since they didn't give any useful information (i.e. variances are smaller than 1×10^{-6}), leaving the remaining set of 14 sensors which numbered as sensor numbers: 2,3,4,7,8,9,11,12,13,14,15,17,20, and 21. Following [42], we choose the FD001 and FD003 sub-datasets to be evaluated in this test case since they share the same operating conditions but have different fault modes. Since the problem is a time-series prediction, we employed a sliding window embedding method to generate the training and testing samples, where the window length is set to 30. In pre-processing the target label, we adopted a piecewise linear function with a constant RUL equal to 125 in the early stage [41, 42].

A modifie and shortened architecture of ResNet, called MResNet9 [42], shown in Figure 3 is used as the main neural network architecture to predict the RUL. The number before *Conv2d* represents the convolution kernel size, the number after *Conv2d* is the quantity of the convolution kernels, and $/2$ means the dimensions are reduced by half. *BN* represents the batch normalization, *ReLU* represents the rectifie linear unit activation function, and *Dropout* represents the dropout layer with 0.2 rate.

3.2.1 Light-INN approach

In this section, we evaluate the ensemble of light-INNs while choosing MC Dropout and deep ensembles as baselines. The ensemble number is set as $M = 5$. To evaluate the model's performance, we primarily consider two aspects: prediction accuracy and uncertainty. To ensure a fair comparison of accuracy, we employ two metrics. First, following the original C-MAPSS data challenge [43], we utilize the scoring function define in equation 13, where N represents the number of engines in the test set, and $h = (\text{Estimated RUL} - \text{True RUL})$.

$$\text{SF} = \begin{cases} \sum_{i=1}^{N} \left(e^{-\frac{h_i}{13}} - 1 \right), & \text{for } h_i < 0 \\ \sum_{i=1}^{N} \left(e^{\frac{h_i}{10}} - 1 \right), & \text{for } h_i \geq 0 \end{cases}. \tag{13}$$

The scoring function (SF) is intentionally designed to be asymmetric, penalizing late predictions more severely than early ones. This design choice reflect the aerospace industry's priority, where delayed maintenance can be significantl more detrimental than the premature allocation of maintenance resources. In addition to the scoring function, we employ the standard root mean

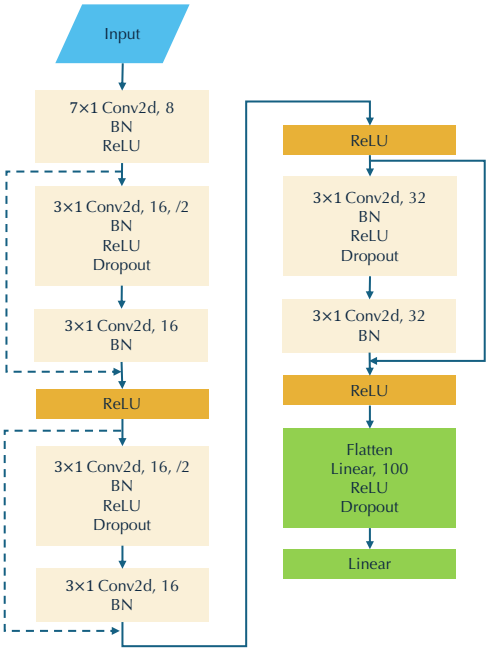


Figure 3: MResNet9 architecture.

squared error (RMSE) of the predicted RUL, define as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} h_i^2}, \tag{14}$$

A Log-likelihood function is often used to assess the prediction uncertainty quality. However, this approach assumes that the uncertainty estimation outputs a normal distribution with mean $\hat{f}(\boldsymbol{x})$ and standard deviation σ_{pred}. Due to the nature of interval prediction where the output distribution is unknown, this approach is not suitable. Instead, we employ the prediction interval coverage probability (PICP), define in equation (3).

The main drawback of the PICP comes when the model returns a very wide prediction interval, thus each of the test set observations falls inside the prediction interval. While the PICP score will be exceptional, this behaviour is unwanted since such behaviour is essentially a completely random process. Thus, we also employ sharpness, or the average interval length in conjunction with PICP to indicate the width of the interval, define as follows:

$$\text{IL} = N^{-1}\sum_{i=1}^{N} \hat{y}_i^U - \hat{y}_i^L. \tag{15}$$

Where IL indicates the length of the predicted $[\hat{y}^L\ \hat{y}^U]$ of the ensemble of light-INNs. In the context of deep ensembles and MC Dropout, the prediction interval applied in equation (15) (2) is obtained from $[\mu_* - \sigma_*\ \mu_* + \sigma_*]$. The best model would be the one that maximizes the PICP while keeping the interval length minimum.

The comparison of precision metrics given in Table 1 shows that the deep ensemble ensemble and the ensemble of light-INNs are very close to each other in terms of predictive performance. While the MC Dropout, despite having a similar architecture is not as accurate as the others. In terms of uncertainty estimation shown in Table 2 and Figures 4 and 5, the ensemble of

Table 1: Accuracy metrics comparison between methods.

Datasets	FD001		FD003	
Metrics	RMSE	SF	RMSE	SF
MC Dropout	15.13 ± 1.02	579.75 ± 498.51	15.05 ± 0.77	534.58 ± 120.21
Deep ensembles	12.65 ± 0.17	247.61 ± 8.26	12.48 ± 0.41	361.57 ± 60.60
Ensemble of light-INNs	12.96 ± 0.50	260.63 ± 31.31	13.11 ± 0.61	483.64 ± 178.88

Table 2: Uncertainty metrics comparison between methods.

Datasets	FD001		FD003	
Metrics	PICP	IL	PICP	IL
MC Dropout	0.75 ± 0.03	38.74 ± 1.27	0.79 ± 0.03	38.96 ± 1.48
Deep ensembles	0.88 ± 0.01	39.50 ± 0.42	0.90 ± 0.01	39.19 ± 0.61
Ensemble of light-INNs	0.35 ± 0.03	12.55 ± 0.30	0.42 ± 0.03	11.99 ± 0.39

light-INNs produces a rather over-confiden estimation compared to the other methods. In addition, our experiments indicate that the performance of light-INN method is more sensitive to hyperparameters and training configurations This heightened sensitivity primarily stems from the finel designed loss function (equation (4)) used in light-INN training, which necessitates a careful balance between the accuracy of mean prediction and the quality of PCIP.

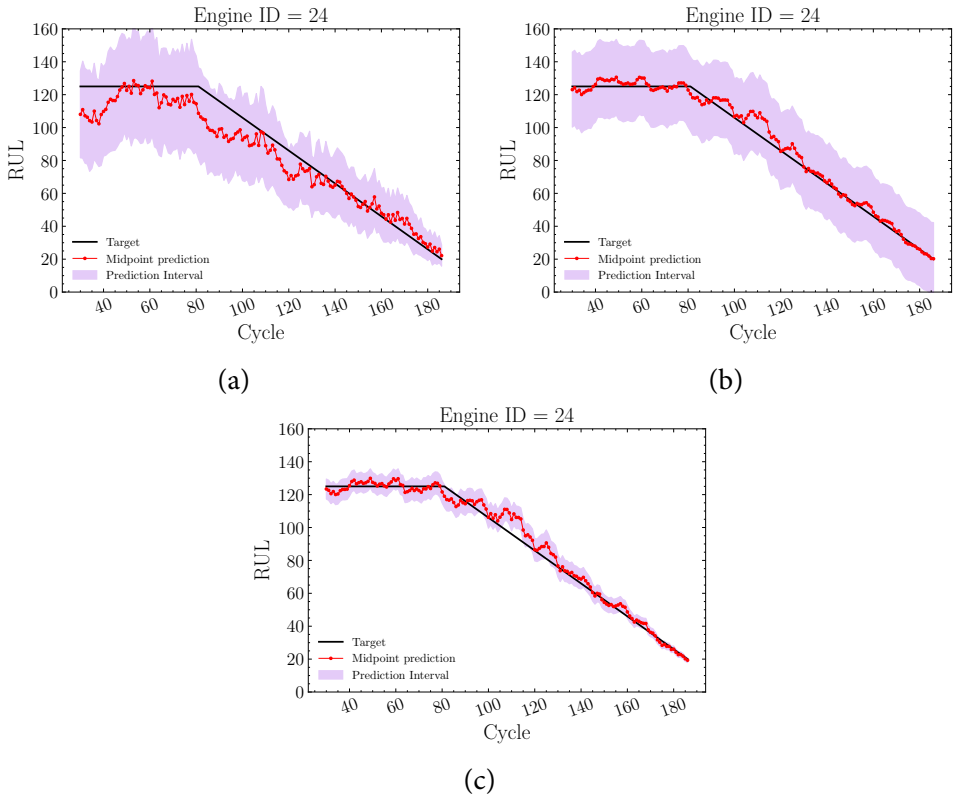


Figure 4: Test data FD001 engine 24 RUL prediction using (a) MC Dropout, (b) deep ensembles, and (c) ensemble of light-INNs.

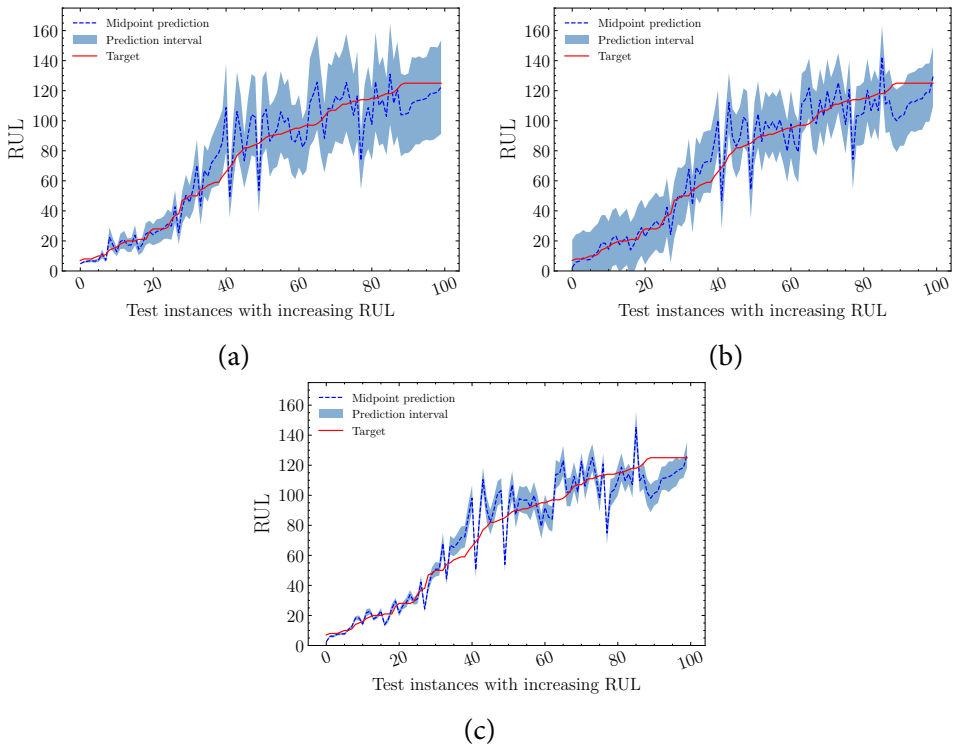


Figure 5: Test data FD001 instances prediction with increasing RUL using (a) MC Dropout, (b) deep ensembles, and (c) ensemble of light-INNs.

3.2.2 Full-INN approach

In this section, we explore the capability of full-INNs to handle imprecise data (interval-valued data). In the RUL test case, we modify the existing dataset to have an imprecise value by adding and subtracting the existing data with a random value between 10 and 20, written as:

$$\begin{aligned} y_i^{\mathrm{U}} &= y_i + \mathcal{U}(10, 20) \\ y_i^{\mathrm{L}} &= y_i - \mathcal{U}(10, 20) \end{aligned}, \tag{16}$$

where the minimum and maximum values sampled from the uniform distribution $\mathcal{U}$ are arbitrary.

Converting the modified ResNet architecture in Figure 3 into the full-INN architecture can be done by replacing the corresponding layers and activation function types with their interval-valued layers counterpart [21, 22]. To train the full interval-version MResNet9, we employ the loss function in equation (11).

To evaluate the full-INN, we propose several different metrics than the light-INN. Since we have interval-valued data (IVD) in the full-INN, we first propose to use a modified version of RMSE, where the final value is the average of the lower and upper RMSE, written as:

$$\text{I-RMSE} = \frac{1}{2}\text{RMSE}\left(\hat{y}^{\mathrm{L}} - y^{\mathrm{L}}\right)^2 + \frac{1}{2}\text{RMSE}\left(\hat{y}^{\mathrm{U}} - y^{\mathrm{U}}\right)^2. \tag{17}$$

We also slightly modified the scoring function (SF) in equation 13 to be suited for IVD. At its core, we still follow the original SF paradigm, which penalizes late predictions more than early predictions. However, we also define a safe space, which is defined as the area between the upper

Table 3: Accuracy metrics comparison between methods.

Datasets	FD001		FD003	
Metrics	I-RMSE	I-SF	I-RMSE	I-SF
full-INN	16.22 ± 1.00	558.53 ± 184.83	15.34 ± 0.78	616.24 ± 151.08

bound and the lower bound ground truth. Therefore, early prediction of the upper bound and late prediction of the lower bound are not penalized. The interval SF is define as:

$$\text{I-SF} = \begin{cases} \sum_{i=1}^{N} \left(e^{-\frac{h^L}{13}} - 1\right) + \left(e^{\frac{h^U}{10}} - 1\right), & \text{for } h^L < 0 \;\&\; h^U \geq 0 \\ \sum_{i=1}^{N} \left(e^{-\frac{h^L}{13}} - 1\right), & \text{for } h^L < 0 \;\&\; h^U < 0 \\ \sum_{i=1}^{N} \left(e^{\frac{h^U}{10}} - 1\right), & \text{for } h^L \geq 0 \;\&\; h^U \geq 0. \\ 0, & \text{else}. \end{cases} \tag{18}$$

PICP and IL metrics are used to quantify the quality of uncertainty estimation in the light-INN case in section 3.2.1. However, since the problem is interval-valued we propose to use two metrics to evaluate the uncertainty estimation in full-INNs. First, we introduce the area coverage ratio (ACR) define as the ratio of the overlapping area between the predicted interval and the true interval relative to the total area of the true interval. The expression is written as:

$$\begin{aligned} \text{ACR} &= \frac{A_{\text{overlap}}}{A_{\text{true}}} \\ A_{\text{true}} &= \int_{x \in X} (y_{\text{up}}(x) - y_{\text{low}}(x)) dx \\ A_{\text{overlap}} &= \int_{x \in X} \max(0, \min(\hat{y}_{\text{up}}(x), y_{\text{up}}(x)) - \max(\hat{y}_{\text{low}}(x), y_{\text{low}}(x))) \, dx \end{aligned} \tag{19}$$

The ACR can be interpreted as how much the model correctly predicts the true interval. Thus, a higher ACR indicates a better model. However, ACR can be easily achieved if the model predicts a very wide interval that encloses the whole space. Thus, to avoid such conditions, we introduce the non-overlap (outside) area ratio (OAR). The OAR is define as the non-overlapping area of the predicted interval relative to the total area of the predicted interval, written as:

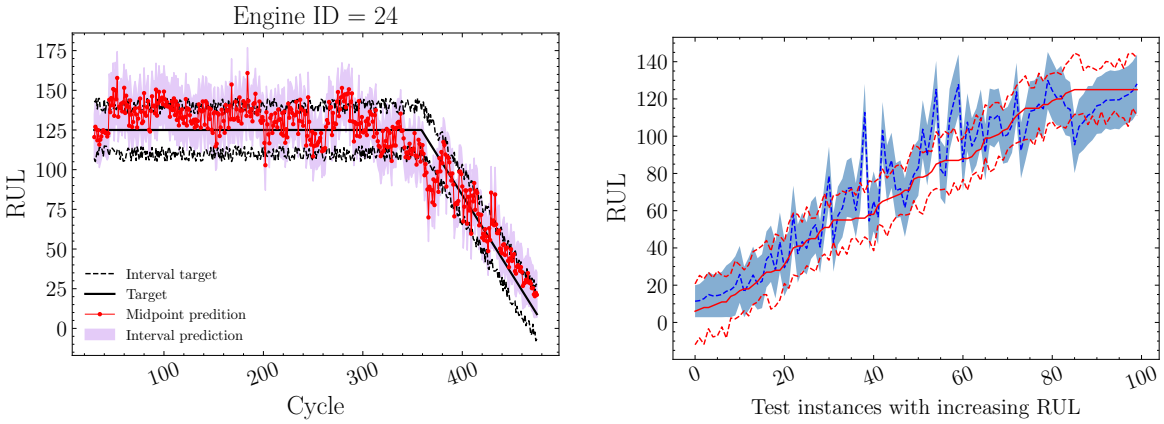
$$\begin{aligned} \text{OAR} &= \frac{A_{\text{pred}} - A_{\text{overlap}}}{A_{\text{pred}}} \\ A_{\text{pred}} &= \int_{x \in X} (\hat{y}_{\text{up}}(x) - \hat{y}_{\text{low}}(x)) dx \end{aligned} \tag{20}$$

Intuitively, the OAR measures how much the model mispredicts the true interval. Thus, lower OAR indicates a better model.

The prediction plot is given in Figure 6. Tables 3 and 4 show the value of the proposed metrics for the full-INN model with 10 repetitions. However, because there are no comparable methods and models to learn from RUL data, no methodological comparison is provided. Instead, we provide the result as a benchmark for future research.

Table 4: Uncertainty metrics comparison between methods.

Datasets	FD001		FD003	
Metrics	ACR	OAR	ACR	OAR
full-INN	0.60 ± 0.03	0.37 ± 0.02	0.62 ± 0.03	0.35 ± 0.02



(a) (b)

Figure 6: (a) Interval prediction on engine 24, and (b) Test instances prediction with sorted increasing RUL.

4 CONCLUSION

In this study, we evaluated two types of interval neural networks (INNs)—light-INNs and full-INNs—for uncertainty quantificatio in regression tasks. Our evaluation covered standard regression, a realistic remaining useful lifetime (RUL) prediction task, and the RUL task under interval-valued data (IVD) settings. The results suggest that light-INNs have the potential to serve as an alternative approach for uncertainty quantification However, they require a careful trade-off between mean prediction performance and the quality of prediction interval coverage probability, particularly when handling complex problems.

Furthermore, the capability of full-INNs to process interval-based data and integrate with deep architectures was validated through the RUL task with IVD settings. However, their performance on standard datasets in our test cases was suboptimal, indicating the need for further investigation into training strategies and the inherent behavior of these models.

For future work, we aim to deepen our understanding of the training dynamics and loss function design of INNs. Additionally, we plan to explore their performance in time-series forecasting and partial differential equation problems.

ACKNOWLEDGMENT

This work was supported by granted H2020 FETOPEN-2018-2019-2020-01 European project, *Epistemic AI* under grant agreement No. 964505 (E-pi).

REFERENCES

[1] Houyu Lu, Sergio Cantero-Chinchilla, Xin Yang, Konstantinos Gryllias, and Dimitrios Chronopoulos. Deep learning uncertainty quantificatio for ultrasonic damage identificatio in composite structures.

Composite Structures, 338:118087, 2024.

[2] Lavi Rizki Zuhal, Ghifari A. Faza, Pramudita S. Palar, and Rhea P. Liem. Fast and adaptive reliability analysis via kriging and partial least squares. In *AIAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, January 2021.

[3] Gang Sun and Shuyue Wang. A review of the artificia neural network surrogate modeling in aerodynamic design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(16):5863–5872, 2019.

[4] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[5] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introductionto concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[7] Aryan Mobiny, Pengyu Yuan, Supratik K Moulik, Naveen Garg, Carol C Wu, and Hien Van Nguyen. Dropconnect is effective in modeling uncertainty of Bayesian deep networks. *Scientific Reports*, 11(1):1–14, 2021.

[8] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[9] Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

[10] Manuel A Vega and Michael D Todd. A variational Bayes ian neural network for structural health monitoring and cost-informed decision-making in miter gates. *Structural Health Monitoring*, 21(1):4–18, 2022.

[11] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantifica tion using Bayesian neural networks in classific tion: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

[12] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

[13] Vincent Fortuin. Priors in Bayesian deep learning: A review. *International Statistical Review*, 2022.

[14] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International conference on machine learning*, pages 4075–4084. PMLR, 2018.

[15] Krasymyr Tretiak, Georg Schollmeyer, and Scott Ferson. Neural network model for imprecise regression with interval dependent variables. *Neural Networks*, 161:550–564, 2023.

[16] Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F. Atiya. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks*, 22(3):337–346, 2011.

[17] Tárik S. Salem, Helge Langseth, and Heri Ramampiaro. Prediction intervals: Split normal mixture from quality-driven deep ensembles. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1179–1187. PMLR, 03–06 Aug 2020.

[18] Yuandu Lai, Yucheng Shi, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Exploring uncertainty in regression neural networks for construction of prediction intervals. *Neurocomputing*, 481:249–257, 2022.

[19] Hisao Ishibuchi, Hideo Tanaka, and Hidehiko Okada. An architecture of neural networks with interval weights and its application to fuzzy regression analysis. *Fuzzy Sets and Systems*, 57(1):27–39, 1993.

[20] Z.A. Garczarczyk. Interval neural networks. In *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages 567–570 vol.3, 2000.

[21] Luis Oala, Cosmas Heiß, Jan Macdonald, Maximilian März, Wojciech Samek, and Gitta Kutyniok. Interval neural networks: Uncertainty scores, 2020.

[22] Kaizheng Wang, Keivan Shariatmadar, Shireen Kudukkil Manchingal, Fabio Cuzzolin, David Moens, and Hans Hallez. CreINNs: Credal-Set Interval Neural Networks for Uncertainty Estimation in Classificatio Tasks. *Neural Networks*, 185:107198, 2025.

[23] L. Billard and E. Diday. Regression analysis for interval-valued data. In *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 369–374. Springer Berlin Heidelberg, 2000.

[24] Eufrásio de A. Lima Neto and Francisco de A.T. de Carvalho. Constrained linear regression models for symbolic interval-valued variables. *Computational Statistics and Data Analysis*, 54(2):333–347, February 2010.

[25] Ghifari A. Faza, Keivan Shariatmadar, Hans Hallez, and David Moens. Interval reduced order surrogate modelling framework for uncertainty quantification In *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, January 2024.

[26] Ronay Ak, Valeria Vitelli, and Enrico Zio. An interval-valued neural network approach for uncertainty quantificatio in short-term wind speed prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 26(11):2787–2800, 2015.

[27] David Betancourt and Rafi L Muhanna. Interval deep learning for computational mechanics problems under input uncertainty. *Probabilistic Engineering Mechanics*, 70:103370, 2022.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

[30] Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[31] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[32] Hendrik A Mehrtens, Alexander Kurz, Tabea-Clara Bucher, and Titus J Brinker. Benchmarking common uncertainty estimation methods with histopathological images under domain shift and label noise. *Medical image analysis*, 89:102914, 2023.

[33] Bálint Mucsányi, Michael Kirchhof, and Seong Joon Oh. Benchmarking uncertainty disentanglement: Specialized uncertainties for specialized tasks. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[34] Kaizheng Wang, Fabio Cuzzolin, Shireen Kudukkil Manchingal, Keivan Shariatmadar, David Moens, and Hans Hallez. Credal deep ensembles for uncertainty quantification In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[35] Kaizheng Wang, Fabio Cuzzolin, Keivan Shariatmadar, David Moens, and Hans Hallez. Credal wrapper of model averaging for uncertainty estimation in classification In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Timothy Hickey, Qun Ju, and Maarten H Van Emden. Interval arithmetic: From principles to implementation. *Journal of the ACM (JACM)*, 48(5):1038–1068, 2001.

[37] Hisao Ishibuchi, Hideo Tanaka, and Hidehiko Okada. An architecture of neural networks with interval weights and its application to fuzzy regression analysis. *Fuzzy Sets and Systems*, 57(1):27–39, 1993.

[38] Leto Peel. Data driven prognostics using a kalman filte ensemble of neural network models. In *2008 International Conference on Prognostics and Health Management*, pages 1–6, 2008.

[39] Felix O. Heimes. Recurrent neural networks for remaining useful life estimation. In *2008 International Conference on Prognostics and Health Management*, pages 1–6, 2008.

[40] Giduthuri Sateesh Babu, Peilin Zhao, and Xiao-Li Li. *Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life*, page 214–228. Springer International Publishing, 2016.

[41] Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering and System Safety*, 172:1–11, April 2018.

[42] Zhibin Zhao, Jingyao Wu, David Wong, Chuang Sun, and Ruqiang Yan. Probabilistic remaining useful life prediction based on deep convolutional neural network. *SSRN Electronic Journal*, 2020.

[43] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9, 2008.