LSTM and Transformers based methods for Remaining Useful Life Prediction considering Censored Data

Jean-Pierre NOOT[1,2]
Mikaël MARTIN[2]
Etienne BIRMELE[1]

[1] *Institut de Recherche Mathématique Avancée, UMR 7501 Université de Strasbourg et CNRS*
7 rue René-Descartes, 67000 Strasbourg, France
jnoot@unistra.fr
birmele@unistra.fr
[2] *Liebherr Components Colmar, Haut-Rhin, 68000, FRANCE*
jean-pierre.noot@liebherr.com
mikael.martin@liebherr.com

ABSTRACT

Predictive maintenance deals with the timely replacement of industrial components relatively to their failure. It allows to prevent shutdowns as in reactive maintenance and reduces the costs compared to preventive maintenance. As a consequence, Remaining Useful Life (RUL) prediction of industrial components has become a key challenge for condition based monitoring. In many applications, in particular those for which preventive maintenance is the general rule, the prediction problem is made harder by the rarity of failing instances. Indeed, the interruption of data acquisition before the occurrence of the event of interest leads to right censored data. Recent deep-learning architectures, that show the best results of the literature for complete-life data, most often do not consider censoring, even though its rate in the industrial environment may be high.

The present article introduces a method which considers censored data for the Dual Aspect Self-Attention based on Transformer proposed by (Z. Zhang, Song, & Li, 2022), and puts it into competition a modified version of the ordinal-regression based LSTM of (Vishnu, Malhotra, Vig, & Shroff, 2019). The evaluation of the resulting method on the C-MAPSS and N-CMAPSS benchmark dataset shows that it is competitive compared to the state-of-the-art RUL prediction methods for a low censoring rate and more efficient for a high rate of censoring in large enough data sets. Finally, conformal prediction is used to estimate confidence intervals for the predictions.

https://doi.org/10.36001/IJPHM.2025.v16i2.4260

1. INTRODUCTION

Technology and electronic developments of sensors nowadays allow the collection of huge amounts of data on mechanical and industrial equipment, in particular time series measuring their evolution over time. The definition of the maintenance schedule, which is crucial for the industry, therefore shifts to predictive maintenance, or condition-based maintenance (CBM). The latter is defined by opposition to the historical preventive maintenance, for which the maintenance schedule is pre-defined, each component being replaced at fixed time intervals. CBM avoids replacement of healthy components and therefore reduces costs, by determining a dynamic schedule depending on the real-time monitoring of the system. A crucial step is therefore the estimation, given the actual status of the system, of the Remaining Useful Lifetime (RUL) of a component, that is the time before its failure.

Preliminary results were presented at the European Conference of the Prognostics and Health Management Society 2024 (Jean-Pierre, Birmelé, & François, 2024). This journal article extends that work by evaluating the proposed method on a different dataset and by incorporating conformal prediction to provide calibrated uncertainty estimation.

Several approaches exist to create CBM models for RUL estimation (Arena, Collotta, Luca, Ruggieri, & Termine, 2021), most of them being physical model-based methods, data-driven methods or hybridisation of those approaches.

Model-based methods consider the physical phenomenon, for instance corrosion or fatigue, that leads to the failure. A mathematical model is used to simulate the studied mechanism and to get a RUL prediction (Tinga & Loendersloot,

2019). A precise physical and mechanical knowledge is however needed to build physical-based models. Moreover, this approach results in highly complex models when applied to large scale industrial systems composed with a lot of subsystems (Casenave, 2024).

Data-driven methods regroup approaches that rely on stochastic models or statistical analysis to create fault detection models not directly mimicking the underlying physics. It may consist in statistical algorithms to diagnose battery fault (Y. Zhao, Liu, Wang, & Hong, 2017), stochastic processes to mimic the degradation processes (Garay & Diedrich, 2019) or evolving fuzzy models for semiconductor health management (Boutrous, Bessa, Puig, Nejjari, & Palhares, 2022).

Data-driven methods include machine learning algorithms, which have been extensively used by the Prediction and Health Management community to establish predictive maintenance rules. Multiple linear regression durability models were for instance used to predict the fatigue life of automotive coil (Kong, Abdullah, Schramm, Omar, & Haris, 2019), or SVM classifiers for fault detection in vehicle suspensions (Jeong & Choi, 2019). In (Vasavi, Aswarth, Pavan, & Gokhale, 2021), a *k*NN classifier is used to detect fault by predicting vehicle health using real time data, while (Patil et al., 2018) relies on decision trees and gradient boosting regressor for RUL prediction. Another modelisation choice is gaussian processes (Groot & Lucas, 2012) that have been applied to RUL prediction of batteries (Liu & Chen, 2019; Jia et al., 2020) or turbofans (Benker, Bliznyuk, & Zaeh, 2021)

Deep learning, like machine learning methods, allow to have no physical or mechanical knowledge of the studied system. In recent years, numerous articles have demonstrated the effectiveness of those methods for RUL prediction. The data at hand being mainly time series, the developed methods focus on architectures widely used to treat sequential data. Recurrent neural networks like Long-short-time-memory (LSTM) (Zheng, Ristovski, Farahat, & Gupta, 2017), or Convolutional neural network (CNN) (Sateesh Babu, Zhao, & Li, 2016) and recently Transformers (Z. Zhang et al., 2022), which were adapted from the original Transformer architecture (Vaswani et al., 2017) to deal with time series are popular method used to perform RUL predictions.

Besides the choice of a physical, statistical or deep-learning modelisation, another important characteristic for RUL prediction methods is their ability to take right-censored data into account (Lillelund, Pannullo, Jakobsen, Morante, & Pedersen, 2024). Indeed, when the current policy for an industrial application is predictive maintenance, equipment's are renewed before failure, leading to numerous time-series in the dataset for which the RUL is unknown. One way to deal with such data is to use the survival approach based on Cox models that has been successfully transposed from medical analysis to maintenance analysis (Rahat, Kharazian, Mashhadi, Rögnvaldsson, & Choudhury, 2023; Yang, Kanniainen, Krogerus, & Emmert-Streib, 2022a), and combined with dimension reduction (Wang, Zhao, Yang, Xu, & Ge, 2022) or neural networks (Katzman et al., 2018; Chen et al., 2020). (Aggarwal et al., 2018) propose a modelisation based on a Weibull distribution for the RUL, which estimation takes account right-censored data. Another alternative is the ordinal regression (OR) approach where the RUL prediction is replaced by a vector of predictions encoding the failure time (Vishnu et al., 2019).

A third concern in RUL prediction in industrial systems is that they are often subject to uncertainties arising from data noise, stochastic failures or model limitations. As an error may be of high cost if the failure happens before the prediction, it is crucial to quantify the uncertainty associated to a RUL prediction. Some methods, for instance based on Bayesian methods (Caceres, Gonzalez, Zhou, & Droguett, 2021) or previously cited gaussian processes, natively define a probabilistic law for the RUL and thus an uncertainty measure. Another option is to try and predict the quantiles of the RUL distribution rather than the RUL itself (Chen, Shi, Shen, Feng, & Tao, 2023; T. Zhang & Wang, 2024). Finally, prediction methods which outcome is a single procedure can predict a confidence interval, without strong assumptions about the underlying data distribution, by using conformal prediction (CP) (Angelopoulos, Bates, & al, 2023), as illustrated in (Javanmardi & Hüllermeier, 2023).

The aim of the present paper is to build upon the recent Transformer based DAST model (Z. Zhang et al., 2022) which was proven to be among the most performant models in terms of prediction accuracy for RUL prediction on non-censored multi-dimensional time series. The model is adapted using ordinal regression in order to be able to take right-censored data into account and put into competition with an improved version of the LSTM-OR model of (Vishnu et al., 2019). To illustrate the performance of the obtained model, the proposed method is run on the C-MAPSS Turbofan NASA benchmark dataset, and completed by an uncertainty quantification using conformal prediction. The predictions are compared to state-of-the-art methods, able to consider censored data or not. The benchmark dataset being characterized by the absence of censoring, the latter is artificially introduced at various levels. The results highlight that the best choice between LSTM and Transformer architectures is dependent on the operating conditions and fault modes. Moreover, the proposed method is shown to be comparable to the best methods on non-censored data and better when a significant amount of data is censored.

2. RELATED WORK

This section introduces the main ideas of the DAST (Z. Zhang et al., 2022) and LSTM-OR (Vishnu et al., 2019) architectures,

and then builds upon those ideas to propose a novel method for RUL estimation on censored data.

Beforehand, let us introduce the notations which will be used throughout the paper.

For a given unit, we denote by:

- $T^\star$ the time of failure,
- C the censoring time if relevant, that is if the unit is replaced before failure,
- $T = \min(C, T^\star)$ the observed time of replacement,
- X the time series of the p sensors data, $x_{k,t}$ being the measure of sensor k at time t,
- Z the optional of vector covariates, that is characteristics of the unit which are not varying with time.

We denote the training, calibration, validation, and test sets by $\mathscr{D}_{\text{train}}$, $\mathscr{D}_{\text{calib}}$, $\mathscr{D}_{\text{valid}}$, and $\mathscr{D}_{\text{test}}$, respectively.

Let us fix a maximum value R_{max} for the RUL estimation, which is standard procedure (Heimes, 2008; H. Li, Zhao, Zhang, & Zio, 2020) and is relevant for the applications, as it focuses on the precision of the method on the period preceding the failure. At a given time point t, The the remaining lifetime to predict is defined by

$$R_t = \min(T^\star - t, R_{max})$$

Note that this lifetime is observed in the training set only when $T^\star = T$. If not, the only available information is that $R_t \geq \min(C - t, R_{max})$.

All the variables in that section are in fact indexed by the number i of the considered unit, for instance when computing a loss. That index is omitted unless necessary for reading purposes.

2.1. Dual Aspect Self-Attention based on Transformer (DAST)

The DAST model is an encoder-decoder, with the specificity of a double encoding, using a time step encoder and the sensor encoder.

The input data of the DAST architecture consists in a decomposition of the times series X by a sliding window processing of width W, as shown in Figure 2. The input is thus a list of matrices (X_t), each of size (p, W):

$$X_t = \begin{pmatrix} x_{1,t} & \cdots & x_{1,t+W} \\ \vdots & \ldots & \vdots \\ x_{p,t} & \cdots & x_{p,t+W} \end{pmatrix} \tag{1}$$

The data of each window are normalized to equalize the amplitude for each sensor and completed with positional encoding to keep track of the relative time positions of the

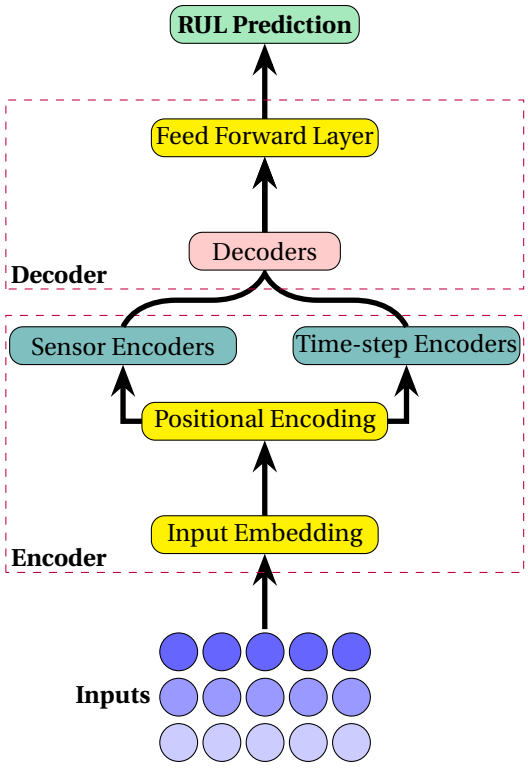


Figure 1. Original DAST architecture (Z. Zhang et al., 2022)

columns as well as constant lines corresponding to the covariates Z. It is also enriched for each sensor by the mean value and the slope of the linear regression as a function of time, as proposed in (Song et al., 2020).

The originality of DAST is to consider these inputs in two dimensions. On the one hand, the enriched matrix X_t is given as input of the time step encoder, which encodes through self-attention scores per time point the dependency between the vectors of data at different time points. On the other hand, its transpose X_t^T is given as input to the sensor encoder which uses the same architecture to encode and capture the dependency information between the sensors. A final fusion layer finally allows to mix both encodings into a final one, which contains the importance of different combinations of sensors and time steps at the same time. That information is valuable in the context of RUL estimation and is processed by the decoder part of the architecture to obtain a prediction.

As the prediction is a scalar corresponding to $\hat{R}_t$, the model is trained using a RMSE loss, that is the square root of the mean squared prediction error when summing over all units i and time points t.

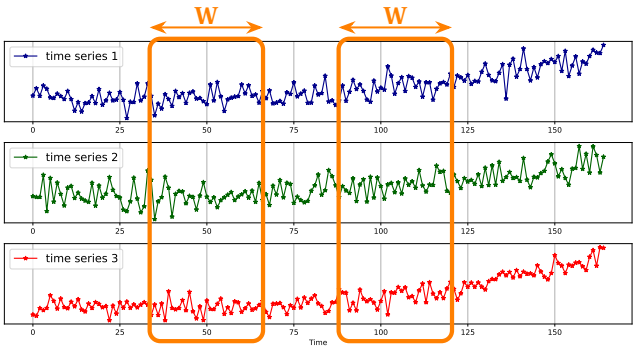


Figure 2. Example of sliding window of size W for time-series on 3 sensors

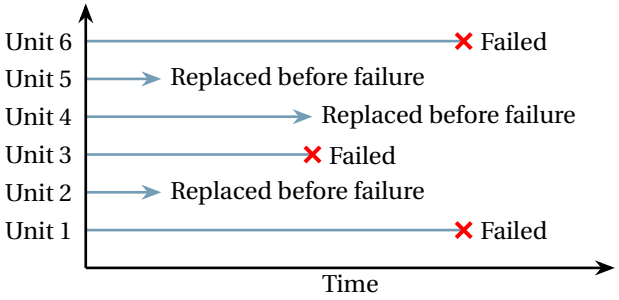


Figure 3. Right-censored data: unit 2, 4 and 5 are censored to the right, they were still healthy when replaced

2.2. Ordinal Regression for RUL Estimation with censored data

In various applications, the complete lifetime of the units is not systematically available as the components may be changed before failure, leading to right-censored lifetime. Direct RUL estimation requires the complete lifetime of the components in the learning data set and thus discards such data, which may represent most of the available data. One possible method to integrate both right-censored and uncensored lifetime data, is the ordinal regression approach developed in (Vishnu et al., 2019).

The key idea is to discretize the object to be estimated, by replacing the RUL $R_t^\star$ by a binary vector of the component status in the future. To do so, two integers L and K are fixed and the status of the unit is checked one time every L cycles (or time points in the time series). The new target is then a vector Y_t of length K where

$$y_{t,k} = \begin{cases} 0 & \text{if} \quad T > t + kL, \\ & \text{i.e. the unit is healthy after } k * L \text{ cycles}, \\ 1 & \text{if} \quad T \leq t + kL \quad \text{and} \quad T = T^\star, \\ & \text{i.e. the unit has failed before } k * L \text{ cycles}, \\ - & \text{if} \quad T \leq t + kL \quad \text{and} \quad T = C, \\ & \text{i.e. the unit status is unknown after } k * L \text{ cycles} \end{cases}$$

t is the time of the current time step and k is the index of Y_t.

Let us for example consider L = 10 and K = 10:

- if the component fails after 75 cycles, $Y_t = (0,0,0,0,0,0,0,1,1,1)$,
- if the component is replaced after 75 cycles but before failure $Y_t = (0,0,0,0,0,0,0,-,-,-)$. The last three elements are masked as no status appropriate for learning is available.

A learning phase applied on the binary vectors of the training set allows then to obtain a prediction rule, as initially proposed using an LSTM architecture (Vishnu et al., 2019). The prediction for a given unit at time t, denoted by $\hat{Y}_t$, is a vector of K probabilities indicating the probability of failure before the corresponding time steps.

As the problem has become a binary classification problem, the learning is done using binary cross-entropy (BCELoss). However, it is adjusted for right-censored data by discarding all coordinates equal to - in the Y vectors. For example, if $Y_t = (0,0,0,0,0,0,0,-,-,-)$, its contribution to the loss is only computed on the seven first coordinates. In other terms, the loss is the sum over all units i and times t of

$$\mathrm{BCE}(Y_t, \hat{Y}_t) = -\sum_{k=1}^{K} \left(y_{t,k} log(\hat{y}_{t,k}) + (1 - y_{t,k}) log(1 - \hat{y}_{t,k}) \right) \tag{2}$$

where the term in the sum is set to 0 whenever $y_{t,k}$ is masked.

2.3. Conformal prediction

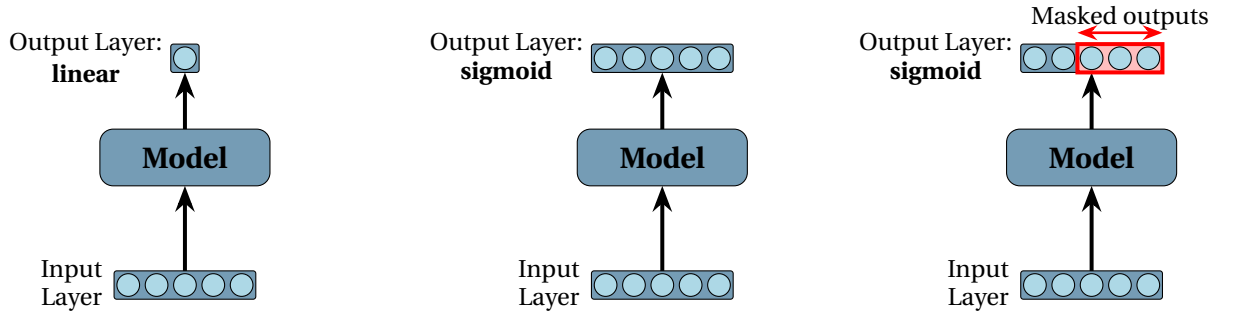
Let α be a user-defined error rate. The aim of conformal prediction for RUL prediction is to provide not only a prediction $\hat{R}_t$ of R_t, but also a prediction interval $C_\alpha(t)$ that contains R_t with a coverage rate $1-\alpha$, that is

$$P\left(R_t \in C_\alpha(t)\right) \geq 1 - \alpha.$$

The Split Conformation Prediction procedure for RUL prediction ((Javanmardi & Hüllermeier, 2023)) consists in randomly splitting the initial training data $\mathscr{D}_{\text{train}}$ into two disjoint subsets: the effective training set, again denoted by $\mathscr{D}_{\text{train}}$, and a calibration set $\mathscr{D}_{\text{calib}}$. A predictive model is trained on $\mathscr{D}_{\text{train}}$, and a non-conformity score function is defined as the absolute error of the prediction:

$$NC_t = |\hat{R}_t - R_t|.$$

The computation of the non-conformity scores NC for all time points in $\mathscr{D}_{\text{calib}}$ allows to estimate its distribution in an independent dataset, and in particular to define a critical score q as the $(1-\alpha)$-quantile of this distribution. The prediction interval for a new query point is then defined as



(a) Metric regression (b) Ordinal regression (c) Ordinal regression with censored data

Figure 4. Metric regression compared to ordinal regression

$$C_\alpha(t) = [\hat{R}_t - q, \hat{R}_t + q].$$

Under the hypothesis that the different times series can be considered as generated independently and under the same distributions, this method ensures that the probability of the true RUL being outside the interval is at most α (Angelopoulos et al., 2023).

2.4. The proposed method

A framework is considered to deal with censored data using the OR encoding with the following step:

1. Adapt the DAST architecture to the OR framework by adding a sigmoid layer, leading to the **DAST-OR** architecture. After training, it outputs a vector $(\hat{Y}_t)$ of length K for every time point in a time series.
2. Following (Chaoub, Voisin, Cerisara, & Iung, 2021) which studies LSTM for RUL prediction, a feed-forward-layer is added in the LSTM-OR architecture, between the LSTM and the sigmoid output layer.This model is denoted as **LSTM-MLP-OR**. It outputs an alternative vector $(\hat{Y}_t)$ of length K for every time point in a time series.
3. Map every vector $\hat{Y}_t$ into a predicted RUL $\hat{R}_t$, following (Vishnu et al., 2019):

$$\hat{R}_t = R_{max}(1 - \frac{1}{K}\sum_{k=1}^{K} \hat{y}_{t,k}) \quad (3)$$

   with $R_{max} = KL$ being chosen as the length of the time interval covered by $\hat{Y}_t$.
4. Select the best model in terms of RMSE loss of this RUL prediction on $\mathscr{D}_{valid}$.
5. Use conformal prediction to obtain a prediction interval.

Note that the RUL estimation introduced step 3 is of practical use, but also allows comparison with methods in the literature estimating directly the RUL.

Moreover, to reduce randomness, 10 train of each model are performed, leading to two options:

1. **The simple model:** The model with the best loss on $\mathscr{D}_{valid}$ is chosen.
2. **The ensemble model:** We consider an ensemble of models, the final prediction corresponding to the average prediction of the 6 best models among the 10 models trained.

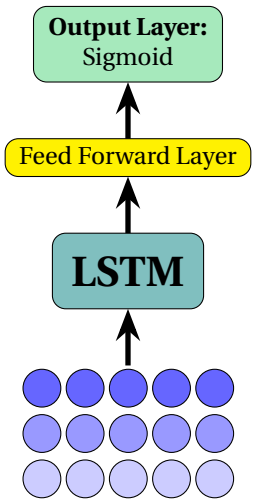


Figure 5. LSTM-MLP-OR architecture

3. EXPERIMENTAL EVALUATION

The performance of the proposed method is evaluated on the **C-MAPSS** (Commercial Modular Aero Propulsion System Simulation) dataset, which is used as a benchmark for RUL estimation methods. It simulates run-to-failure trajectories of turbofan engines (Saxena, Goebel, Simon, & Eklund, 2008) in two different operating conditions and two failure modes, leading to four sub-datasets FD001, FD002, FD003 and FD004. The characteristics of the four sets are summarized in Table 1. Each trajectory contains the following variables:

1. a unit number corresponding to the component identifier,
2. a time variable corresponding to the number of cycles performed,
3. the simulation parameters (operating condition and failure modes),
4. the simulated data from 21 sensors.

Table 1. Summary of C-MAPSS dataset

C-MAPSS sub-datasets	FD001	FD002	FD003	FD004
Train trajectories	100	260	100	249
Test trajectories	100	259	100	248
Operating condition	1	6	1	6
Fault modes	1	1	2	2

3.1. Data preprocessing and censoring

Sensors having a constant value during the experiment are removed, leaving 14 sensors for datasets FD001 and FD003, and 21 for datasets FD002 and FD004. No other feature selection is performed. Data standardization is processed on the remaining sensors by removing the mean and scaling by standard deviation.

Right-censoring is artificially added to the data, with rates $p_c \in [0\%, 20\%, 50\%, 70\%, 90\%]$. More precisely, for every censor rate p_c, the corresponding fraction of the units are randomly chosen, and, for each selected unit, the time series are truncated prior to failure at a random moment. When $p_c = 90\%$, it leads to a train set where only 10% of the units have a known RUL, and approximately 45% of the initial data is kept.

Finally, to be able to chose the best models during the training, each sub-dataset is divided into $\mathscr{D}_{\text{train}}$ and $\mathscr{D}_{\text{valid}}$, 20% of randomly chosen units joining $\mathscr{D}_{\text{valid}}$.

3.2. Trained models

Three architectures are trained on the four datasets of the CMAPSS data:

1. **DAST for RUL** (Z. Zhang et al., 2022),
2. **LSTM-MLP-OR**,
3. **DAST-OR**.

Moreover, each of them are trained ten times, and those results are used to derive a single and an ensemble model for each architecture. Ensemble models are denoted with the addition of a final E, for instance DAST-ORE for the ensemble version of DAST-OR.

We also consider the model **BEST-ORE** which is chosen between DAST-ORE and LSTM-MLP-ORE based on the RMSE on $\mathscr{D}_{\text{valid}}$.

Seven different models are thus obtained, which can be fairly compared, on exactly the same preprocessing, censoring, as well as training, validation and test sets.

For all models, $R_{max} = 130$ is used, and for the methods relying on ordinal regression, we consider Y vectors consisting on K = 13 coordinates corresponding to the status every L = 10 cycles.

3.3. Hyperparameters of the models

The hyperparameters employed for the DAST are those described in the original article (Z. Zhang et al., 2022), except for number of epochs that is set to 250 with early stopping. They are summarized table 2.

Table 2. Hyperparameters of DAST

Hyperparameter	Value
Input embedding	1 MLP layer with 64 neurons, activation: Linear
Sensor encoder	N = 2 Sensor encoder blocks with H = 4 heads
Time step encoder	N = 2 Timestep encoder blocks with H = 4 heads
Decoder	N = 1 Decoder block with H = 4 heads
Output layer	1 MLP layer with 64 neurons, activation: ReLU
Final output layer	1 MLP layer with 1 neurons, activation: Linear
Learning Rate	0.001
Batch Size	256
Dropout	0.2
Window Size	40 for FD001 and FD003, 60 for FD002 and FD004
Optimizer	Rectified Adam
Loss	RMSE

The hyperparameters for DAST-OR are identical, except for the output layer, which was modified for the OR method, and the sliding window size, which was increased for DAST-OR. Table 3 presents those which are specific to DAST-OR (sigmoid output layer and loss) or are chosen different (a manual tuning on the window size gave better results). The number of epochs is set to 500 with early stopping.

Table 3. Hyperparameters of DAST-OR

Hyperparameter	Value
Final output layer	1 MLP layer with 13 neurons, activation: Sigmoid
Window Size	60 for FD001 and FD003, 80 for FD002 and FD004
Loss	BCELoss

The hyperparameters of the LSTM-MLP-OR model mainly correspond to the article introducing LSTM-OR (Vishnu et al., 2019). However, not all parameters being explicitly detailed in the original article, manual tuning has been applied

to the LSTM-MLP-OR model with a few trials on $\mathscr{D}_{valid}$.

3.4. Results on uncensored data

This part focuses on the comparison of the results obtained on data without censoring. Table 4 shows the results for the seven trained models on each of the four datasets, with various state-of-the-art methods. Note that the seven methods are trained with the same preprocessing and separation into $\mathscr{D}_{train}$ and $\mathscr{D}_{valid}$, whereas the reported values for other methods correspond those indicated in the corresponding publications. Small variations may therefore not be significant.

On FD001, results of DAST-ORE are equivalent to the results of DAST and F+T. On FD002 results of LSTM-MLP-OR and LSTM-MLP-ORE are significantly better than the results obtain with DAST, and equivalent to results obtain with MLP+LSTM and F+T. On FD003 DAST-ORE perform better than other models of the state of the art. The results of LSTM-MLP-ORE are equivalent to the result of MLP+LSTM and F+T. On FD004 LSTM-MLP-ORE perform significantly better than other models. All the OR method proposed are significantly better than the LSTM-ORCE.

Two main conclusions can be drawn from those results. The first is that, even if OR models were designed to handle right-censored data, the obtained results on uncensored data are equivalent to those found in the literature with models specifically made for direct RUL estimation. The second interesting fact to note is the dependence on the number of operating conditions in the dataset (cf Table 1). Differences between sets FD001 and FD003, with a unique operating condition, and sets FD002 and FD004, with six different ones, are commonly found in the state of the art (C. Zhao, Huang, Li, & Yousaf Iqbal, 2020) (Sateesh Babu et al., 2016) (C. Zhang, Lim, Qin, & Tan, 2016) (Zheng et al., 2017) (X. Li, Ding, & Sun, 2018). Furthermore, the number of inputs used between is different. In this study, it appears that DAST-based methods are more powerful when the operating condition is unique, while LSTM-based outperform them when there are 6 operating conditions. Learning both architectures and keeping the best on $\mathscr{D}_{valid}$, as does BEST-ORE, is therefore useful.

3.5. Results on censored data

The results on the C-MAPSS dataset for each right-censored rate are detailed in Tables 6 and 5. The former compares the proposed ensemble methods to the ensemble LSTM-ORCE method (Vishnu et al., 2019) for the data subsets (FD001 and FD004) and censoring rates studied in that article. The train and validation sets being different, small variations should not be interpreted. However, it clearly indicates a better performance of DAST-ORE on FD001 and a significant improvement with LSTM-MLP-ORE due to the supplementary MLP layer on FD004.

Table 6 shows the results for the models listed in section 3.2 trained on the same training and validation data. For readability, BEST-ORE is not indicated, but the associated RMSE is always the lowest among the RMSEs of LSTM-MLP-ORE and DAST-ORE.

The FD001 dataset has more simple operating conditions and more simple failure modes than the other C-MAPSS sub-datasets. On FD001 the DAST-ORE model has the best RMSE for each percentage of right-censored value. With the increase of p_c, the RMSE is slowly deteriorating and reach it's worst value at $p_c = 90\%$, which is not a surprise as the learning data becomes less informative. Other models, especially LSTM-based ones, show a bigger deterioration with increasing censoring.

The results are similar on FD003, which has also only one operating condition but two failure modes. The best overall results are obtained with DAST-ORE. Moreover, the increasing of the RMSE for highly censored data is milder for DAST-ORE than for competing methods.

FD002 and FD004 are considered more complex than FD001 and FD003, because they mix several operating conditions. In both cases, the LSTM-based models outperform the DAST-based ones, as for uncensored data, with a small advantage for the LSTM-MLP-ORE ensemble method. In those two cases, the decrease of performance with growing censoring is remarkably low.

The conclusion of this study is therefore two-fold. First, the competition between LSTM and DAST-based architectures remains relevant with censored data, as different conditions may lead to different rankings of those methods. Second, OR-based methods allow to obtain a reasonable loss of performance when the real time of failure is missing for most of the learning data.

As prescribed in (Saxena et al., 2008), the results were evaluated by the RMSE on the predictions of the last time-point of the time-series in the test set. To illustrate more visually the results of the different methods, Figure 6 provides some plots of the predictions of the ensemble methods for randomly picked time series on different datasets and censoring rates.

3.6. Asymmetric score evaluation

Prediction on C-MAPSS should also be evaluated by the asymetric score (Saxena et al., 2008) defined by

$$\text{Score} = \begin{cases} e^{\frac{\hat{\mathrm{R}}_t - \mathrm{R}_t}{13}} - 1 & \text{if } \hat{\mathrm{R}}_t - \mathrm{R}_t \geq 0 \\ e^{-\frac{\hat{\mathrm{R}}_t - \mathrm{R}_t}{13}} - 1 & \text{if } \hat{\mathrm{R}}_t - \mathrm{R}_t < 0 \end{cases}$$

Table 4. RMSE results on the C-MAPSS dataset without censoring

Model	**FD001**	**FD002**	**FD003**	**FD004**	Average RMSE
LSTM-MLP-OR	14.24	**12.00**	17.27	15.35	14.94
LSTM-MLP-ORE	13.20	12.77	13.84	14.75	13.64
DAST	12.35	16.48	13.43	19.89	15.54
DAST-E	12.22	15.44	12.89	16.14	14.17
DAST-OR	12.16	15.62	9.64	16.20	13.41
DAST-ORE	11.57	15.55	**8.54**	18.01	13.42
BEST-ORE	11.57	12.77	**8.54**	14.75	**11.91**
DAST (Z. Zhang et al., 2022)	**11.43**	15.25	11.32	18.36	14.09
LSTM-ORCE (Vishnu et al., 2019)	14.62	-	-	27.47	-
MLP+LSTM (Chaoub et al., 2021)	13.26	12.49	13.11	**13.97**	13.21
F+T (Lai, Liu, Pan, & Chen, 2022)	**11.43**	13.32	11.47	14.38	12.65

Table 5. Results RMSE on C-MAPSS

	FD001					
p_c	LSTM-MLP-OR	LSTM-MLP-ORE	DAST	DAST-E	DAST-OR	DAST-ORE
0%	14.24	13.20	12.35	12.22	12.16	**11.57**
20%	15.42	14.01	13.69	12.59	12.73	**12.51**
50%	15.09	15.96	15.41	13.37	13.39	**12.99**
70%	17.83	17.97	15.38	14.08	14.28	**12.51**
90%	30.02	26.76	16.78	17.17	17.01	**15.80**
	FD002					
p_c	LSTM-MLP-OR	LSTM-MLP-ORE	DAST	DAST-E	DAST-OR	DAST-ORE
0%	**12.00**	12.77	16.48	15.44	15.62	15.55
20%	15.43	**13.01**	14.09	13.80	16.37	18.51
50%	13.71	**13.15**	15.08	14.18	15.39	16.58
70%	14.24	**13.24**	16.10	14.74	16.71	17.73
90%	16.44	**13.61**	15.85	15.08	25.23	17.00
	FD003					
p_c	LSTM-MLP-OR	LSTM-MLP-ORE	DAST	DAST-E	DAST-OR	DAST-ORE
0%	17.27	13.84	13.43	12.89	9.64	**8.54**
20%	15.69	12.80	13.55	12.53	10.03	**8.81**
50%	13.97	13.46	16.04	12.57	11.69	**10.14**
70%	21.72	21.13	20.88	15.32	13.46	**12.20**
90%	38.74	30.66	22.34	22.88	19.59	**16.09**
	FD004					
p_c	LSTM-MLP-OR	LSTM-MLP-ORE	DAST	DAST-E	DAST-OR	DAST-ORE
0%	16.23	**14.75**	19.89	16.14	16.20	18.01
20%	15.66	**14.42**	18.32	15.23	18.01	16.93
50%	16.00	**14.67**	17.46	15.66	17.43	19.49
70%	16.59	**15.11**	17.32	17.10	14.84	20.83
90%	18.85	**15.47**	19.79	17.21	22.41	22.14

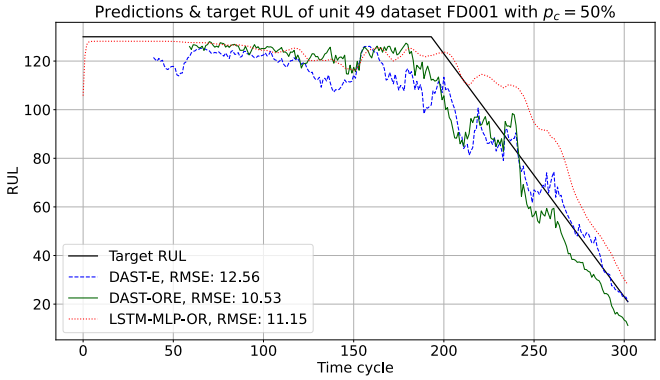
Table 6. RMSE results on FD001-FD004 with censor

p_c	LSTM-MLP-ORE	DAST-ORE	LSTM-ORCE (Vishnu et al., 2019)
	FD001		
50%	15.96	**12.99**	15.98
70%	17.97	**12.51**	16.57
90%	26.76	**15.80**	20.38
	FD004		
50%	**14.67**	19.49	30.62
70%	**15.11**	20.83	31.27
90%	**15.47**	22.14	38.41

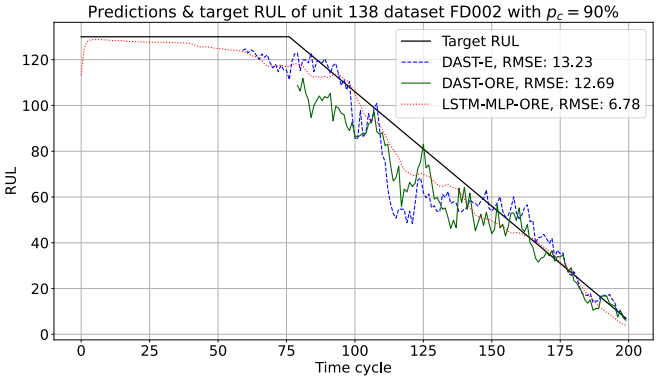
That score corresponds to a higher penalty for overestimation rather underestimation of the RUL.

Table 7 shows the scores for the ensemble methods DAST-E, DAST-ORE and LSTM-ORE. If the results are rather coherent with the RMSE comparisons for FD001 and FD003, the advantage of LSTM-based methods compared to DAST-E is less clear for FD002 and FD004.

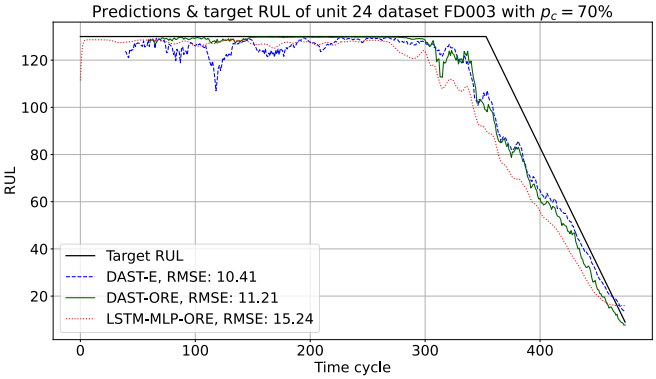
However, it has to be noted that the OR-based methods were trained with a symmetric BCELoss which does not take into consideration a different penalty for over and under-estimations. In terms of binary vectors, it means a higher penalty for a close to 0 coordinate in $\hat{Y}_t$ when the truth is 1 (the fan is predicted running when it actually failed, which is an over-estimation of the RUL) than for a prediction close to 1 when the truth is 0.



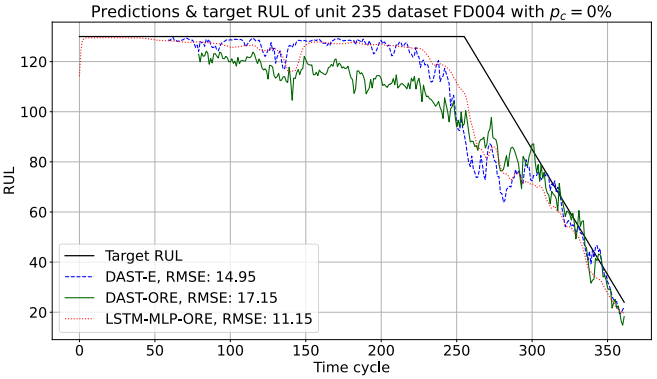
(a) Results of unit 49 (FD001) with $p_c = 50\%$



(b) Results of unit 138 (FD002) with $p_c = 90\%$



(c) Results of unit 24 (FD003) with $p_c = 70\%$



(d) Results of unit 235 (FD004) with $p_c = 0\%$

Figure 6. Example of results on one units of each sub-dataset

A possibility to introduce this asymmetry would be to consider a modified loss by replacing Equation 2 by

$$\mathrm{BCE}(\mathrm{Y}_t, \hat{\mathrm{Y}}_t) = -\sum_{k=1}^{\mathrm{K}} \left(\alpha y_{t,k} log(\hat{y}_{t,k}) + (1 - y_{t,k}) log(1 - \hat{y}_{t,k})\right) \tag{4}$$

where $\alpha > 1$ is a hyperparameter to be optimized.

3.7. Conformal prediction

The Conformal Prediction (CP) framework is applied to our models DAST, DAST-OR, and LSTM-OR on the C-MAPSS dataset. To achieve this, the inital training dataset is divided into three subsets: $\mathscr{D}_{\text{train}}$, $\mathscr{D}_{\text{calib}}$, and $\mathscr{D}_{\text{valid}}$. The three sets contain respectively 70%, 10% and 20% of the initial training data.

The models are trained on $\mathscr{D}_{\text{train}}$, confidence intervals are constructed using $\mathscr{D}_{\text{calib}}$ by evaluating the quantiles of order α on the prediction errors calculated on $\mathscr{D}_{\text{calib}}$, and the models are selected based on their performance scores on $\mathscr{D}_{\text{valid}}$. Note that the quantiles for different values of α can be evaluated directly on $\mathscr{D}_{\text{calib}}$ without the need to retrain the models.

Table 8 presents the coverage results of the best ORE model on the CMAPSS dataset across the four subsets FD001, FD002, FD003, and FD004. The parameter $1-\alpha$ represents the theoretical or expected coverage rate. The table is structured by varying the proportion parameter p_c and the confidence level $1-\alpha$. The bold values indicate cases where the observed coverage exceeds $1-\alpha-0.05$, demonstrating good alignment with the theoretical expectations.

The results demonstrate that the ORE model performs robustly, with higher values of $1-\alpha$ and p_c contributing to improved observed coverage. However, the performance varies across datasets, which may reflect differences in model performance or dataset characteristics.

While achieving a coverage rate on the test data that is close to $1-\alpha$ is desirable, it is not sufficient to ensure model performance. A trivially large interval size relative to the studied problem would guarantee a high coverage rate but would result in an impractical and ineffective model. Furthermore, the method used to compute prediction intervals implies that larger model errors lead to wider intervals, which in turn increases the coverage rate. As a result, a high coverage rate alone does not necessarily indicate a well-calibrated or

Table 7. Score results on C-MAPSS

	FD001		
p_c	LSTM-MLP-ORE	DAST-E	DAST-ORE
0%	341.00	**201.17**	206.48
20%	410,63	**232,49**	269,68
50%	631,91	**252,89**	296,04
70%	1279,51	458,45	**261,79**
90%	4566,18	808,5	**500,21**
	FD002		
p_c	LSTM-MLP-ORE	DAST-E	DAST-ORE
0%	708,28	**638,34**	978,48
20%	753,57	**531,06**	1891,21
50%	751,84	**544,1**	1412,07
70%	786,58	**647,39**	1362,25
90%	860,19	**849,25**	1286,1
	FD003		
p_c	LSTM-MLP-ORE	DAST-E	DAST-ORE
0%	322,06	206,94	**103,65**
20%	250,9	192,28	**111,17**
50%	267,75	223,95	**143,29**
70%	1611,57	437,85	**272,35**
90%	3100,45	2797,84	**447,64**
	FD004		
p_c	LSTM-MLP-ORE	DAST-E	DAST-ORE
0%	**1741,67**	2262,98	2739,62
20%	1772,3	**1518,44**	2591,59
50%	**1434,33**	2206,4	2788,02
70%	**2096,04**	2470,12	3863,9
90%	1689,67	**1194,16**	2903,72

Table 8. Coverage Results on CMAPSS

		Best ORE Model Coverage			
p_c	$1-\alpha$	**FD001**	**FD002**	**FD003**	**FD004**
0%	0.50	0.34	**0.72**	0.20	**0.62**
0%	0.75	0.61	**0.92**	0.33	**0.82**
0%	0.95	**0.91**	**0.98**	0.60	**0.97**
20%	0.50	0.44	**0.58**	0.32	**0.60**
20%	0.75	0.67	**0.83**	0.66	**0.78**
20%	0.95	**0.91**	**0.95**	**0.98**	**0.96**
50%	0.50	0.40	**0.46**	**0.56**	**0.63**
50%	0.75	0.64	**0.95**	**0.81**	**0.83**
50%	0.95	**0.91**	**0.95**	**0.97**	**0.97**
70%	0.50	**0.50**	**0.46**	**0.51**	**0.56**
70%	0.75	**0.70**	**0.77**	**0.70**	**0.77**
70%	0.95	**0.94**	**0.95**	**0.93**	**0.97**
90%	0.50	**0.51**	**0.54**	**0.46**	**0.65**
90%	0.75	0.65	**0.77**	0.61	**0.83**
90%	0.95	0.82	**0.95**	0.73	**0.97**

useful model.

Figure 7 shows the interval width for different models and datasets. The first and last point of the green curve on FD001 can be read as the fact that a prediction by DAST-ORE in that condition has a nominal rate of 95% to be less than 28 hours away from the true RUL, and of 50% to be less than 10 hours away. The figure shows that the predicted interval size is decreasing with α, as expected. However, interval sizes considerably vary between models and datasets.

3.8. The N-CMAPSS dataset

N-CMAPSS ((Arias Chao, Kulkarni, Goebel, & Fink, 2021)) is an enhanced and expanded version of the CMAPSS dataset, aimed to simulate more realistic data for aircraft engines. As a next-generation version, it includes additional data to make the simulations more robust and closer to real-world conditions.

N-CMAPSS introduces several improvements, including the inclusion of new engine configurations, more diverse simulated failures, and more complex environments to train predictive models. Moreover, N-CMAPSS consists of less than ten units per condition, each unit containing a large amount of data. In contrast, C-MAPSS comprises several hundred units per condition, with each unit containing a relatively smaller amount of data.

The same preprocessing applied to C-MAPSS is applied to N-CMAPSS. Right-censoring is artificially applied to the data, with censoring rates $p_c \in [0\%, 20\%, 50\%, 70\%, 90\%]$, following the same procedure as used in the C-MAPSS dataset.

Three architectures are trained on the four datasets of the N-CMAPSS data:

1. **DAST-E for RUL** (Z. Zhang et al., 2022),
2. **LSTM-MLP-ORE**,
3. **DAST-ORE**.

The exact same procedure as used for the C-MAPSS dataset is followed, and the same hyperparameters are applied.

	N-CMAPSS DS02		
p_c	DAST-E	DAST-ORE	LSTM-ORE
0%	6.24	7.16	13.82
20%	7.42	8.27	14.35
50%	11.92	14.44	17.07
66%	6.20	8.04	17.08

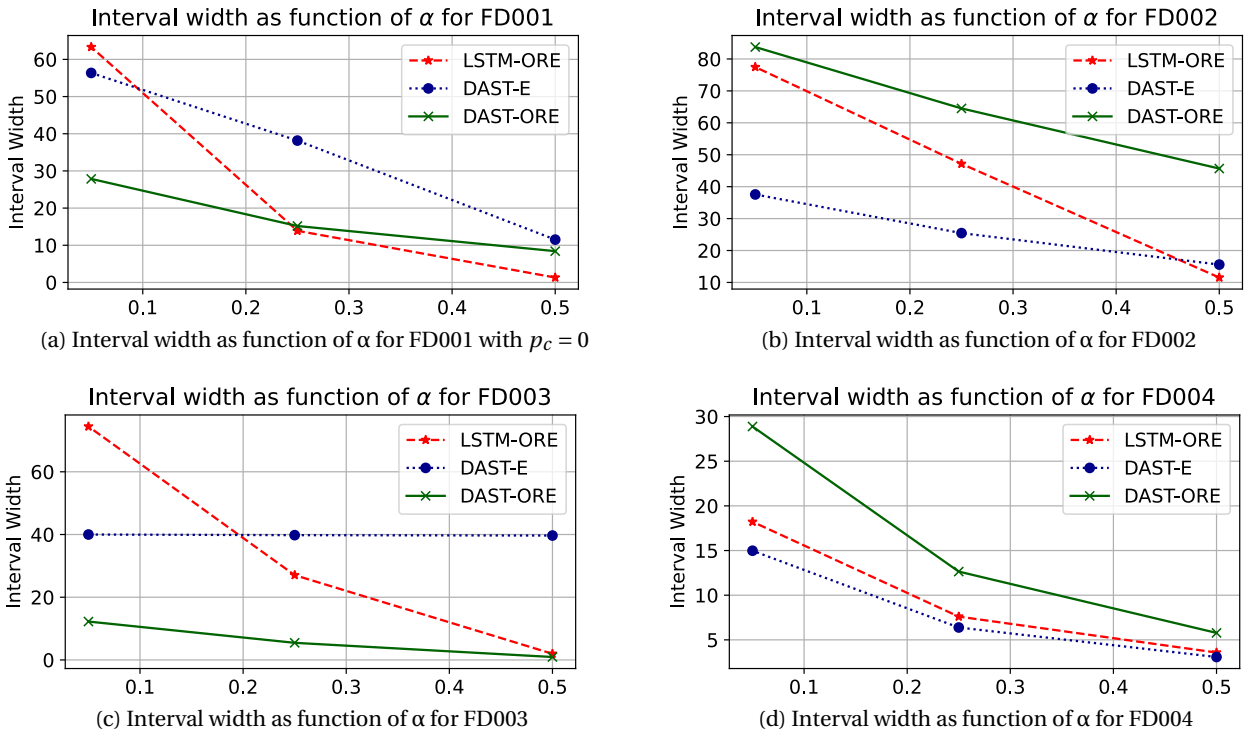
Table 9. Comparison of RMSE between different models for DS02

Table 9 shows the RMSE results on the condition DS02 of N-CMAPSS, consisting of 6 units. On this dataset, DAST-E is the best model, even for censored data.

This observation may be explained by the fact that the discretisation step in OR methods implies some loss of information, which is hopefully compensated by the gain represented by the censored data. However, for such small datasets, the number of censored units remains very low, and the gain of information is not sufficient. This finding suggests that, in order to improve performance by taking censored data into account and using ORE models, a sufficiently large number of censored units is necessary.

4. CONCLUSION

This work addresses the challenge of estimating the Remaining Useful Life (RUL) of industrial components from time series data with no prior physical model of the system and a high rate of censored data. It does so by considering two



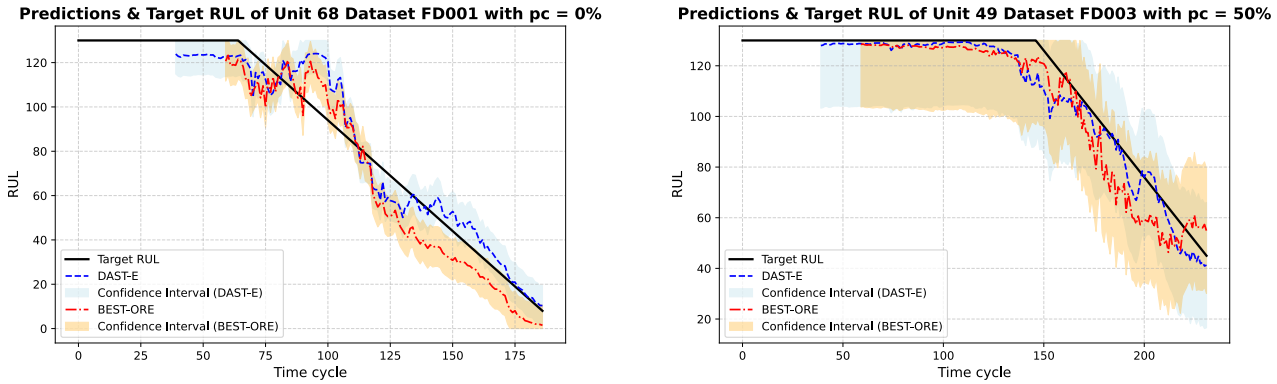
(a) Interval width as function of α for FD001 with $p_c = 0$

(b) Interval width as function of α for FD002

(c) Interval width as function of α for FD003

(d) Interval width as function of α for FD004

Figure 7. Interval width as function of α for each CMAPSS sub-dataset for $p_c = 90\%$, the interval width is expressed in time cycle.



(a) Results predicted RUL and true RUL with confidence interval on unit 8 (FD001)

(b) Results predicted RUL and true RUL with confidence interval on unit 72 (FD003)

Figure 8. Results predicted RUL and true RUL with confidence interval ($1 - \alpha = 0.75$) on the C-MAPSS dataset

data-driven deep-learning architectures relying on the ordinal regression approach introduced in (Vishnu et al., 2019) for RUL estimation. One of them is an improved version of the LSTM-OR method by (Vishnu et al., 2019), the second is an adaptation to censored data of the DAST model introduced in (Z. Zhang et al., 2022). Moreover, the application of conformal prediction allows to obtain prediction intervals for the presented models.

These approaches are shown to perform as well on the C-MAPSS data as the existing direct RUL estimation methods found in the literature on uncensored data, and better on censored data. Their application on the N-CMAPSS dataset however highlights that they are ineffective if only a handful of censored data are available.

Furthermore, the two proposed architectures are shown to be complementary, as they outperform each other depending on the complexity of the dataset. Therefore, in the context of estimating the lifespan of components, it is interesting to put them in competition, considering that this approach should yield favorable results regardless of the complexity of the data and the rate of right-censored data.

From a computational perspective, such models take some time to be learnt but are then of immediate application for new data. The OR method significantly increases training time due to its underlying mechanism, which involves transforming the RUL prediction task into a multi-class classification problem with ordinal targets. This requires computing and optimizing multiple output probabilities per time step and testing each target vector to determine its size and the appropriate truncation point for the prediction 2.2. This added complexity leads to higher computational demands. As a result, DAST-OR is five to ten times slower than DAST, while LSTM-OR is approximately twenty to twenty-five times slower. A reasonable deployment in industry would consist in a periodical centralized learning given all gathered data giving raise to regular updates for the prediction model.

Finally, the present article focused on right censored data, which corresponds to the missing data situation created by preventive maintenance. However, (Yang, Kanniainen, Krogerus, & Emmert-Streib, 2022b) highlights the presence of various types of censoring (left, interval, right) in maintenance data, which complicates statistical analysis and RUL estimation. The ordinal regression framework could easily be extended to the case of gaps in the learning data. The vector to predict could then be of the form $Y_t = (0, 0, -, 0, -, 0, 0, 1, 1, 1)$ for example, the $-$ representing missing data, and the error defined as previously only on the observed data segments. Studying the influence of the missing data rate and time window size in that case is an open research question.

CODE AVAILABILITY

The code was written in Pytorch and is available at https://gitlab.math.unistra.fr/jnoot/rul_estimation_cmapss

REFERENCES

Aggarwal, K., Atan, O., Farahat, A. K., Zhang, C., Ristovski, K., & Gupta, C. (2018). Two birds with one network: Unifying failure event prediction and time-to-failure modeling. In *2018 ieee international conference on big data (big data)* (pp. 1308–1317).

Angelopoulos, A. N., Bates, S., & al. (2023). Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, *16*(4), 494–591.

Arena, F., Collotta, M., Luca, L., Ruggieri, M., & Termine, F. G. (2021). Predictive maintenance in the automotive sector: A literature review. *Mathematical and Computational Applications*, *27*(1), 2.

Arias Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, *6*(1), 5.

Benker, M., Bliznyuk, A., & Zaeh, M. F. (2021). A gaussian process based method for data-efficient remaining useful life estimation. *IEEE Access*, *9*, 137470–137482.

Boutrous, K., Bessa, I., Puig, V., Nejjari, F., & Palhares, R. M. (2022). Data-driven prognostics based on evolving fuzzy degradation models for power semiconductor devices. In *Phm society european conference* (Vol. 7, pp. 68–77).

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

Casenave, F. (2024). Learning large-scale industrial physics simulations. *European Mathematical Society Magazine*(134), 5–13.

Chaoub, A., Voisin, A., Cerisara, C., & Iung, B. (2021). Learning representations with end-to-end models for improved remaining useful life prognostics. *arXiv preprint arXiv:2104.05049*.

Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S., & Syntetos, A. A. (2020). Predictive maintenance using cox proportional hazard deep learning. *Advanced Engineering Informatics*, *44*, 101054.

Chen, C., Shi, J., Shen, M., Feng, L., & Tao, G. (2023). A predictive maintenance strategy using deep learning quantile regression and kernel density estimation for failure prediction. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–12.

Garay, J. M., & Diedrich, C. (2019). Analysis of the applicability of fault detection and failure prediction based on unsupervised learning and monte carlo simulations for real devices in the industrial automobile production. In *2019 ieee 17th international conference on industrial informatics (indin)* (Vol. 1, pp. 1279–1284).

Groot, P., & Lucas, P. (2012). Gaussian process regression with censored data using expectation propagation. In *6th european workshop on probabilistic graphical models, pgm 2012* (pp. 115–122).

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management* (pp. 1–6).

Javanmardi, S., & Hüllermeier, E. (2023). Conformal prediction intervals for remaining useful lifetime estimation. *arXiv preprint arXiv:2107.07511*.

Jean-Pierre, N., Birmelé, E., & François, R. (2024). Lstm and transformers based methods for remaining useful life

prediction considering censored data. In *Phm society european conference* (Vol. 8, pp. 10–10).

Jeong, K., & Choi, S. (2019). Model-based sensor fault diagnosis of vehicle suspensions with a support vector machine. *International Journal of Automotive Technology*, *20*, 961–970.

Jia, J., Liang, J., Shi, Y., Wen, J., Pang, X., & Zeng, J. (2020). Soh and rul prediction of lithium-ion batteries based on gaussian process regression with indirect health indicators. *Energies*, *13*(2), 375.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, *18*, 1–12.

Kong, Y., Abdullah, S., Schramm, D., Omar, M., & Haris, S. (2019). Development of multiple linear regression-based models for fatigue life evaluation of automotive coil springs. *Mechanical Systems and Signal Processing*, *118*, 675–695.

Lai, Z., Liu, M., Pan, Y., & Chen, D. (2022). Multi-dimensional self attention based approach for remaining useful life estimation. *arXiv preprint arXiv:2212.05772*.

Li, H., Zhao, W., Zhang, Y., & Zio, E. (2020). Remaining useful life prediction using multi-scale deep convolutional neural network. *Applied Soft Computing*, *89*, 106113.

Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, *172*, 1–11.

Lillelund, C. M., Pannullo, F., Jakobsen, M. O., Morante, M., & Pedersen, C. F. (2024). A probabilistic estimation of remaining useful life from censored time-to-event data. *arXiv preprint arXiv:2405.01614*.

Liu, J., & Chen, Z. (2019). Remaining useful life prediction of lithium-ion batteries based on health indicator and gaussian process regression model. *Ieee Access*, *7*, 39474–39484.

Patil, S., Patil, A., Handikherkar, V., Desai, S., Phalle, V. M., & Kazi, F. S. (2018). Remaining useful life (rul) prediction of rolling element bearing using random forest and gradient boosting technique. In *Asme international mechanical engineering congress and exposition* (Vol. 52187, p. V013T05A019).

Rahat, M., Kharazian, Z., Mashhadi, P. S., Rögnvaldsson, T., & Choudhury, S. (2023). Bridging the gap: A comparative analysis of regressive remaining useful life prediction and survival analysis methods for predictive maintenance. In *Phm society asia-pacific conference* (Vol. 4).

Sateesh Babu, G., Zhao, P., & Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Database systems for advanced applications: 21st international conference, dasfaa 2016, dallas, tx, usa, april 16-19, 2016, proceedings, part i 21* (pp. 214–228).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).

Song, Y., Gao, S., Li, Y., Jia, L., Li, Q., & Pang, F. (2020). Distributed attention-based temporal convolutional network for remaining useful life prediction. *IEEE Internet of Things Journal*, *8*(12), 9594–9602.

Tinga, T., & Loendersloot, R. (2019). Physical model-based prognostics and health monitoring to enable predictive maintenance. *Predictive Maintenance in Dynamic Systems: Advanced Methods, Decision Support Tools and Real-World Applications*, 313–353.

Vasavi, S., Aswarth, K., Pavan, T. S. D., & Gokhale, A. A. (2021). Predictive analytics as a service for vehicle health monitoring using edge computing and ak-nn algorithm. *Materials Today: Proceedings*, *46*, 8645–8654.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vishnu, T., Malhotra, P., Vig, L., & Shroff, G. (2019). Data-driven prognostics with predictive uncertainty estimation using ensemble of deep ordinal regression models. *International Journal of Prognostics and Health Management*, *10*(4).

Wang, Y., Zhao, J., Yang, C., Xu, D., & Ge, J. (2022). Remaining useful life prediction of rolling bearings based on pearson correlation-kpca multi-feature fusion. *Measurement*, *201*, 111572.

Yang, Z., Kanniainen, J., Krogerus, T., & Emmert-Streib, F. (2022a). Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Scientific Reports*, *12*(1), 8529. Retrieved from https://doi.org/10.1038/s41598-022-12572-z doi: 10.1038/s41598-022-12572-z

Yang, Z., Kanniainen, J., Krogerus, T., & Emmert-Streib, F. (2022b). Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Scientific reports*, *12*(1), 8529.

Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2016). Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE transactions on neural networks and learning systems*, *28*(10), 2306–2318.

Zhang, T., & Wang, H. (2024). Quantile regression network-based cross-domain prediction model for rolling bearing remaining useful life. *Applied Soft Computing*, *159*, 111649.

Zhang, Z., Song, W., & Li, Q. (2022). Dual aspect self-attention based on transformer for remaining useful

life prediction. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–11.

Zhao, C., Huang, X., Li, Y., & Yousaf Iqbal, M. (2020). A double-channel hybrid deep neural network based on cnn and bilstm for remaining useful life prediction. *Sensors*, *20*(24), 7109.

Zhao, Y., Liu, P., Wang, Z., & Hong, J. (2017). Electric vehicle battery fault diagnosis based on statistical method. *Energy Procedia*, *105*, 2366–2371.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 ieee international conference on prognostics and health management (icphm)* (pp. 88–95).