A General Data-driven Framework for Remaining Useful Life Estimation with Uncertainty Quantification Using Split Conformal Prediction

Weijun Xu
Department of Energy
Politecnico di Milano
Milan, Italy
weijun.xu@polimi.it

Enrico Zio
MINES Paris
PSL University
Sophia Antipolis, France
enrico.zio@mines-paristech.fr
Department of Energy
Politecnico di Milano
Milan, Italy
enrico.zio@polimi.it

***Abstract*—Remaining useful life estimation utilizes historical data and advanced analytic techniques to estimate the remaining lifespan of systems, components or assets, for enabling proactive decision-making and efficient resource management. However, existing data-driven prognostic technologies often provide deterministic values of remaining useful life which do not represent uncertainties. In practice, uncertainties are due to factors such as noise, measurement errors in the prediction and limited availability of data to train the predictive models. In this work, we propose a novel framework that addresses the issue of prognostic unertainty by generating prediction intervals for single-point estimators, thereby enabling robust uncertainty quantification. The proposed framework overcomes the drawbacks of previous conformal prediction methods, which lacked interval adaptivity and relied on the assumption of data exchangeability. To validate our framework, we performed experiments using the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) datasets. Results demonstrate that both conventional machine learning estimators and modern deep learning estimators yield reasonable prediction intervals within the proposed framework. The superiority of our framework over previous conformal prediction methods is demonstrated.**

***Keywords*—Remaining useful life, uncertainty quantification, split conformal prediction, prediction interval, data exchangeability**

I. Introduction

With the rapid advancement of sensor technology, it has become possible to perform online monitoring of the health conditions of various industrial devices and assets. The availability of data provides support for the development of prognostics and health management (PHM) technologies [1]. In particular, accurately forecasting the remaining useful life (RUL) of a gradually degrading system allows for the development of condition-based and predictive maintenance plans, that can be both cost and safety effective. Prognostic approaches to RUL estimation can be broadly classified into two main categories: model-based and data-driven [2]. Model-based approaches rely on the thorough understanding, and consequent modeling of the degradation mechanisms and the governing laws of the progression of deterioration. However, these approaches often face challenges in practice, due to the complexity of the failure mechanisms and the operational environment. In the end, it can be difficult to establish accurate models, especially when the underlying mechanisms are not fully understood or when there are uncertainties in the system operational behavior. In contrast, data-driven approaches primarily depend on the amount and quality of the available data, but are not constrained by specific model structures. These approaches harness the power of machine learning algorithms and statistical techniques to extract patterns and relationships from the data. They do not explicitly rely on in-depth knowledge of the underlying degradation mechanisms. This makes data-driven approaches more adaptable to diverse systems and less dependent on complex modeling assumptions. By learning directly from the data, these approaches can capture complex relationships and patterns that may not be easily discernible through traditional model-based approaches [3].

Most data-driven approaches provide single-point estimates of the RUL, thus failing to capture the uncertainties associated with prognostics in real-world scenarios [4]. In the absence of a thorough comprehension of the prediction's uncertainty, solely relying on determinstic predictions of RUL can potentially lead to unsuitable maintenance decisions. Thus, RUL prediction intervals with a specified level of confidence must be provided, rather than single values [5].

In ideal circumstances, prediction intervals should be as narrow as possible, reflecting the degree of difficulty in making accurate predictions, which is referred to as the adaptive property. A wider interval indicates higher uncertainty or greater difficulty in making precise predictions. Furthermore, it is important for the predicted interval to have a specified probability of containing the true value of the response variable, which is known as the coverage property [6]. Conformal

prediction (CP) is a popular technique that provides valid predictive inference for arbitrary machine learning models [7].

In this paper, we present a novel framework that focuses on generating prediction intervals for single-point estimators, allowing for robust uncertainty quantification. By doing so, we address the lack of interval adaptivity and overcome the assumption of data exchangeability present in previous approaches. In order to assess the effectiveness of our framework, we performed experiments utilizing the widely acknowledged Commercial Modular Aero-Propulsion System Simulation (CMAPSS) datasets [28]. The results demonstrate that both conventional machine learning estimators and popular deep learning estimators produce reasonable prediction intervals when integrated into our proposed framework. Additionally, we provide evidence to support the superiority of our framework over previous conformal prediction methods. The validation of our framework showcases its potential for practical application in uncertain prediction tasks.

The remainder of this paper is organized as follows. Some related works are recalled in Section II. In Section III, basic theory of conformal prediction and the proposed framework are presented, followed by the case study in Section IV. Section V concludes this paper.

II. Background and Related Work

A. RUL Estimation Methods

Model-based methods of RUL estimation rely on models to describe the stochastic degradation process of a system. There are two main techniques of uncertainty quantification in model-based methods, which include filtering methods and stochastic processes. By employing Bayesian tracking techniques, Kalman Filter, Particle Filter, and Extended Kalman Filter can quantify the uncertainty associated to predictions through the estimation of variance or covariance [8, 9]. Stochastic processes are common approaches to include uncertainties. As one of the typical stochastic processes, the Wiener process is widely used for quantifying uncertainty in model-based methods [10]. Data-driven approaches to RUL estimation leverage the available data to build the relationships between the influencing factors and the system state of degradtation. These approaches encompass a wide range of techniques, including classical machine learning methods and modern deep learning methods. Classical machine learning methods, such as random forest (RF), support vector machines (SVM), K-nearest neighbors (KNN) and gradient boosting (GB), have been widely used for RUL estimation [11–15]. In recent years, deep learning techniques have gained great attention. Deep Neural Networks (DNN), deep convolutional neural networks (DCNN), recurrent neural networks (RNN), and long short-term memory networks (LSTM) have demonstrated promising performance in capturing complex patterns and dependencies within the data [16–19]. These deep learning models are capable of automatically learning hierarchical representations from raw data, enabling more accurate and robust RUL predictions.

B. Gaps in Existing Knowledge

Bayesian methods are used in both Gaussian Processes [20] and Bayesian neural networks [21]. These methods adopt prior distributions over model parameters, which are then updated to posterior distributions based on the observed data. Ensemble methods, on the other hand, involve training multiple machine learning models simultaneously and utilizing mean and variance of predictions to quantify uncertainty [22]. The upper and lower quantiles of the response variables based on the input features are utilized to generate intervals in quantile regression models [23]. Each of these methods possesses unique strengths and weaknesses, and the selection of which one to use relies on the specific demands and attributes of the RUL estimation problem at hand. Bayesian methods provide formal guarantees but may be sensitive to model misspecification. Ensemble methods offer simplicity and efficiency but lack interpretability. Quantile regression models provide direct interval estimates but may not ensure coverage for all cases. In summary, there is a lack of a general framework for uncertainty quantification in data-driven RUL estimation. This paper aims to establish such a general framework.

III. Methodology

The estimation of the RUL using data-driven approaches can be framed as a regression problem on time series data. Conformal prediction (CP) is a method that provides prediction regions for regression, which encompass the target value with a predetermined level of confidence [24].

In split conformal prediction (SCP), the training data is divided into subsets for training the regression model and calibrating the prediction intervals for the test. The training subset is employed for training the regression model, whereas the calibration subset is utilized to assess and quantify the uncertainty in the predictions. On the one hand, the original SCP suffers from a limitation that the prediction intervals have the constant width for each instance in the test set which means a lack of adaptivity. Furthermore, when constructing the prediction interval, in the SCP method each point of the calibration set is equally weighted, negelecting the fact that RUL prediction is actually a time series regression problem with ordered data.

A. Split Conformal Prediction

SCP randomly splits the original training data $D = D_{train} \cup D_{calibration}$ into two separate subsets: a *proper* training set $D_{train}\,\{(\mathbf{X}_i, y_i)\}_{i=1}^{m}$ and a calibration set $D_{calibration}\,\{(\mathbf{X}_j, y_j)\}_{j=1}^{n}$. The *proper* training set D_{train} is used to train the predictive regression model $\mathcal{M}$, whereas the calibration set $D_{calibration}$ is used to obtain the nonconformity scores, which serve as a measure of how unusual or atypical a given example is, compared to previously observed data. In regression, the absolute residual error between true value y_j and corresponding prediction $\tilde{y}_j = \mathcal{M}(\mathbf{X}_j)$ generated by the underlying model is a commonly used nonconformity score [25]:

$$S_j = |y_j - \tilde{y}_j| = |y_j - \mathcal{M}(\mathbf{X}_j)| \tag{1}$$

By caculating the nonconformity scores for each point in the calibration set, we obtain a distribution of scores that reflects the range of nonconformity observed in the training data. This distribution serves as reference for evaluating the nonconformity of new queries. Given a new query instance in the test set $D_{test}\left\{(\mathbf{X}_k, y_k)\right\}_{k=1}^{p}$, the prediction interval can be computed as:

$$\begin{aligned} C_{SCP}^{\alpha}(\mathbf{X}_k) &= \mathcal{M}(\mathbf{X}_k) \pm \left(\lceil (1-\alpha)(n+1) \rceil \text{-th} \right. \\ &\qquad \left. \text{of } R_{asc}\left\{S_j\right\}_{j=1}^{n}\right) \\ &= \mathcal{M}(\mathbf{X}_k) \pm q \end{aligned} \tag{2}$$

where $\lceil \bullet \rceil$ is the ceiling function and α is the error rate (miscoverage rate), $R_{asc}\left\{\bullet\right\}$ denotes the ranking in ascending order of the elements of the set, q is defined as the smallest value among the nonconformity scores. Equivalently, we can write:

$$C_{SCP}^{\alpha}\left(\mathbf{X}_k\right) = \mathcal{M}\left(\mathbf{X}_k\right) \pm Q_{1-\alpha}\left(\frac{1}{n+1}\delta_{+\infty} + \sum_{j=1}^{n}\frac{1}{n+1}\delta_{S_j}\right) \tag{3}$$

where $Q_{\tau}\left(\bullet\right)$ denotes the τ-quantile of its argument, and δ_a denotes the point mass at a. One drawback of the existing method is lacking adaptivity, as the width of the prediction intervals remains constant at $2q$ for all query instances, regardless of their difficulty of prediction. Indeed, we would expect the width of the intervals to vary based on the difficulty of making predictions for different data points.

To tackle this concern, one strategy is to adapt the nonconformity measure itself by integrating the difficluty of the prediction task [26]. For instance, a normalized nonconformity measure can be defined as a ratio between the nonconformity score and a measure of data point prediction difficulty. This difficulty measure can be based on various factors such as the data distribution, the characteristics of the query instance, or any other relevant information that indicates the difficulty of the prediction task. In this work, a normalized nonconformity is defined as:

$$S_j = \frac{\left|y_j - \tilde{y}_j\right|}{\tilde{\mathcal{M}}\left(x_j\right)} \tag{4}$$

$\forall\left(\mathbf{X}_j, y_j\right) \in D_{calibration}$, where $\tilde{\mathcal{M}}$ denotes a new regression model trained by a new dataset $\left\{(\mathbf{X}_i, |y_i - \tilde{y}_i|) : (\mathbf{X}_i, y_i) \in D_{train}\right\}_{i=1}^{m}$.

B. Nonexchangeable Split Conformal Prediction

The effectiveness of SCP in providing valid prediction intervals relies on the crucial assumption of exchangeability, which refers to the property whereby the order of observations does not affect the statistical properties of the data [27]. For a sequence of random variables $(X_1, X_2, ..., X_n)$ to be exchangeable, it means that for any permutation $\sigma\left(\bullet\right)$ of the indices (i.e., any rearrangement of the observations), the joint probability distribution of the random variables remains the same:

$$P\left(X_1, X_2, ..., X_n\right) = P\left(X_{\sigma(1)}, X_{\sigma(2)}, ..., X_{\sigma(n)}\right) \tag{5}$$

Nevertheless, it is crucial to acknowledge that exchangeability is not universally guaranteed in real-world data. In many cases, the assumption of exchangeability may be violated due to various factors, such as temporal or spatial dependencies, trends or structural changes in the data. In such situations, alternative modeling approaches or modifications to existing methods may be required to account for the lack of exchangeability and allowing to capture the underlying patterns of the data more accurately. In the method proposed in [27], the SCP framework is modified to address the issue of violating the exchangeability assumption. This modification allows for the generation of valid prediction intervals even in situations where exchangeability does not hold, such as when dealing with ordered data like time series.

The relevant random variables in split conformal prediction consist of calibration data $D_{calibration}\left\{(\mathbf{X}_j, y_j)\right\}_{j=1}^{n}$ and test data $D_{test}\left\{(\mathbf{X}_k, y_k)\right\}_{k=1}^{p}$. Consider a new instance $(\mathbf{X}_{new}, y_{new})$ in the test set and assume that the instances in the combined $D_{calibration} \cup (\mathbf{X}_{new}, y_{new})$ are not exchangeable. In this case, if we have prior knowledge or information about the underlying similarity between the distribution of $(\mathbf{X}_{new}, y_{new})$ and the instances in $D_{calibration}\left\{(\mathbf{X}_j, y_j)\right\}_{j=1}^{n}$, we can assign weights $w_j \in [0, 1]$ to each $(\mathbf{X}_j, y_j)$ in $D_{calibration}\left\{(\mathbf{X}_j, y_j)\right\}_{j=1}^{n}$. Higher weights indicate a higher level of similarity, suggesting that the corresponding calibration instances are more relevant for generating intervals of the new test instances. For example, if the new test instance $(\mathbf{X}_{new}, y_{new})$ is closer in time to certain instances in the calibration set, those instances might be considered more similar and assigned higher weights. Incorporating the weights into the conformal prediction framework, allows for more adaptive and flexible prediction intervals that can better capture the varying levels of prediction similarity in different regions of the data space.

Given a new instance in the test set, we define a weight set which represent the similarity between test instance and calibration instance:

$$w_j = 0.9^{|t(y_{new}) - t(y_j)|} \tag{6}$$

where $t(y)$ is the time index of y [27]. In our study, we take the RUL value of calibration set and the predicted RUL value of test set as time index, which is under the assumption that the degradation process is related to a single operating condtion and a single fault mode. Under such assumption, the two RUL values can reflect the relative time position in the whole degradation process. Then, we define a normalized weight:

$$\tilde{w}_j = \frac{w_j}{1 + \sum_{j=1}^{n} w_j} \tag{7}$$

From a distribution perspective, the normalized weights can be employed to adjust the empirical distribution of nonconformity scores:

$$\frac{1}{1 + \sum_{j=1}^{n} w_j}\delta_{+\infty} + \sum_{j=1}^{n}\frac{1}{1 + \sum_{j=1}^{n} w_j}\delta_{S_j} \tag{8}$$

C. Nonexchangeable Split Conformal Prediction with Normalized Nonconformity Measure

Accordingly, the intervals of split conformal prediction (SCP), split conformal prediction with normalized nonconformity measure (SCPN), nonexchangeable split conformal prediction (NSCP) and nonexchangeable split conformal prediction with normalized nonconformity measure (NSCPN) can be calculated as follows respectively:

$$C^{\alpha}_{SCP}\left(\mathbf{X}_k\right)=\mathcal{M}\left(\mathbf{X}_k\right)\pm Q_{1-\alpha}\left(\frac{1}{n+1}\delta_{+\infty}+\sum_{j=1}^{n}\frac{1}{n+1}\delta_{S_j}\right) \tag{9}$$

$$C^{\alpha}_{SCPN}\left(\mathbf{X}_k\right)=\mathcal{M}\left(\mathbf{X}_k\right)\pm Q_{1-\alpha}\left(\frac{1}{n+1}\delta_{+\infty}+\sum_{j=1}^{n}\frac{1}{n+1}\delta_{S_j}\right)\times\tilde{M}\left(\mathbf{X}_k\right) \tag{10}$$

S_j in (9) and (11) can be substituted by (1), whereas S_j in (10) and (12) can be substituted by (4):

$$C^{\alpha}_{NSCP}\left(\mathbf{X}_k\right)=\mathcal{M}\left(\mathbf{X}_k\right)\pm Q_{1-\alpha}\left(\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{+\infty}+\sum_{j=1}^{n}\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{S_j}\right) \tag{11}$$

$$C^{\alpha}_{NSCPN}\left(\mathbf{X}_k\right)=\mathcal{M}\left(\mathbf{X}_k\right)\pm Q_{1-\alpha}\left(\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{+\infty}+\sum_{j=1}^{n}\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{S_j}\right)\times\tilde{M}\left(\mathbf{X}_k\right) \tag{12}$$

The procedure of nonexchangeable split conformal prediction with normalized nonconformity measure (NSCPN) can be summarized as follows:

- Step 1. A regression algorithm $\mathcal{A}$ is trained on D_{train} to build model $\mathcal{M}$.
- Step 2. After computing the absolute residual error between the prediction of the underlying model $\tilde{y}_i = \mathcal{M}\left(\mathbf{X}_i\right)$ and the ture value y_i, another regression algorithm $\tilde{\mathcal{A}}$ is trained on $\{(\mathbf{X}_i,|y_i-\tilde{y}_i|):(\mathbf{X}_i,y_i)\in D_{train}\}_{i=1}^{m}$ to build the model $\tilde{\mathcal{M}}$.
- Step 3. For each data point in $D_{calibration}\,\{(\mathbf{X}_j,y_j)\}_{j=1}^{n}$, calculate the absolute residual error between the prediction of the underlying model $\tilde{y}_j = \mathcal{M}\left(\mathbf{X}_j\right)$ and the true value y_j; then, calculate the prediction $\tilde{\mathcal{M}}\left(\mathbf{X}_j\right)$. The nonconformity scores S_j are computed as $S_j=\frac{|y_j-\tilde{y}_j|}{\tilde{\mathcal{M}}(x_j)}$.
- Step 4. For every data point in $D_{test}\,\{(\mathbf{X}_k,y_k)\}_{k=1}^{p}$, calculate $\mathcal{M}\left(\mathbf{X}_k\right)$ and $\tilde{\mathcal{M}}\left(\mathbf{X}_k\right)$,respectively. Then, calculate the quantile of $Q_{1-\alpha}\left(\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{+\infty}+\sum_{j=1}^{n}\frac{1}{1+\sum_{j=1}^{n}w_j}\delta_{S_j}\right)$ based on S_j and w_j. Finally, according to (12) the predictioin interval can be constructed.

Figure 1 shows the general data-driven framework for RUL estimation with uncertainty quantification using split conformal prediction.

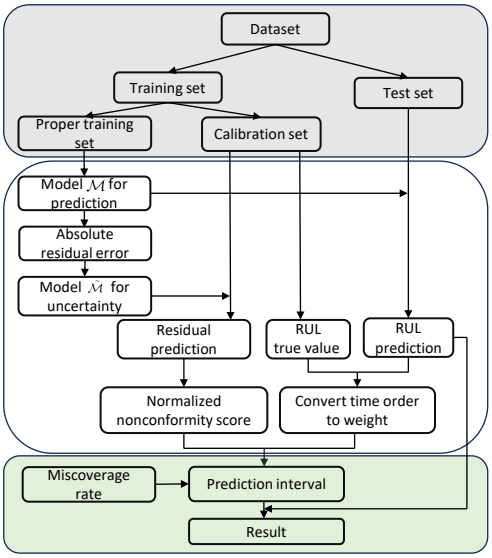


Fig. 1: The general data-driven framework for RUL estimation with uncertainty quantification using split conformal prediction.

IV. Case Study

A. Datasets

The datasets utilized in this study are sourced from the NASA Ames prognostics data repository. They include run-to-failure data of turbofan engines which are simulated by the CMAPSS simulation environment. Four datasets of multivariate time series data are under different combinations of operational conditions and fault modes. They have been used as a standardized benchmark for evaluating RUL estimation methods on multivariate time series data. For in-depth information regarding the simulation settings and the platform employed for data collection, the details can consult [28].

For our case study, we consider only dataset FD001 because the RUL can be used as time index, based on the assumption of single operating condition and single failure mode. Dataset FD001 is single operating condition and single failure mode. The training set is partitioned into a proper training set and a calibration set, with data proportions of 80% and 20%, respectively.

The analysis reveals that seven out of the twenty-one sensor measurements do not contribute any valuable information for the RUL estimation. Therefore, it is decided to remove these seven sensors from the dataset and retain fourteen sensors. By using piecewise linear rectified labels, the maximum RUL value is constrained, providing a more realistic representation of the engine's RUL. In line with the approach described in [29], a common choice for the maximum RUL value is set to 125.

The dataset and code for our work are publicly available at https://github.com/xuweijun-npuer/Nonexchangeable-Split-Conformal-Prediction-of-RUL.

B. RUL Estimators

To validate the feasibility of the proposed framework as a general uncertainty quantification tool, we employ six data-driven estimators including conventional machine learning estimators and modern deep learning estimators. Support Vector Regression (SVR), Random Forest (RF) and Gradient Boosting (GB) are adopted as conventional machine learning estimators whereas Convolutional Neural Networks (CNN), Deep Neural Networks (DNN) and Long Short-Term Memory Networks (LSTM) are considered as modern deep learning estimators. To compare the performance of our framework to previous conformal prediction methods, we conformalize six estimators using Split Conformal Prediction (SCP), Split Conformal Prediction with normalized nonconformity measure (SCPN), Nonexchangeable Split Conformal Prediction (NSCP) and Nonexchangeable Split Conformal Prediction with normalized nonconformity measure (NSCPN).

The general settings for each estimator are as follows (For more detailed settings, please refer to the implementation code).

- RF: The model is set up with 300 trees and employs the square root of the total number of features to identify the optimal split at each node. Additionally, another Random Forest Regression model with default settings is utilized for training purposes of all 6 estimators.
- SVR: The regularization parameter is set to 10. The epsilon parameter, which controls the size of the epsilon-tube around the predicted values, is set to 10. The kernel function used for SVR is a radial basis function.
- GB: The mean square error is employed as the loss function. Other settings are default.
- CNN: The neural network architecture includes a total of five convolutional layers. The first four of these layers have ten filters each with dimensions of 10 × 1, while the final layer consists of a single filter with dimensions 3 × 1. Additionally, the architecture comprises a flatten layer and a fully connected layer. The dropout rate is set as 0.5 to prevent overfitting,. The convolutional layers utilize zero-padding to maintain the data dimension. Tanh activation function is employed in all layers except for the final neuron, which employs a linear activation function. The Adam optimizer is used to minimize the mean squared error loss function.
- DNN: The architecture of the deep neural network includes four hidden layers with different neuron counts: 500, 400, 300, and 100. There is a flatten layer after four hidden layers. The dropout rate of 0.5 is set to mitigate overfitting. The mean squared error (MSE) loss function is employed in conjunction with the Adam optimizer to optimize the model's parameters.
- LSTM: The LSTM architecture consists of three LSTM layers followed by two fully connected layers with 96 and 128 neurons respectively, and a linear output neuron for RUL estimation.

For coventional machine learning estimators, the input size is (*numbers_samples*, *numbers_features*), and (*numbers_samples*, *numbers_time_steps*, *numbers_features*) for deep learning estimators. In this paper, the *numbers_time_steps* is set to 30 using the sliding time window technique. The *numbers_features* is 14 as mentioned before.

C. Metrics

In this investigation, two evaluation metrics, namely the scoring function and root mean square error (RMSE), are employed to assess the performance of the RUL estimators. The scoring function is utilized to gauge the precision of the predicted values [30]:

$$s = \sum_{i=1}^{N} s_i, s_i = \begin{cases} e^{-\frac{d_i}{13}} - 1, d_i < 0 \\ e^{-\frac{d_i}{10}} - 1, d_i \geq 0 \end{cases} \tag{13}$$

where s denotes the values and N is the total number of test data, d_i is the error between the estimated RUL and the actual RUL for the i-th test data. RMSE measures the average error between the predicted values and the actual values. It provides an indication of the overall accuracy of the prognostic method, with lower RMSE values indicating better performance:

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

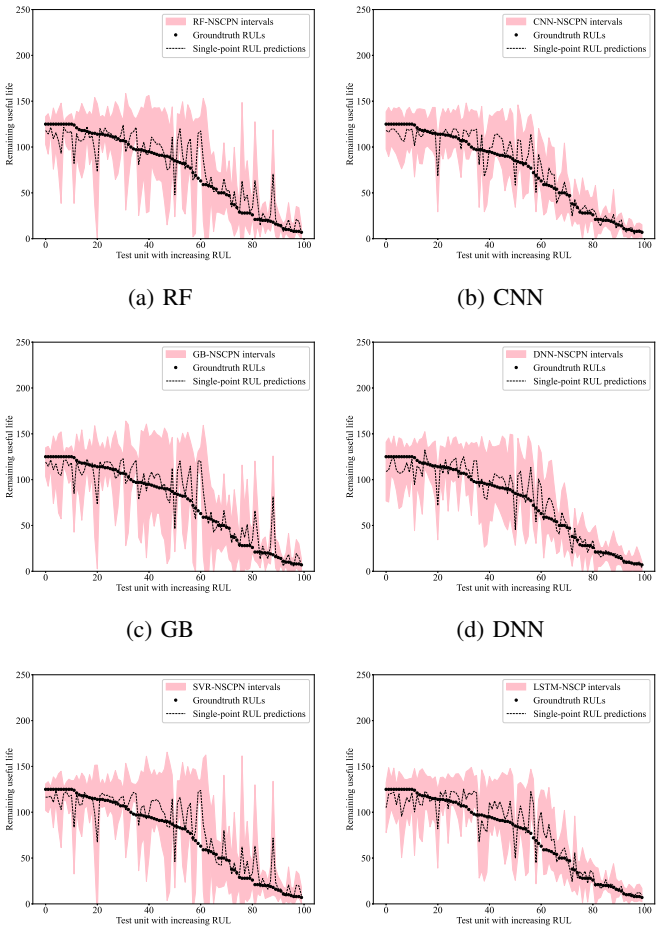
Two additional metrics are employed to evaluate the uncertainty quantification of conformal prediction: average coverage and average width. The average coverage represents the percentage of test data falling within the predicted range in relation to the total number of test data, the average width denotes the average width of the prediction intervals. The miscoverage rate α in this paper is set to 0.1.

D. Results and Discussion

Firstly, we conformalize six estimators using Nonexchangeable Split Conformal Prediction with normalized nonconformity measure (NSCPN). Figure 2 shows the results of the conventional machine learning estimators (RF/SVR/GB) on the left side and modern deep learning estimators (CNN/DNN/LSTM) on the right side. It can be observed that the NSCPN framework can provide prediction intervals for both conventional machine learning estimators and modern deep learning estimators. This proves that NSCPN can be taken as a general RUL estimation framework with uncertainty quantification. Additionally, we use the same algorithm in the RF case, validating that arbitrary regression algorithms can be effective within the proposed framework, which again reflects generality and scalability.

We can analyze the results in three stages, initial degradation process (test units from 0 to 30), progressive degradation process (test units from 30 to 70) and near to failure (test units from 70 to 100). Deep learning can give more accurate prediction results in the stage of near to failure because of the strong capability of capturing complex temporal dependencies. During the other two stages, both conventional machine learning and modern deep learning predict unstable results,

especially in the progressive degradation process. Therefore, we can expect to see narrower prediction intervals in the stage of near to failure but wider prediction intervals in the stage of progressive degradation process, which is the behavior expected from our proposed framework.

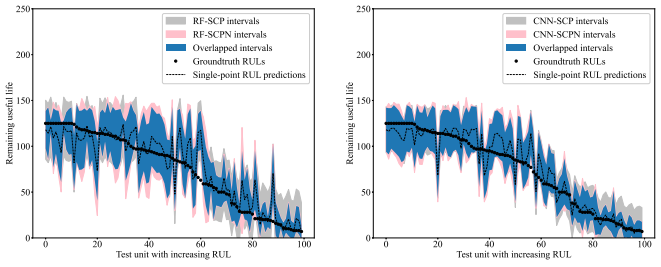


(a) RF (b) CNN

(c) GB (d) DNN

(e) SVR (f) LSTM

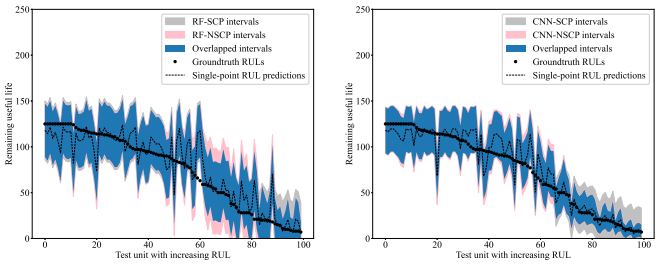
Fig. 2: Sorted RUL single-point estimation with predicted intervals from NSCPN by different estimators.



(a) RF (b) CNN

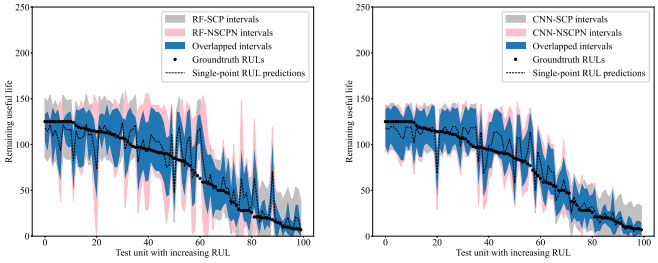
Fig. 3: RF and CNN applied with SCP and SCPN.

We also apply the SCPN, NSCP and NSCPN frameworks to each estimator and compare it with SCP. After experimenting



(a) RF (b) CNN

Fig. 4: RF and CNN applied with SCP and NSCP.



(a) RF (b) CNN

Fig. 5: RF and CNN applied with SCP and NSCPN.

with six estimators, we find the same pattern. Due to space constraints, we solely present and discuss the results of RF and CNN in this paper. Figure 3 illustrates that, for those areas where the prediction results are accurate, SCPN gives a narrower prediction interval, proving that we have strong confidence in the prediction results. Data points closer to the failure time are relatively easier to predict compared to those with larger actual RUL values. Therefore, we see narrower prediction intervals for these data points, which aligns with the behavior expected from score normalization. In Figure 4, it is evident that for those areas where the prediction results are not accurate, especially those peaks which are far away from real values, NSCP gives a wider prediction interval, proving that we have weak confidence in the prediction results. This observation indicates that converting time order to weight employs similarity to keep the similar shape as original SCP. Figure 5 shows that both score normalization and nonexchangeability improve the split conformal prediction. They give more adaptive and flexible prediction intervals that can better capture the varying levels of prediction difficulty or uncertainty in different regions of the data space. This performance characteristic is very useful in safety-critical scenarios for decision-making evaluation.

Considering the four metrics mentioned earlier, we run the experiments randomly ten times. Table I and Table II show the function score, root mean square error, average coverage and average width of one random time and the average of ten random times, respectively.

Based on the randomized experiment presented in Table I, it is evident that while the accuracy of the estimators may differ,

the average coverage of the generated intervals remains around 0.9. This indicates the robustness of NSCPN, whose coverage is guaranteed by 1-α. In general, deep learning estimators demonstrate superior prediction accuracy compared to traditional machine learning estimators when averaging random experimental results. This is evidenced by the slightly narrower average width of the prediction intervals, as demonstrated in Table II.

TABLE I: Metrics of One-time Random Experiment

	Score	RMSE	Average Coverage	Average Width
CNN	264.13	12.83	0.95	46.98
DNN	292.01	13.38	0.92	53.28
LSTM	413.4	14.69	0.90	49.65
RF	941.3	17.14	0.91	61.29
GB	1495.29	17.58	0.89	64.24
SVR	1336.24	17.86	0.89	61.59

TABLE II: Average Metrics of Ten-time Random Experiment

	Score	RMSE	Average Coverage	Average Width
CNN	294.62	12.65	0.89	39.96
DNN	302.69	13.48	0.89	42.65
LSTM	411.65	14.47	0.88	44.27
RF	795.01	16.83	0.90	56.41
GB	946.76	17.32	0.89	56.92
SVR	1021.15	17.50	0.88	54.60

V. Conclusion

In this work, we have introduced a new framework that addresses the limitations of previous conformal prediction methods by generating prediction intervals for single-point estimators. We conducted experiments using the CMAPSS datasets to validate the effectiveness of our approach.

The results of our experiments demonstrate that both traditional machine learning estimators and modern deep learning estimators produce reliable prediction intervals when used within our framework. We also compared our framework to previous conformal prediction methods and found that it outperforms them in terms of interval adaptivity that can better capture the varying levels of prediction difficulty or uncertainty in different regions of the data space.

Overall, our proposed framework offers a promising solution for generating prediction intervals with improved adaptivity and reliability, as shown through the experiments conducted on the CMAPSS datasets.

Further research will be focused on two aspects. On the one hand, the proposed uncertainty framework should be compared with those methods that already have the ability of uncertainty quantification, for example, Gaussian process in conventional machine learning and Deep Bayesian Neural Network in modern deep learning. On the other hand, the proposed framework is based on an assumption of single operating condition and single failure mode. The uncertainty quantification of RUL predictioin under multiple operating conditons and multiple failure modes should also be further studied.

Acknowledgment

Weijun Xu gratefully acknowledges the financial support from the China Scholarship Council (No. 202108610087).

References

[1] J. E. Office, N. Gebraeel, Y. Lei, N. Li, X. Si, and E. Zio, "Prognostics and Remaining Useful Life Prediction of Machinery: Advances, Opportunities and Challenges," Journal of Dynamics, Monitoring and Diagnostics, vol. 2, no. 1, Art. no. 1, Feb. 2023, doi: 10.37965/jdmd.2023.148.

[2] Y. Hu, X. Miao, Y. Si, E. Pan, and E. Zio, "Prognostics and health management: A review from the perspectives of design, development and decision," Reliability Engineering & System Safety, vol. 217, p. 108063, Jan. 2022, doi: 10.1016/j.ress.2021.108063.

[3] E. Zio, "Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice," Reliability Engineering System Safety, vol. 218, p. 108119, Feb. 2022, doi: 10.1016/j.ress.2021.108119.

[4] J. Caceres, D. Gonzalez, T. Zhou, and E. L. Droguett, "A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties," Structural Control and Health Monitoring, vol. 28, no. 10, p. e2811, 2021, doi: 10.1002/stc.2811.

[5] S. Sankararaman, "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction," Mechanical Systems and Signal Processing, vol. 52–53, pp. 228–247, Feb. 2015, doi: 10.1016/j.ymssp.2014.05.029.

[6] A. N. Angelopoulos and S. Bates, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification." arXiv, Dec. 07, 2022. doi: 10.48550/arXiv.2107.07511.

[7] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-Free Predictive Inference for Regression", Journal of the American Statistical Association, 113:523, 1094-1111, doi: 10.1080/01621459.2017.1307116

[8] B. Duan, Q. Zhang, F. Geng, and C. Zhang, "Remaining useful life prediction of lithium-ion battery based on extended Kalman particle filter," International Journal of Energy Research, vol. 44, no. 3, pp. 1724–1734, 2020, doi: 10.1002/er.5002.

[9] D. An, J.-H. Choi, and N. H. Kim, "Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab," Reliability Engineering & System Safety, vol. 115, pp. 161–169, Jul. 2013, doi: 10.1016/j.ress.2013.02.019.

[10] J. Yang, S. Tang, P. Fang, F. Wang, X. Sun, and X. Si, "Remaining useful life prediction of implicit linear Wiener degradation process based on multi-source information," Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and

Reliability, p. 1748006X221132606, Nov. 2022, doi: 10.1177/1748006X221132606.

[11] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Remaining useful life estimation based on nonlinear feature reduction and support vector regression," Engineering Applications of Artificial Intelligence, vol. 26, no. 7, pp. 1751–1760, Aug. 2013, doi: 10.1016/j.engappai.2013.02.006.

[12] Y. Zhou, M. Huang, and M. Pecht, "Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization," Journal of Cleaner Production, vol. 249, p. 119409, Mar. 2020, doi: 10.1016/j.jclepro.2019.119409.

[13] X. Chen, G. Jin, S. Qiu, M. Lu, and D. Yu, "Direct Remaining Useful Life Estimation Based on Random Forest Regression," in 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai), Oct. 2020, pp. 1–7. doi: 10.1109/PHM-Shanghai49105.2020.9281004.

[14] Z. Zheng et al., "A Novel Method for Lithium-Ion Battery Remaining Useful Life Prediction Using Time Window and Gradient Boosting Decision Trees," in 2019 10th International Conference on Power Electronics and ECCE Asia (ICPE 2019 - ECCE Asia), May 2019, pp. 3297–3302. doi: 10.23919/ICPE2019-ECCEAsia42246.2019.8797021.

[15] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," in Database Systems for Advanced Applications, S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 214–228. doi: 10.1007/978-3-319-32025-0_14.

[16] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2306–2318, Oct. 2017, doi: 10.1109/TNNLS.2016.2582798.

[17] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," Reliability Engineering & System Safety, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/j.ress.2017.11.021.

[18] S. Zhao, Y. Zhang, S. Wang, B. Zhou, and C. Cheng, "A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method," Measurement, vol. 146, pp. 279–288, Nov. 2019, doi: 10.1016/j.measurement.2019.06.004.

[19] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," in 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Jun. 2017, pp. 88–95. doi: 10.1109/ICPHM.2017.7998311.

[20] K. Vuckovic and S. Prakash, "Remaining Useful Life Prediction using Gaussian Process Regression Model," Annual Conference of the PHM Society, vol. 14, no. 1, Art. no. 1, Oct. 2022, doi: 10.36001/phmconf.2022.v14i1.3220.

[21] A. Rivas, G. K. Delipei, and J. Hou, "Predictions of component Remaining Useful Lifetime Using Bayesian Neural Network," Progress in Nuclear Energy, vol. 146, p. 104143, Apr. 2022, doi: 10.1016/j.pnucene.2022.104143.

[22] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, "Ensemble of optimized echo state networks for remaining useful life prediction," Neurocomputing, vol. 281, pp. 121–138, Mar. 2018, doi: 10.1016/j.neucom.2017.11.062.

[23] M. Zhang, D. Wang, N. Amaitik, and Y. Xu, "A Distributional Perspective on Remaining Useful Life Prediction With Deep Learning and Quantile Regression," IEEE Open Journal of Instrumentation and Measurement, vol. 1, pp. 1–13, 2022, doi: 10.1109/OJIM.2022.3205649.

[24] C. Xu and Y. Xie, "Conformal prediction for time series." arXiv, Feb. 16, 2023. doi: 10.48550/arXiv.2010.09107.

[25] Y. Romano, E. Patterson, and E. J. Candès, "Conformalized Quantile Regression." arXiv, May 08, 2019. doi: 10.48550/arXiv.1905.03222.

[26] H. Papadopoulos, V. Vovk, and A. Gammerman, "Regression Conformal Prediction with Nearest Neighbours," jair, vol. 40, pp. 815–840, Apr. 2011, doi: 10.1613/jair.3198.

[27] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability." arXiv, Mar. 16, 2023. doi: 10.48550/arXiv.2202.13415.

[28] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in 2008 International Conference on Prognostics and Health Management, Oct. 2008, pp. 1–9. doi: 10.1109/PHM.2008.4711414.

[29] E. Ramasso, "Investigating Computational Geometry for Failure Prognostics in Presence of Imprecise Health Indicator: Results and Comparisons on C-MAPSS Datasets," PHM Society European Conference, vol. 2, no. 1, Art. no. 1, 2014, doi: 10.36001/phme.2014.v2i1.1460.

[30] P. Wang, B. D. Youn, and C. Hu, "A generic probabilistic framework for structural health prognostics and uncertainty management," Mechanical Systems and Signal Processing, vol. 28, pp. 622–637, Apr. 2012, doi: 10.1016/j.ymssp.2011.10.019.