SurvLoss: A New Survival Loss Function for Neural Networks to Process Censored Data

Mahmoud Rahat[1], and Zahra Kharazian[2]

[1] *Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden*
mahmoud.rahat@hh.se

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

Abstract

This paper presents SurvLoss, a novel asymmetric partial loss and error calculation function for survival analysis and regression, enabling the inclusion of censored samples. An observation in a dataset for which the complete information regarding an event of interest is not available is called censored. Censored samples are ubiquitous in the industry and play a crucial role in Prognostics and Health Management (PHM) by providing a realistic representation of data, improving the accuracy of analyses, and supporting better decision-making in various industries and the healthcare sector. The proposed approach can effectively equip the conventional regression loss functions such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) with the ability to process censored samples. This can impact the field hugely by providing a more accessible usage of neural network models in survival analysis. The proposed survival loss incorporates censored samples by penalizing predictions outside the censoring region and skipping them otherwise. Then, it uses weighted averaging to aggregate the loss from censored samples with the loss from event samples. Unlike many other methods in the field, the proposed model distinguishes itself by avoiding superficial assumptions and exclusively relies on the available information, considering the entirety of the data.

We compared the proposed loss function with its baseline on two publicly available datasets. The first dataset, called C-MAPSS, is from NASA Turbofan Jet Engines simulation, and the second is a recently published real-world dataset from SCANIA trucks. The goal of both datasets is to predict the remaining useful life (RUL) of the machines. The experimental results show that optimization algorithms for training deep neural networks like Adam can effectively utilize the proposed loss function to calculate gradients, update the model's weights, and reduce training and test errors. Moreover, the proposed model outperformed the baseline by taking advantage of the censored samples. The proposed loss function paves the way for the employment of advanced architectures of neural networks with bigger training sizes in survival analysis.

1. Introduction

This paper deals with the problem of time-to-event prediction. Specifically, the prediction of time until a component fails or, in other words, the component can no longer perform its intended functionality. The literature has three main directions for tackling this problem. The remaining useful life prediction (Revanur, 2020; Altarabichi, 2020; Karlsson, 2023), risk classification (Rahat, Pashami, Nowaczyk, & Kharazian, 2020), and survival analysis (Wang, Li, & Reddy, 2019). While each of these directions has benefits and drawbacks, a shared challenge among all three approaches is dealing with the censored samples. While some methods, like Cox proportional hazards (Cox, 1972), consider them using the partial likelihood function, most methods simply ignore censored samples. Nonetheless, censoring is an inherent aspect of time-to-event prediction, especially in long-term studies. Censored samples refer to data points for which the event of interest (such as death or failure) has not been observed by the end of the study or at the time of analysis. Censoring can happen, for instance, when the exact time of the event is unknown; typically, since the event has not occurred yet, the subject has been lost to follow-up, or the study finishes before we observe the event. With the censored samples, we have incomplete information about the individual, i.e., we don't know when the event happened. Still, we know the event has not occurred during a specific period.

Generally, three types of censoring are recognized in survival analysis (Kleinbaum & Klein, 1996). The right censoring where the event of interest has not happened for some individuals by the end of the study, but it may occur at an unknown time in the future. In other words, the event time for these individuals is known to be longer than the observed follow-up time. The left censoring happens when we know that the event of interest happened before a specific time, but we don't know precisely when, i.e., the survival time is shorter than the study's start time. Finally, we have interval censoring, where we only observe the event that has happened within a specific time window, but again, we don't know precisely when. Right censoring is considered the most common type; hence, it is regarded as the primary focus of this paper. It is worth mentioning that although we only looked at the right censoring, the proposed loss function can be generalized similarly to all three types.

Censoring is prevalent in the industry for many reasons, such as the long life of components or changes in the conditions of the equipment, like the expiry of a warranty period. In real-world industrial applications, it is common to access much more censored samples than those that fail. Depending on the cases, the ratio of censored to death varies, but this ratio can reach, for instance, 30 censored samples per 1 death or more. Most available methods in the field struggle with consuming censored samples and end up ignoring these samples. This means that they essentially ignore a significant portion of their available data.

An exception to this general rule is methods based on survival analysis designed to take advantage of the censored information. However, survival analysis is much more common in clinical studies than in industry due to its practical limitations, such as the inability to process big data sizes or their shortage in handling temporal information, like sequences of observations from the same individual over time. Additionally, they mostly rely on the Cox proportional assumption, which is known to be naive and not genuine in many real-world cases. On the other hand, there is no way to validate the survival functions produced by these models as we only have access to the time of the event, and the actual degradation curves are unknown. That is why, in many industrial applications, we rely on the median or the mean point of the survival functions. Concordance index (C-index) (Harrell, Califf, Pryor, Lee, & Rosati, 1982) is the typical survival analysis evaluation metric which only considers the order of the events and is known to be biased (Hartman, Kim, He, & Kalbfleisch, 2023; Alabdallah, Ohlsson, Pashami, & Rögnvaldsson, 2024). Most survival datasets are clinical records of patients with a very small number of data points (around 1000) and features (around 10), and there are limitations regarding the proportion of censored data compared to event instances. In most studies, clinical researchers maintain this percentage below 50%.

Additionally, specific constraints are related to applying neural networks within the survival analysis domain. While a handful of methods have been proposed to merge the capabilities of neural networks with survival analysis (Katzman et al., 2018; Kvamme, Borgan, & Scheel, 2019), the prediction accuracy for deep learning methods remains comparable to the classical methods such as Cox and Random Survival Forest (Ishwaran, Kogalur, Blackstone, & Lauer, 2008) in many datasets. It is shown that the performance gain using deep learning or neural network-based approaches is often around 0.02 to 0.03 in concordance index (Chen, 2020). This is primarily due to various underlying assumptions made by methods, such as the constant ratio of risk over time or the small sizes of the standard survival analysis datasets. On the other hand, the neural network field is growing rapidly, and it is crucial to search for new ways to employ their incredible computational power in fields such as time-to-event prediction.

This paper contributes to the mentioned challenge by introducing a new loss function called survival loss, which essentially enables conventional neural networks to process censored samples along with the standard event samples. The idea is to penalize the model in accordance with the information available. For the event samples, the survival loss performs similarly to an ordinary loss by considering the distance between the model predictions and actual values. For the censored samples, the survival loss only penalizes the model if the predicted value falls outside the censoring time interval. As an example for the right censored samples, the model gets penalized only if its prediction is below censoring time (which we already know the event has not happened in that period). On the other hand, if the model's prediction is larger than the censoring time, the proposed survival loss effectively ignores that sample in the loss calculation as there is no evidence of the precise event time. Ultimately, the final loss value will be reported as a weighted average of the error values from censored and event samples. The weighting of the two error values takes place in accordance with the number of considered samples in each part. The following section defines the proposed loss function in more detail.

2. METHODOLOGY

We first visually explain the intuition behind the proposed survival loss and later define it mathematically. Figure 1 describes how the proposed survival loss function calculates the amount of error for an event sample, right-censored sample, left-censored sample, and interval-censored sample. The follow-up region (represented in red) indicates the period where we monitored the individual, and we know that the event of interest has not occurred inside it. The censoring region (represented in green) displays the period where the event has happened or will happen inside it, but we don't know precisely when. Finally, the red-dotted region represents the time in the future after the event has occurred.

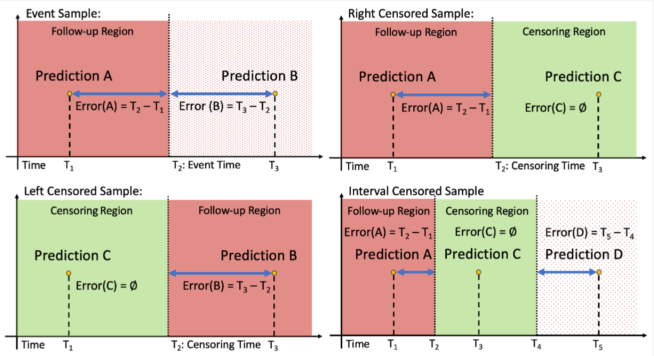

Figure 1. Visual explanation of the proposed loss function.

The loss calculation for an event sample is straightforward. The distance between prediction and actual event time indicates the amount of error. Here, we can use any loss calculation formula like mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), etc. For the sake of simplicity, this paper primarily focuses on the MAE, but any other mentioned error measurement functions can interchangeably be employed.

Unlike the event samples, the proposed loss uses an asymmetric error calculation algorithm for the right censored sample. If the prediction falls within the follow-up region (i.e., the outside of the censoring region), we penalize the prediction according to its distance from the censoring time. Note that this error partially captures the whole error of the prediction as the event happens sometime after censoring time, and the actual error is greater than the proposed partial one. Since calculating the exact error for these samples is impossible, we resort to partial penalization. This partial error from the censored samples can help the optimization algorithm calculate gradients more accurately. Conversely, if the prediction falls within the censoring time, the amount of error is set to null. Note that a null error differs from a zero error since, in the former, we remove the sample from the total number of samples in the batch and, therefore, from the denominator of the averaging function. Finally, a similar logic can also handle the left-censored and interval-censored samples. Again, for the sake of simplicity, we primarily focus on the right censored samples and mean absolute error. Still, the method can be generalized to all censoring types and error calculation functions.

Figure 1 also provides some example prediction points and their associated error values for varying conditions. The errors for prediction A, B, C, and D are respectively $T_2 - T_1$, $T_3 - T_2$, $\emptyset$, and $T_5 - T_4$, where $\emptyset$ means we ignore the sample in the loss calculation.

Assume (X, t, δ) represents a random survival data point where $X \in \mathcal{R}^d$ is a d-dimensional covariate vector and $\delta \in \{0, 1\}$ is an event indicator such that $\delta = 1$, if we observed the event and $\delta = 0$ in case of censoring. Moreover, $t = \min(y, c)$ is the observed time, where $y \in \mathcal{R}^+$ is the actual event time and $c \in \mathcal{R}^+$ is the censoring time.

In the context of PHM and without loss of generality, we define the set of failed samples where the event of failing happened for them (i.e., $\delta_i = 1$) as:

$$(X_i, y_i)_{i=1}^{N_{Failed}} \tag{1}$$

and similarly the set of censored samples (i.e., $\delta_i = 0$) as:

$$\left(\widetilde{X}_j, c_j\right)_{j=1}^{N_{Censored}} \tag{2}$$

which means we divide the samples into two groups of failed (aka event) and censored samples where the total number of samples is $N = N_{Failed} + N_{Censored}$. Then, for a given predictive model f, we define $\hat{y}_i = f(X_i)$ and similarly, $\hat{c}_j = f(\widetilde{X}_j)$ as the output of the predictive model for the failed and censored samples. Note that we have access to the ground truth values for $f(X_i)$, but ground truth values for $f(\widetilde{X}_j)$ are unknown, and the only information we have regarding them is that the actual event time is greater than c_j. This means that we can only penalize the model if the prediction of the model $\hat{c}_j$ is less than c_j; otherwise, we ignore the sample in our loss calculation. Note that ignoring a sample in the loss calculation is different from having an error equal to zero for that sample since by ignoring the sample, we do not consider it in the total number of samples in the denominator of the loss function.

The new survival loss function is defined as a weighted sum of the error for the censored and failed samples:

$$E = \frac{N_{Failed} \times E_{Failed} + N_{Censored'} \times E_{Censored'}}{N_{Failed} + N_{Censored'}} \tag{3}$$

where $Censored'$ represents the set of samples for which the model predicts a survival time less than the censoring time, i.e., $\hat{c}_j < c_j$ and is defined as follows:

$$N_{Censored'} = \left\| \left(\tilde{X}_j, c_j\right) \quad given \quad \hat{c}_j < c_j \right\| \tag{4}$$

this means that $N_{Censored'} \leq N_{Censored}$ since:

$$\{\left(\widetilde{X}_j, c_j\right) \quad given \quad \hat{c}_j < c_j\} \subseteq \{\left(\widetilde{X}_j, c_j\right)\} \tag{5}$$

Then, equations 6 and 7 represent the standard calculation of the mean absolute error for the two groups of $Failed$ and $Censored'$.

$$E_{Failed} = \frac{1}{N_{Failed}} \times \sum_{i=1}^{N_{Failed}} |y_i - \hat{y}_i| \tag{6}$$

$$E_{Censored'} = \frac{1}{N_{Censored'}} \times \sum_{j=1}^{N_{Censored'}} |c_j - \hat{c}_j| \; given \, \hat{c}_j < c_j \tag{7}$$

and if we plug in equations 6 and 7 into equation 3, we get Equation 8 that defines Survival Mean Absolute Error (S-MAE). Equation 8 can easily be modified for Mean Squared Error (MSE) and Root Means Squared Error (RMSE). Note that in the extreme cases of very high censoring ratio, the denominator of the Equation 8 can become zero. To avoid such cases, we recommend reducing the censoring ratio by randomly skipping some of the censored samples or increasing the batch size of the gradient descent algorithm.

3. Experimental Results

We first provide some general information about the experiments and the two datasets used in section 3.1. This is followed by the experimental results and discussion for the first dataset in section 3.2, and that of the second dataset in section 3.3.

3.1. Experimental Setup

For the experiments, we used two public run-to-failure datasets, one on lab-simulated data and the other on real-world data from the field. The first dataset is the well-known NASA Commercial Modular Aero-Propulsion System Simulation, also known as C-MAPSS (Saxena, Goebel, Simon, & Eklund, 2008). It is a widely used benchmark dataset in prognostics and health management (PHM) developed by NASA to support research in aircraft engine health monitoring and prognostics and to estimate the remaining useful life. The dataset consists of simulation measurements from turbofan jet engines with multiple subsets, each corresponding to different operating conditions and engine fault modes. It includes sensor measurements collected from various sensors installed on the engine, along with information about the engine's health and remaining useful life. The C-MAPSS dataset contains temporal information in the form of multiple observations during time from each engine.

The C-MAPSS dataset originally did not contain censored samples as it is a simulated dataset, and the actual failing time for all the engine cases is provided. Since the goal of this paper is to study the effect of having censored samples in the dataset, we used the algorithm introduced in (Rahat, Kharazian, Mashhadi, Rögnvaldsson, & Choudhury, 2023) to transform the dataset into survival settings by defining a specific study period and labeling all the failed samples after the end of the study as censored.

The second dataset is the recently published SCANIA Component X Dataset (Kharazian, Lindgren, Magnússon, Steinert, & Reyna, 2024; Lindgren, Steinert, Andersson Reyna, Kharazian, & Magnússon, 2024). This dataset is collected from an unknown engine component (called component X) of a fleet of trucks. We refer to the second dataset as the Scania dataset. This dataset contains sensor measurements from 21278 censored trucks and 2272 instances of trucks where their component X failed. We define the censoring ratio for a dataset as the percentage of censored samples to the total samples, i.e., the number of censored samples divided by the total number of samples. Therefore, in this dataset, the censoring ratio is 90%, which is way beyond the common censoring ratios in the survival analysis domain. Looking at the literature, it is very rare to see a dataset that contains more than 50% censored samples. Similar to C-MAPSS, this dataset includes temporal measurements from trucks. The only difference compared to C-MAPSS is that in C-MAPSS, the intervals between the observations are the same, but here they vary.

The predictive model used for the experiments is a multilayer Perceptron neural network that contains an input layer followed by five dense layers, each containing 14 neurons, followed by a single neuron as the output, where all layers use the ReLU activation function. The purpose of the network is to predict the remaining useful life of a piece of equipment according to the covariate features received as input. All the networks are trained using the Adam optimization algorithm, and the batch size for all experiments is 32.

There is no need to spend too much time optimizing the neural network's architecture, as both the baseline and the proposed model use the same architecture in terms of fairness of the comparisons. We also tweaked the network's architecture and confirmed that the proposed loss function is not sensitive to the architecture and can perform robustly regardless of its structure. Two models are compared in the following sections. The first model uses the mentioned neural network architecture with S-MAE as the loss function and is referred to as the proposed model. Due to the use of S-MAE, the proposed model can consume censored samples. The second model also uses the mentioned neural network architecture. The only difference is using the standard MAE loss function, which makes the second model unable to render censored samples as there are no ground truth target values associated with them in the dataset. We refer to the second model as the baseline model. The code is implemented in Python using Keras running on a TensorFlow backend. A code implementation of the S-MAE function in TensorFlow is also provided in the appendix section.

$$E_{S-MAE} = \frac{\left(\sum_{i=1}^{N_{Failed}} |y_i - \hat{y}_i|\right) + (\sum_{j=1}^{N_{Censored'}} |c_j - \hat{c}_j| \; given \; \hat{c}_j < \; c_j)}{N_{Failed} + N_{Censored'}} \tag{8}$$

3.2. SCANIA Dataset

We used the SCANIA dataset for the first experiment. Here, the number of censored to failed samples is enormous, which means most of the components did not fail during the study period. As mentioned before, the ratio of censored vehicles is 90%. In other words, we have around 9 censored vehicles per failed one. Including all censored samples in the experiments is technically impossible, as this will cause the loss to become zero for almost all batches, and consequently, the gradients will become NaN. We randomly down sample censored instances to 500 vehicles. We also include all 2272 instances of failed trucks. The resulting dataset has a censoring ratio of 18%. Additionally, we ignored the temporal information and randomly picked one observation per truck for this experiment. The number of independent features in this dataset is 105, and the goal is to predict the remaining useful life of component X.

The objective of the experiment is to investigate how the inclusion of censored samples impacts a model employing the proposed survival loss function compared to a standard loss function. We conducted two experiments with the model outlined in the experimental setup. In the first experiment, the model is equipped with the ability to incorporate censored samples through the proposed S-MAE loss function. Conversely, the second iteration excludes censored samples from the training data, as the model employs a conventional loss function, making it unable to process partially observed instances. Both models are trained for 10 epochs. The test data used for evaluating both models contains both censored and failed cases, and the S-MAE loss function is used to report the models' performance.

Figure 2 represents the five-fold cross-validation results for the Scania dataset. The two models' average training and test curves across five folds are presented using lines. The shaded confidence bands visualize the respective standard deviations of the test data across five folds. The green curves represent the model's performance using S-MAE, and the red curves represent the conventional MAE loss function. The y-axis shows the value of error using S-MAE. Similar to the standard MAE, the lower values of S-MAE indicate better performances. As you can see in the figure, the model that used the proposed S-MAE consistently outperforms the conventional MAE. Furthermore, the standard deviation of both models decreased during the training epochs.

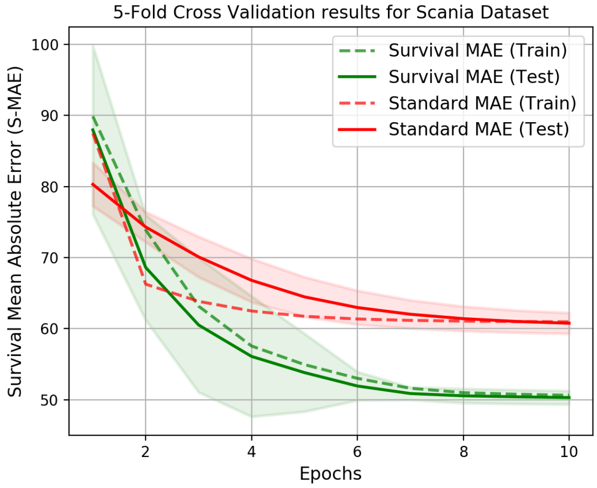

Figure 2. Five-fold cross-validation results for Scania dataset.

3.3. C-MAPSS Dataset

In the second experiment, we performed an assessment similar to the first experiment but with the C-MAPSS dataset. We used the dataset related to the first operational setting (FD001) representing condition one (Sea Level) with 14 covariate features and the remaining useful life of the equipment as the target. The train and test trajectories each contain 100 units, and a varying number of readouts is available in the dataset for each unit. We merged the train and test units from the original dataset to get a dataset with 200 units and reduced the samples by randomly picking 20 readouts from each engine unit. Again, we employed a 5-fold cross-validation approach to evaluate the performance of the proposed method and compare it to our baseline model. Both models are trained for 15 epochs. To simulate censoring, we used the technique described in (Rahat et al., 2023) and set the end-study parameter to 200 in this experiment. The simulation resulted in 1140 event (failed) samples and 646 censored samples with a censoring ratio of 36%.

Figure 3 displays the train (shown with a dashed line) and test (shown with a solid line) learning curves for the baseline and the proposed models. The green curves show the proposed model learning curves, while red is used for the baseline. The shaded area around the test curves represents the standard deviations between the results from the five folds. The standard deviations of the training curves are not visualized to avoid overcrowding and maintain readability. The models are trained until the learning curves flatten after about 15 epochs. As can be seen, the proposed model outperforms the baseline

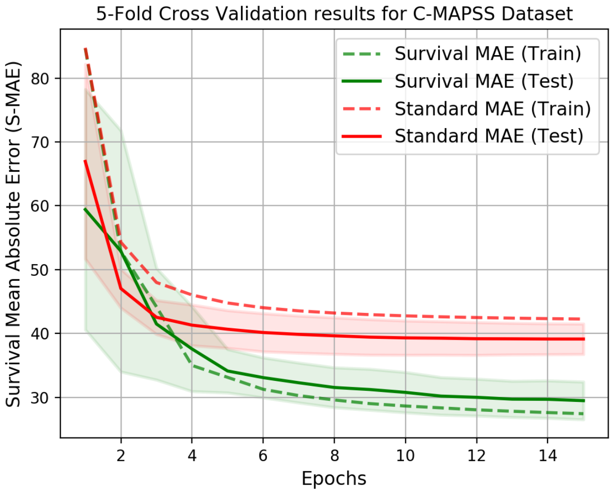

Figure 3. Five-fold cross-validation results for C-MAPSS dataset.

Table 1. The final performance of two models on the test data averaged over 5 folds.

	Scania Dataset	C-MAPSS Dataset
S-MAE	50.32 ± 0.85	29.49 ± 2.64
MAE	60.77 ± 1.31	39.14 ± 2.13

by a significant margin.

Table 1 compared the final performance of two models on the test data averaged over five folds. The proposed loss function significantly outperforms the standard mean absolute error with a margin of 10.45 units in the Scania dataset and with a margin of 9.65 units for the C-MAPSS dataset. There is no need to run statistically significant tests as the standard deviation of the models compared to the net improvement level is little.

4. Conclusion

We presented a novel loss and error calculation method that partially considers censored samples in the context of survival analysis and remaining useful life prediction. The proposed loss function can be used with any standard regression error function and can handle right, left, or interval-censored samples. To assess the algorithm, we tested it using a flat regressor on two public industrial datasets to predict the remaining useful life of engine equipment. The results indicated that the proposed loss function can significantly reduce the model loss on the test data compared to the baseline. The experiments only looked at the mean absolute error function and right censored samples. The application of other regression loss functions with varying censored settings is left for future work. Another suggested future work is to use the proposed loss function on advanced types of neural networks, such as long short-term memory networks (LSTM) or Gated recurrent units (GRU), to incorporate temporal information in survival analysis.

Acknowledgment

The work was supported by research grants from KK-Foundation, Scania CV AB, and the Vinnova Program for Strategic Vehicle Research and Innovation (FFI).

References

Alabdallah, A., Ohlsson, M., Pashami, S., & Rögnvaldsson, T. (2024). The concordance index decomposition: A measure for a deeper understanding of survival prediction models. *Artificial Intelligence in Medicine*, *148*, 102781.

Altarabichi, e. a., Mohammed Ghaith. (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).

Chen, G. H. (2020). Deep kernel survival analysis and subject-specific survival time prediction intervals. In *Machine learning for healthcare conference* (pp. 537–565).

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*(2), 187–202.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, *247*(18), 2543–2546.

Hartman, N., Kim, S., He, K., & Kalbfleisch, J. D. (2023). Pitfalls of the concordance index for survival outcomes. *Statistics in Medicine*, *42*(13), 2179–2190.

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

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

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.

Kharazian, Z., Lindgren, T., Magnússon, S., Steinert, O., & Reyna, O. A. (2024). Scania component x dataset: A real-world multivariate time series dataset for predictive maintenance. *arXiv preprint arXiv:2401.15199.*

Kleinbaum, D. G., & Klein, M. (1996). *Survival analysis a self-learning text.* Springer.

Kvamme, H., Borgan, Ø., & Scheel, I. (2019). Time-to-event prediction with neural networks and cox regression. *Journal of machine learning research*, *20*(129), 1–30.

Lindgren, T., Steinert, O., Andersson Reyna, O., Kharazian, Z., & Magnússon, S. (2024). *SCANIA Component X Dataset: A Real-World Multivariate Time Series Dataset for Predictive Maintenance.* Scania CV AB. Retrieved from `https://doi.org/10.58141/1w9m-yz81` doi: 10.58141/1w9m-yz81

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).

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, e. a., Vandan. (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).

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, *51*(6), 1–36.

APPENDIX

SurvLoss Mean Absolute Error (S-MAE) implementation in Python using TensorFlow. Note that S-MAE receives three values as input, ground truth observed times, predictions, and censoring flag that is either True or False.

```python
import tensorflow as tf

def SurvLoss_MAE(y_true, y_pred, censore_flags):
    y_true_event = y_true[censore_flags]
    y_pred_event = y_pred[censore_flags]
    y_true_censored = y_true[~censore_flags]
    y_pred_censored = y_pred[~censore_flags]
    y_pred_event = tf.squeeze(y_pred_event)
    count1 = y_true_event.shape[0]
    error1 = tf.reduce_mean( tf.abs( y_true_event - y_pred_event ) )
    error1 = tf.where(tf.math.is_nan(error1), tf.zeros_like(error1), error1)
    y_pred_censored = tf.squeeze(y_pred_censored)
    mask = tf.cast(y_true_censored > y_pred_censored, tf.float32)
    count2 = tf.reduce_sum(mask)
    error2 = tf.reduce_mean( tf.abs( tf.multiply( mask, (y_true_censored - y_pred_censored) ) ) )
    error2 = tf.where(tf.math.is_nan(error2), tf.zeros_like(error2), error2)
    survloss_mae = (error1 * count1 + error2 * count2) / (count1 + count2)
    return survloss_mae
```