Bridging the Gap: A Comparative Analysis of Regressive Remaining Useful Life Prediction and Survival Analysis Methods for Predictive Maintenance

Mahmoud Rahat[1], Zahra Kharazian[2], Peyman Sheikholharam Mashhadi[1], Thorsteinn Rögnvaldsson [1], and Shamik Choudhury[3]

[1] *Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden*
{mahmoud.rahat, peyman.mashhadi, thorsteinn.rognvaldsson}@hh.se

[2] *Department of Computer and System Science (DSV), Stockholm University, Stockholm, Sweden*
zahra.kharazian@dsv.su.se

[3] *Volvo Group Trucks Technology Gothenburg, Vastra Gotaland County, Sweden*
shamik.choudhury@consultant.volvo.com

ABSTRACT

Regressive Remaining Useful Life Prediction and Survival Analysis are two lines of research with similar goals but different origins; one from engineering and the other from survival study in clinical research. Although the two research paths share a common objective of predicting the time to an event, researchers from each path typically do not compare their methods with methods from the other direction. Given the mentioned gap, we propose a framework to compare methods from the two lines of research using run-to-failure datasets. Then by utilizing the proposed framework, we compare six models incorporating three widely recognized degradation models along with two learning algorithms. The first dataset used in this study is C-MAPSS which includes simulation data from aircraft turbofan engines. The second dataset is real-world data from streamed condition monitoring of turbocharger devices installed on a fleet of Volvo trucks.

1. INTRODUCTION

In prognostics and health management (PHM), accurately predicting the failure time of devices is paramount. Estimating functional lifetime can help industries minimize downtime, optimize their maintenance plans and resource allocation, thereby reducing costs. Different statistical and machine learning techniques can be used to achieve this goal, two most important of which are regressive Remaining Useful Life (RUL) prediction (Altarabichi et al., 2020; Karlsson et al., 2023) and Survival Analysis (SA) (Alabdallah, Ohlsson, Pashami, & Rögnvaldsson, 2022).

The Survival Analysis originates from the medical domain, and the regressive Remaining Useful Life (RUL) prediction originates from the Through-life Engineering Services field. Although with completely different origins, both approaches are applicable to Predictive Maintenance with slightly different problem formulations. Survival models generate survival curves, also referred to as survival functions, and are primarily developed to handle censored data. Censored data represents samples that have not encountered the event of interest, such as failure, within the study period. For instance, (Voronov, Frisk, & Krysander, 2018) used survival analysis to predict the battery lifetime in heavy-duty trucks due to the dataset's high censoring rate (80%). Another example can be seen in (Yang, Kanniainen, Krogerus, & Emmert-Streib, 2022), where the survival analysis is used to estimate mobile work assets' survival probabilities and hazard functions. They used Kaplan Meier from non-parametric methods to find the survival functions that provide information about the remaining useful life of assets. On the other hand, RUL prediction models usually ignore censored data and settle for point-wise lifetime estimation from regressors (Rahat et al., 2022). Given the differences in the problem formulation, the researchers in each field tend to consider only studies in the same field, and there are few attempts to cross borders and compare models from both fields.

In order to bridge the identified gap, we propose a straightforward yet impactful framework for transforming run-to-failure historical data, which is a commonly used data format in remaining useful life (RUL) prediction, into a Survival Analysis formulation. Then we compare six models with different configurations. Two learning models, namely Random Forest (RF) and Gradient Boosting Trees (GBT), are utilized to train both regression and survival models. Additionally, two configurations are taken into account for the regression models, involving linear and piecewise linear degradations.

Due to the existence of censored samples in survival analysis, it is not possible to use most of the standard machine learning evaluation measures. Alternatively, Harrell's Concordance index (C-index) is commonly used since it can consider censored samples by limiting the evaluations to the possible pairs. However, it is known that C-index is susceptible to various bises (Hartman, Kim, He, & Kalbfleisch, 2023; Alabdallah et al., 2022) e.g., it has an upward bias when there is a high degree of censoring in the test data (Uno, Cai, Pencina, D'Agostino, & Wei, 2011). Considering the issues with C-index, in this work, we resort to evaluate the models using Mean Absolute Error (MAE) which is inherently an unbiased measure. The application of MAE to measure the performance of regressive RUL models is straightforward. However, to be able to apply this measure to the output of the survival models, we compare the actual failure time of the units with the median survival time calculated from the projected survival curve. This procedure is explained in detail in the methodology section.

The following describes our two main research questions:

1. What are the performance differences among the RUL prediction and Survival Analysis models, incorporating three widely recognized degradation models and two learning algorithms, in predicting time to an event?
2. How does changing the number of censored samples impact the models' performance?

2. Problem Formulation and Methodology

Assume $X_t^u \in R^D$ is an observation where D is the number of covariants, $t \in T$ indicates time, and $u \in U$ represents the unit number. We consider temporal data from a machine as stand-alone observations; therefore, each row of the data is an independent data point regardless of its unit number. Given that without loss of generality, we can reformulate a data point as X_i where i represents the index of that data point in the dataset. Furthermore, $t_i^{readout}$ and $t_i^{failure}$ represent readout time and failure time for the i^{th} data point, respectively. Consequently, we define:

$$t_i^{lifetime} = t_i^{failure} - t_i^{readout} \tag{1}$$

and

$$t_i^{observed} = min(t_i^{lifetime}, EOS - t_i^{readout}) \tag{2}$$

Where $t_i^{lifetime} \in T$ illustrates the lifetime of a sample (also known as remaining useful life), $EOS \in T$ is an arbitrary end of study time, and $t_i^{observed} \in T$ is the observed time for the i^{th} observation according to the EOS, and min is a function that returns the smallest input value.

Algorithm 1 Converting Survival Curve to RUL

```
Require:
    $X_i^{test}$ - test sample
    $SurvivalModel$ - previously trained survival model
Ensure:
    $t'$ - predicted remaining useful life where $t' \in R^+$
1: $S(t) \leftarrow$SurvivalModel($X_i^{test}$)
2: if $S(t)$ does not include 0.5 then
3:     $S(t) \leftarrow Extrapolate(S(t))$
4: end if
5: Return $t'$ where $s(t') = 0.5$
```

We formulate the data points for the regression model using (X_i, y_i) where $y_i = t_i^{lifetime}$. On the other hand, similar to (Fotso et al., 2019) the survival analysis data points are characterized as (X_i, e_i, y_i) where $\forall i$:

- X_i is a D dimensional feature vector.
- e_i is event indicator such that $e_i = 1$, if we observe the event and $e_i = 0$, in case of censoring.
- $y_i = t_i^{observed}$.

We further formulate the target regression y_i for the two degradation models (linear, piecewise), as well as the survival model as follows:

$$y_i^{linear} = \begin{cases} t_i^{lifetime}, & if \quad t_i^{lifetime} \leq EOS \\ \emptyset, & otherwise \end{cases} \tag{3}$$

The linear degradation model requires an exact target value. As a result, it ignores censored samples with a lifetime greater than the end of the study, as shown in Eq. (3).

$$y_i^{piecewise} = \begin{cases} t_i^{lifetime}, & if \quad t_i^{lifetime} \leq EOS \\ EOS, & otherwise \end{cases} \tag{4}$$

While the piecewise degradation model includes censored samples by considering an early RUL value equal to *EOS* (see Eq. (4)).

$$y_i^{survival} = t_i^{observed} \tag{5}$$

The value of target regression y_i for survival model is shown in Eq. (5) which is equal to $t_i^{observed}$. This value can be calculated from Eq. (2).

$$e_i^{survival} = \begin{cases} 1, & if \quad t_i^{lifetime} \leq EOS \\ 0, & otherwise \end{cases} \tag{6}$$

Finally, Eq. (6), $e_i^{survival}$ provides the event indicator flag required for training the survival models.
The most common output of the survival analysis is called survival curve $S(t)$, or survival function, which represents the probability that the event of interest has not occurred by some time t.

In order to facilitate a comparison between the outcomes of a regression and a survival analysis model, we calculate the median survival time of the input sample from its estimated survival curve. This is equivalent to the point where the survival function crosses 0.5, i.e. the probability of failure becomes greater than the probability of surviving (Goel, Khanna, & Kishore, 2010).

The procedure for this conversion is provided in Algorithm 1, where we first predict the survival function for a test sample. Then we extrapolate the curve if it does not cross the median point. Lastly, we find the corresponding point when the survival curve crosses the median point and return the mentioned point in time as the estimated remaining useful life value.

Figure 1 visualizes a projected survival curve. Note that the curve concludes prior to 0.5 (curve fraction), which happens due to considering a specific study period (EOS) and the fact that the survival models do not extrapolate beyond the duration of the study period. In such cases, we extrapolate our survival curve using "xgbse.extrapolation" implementation (Vieira, Gimenez, Marmerola, & Estima, 2021), which considers a constant risk extrapolation strategy.

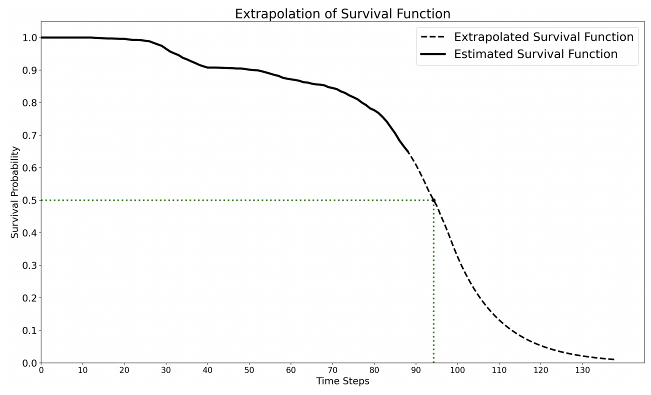

Figure 1. Extrapolation

3. Dataset

In this study, two time-series run-to-failure datasets from the field of prognostics and health management (PHM) have been selected. The first dataset is publically available and contains synthetic degradation data from commercial turbofan engines simulated by NASA (Saxena, Goebel, Simon, & Eklund, 2008). Within the provided dataset known as C-MAPSS, multiple engines are operated concurrently. If the engine's deterioration surpasses a certain threshold, it is considered a failure. The original dataset contains four distinct experiments labeled *FD001* to *FD004* with varying fault modes and conditions. In each experiment, both training and test data sets were gathered. In this work, we only used data from one of the experiments (*FD001*) and kept the original train test split.

The second dataset contains operational signal measurements showing the degradation of 461 Turbocharger devices installed on a fleet of Volvo trucks. The dataset has 30746 samples and 264 columns. The dataset is recorded in a similar format to the C-MAPSS dataset, i.e. each sample represents one readout (observation) from a truck at a specific time. For the rest of the paper, we will refer to this dataset as the *Turbocharger* dataset. It is not possible for us to release the Turbocharger dataset for confidentiality reasons, but the code used for running the experiments for both Turbocharger and the C-MAPSS dataset is the same.

A turbocharger plays an essential role in boosting engine power and efficiency by increasing air intake and improving combustion. When a turbocharger fails, engine performance is compromised, resulting in reduced power output, decreased fuel efficiency, and increased emissions. Furthermore, this decline in performance often leads to complete stoppages, as it is unsafe to operate a loaded truck with a faulty turbocharger. Predicting turbocharger failure ensures uninterrupted system performance, optimizing power output, and maintaining optimal fuel economy. Several prior studies investigated the predictive maintenance of turbocharger devices (Revanur, Ayibiowu, Rahat, & Khoshkangini, 2020; Rahat, Pashami, Nowaczyk, & Kharazian, 2020; Rahat et al., 2022).

3.1. Considering Temporal Information of Dataset

Run-to-failure is a time-series data including temporal information. There are various ways to include temporal information in the model, e.g., windowing. Here, we treat each machine's readouts as a standalone independent sample. As an example, assume there are 100 units in a dataset, and each unit has a fixed temporal length of 20 i.e., we have 20 readouts per unit over time. Considering each readout as a standalone unit will give us 2000 independent units. This means the proposed model considers the history of the units by assuming it is a standalone machine. This approach is similar to applying windowing with window length = 1.

4. Experiments and Results

The "*train_FD001.txt*" data originally contains 20631 rows and 26 columns, including *unit_number* and *time_in_cycle*, where the former shows the engine number and the latter shows the operational cycle number (time steps). The rest of the columns include three operational settings and 21 sensor measurements. We skipped six of the sensor measurement columns that exhibited a very low or literally zero variance for the training of the models. Similarly, we skipped the operational settings. This left us with 15 columns in the test and training files. The train and test data each contain 100 engines, and the number of samples in the test data is 13096.

In the Turbocharger dataset, removing the independent and redundant columns gives us 256 features. Then we split the data into train and test by randomly picking 300 vehicles for training, and the remaining 161 vehicles are used for test. The number of samples in the training and test split is 20068 and 11114, respectively.

We ran the same experiment twice, once with C-MAPSS and then with the Turbocharger dataset. In each experiment, we train six models as follows:

1. A **Random Forest regressor** that receives covariate features and estimated RUL as a target. The target label is calculated using a **Linear Degradation**.
2. **Random Forest regressor** that receives covariate features and estimated RUL as a target. The target label is calculated using a **Piecewise Linear Degradation**. The early RUL value is set equal to EOS for each experiment.
3. A **Random Survival Forest** (Ishwaran, Kogalur, Blackstone, & Lauer, 2008) model that receives covariate features and event indicators and estimates a survival function.
4. A **Gradient Boosting regressor** that receives covariate features and estimates RUL as a target. The target label is calculated using a **Linear Degradation** and the model minimizes a **Squared Regression** loss.
5. A **Gradient Boosting regressor** that receives covariate features and estimates RUL as a target. The target label is calculated using a **Piecewise Linear Degradation**. The early RUL value is set equal to EOS for each experiment. The model minimizes a **Squared Regression** loss.
6. A **Gradient Boosting survival** (Friedman, 2001) model that receives covariate features and event indicators and estimates a survival function. The model minimizes a partial likelihood loss of **Cox's Proportional Hazards model**.

We use the Random Forest and Gradient Boosting implementations in the *scikit-survival* (Pölsterl, 2020) and *scikit-learn* (Buitinck et al., 2013) packages. The parameters used for the Random Forest models are as follows: The number of trees in the Forest (n_estimators) is 100, The minimum number of samples required to split an internal node (min_samples_split) is 10, and The minimum number of samples required to be at a leaf node (min_samples_leaf) is 15. The training parameters for the Random Survival Forest are exactly the same as the one reported before, and we employed the same package implementation for both regression and survival models. This ensures that the comparison between models is not affected by employing different package implementations or parameters.

The parameters used for the Gradient Boosting Regressor are as follows: The number of boosting stages (n_estimators) is 100, The learning rate is 10, and The Maximum depth of the individual regression estimators (max_depthint) is 3. We used Squared Regression loss for the regressor models. The training parameters for the Gradient Boosting Survival model are exactly the same as the one reported before, except using Cox's Proportional Hazards loss. We employed the same code implementation for both regression and survival models.

Aligned with the goal described in the research questions, we altered the number of censored samples in the dataset by changing the study duration and analyzed its impact on the performance of the models. The end of the study (EOS) parameter indicates the study duration, and for the C-MAPSS dataset, it has been changed from 70 to 290. For the mentioned EOS range, the percentage of the censored data varied between 0.66 to 0.01. We calculated the percentage of the censored data by dividing the number of samples flagged as censored ($e_i = 0$) by the total number of samples. Both the EOS and the percentage of the censored samples are represented on the x-axis of the plot in Figure 2.

The linear degradation model ignores the censored data; therefore, the number of its training samples is essentially less or equal to the number of training samples of the other two models. The readers can get the number of training samples for the linear degradation model by multiplying the total number of samples by one minus the censoring percentages provided in the plots. For example, according to Fig 3, the censored data percentage for the EOS = 160 is 0.23. This means that the number of training samples for the models that use linear degradation is $20631 \times (1 - 0.23) = 15886$. Note that 20631 is the total number of training samples in the first dataset. Similarly, the percentage of the censored samples for the Turbocharger dataset starts at 0.54 and ends at 0.0, meaning the end of life for all the samples is observed. We did not report the EOS range for the Turbocharger dataset for confidentiality reasons.

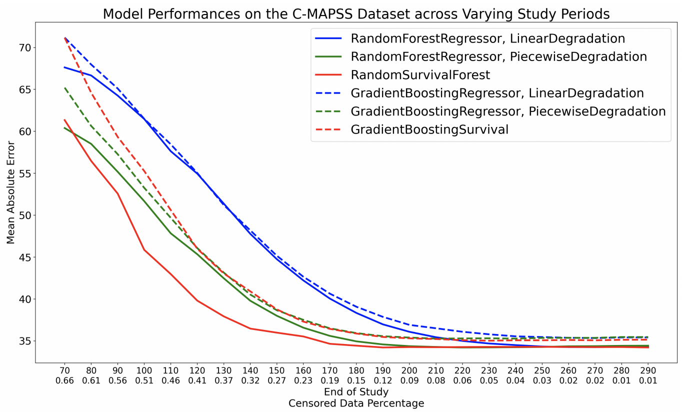

Figure 2. Model performances on the CMAPSS dataset across varying study periods

Figure 2 demonstrates the performances of different learning and degradation models on the CMAPSS dataset in terms of MAE with different amounts of censored data. Similar results for the turbocharger dataset are depicted in Fig 3. Equation 7

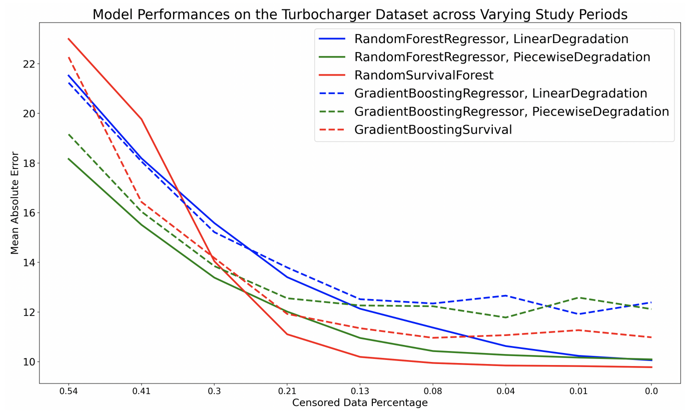

Figure 3. Model performances on the Turbocharger dataset across varying study periods

represents Mean Absolute Error (MAE) where $\hat{y}_i$ is the prediction of the model and y_i is the true RUL.

$$MAE = \frac{1}{n}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{7}$$

Table 1. Average and Standard Deviation of MAE for different models across varying study periods

Learning Model	**Degradation Model**	**C-MAPSS**	**Turbocharger**
RandomForest	Regressor, LinearDegradation	44.4 ± 11.92	13.68 ± 3.99
	Regressor, PiecewiseDegradation	40.18 ± 8.61	**12.34 ± 2.85**
	RandomSurvivalForest	**38.74 ± 7.88**	13.06 ± 4.97
GradientBoosting	Regressor, LinearDegradation	45.3 ± 12.02	14.46 ± 3.2
	Regressor, PiecewiseDegradation	41.46 ± 9.29	13.62 ± 2.45
	GradientBoostingSurvival	42.0 ± 10.76	13.38 ± 3.82

Table 1 summarizes the average and standard deviation of mean absolute error (MAE) over varying study periods. The reported average and standard deviation values are calculated over 23 runs for the C-MAPSS dataset and nine runs for the Turbocharger dataset.

5. Disscussion

After analyzing the plots depicted in Figures 2 and 3 and Table 1, we have derived the following findings and observations: It is apparent that the inclusion of censored samples in the training data has significantly enhanced the performance of Survival Analysis models and the Random Forest model incorporating Piecewise Linear degradation compared to the Linear Degradation model. In other terms, the linear degradation model consistently performs worse than both the piecewise linear and survival degradation models in both datasets. Next, you can observe that as the study duration increases (i.e. the percentage of censored data decreases), the Mean Absolute Error (MAE) in all six models reduces. This improvement in the models' performances is happening due to providing more information to the models and having fewer censored samples. Moreover, with a decrease in the number of censored data, all degradation models tend to converge and produce very similar results. Overall, in both datasets, the Random Forest model slightly outperforms the Gradient Boosting model. The Piecewise Linear and Survival Degradation models exhibit similar performance, with the former demonstrating slightly better results in the real Turbocharger dataset and the latter performing slightly better in C-MAPSS. Finally, we observed that there is very low fluctuation in Mean Absolute Error values across multiple runs. We interpret this as having a robust performance from the models. It is worth mentioning that the effect of the censored percentage on error is more consistent throughout the curves of the CMAPS dataset in comparison to the turbocharger dataset. We argue that this is due to the fact that CMAPSS is a well-curated simulated dataset, while our real-world turbocharger dataset contains a considerable amount of noise.

6. Conclusion

This paper proposed a framework to compare Remaining Useful Life Prediction and Survival Analysis Methods for Predictive Maintenance. We experimented with two learning models of Random Forest and Gradient Boosting Trees once in the context of survival analysis and then regression. The regressors used two degradation configurations: linear and piecewise linear with early RUL. To answer the research questions, we can iterate that both survival analysis and piecewise linear degradation models almost consistently outperformed the linear degradation model in both datasets. We believe this is due to ignoring censored samples in the training of the linear degradation model. It seems that information from censored samples has been able to boost the performance of the survival analysis and piecewise linear models and reduce their MAE. We also noted that performances of all models converge to similar values as we remove the censoring impact (i.e., decreasing the number of censored samples in the data). On the other hand, it is noted that having too many censored samples impacts the survival models' performance conversely. The other somewhat unexpected conclusion from the results is that models that used Random Forest slightly outperformed their Gradient Boosting counterparts. The Random Survival Forest model proved the lowest MAE in the C-MAPSS dataset, while the Random Forest regressor with piecewise linear degradation model outperformed others in Turbocharger dataset.

Acknowledgment

The work was partially supported by grants from The Knowledge Foundation (KK-Stiftelsen).

References

Alabdallah, A., Ohlsson, M., Pashami, S., & Rögnvaldsson, T. (2022). The concordance index decomposition: a measure for a deeper understanding of survival prediction models. *arXiv preprint arXiv:2203.00144*.

Altarabichi, M. G., Sheikholharam Mashhadi, P., Fan, Y., Pashami, S., Nowaczyk, S., Del Moral, P., … Rögnvaldsson, T. (2020). Stacking ensembles of heterogenous classifiers for fault detection in evolving environments. In *30th european safety and reliability conference, esrel 2020 and 15th probabilistic safety assessment and management conference, psam15 2020, venice, italy, 1-5 november, 2020* (pp. 1068–1068).

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., … Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *Ecml pkdd workshop: Languages for data mining and machine learning* (pp. 108–122).

Fotso, S., et al. (2019). *PySurvival: Open source package for survival analysis modeling.* Retrieved from https://www.pysurvival.io/

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Goel, M. K., Khanna, P., & Kishore, J. (2010). Understanding survival analysis: Kaplan-meier estimate. *International journal of Ayurveda research*, *1*(4), 274.

Hartman, N., Kim, S., He, K., & Kalbfleisch, J. D. (2023). Pitfalls of the concordance index for survival outcomes. *Statistics in Medicine*.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests.

Karlsson et al., N. (2023). Baseline selection for integrated gradients in predictive maintenance of volvo trucks' turbocharger. In *Vehicular 2023-iaria.*

Pölsterl, S. (2020). scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, *21*(212), 1-6. Retrieved from http://jmlr.org/papers/v21/20-729.html

Rahat, M., Mashhadi, P. S., Nowaczyk, S., Rognvaldsson, T., Taheri, A., & Abbasi, A. (2022). Domain adaptation in predicting turbocharger failures using vehicle's sensor measurements. In *Phm society european conference* (Vol. 7, pp. 432–439).

Rahat, M., Pashami, S., Nowaczyk, S., & Kharazian, Z. (2020). Modeling turbocharger failures using markov process for predictive maintenance. In *30th european safety and reliability conference (esrel2020) & 15th probabilistic safety assessment and management conference (psam15), venice, italy, 1-5 november, 2020.*

Revanur, V., Ayibiowu, A., Rahat, M., & Khoshkangini, R. (2020). Embeddings based parallel stacked autoencoder approach for dimensionality reduction and predictive maintenance of vehicles. In *Iot streams for data-driven predictive maintenance and iot, edge, and mobile for embedded machine learning: Second international workshop, iot streams 2020, and first international workshop, item 2020, co-located with ecml/pkdd 2020, ghent, belgium, september 14-18, 2020, revised selected papers 2* (pp. 127–141).

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).

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., & Wei, L.-J. (2011). On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, *30*(10), 1105–1117.

Vieira, D., Gimenez, G., Marmerola, G., & Estima, V. (2021). *Xgboost survival embeddings: improving statistical properties of xgboost survival analysis implementation.*

Voronov, S., Frisk, E., & Krysander, M. (2018). Data-driven battery lifetime prediction and confidence estimation for heavy-duty trucks. *IEEE Transactions on Reliability*, *67*(2), 623–639.

Yang, Z., Kanniainen, J., Krogerus, T., & Emmert-Streib, F. (2022). Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment. *Scientific Reports*, *12*(1), 1–20.