On Training Survival Models with Scoring Rules

Philipp Kopper[1,2][0000−0002−5037−7135], David Rügamer[1,2][0000−0002−8772−9202], Raphael Sonabend[3][0000−0001−9225−4654], Bernd Bischl[1,2][0000−0001−6002−6980], and Andreas Bender[1,2][0000−0001−5628−8611] (✉)

[1] Department of Statistics, LMU Munich, 80539 Munich, Germany
{philipp.kopper, david.ruegamer, bernd.bischl, andreas.bender}@stat.uni-muenchen.de
[2] Munich Center for Machine Learning (MCML), LMU Munich, 80539 Munich, Germany
[3] OSPO Now, London, UK raphaelsonabend@gmail.com

Abstract. Scoring rules are an established way to compare predictive performance between model classes. In the context of survival analysis, they require adaptation in order to accommodate censoring and other aspects specific to survival tasks. This work investigates the use of scoring rules for model training rather than evaluation. Doing so, we establish a general framework for training survival models that is model-agnostic and can learn event time distributions parametrically or non-parametrically. In addition, our framework is not restricted to any specific scoring rule. Although we focus on neural network-based implementations, we also provide proof-of-concept implementations using gradient boosting, generalized additive models, and trees. Empirical comparisons on synthetic and real-world data indicate that scoring rules can be successfully incorporated into model training and yield competitive predictive performance with established time-to-event models.

Keywords: Proper Scoring Rules · Survival Analysis · Neural Networks.

1 Introduction

Survival analysis (SA) is an important branch of statistics and machine learning that deals with time-to-event data analysis. Let $Y > 0$ be a random variable representing a time-to-event of interest (e.g., time-to-death after operation) and y its realization. In many studies, Y cannot be observed in all cases due to censoring $C > 0$. Thus, in the presence of right-censoring, we can only observe realizations of $T := \min(Y, C)$ and status indicator $D := I(Y \leqslant C)$. Observed data is then given by tuples $(t_i, d_i, \mathbf{x}_i), i = 1, \ldots, n$, where t_i is an observed event or censoring time, d_i the status indicator and $\mathbf{x}_i^\top = (x_{i1}, \ldots, x_{ip})$ a p-dimensional feature vector.

Notably, while we are interested in inference about Y, we only have realizations of (T, Δ). Therefore, the usual metrics for evaluating predictive performance based on the difference in the true and observed value $(y_i - \hat{y}_i)$ cannot be calculated for the censored data from time to event. For the same reason,

most survival models do not generate predictions $\hat{y}$, but rather probabilistic predictions $\hat{F}_Y(\tau) = \mathbb{P}(Y \leqslant \tau), \tau \in \mathbb{R}_0^+$, or equivalently the survival function $\hat{S}_Y(\tau) = 1 - \hat{F}_Y(\tau)$. At the estimation stage, censoring must be accounted for to obtain unbiased estimates of $S_Y(\tau)$. Common approaches include parametric models that assume a specific distribution for the event times with a censoring-adjusted likelihood (e.g., accelerated failure time models) as well as non- and semi-parametric approaches that partition the follow-up into intervals and estimate the (baseline) hazard rate within each interval (e.g., Kaplan-Meier, Cox, discrete-time approaches).

For predictive modeling, dedicated evaluation metrics that consider the data's survival nature have been proposed in the literature (see [26] for an overview). Such metrics are often model-agnostic to allow comparison of predictive performances across model classes. While concordance-based metrics [e.g. Harrell's C, 14] are popular in practice, they only allow for evaluating how well the model ranks the risk for an event. On the other hand, (strictly proper) scoring rules have been proposed as suitable tools to evaluate probabilistic (distribution) predictions [12]. As these scores often only rely on point-wise survival probability predictions (without requiring a density estimate, for example), scores can be compared across different model classes. One such scoring rule is the continuous rank probability score or integrated brier score [12]. [13] adapted it to the survival setting by weighting the scores concerning the individuals' probabilities of being censored (IPCW). This work refers to it as integrated survival brier score (ISBS). While ubiquitous in practice, recent work suggests that the ISBS is not proper [24, 26, 32], and proper alternatives have been proposed. As scoring rules in survival analysis are established in the context of model evaluation and comparison, so far only few attempts have been made to use them as a loss function for model training.

Our Contributions In this work, we investigate the use of censoring-adapted scoring rules for model training rather than evaluation. The developed framework uses gradient-based optimization of the scoring rule of choice, evaluated at discrete partitions of the follow-up. In contrast to previous contributions, it is scoring-rule agnostic, allows parametric and non-parametric estimation of the event time distribution, and extends scoring rule-based estimation to the important case of competing risks. Additionally, while our main implementation is based on neural networks, we also show that our framework is applicable to other model classes, such as gradient boosting, trees, and generalized additive modeling. We empirically evaluate the approach on synthetic and real-world data, showing competitive predictive performance compared to established state-of-the-art survival models.

2 Related Literature

Scoring rules Scoring rules are established tools for model evaluation and comparison, particularly in the context of probabilistic predictions. A comprehensive

summary is given in [12], who also investigate the role of scoring rules in estimation. Adaptations of scoring rules for survival analysis (see Table 1 for an overview of selected scores) have been pioneered by [13], who defined the ISBS, which weights the integrated brier score by an estimate of the censoring distribution $\hat{G}$, usually using the Kaplan-Meier estimator. Other adaptations are discussed in [9, 24, 32, 26]. [24] propose the right-censored log-likelihood (RCLL) and claim to prove its properness, but its calculation requires an estimate of the density f_Y, which is not readily available for non- and semi-parametric methods that often only return survival probability predictions. The score proposed by [32] is also claimed to be proper but relies on an oracle parameter that is not known in practice. [26] suggest a class of re-weighted scoring rules (Eq. (1)), including the re-weighted ISBS (RISBS) and re-weighted integrated survival log-loss (RISLL):

$$\text{SR}_{R,i}(\tau) = \frac{d_i}{\hat{G}(t_i)} \text{SR}_i(\tau), \tag{1}$$

with d_i being the status indicator, $\text{SR}_i(\tau)$ is a suitable point-wise scoring rule (e.g. the IBS) of observation i at time τ and $\hat{G}(t_i)$ an estimate of the censoring distribution at time t_i, which is estimated beforehand. SR is computed up to a $\tau^* < \tau^{max}$, the largest observed survival time, and [26] recommend to consider all fully observed i still at risk at τ^* with $d_i = 1$.

Survival Models Most of the existing methods model the hazard function non- or semi-parametrically (i.e. without (strong) distributional assumptions) based on prior partitioning or discretization of the follow-up (for example, Cox regression [7], (extensions of) piece-wise exponential models [10, 5] and discrete-time approaches [28]), or use specific distributional assumptions with dedicated loss functions for censored data [e.g., 30]. More recently, adaptations of these approaches based on machine and deep learning have been suggested [cf. 29, 31, for respective reviews]. According to the latter, most deep learning models are adaptations of the Cox model, followed by discrete-time approaches. The latter are popular as they allow for transforming a survival task to a classification task and don't require strong distributional assumptions. Concretely, the follow-up is partitioned into J intervals $(\tau_{j-1}, \tau_j), j = 1, \ldots, J$; $\tau_0 := 0$ and new status indicators are defined for each interval $d_{ij} = I(t_i \in (\tau_{j-1}, \tau_j] \wedge d_i = 1)$. Assuming a Bernoulli distribution for these new event indicators, discrete time methods that optimize the resulting Binomial log-likelihood. Popular methods within this class include DeepHit [23] and *nnet-survial* [11]. Another stream of models that reduces the survival problem to a (Poisson) regression problem through discretization are methods based on piece-wise exponential models. State-of-the-art examples include [20, 3, 21]. While this reduction idea relates to our approach, we do not further review this model class in this contribution, as there are too many disjunctions.

In the context of SA, only a few have suggested using scoring rules at the estimation or training rather than the evaluation step. A notable exception is

[2], who use a survival-adapted continuous rank probability score (SCRPS). Additionally, [24] illustrate how RCLL can be used for training and evaluating survival models. While [2] evaluate the SCRPS to estimate the parameters of a log-normal distribution, the approach by [24] is distribution-free, but requires an estimate of the density f_Y which usually needs to be approximated. The two approaches suggested in this work differ from previous endeavors. In contrast to other methods, our non-parametric approach learns increments of a function for event probabilities (i.e., survival or cumulative incidence) based on a scoring rule, whereas others use a specific likelihood. Our parametric approach is similar to [2] but not restricted to SCRPS or the log-normal distribution. Importantly, we extend the scoring rule-based estimation to the important case of competing risks and illustrate how the proposed estimation routine can be incorporated in various modeling approaches (deep learning, boosting, trees, and additive models).

3 Training with Scoring Rules

We aim to learn F_Y by discretely evaluating an associated scoring rule. While our approach is scoring rule agnostic, we focus on the rules in Table 1. The ISBS is of great historical significance and is a popular evaluation metric in the majority of benchmark experiments for SA. The alternatives in Table 1 have been suggested only recently and therefore have not been applied often in practice. The SCRPS, as implemented in [2], is the ISBS but without weighting contributions by inverse probability of censoring weights. The weighting factor in RISBS and RISLL means that contributions of censored observations are always set to zero. Non-censored observations are weighted by the probability of not being censored until the observed event time.

In order to use scoring rules for training, we partition the follow-up into J equidistant intervals $(\tau_{j-1}, \tau_j], j = 1, \ldots, J$, with $\tau_0 = 0$ and τ_J the largest observed event time. We then minimize the objective O that evaluates scoring rule $\text{SR}_i(\tau_j|\hat{G})$ for observation i at time τ_j, given censoring distribution $\hat{G}$:

$$O = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{J} \sum_{j=1}^{J} \text{SR}_i(\tau_j|\hat{G}). \tag{2}$$

To do so, we need J point-wise estimates of $S(\tau_j|\mathbf{x}_i) = 1 - F(\tau_j|\mathbf{x}_i)$. These can be generated in two ways:

1.) *Parametric Learning*: Estimation of the parameters of an assumed distribution;
2.) *Distribution-free Approach*: Direct estimation of the survival function without distributional assumption.

In both cases, the data transformation is identical as depicted in Figure 1. The shown transformation contains all sufficient information for all scoring rules and model classes discussed in this work. However, for some scoring rules, the

Table 1: Selected model-agnostic scoring rules. Here, $F_i(\tau) := F(\tau|\mathbf{x}_i)$; S_i, f_i equivalently. RCLL is only evaluated at the observed time t_i, while all other rules are evaluated over $[0, \tau^*]$.

Abbreviation (Source)	**Definition**
ISBS [13]	$\int_0^{t_i} \frac{F_i(\tau)^2}{\hat{G}(\tau)} d\tau + \int_{t_i}^{\tau^*} \frac{d_i S_i(\tau)^2}{\hat{G}(t_i)} d\tau$
SCRPS [2]	$\int_0^{t_i} F_i(\tau)^2 d\tau + \int_{t_i}^{\tau^*} S_i(\tau)^2 d\tau$
RISBS [26]	$\int_0^{t_i} \frac{d_i F_i(\tau))^2}{\hat{G}(t_i)} d\tau + \int_{t_i}^{\tau^*} \frac{d_i S_i(\tau)^2}{\hat{G}(t_i)}$
RISLL [26]	$-\frac{d_i}{\hat{G}(t_i)} \left(\int_0^{t_i} \log(F_i(\tau)) + \int_{t_i}^{\tau^*} \log(S_i(\tau)) d\tau \right)$
RCLL [24]	$-\log(d_i f_i(t_i) + (1 - d_i) S_i(t_i))$

computation of the weights (w_j) varies or is not necessary, or only a limited number of intervals is needed. Also, the features do not necessarily need to be transformed as well if the model class can facilitate such mapping internally. For example, neural networks can do this by reshaping.

3.1 Modeling Approaches

Both approaches 1.) and 2.) share the same objective function (2) and only differ in the way the predictions $\hat{S}(\tau_j|\mathbf{x}_j)$ are obtained. Both variants ensure that $\hat{S}$ is monotonically decreasing. Details are given below.

Parametric Learning One way to obtain estimates for $S(\tau|\mathbf{x}_j)$ is by assuming a parametric distribution of event times and learning the distribution's parameters. Let $F(\tau|\boldsymbol{\theta})$ be a distribution suitable to represent event times $Y > 0$, with parameters $\boldsymbol{\theta} \in \mathbb{R}^m$ depending on the input features, i.e., $\boldsymbol{\theta}(\mathbf{x}) = (\theta_1(\mathbf{x}), \theta_2(\mathbf{x}), \ldots, \theta_m(\mathbf{x}))^\top$. Some popular parametric survival distributions include the Weibull, log-logistic, and log-normal distribution. Given parameter estimates $\hat{\boldsymbol{\theta}}(\mathbf{x})$, all quantities of the distribution, including the survival function $\hat{S}(\tau|\hat{\boldsymbol{\theta}}(\mathbf{x})) = 1 - \hat{F}(\tau|\hat{\boldsymbol{\theta}}(\mathbf{x}))$, are fully specified and thus prediction can be obtained at any time point τ. Depending on the distribution, parameters may have restrictions, e.g. for the log-normal distribution $\boldsymbol{\theta}(\mathbf{x}) = (\mu(\mathbf{x}), \sigma(\mathbf{x}))^\top$ with $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$. The distribution parameters $\boldsymbol{\theta}$ are learned by minimizing Eq. (2) w.r.t. the model parameters.

Distribution-free Approach Instead of obtaining an estimate of the survival function by learning the parameters of an assumed distribution, we can also learn the

id	status	time	x
1	0	3.1	2.3
2	1	1.4	2.4
3	0	1.2	-1.3

$\Rightarrow$

id	d_j	τ_j	w_j	x
1	0	0.5	1.70	2.3
1	0	1.5	1.70	2.3
1	0	2.5	1.70	2.3
...	...	...	...	...
2	0	0.5	1.23	2.4
2	1	1.5	1.23	2.4
2	1	2.5	1.23	2.4
...	...	...	...	...
3	0	0.5	0.00	-1.3
3	0	1.5	0.00	-1.3
3	0	2.5	0.00	-1.3
...	...	...	...	...

Fig. 1: Example of the transformation of original survival data into a discretized data set with $\tau_j \in \{0.5, 1.5, 2.5, ...\}$. (Time-constant) Features (x) are simply repeated. The survival indicator d_j switches from 0 to 1 when a failure is observed and remains 1 for the remaning intervals. The weights are computed for the RISBS scoring rule. For other scoring rules, weights can be time-varying. As the first observation is fully observed (censored after $\tau_{max} = 2.9$) it has positive weights ($w_{1j} = \frac{1}{\hat{G}(t=2.9)}$) while the third observation has zero-weights not being fully observed.

survival function by estimating the increments $\alpha_{i,j} := \alpha_{i,j}(\mathbf{x}_i)$ between the survival functions at subsequent discrete time points/intervals τ_{j-1}, τ_j. We require the following properties to obtain a correctly specified survival function:

(a) $S(\tau_j|\mathbf{x}_i)$ needs to be monotonically decreasing, i.e. $\alpha_{i,j} \leqslant 0$;
(b) $S(\tau_j|\mathbf{x}_i) \in [0, 1]$;
(c) $\alpha_{i,j} \in [-1, 0]$.

In order to learn the increments $\alpha_{i,j}$, we require appropriate activation functions $\gamma_u(x) \in [0, 1], u \in \{1, 2\}$, such as the sigmoid or truncated ReLU function $f(\cdot) = \min(1, \max(0, \cdot))$, and a model g_l (e.g., a neural network) for the lth interval. By defining

$$\hat{S}(\tau_j|\mathbf{x}_i) = \gamma_2 \left(\sum_{l=1}^{j} (-\gamma_1(g_l(\mathbf{x}_i)) \right)$$

through increments $\hat{\alpha}_{i,l} := -\gamma_1(\hat{g}_l(\mathbf{x}_i)) \in [-1, 0]$, we obtain a monotonically decreasing survival function $\hat{S}(\tau_j|\mathbf{x}_i) = \gamma_2(\sum_{l=1}^{j} \hat{\alpha}_{i,l}) \in [0, 1]$ for each time interval τ_j with $\tau_0 = 0$ and $\hat{S}(\tau_0|\mathbf{x}_i) = 1$. In contrast to the parametric learning approach, this approach initially only produces discrete survival probabilities $\hat{S}(\tau_j|\mathbf{x})$. However, simple interpolation or smoothing can be applied to obtain meaningful predictions at time points between initial interval points τ_j.

3.2 Competing Risks

In the competing risks setting, we are interested in the time until the first of K competing events is observed. Let $E \in \{1, \ldots, K\}$ be a random variable representing the possible event types with realizations e. In this setting, we are typically interested in estimating $P(Y \leqslant \tau, E = e|\mathbf{x})$, i.e., the probability of

observing an event of type e before time τ given feature set $\mathbf{x}$. This quantity is usually referred to as cumulative incidence function (CIF) and denoted by $\mathrm{CIF}_k(\tau|\mathbf{x}), k = 1, \ldots, K$.

In the case of our parametric framework, we either learn the set of parameters for each competing risk k with separate sub-models for distribution parameters $\boldsymbol{\theta}_k$ or train a single joint model for all parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}\}_{k=1}^K$.

The CIF in the non-parametric case is modeled via

$$\widehat{\mathrm{CIF}}_k(\tau_j|\mathbf{x}_i) = \gamma_2 \left(\sum_{l=1}^{j} (\gamma_1(g_{l,k}(\mathbf{x}_i)) \right), \tag{3}$$

where $g_{l,k}$ are now interval- and risk-specific models.

To evaluate competing risk models, we can use the single-risk scoring rules, but need to define a cause-specific status indicator

$$d_{i,k} = d_i \mathbb{1}(e_i = k) \in \{0, \ldots, K\}, \tag{4}$$

where e_i is the cause observed for subject i. We further constrain

$$\sum_{k=1}^{K} \hat{F}_k(\tau_j|\mathbf{x}_i) \leqslant 1.$$

This can be achieved through the network architecture (by directly constraining the sum of the outputs) or by reweighting the resulting CIFs using their increments. Putting everything together, we optimize the competing risks objective

$$O^{\mathrm{CR}} = \sum_{k=1}^{K} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{J} \sum_{j=1}^{J} \mathrm{SR}_{i,k}(\tau_j|\hat{G}), \tag{5}$$

where $\mathrm{SR}_{i,k}$ is a single-event scoring rule (e.g. Table 1) with the status indicator d_i replaced by the competing risks indicator from Eq. (4). Predictions can be directly obtained from the model or internally reweighted depending on the scoring rule.

3.3 Optimization and Implementation

Gradient-based Optimization For all scoring rules discussed in this paper, first derivatives with respect to an arbitrary weight vector ω of

$$S(\tau|\mathbf{x}_i, \omega) \text{ or } F(\tau|\mathbf{x}_i, \omega)$$

exist. For example, for RISBS and a single observed individual i and interval j

$$\frac{\partial}{\partial \omega} \mathrm{SR}(\tau; \hat{G}) \propto F(\tau|\mathbf{x}_i, \omega) \frac{\partial}{\partial \omega} F(\tau|\mathbf{x}_i, \omega)$$

if $\hat{G}(t_i)$ is considered constant, which is usually the case if it is determined *a priori*. If $\hat{S}(\tau|\mathbf{x}_i)$ is differentiable itself, which is typically the case for neural networks, the model itself is differentiable.

Implementation Our framework can be easily implemented in a neural network. The network trunk can have an arbitrary shape whereas the output layer contains $J \times K$ units for both the parametric and non-parametric variant. The final layer of the parametric variant is a deterministic distributional layer that automatically enforces monotonicity by implementing the cumulative distribution or survival function of a parametric distribution. By the chain rule, backpropagation is given by the derivative of the SR w.r.t. the parameters $\boldsymbol{\theta}_k$ of the chosen survival distribution for each risk k times the gradients of $\boldsymbol{\theta}_k$ w.r.t. the network's weights. For the non-parametric version, the output layer is specified as in Eq. (3). The selected architectures must reflect the general modeling flow shown in Figure 2. Example architectures are depicted in Figure 3.

Overfitting can, e.g., be addressed by dropout layers throughout the network architectures and L2 regularization on the ultimate layer's weights. The parametric framework produces smooth, continuous estimates, the non-parametric one interpolates step-functions, as shown in Figure 3. In many cases, it is reasonable not to choose $\tau^* = \tau^{max}$ but slightly smaller (e.g. the 80th or 90th percentile) as late events have outlier character in small data sets [26].

Alternative Implementations While neural networks achieve the most versatile implementation, the idea can be generalized to arbitrary machine learning models. Particularly, the parametric framework with RISLL or ISLL as a loss function applies to some established machine learning models without further modification. For an assumed lognormal or log-logistic distribution with location μ and scale σ, we only need to model the linear predictor $\log(\tau)/\sigma - \mu/\sigma$, apply a logit (for an assumed log-logistic distribution) or probit (for an assumed lognormal distribution) link, and optimize a weighted binary cross-entropy loss with the weights being determined by the scoring rule used. As $\sigma > 0$ by definition, monotonicity must be enforced on the estimation of $\gamma = \frac{1}{\sigma}$ where γ is the linear coefficient for $\log(\tau)$ (the natural logarithm of the discretized follow-up time). For generalized linear models, linear estimates guarantee (weak) monotonicity. While technically, negative estimates are possible for γ, this doesn't occur in practice in our experience when estimated with maximum likelihood optimization. In boosting applications, monotonicity can be explicitly enforced through constraints on the estimation of the feature $\log(\tau)$. Essentially, we can fit a GLM using the discretized data with the following form:

$$P(d_{ij} = 1|\mathbf{x}_i, \tau) = g^{-1}(\gamma \log \tau + \mathbf{x}_{ij}\nu),$$

where τ is part of the feature matrix, γ is a scalar coefficient, ν is a vector of coefficients and $g()$ is the respective link function. Survival predictions can be directly made from the model. Location and scale parameters can be obtained indirectly via γ. Furthermore, we can use distributional regression software to generate predictions for any model class, e.g. trees, independent of their optimization.

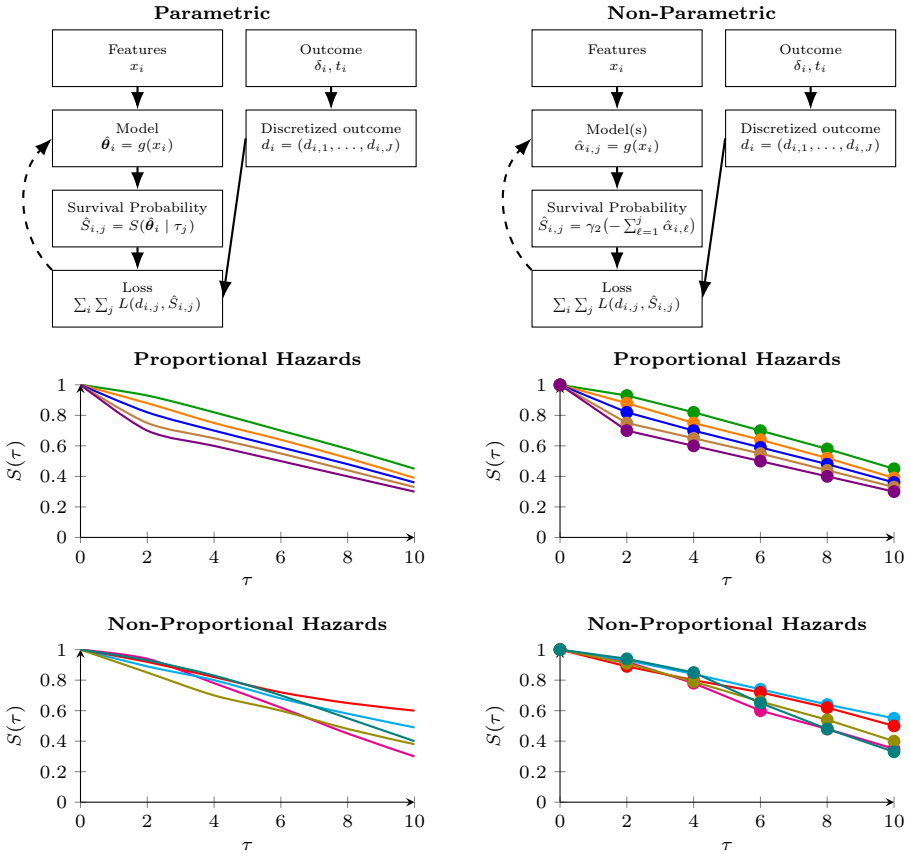

Fig. 2: Schematic model flow graphs (top) for parametric (left) and non-parametric (right) variants with schematic survival predictions (bottom). Features are used to learn model weights, optimized with respect to discretized outcomes and a loss function or **scoring rule**. Some models may require feature transformations (not depicted). In the parametric model, these weights determine a parameter vector ($\boldsymbol{\theta}_i$) for each individual (e.g., location and scale). In the non-parametric approach, survival increments $\alpha_{i,j}$ are estimated directly. The parameters $\boldsymbol{\theta}_i$ generate a continuous prediction of survival probabilities, resulting in smooth predictions (left bottom panels). For optimization, only the subset $\tilde{\mathcal{S}} = \{S_{i,j} \forall j \in 1, ..., J\}$ is needed. In the non-parametric case, this subset is available by construction, leading to point-wise predictions that are linearly interpolated (right bottom panels). Models are said to assume proportional hazards when survival functions do not intersect (upper panel of bottom graph).

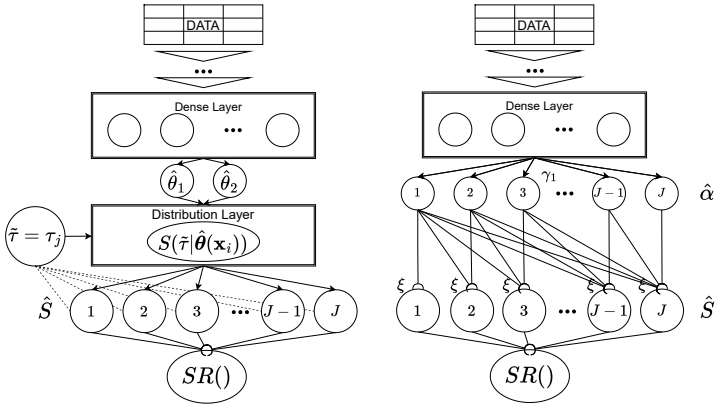

Fig. 3: Examples for architectures of our proposed method in the single risk case. Top: Parametric approach. We pass the data through a fully connected neural network to estimate the parameters (here θ_1 and θ_2) of a survival distribution. We generate predictions for each $\tilde{\tau} = \tau_j$ using the parameterized F. Bottom: Non-parametric approach. We pass the data through a fully-connected neural network to estimate the survival increments α_j and use them to generate survival predictions for each τ_j, where $\xi(\cdot) := \gamma_2(-\sum(\cdot))$.

4 Numerical Experiments

In the following sections, we evaluate our framework empirically. First, we test our approach with simulated data. While our proposed method can represent arbitrarily complex associations, our goal is to show that the proposed method can estimate parametric and semi-parametric SA methods that traditionally optimize likelihoods: Accelerated Failure Time models (AFT) and the Cox proportional hazard model (CPH). Furthermore, we explore how well our framework performs on benchmark data sets commonly used in SA for both single and competing risks. We benchmark our neural network implementation against other deep learning algorithms for a meaningful comparison. However, we also include the oblique random survival forest [ORSF; 18], which has been shown to yield good predictive performance in SA tasks. Last, we illustrate that the framework also applies to learners different from neural networks.

Evaluation and Tuning In all experiments, we make use of repeated subsampling. For all benchmark data sets, we repeat the subsampling 25 times, and, except for KKBox, use 80% of the data for training and 20% for evaluation. Repeated subsampling is preferred over cross-validation as the test set needs to be sufficiently large to estimate the censoring probability for all evaluation metrics. The number of subsamples depends on the complexity of the underlying experiments. For KKBox, 2 percent of the data (ca. 1,000 events) is sufficient for model evaluation. If tuning is necessary, we use a random search with a budget

of 25 configurations. The inner loop of the nested resampling is a five-fold cross-validation. Early stopping is performed when necessary based on the validation error. In all experiments, models are evaluated using the RISBS as our primary evaluation metric at different quantiles (25, 50, and 75 percent) of the follow-up.

4.1 Comparison to Maximum Likelihood Estimation

We first empirically check whether our approach can recover the parameters of a known event time distribution without explicitly using its likelihood for estimation. We compare the goodness of approximation with the true parameters and those estimated through maximum likelihood. To do so, we simulate event times from an AFT model via

$$\log(T_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \theta_2 \epsilon_i \tag{6}$$

with $\boldsymbol{\beta}^\top = (2, 0.5, 0.2, 0)$ and let ϵ follow the (i) Logistic, (ii) Normal, and (iii) Extreme value distribution, implying event times $T \sim F(\theta_1(x_1, x_2, x_3), \theta_2)$ that follow a (i) Loglogistc, (ii) Log-normal and (iii) Weibull distribution, respectively. We only let one parameter of the distribution depend on features, i.e. $\theta_1(x_1, x_2, x_3) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ and set $\theta_2 = \sigma = 0.4$.

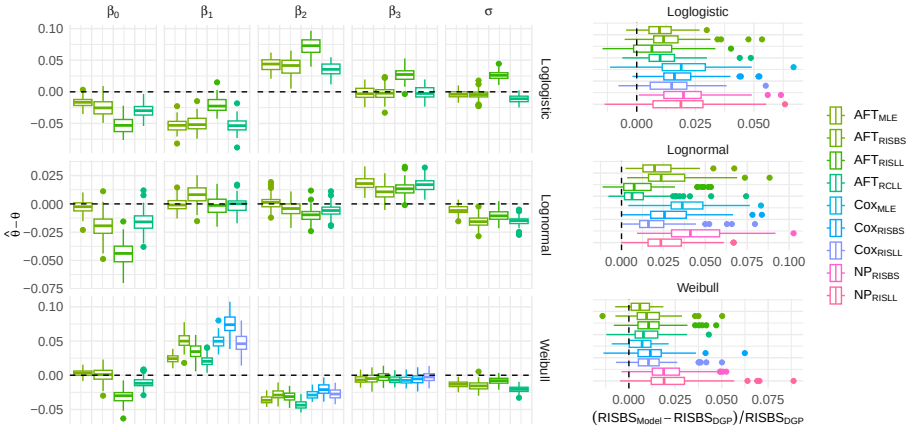

Fig. 4: Results of the comparison to ML estimation. **Left**: difference of estimated parameter $\hat{\theta}$ to oracle parameters θ. Parameter comparison for Cox PH models is limited to the coefficients β_1, β_2 and β_3, and the Weibull distribution. **Right**: Relative difference in the predictive performance w.r.t. the data generating process (DGP). Optimal performance is given by $\text{RISBS}_{\text{DGP}}$, obtained by using true parameters in the correctly specified model.

For the simulation, we draw $n = 1500$ event times from each distribution based on Eq. (6) and introduce censoring assuming a uniform distribution over

the follow-up, resulting in approximately 28% censoring. We repeat this $B = 100$ times, each time splitting the data into train (80%) and test (20%) data. In each iteration, we calculate

$$\beta_j - \hat{\beta}_{j,m};\ j = 0, \ldots, 3;$$

$$m \in \{\text{AFT}_{\text{MLE}}, \text{Cox}_{\text{MLE}}, \text{AFT}_{\text{SR}}, \text{Cox}_{\text{SR}}\},$$

where we either optimize the respective correctly specified AFT models and Cox PH models via maximum likelihood estimation (MLE) or one of the scoring rules (SR) based approaches as proposed in Section 3 . Additionally, we consider the difference between the estimated and true scale $\hat{\sigma} - \sigma$. In addition to recovering coefficients, we report the aggregated predictive performance of all models in terms of the RISBS, RISLL, and ISBS. For predictive performance evaluation, we also fit the non-parametric variant of our framework, NP_{SR} with SR $\in$ {RISBS, RISLL}. The AFT model estimated within our framework with RCLL provides a direct comparison to AFT_{MLE}. This results in a total of 4 AFT models (3 SR and 1 MLE), 2 Cox PH models (2 SR and 1 MLE), and 2 non-parametric models (both SR) for the main analysis.

The experimental results are presented in Figure 4. The methods specified within our framework recover the true coefficients well, with, however, a little approximation error. This approximation error is negligibly small when considering the predictive performance in the right panel. This finding holds for both, AFT and Cox PH model. When considering the model performances, we also see that a simple, untuned, yet regularized, non-parametric scoring-rule-based method performs comparably to the other (correctly specified) methods. In contrast to other deep AFT (e.g. [2]) approaches, our method allows the estimation of a variety of distributions using the parametric framework, including the Weibull distribution, which has repeatedly been reported to suffer from poor computational conditioning [e.g., 2].

4.2 Benchmark Study

In this section, we evaluate the models' predictive performance on synthetic and real-world data for both single (Table 2) and competing risks (Table 3) settings.

Single Risk We compare our framework to other popular deep learning models for survival analysis, namely nnet-survival [11], DeepHit [23], and DeepSurv [19], as well as the Countdown model (with an assumed log-normal distribution) as proposed in [2] ($\text{AFT}^{\text{deep}}_{\text{SCRPS}}$). For our framework we fit both, a non-parametric version NP_{RISBS} and a deep parametric variant $\text{AFT}^{\text{deep}}_{\text{RISBS}}$. Furthermore, we compare with baselines (KM and CPH) and Oblique Random Survival Forests (ORSF). All methods have been tuned over 50 configurations (10 for KKBox) except for the KM and CPH baselines. $\text{AFT}^{\text{deep}}_{\text{SCRPS}}$ approximates the model proposed in [2] (log-normal distribution, SCRPS as scoring rule). The model by [2] itself suffered from computational issues and did not result in adequate predictive performance. While not being perfectly identical to [2] $\text{AFT}^{\text{deep}}_{\text{SCRPS}}$ adapts their idea.

Table 2: Predictive performance of different learning algorithms for different data sets for a single event using the RISBS (smaller is better). We report the mean and standard deviation (in brackets) from 25 distinct train-test splits and highlight the best method in **bold**. The AFT models are also tuned with respect to the distribution family.

		KM	Cox PH	ORSF	DeepSurv	nnet	DeepHit	$\text{AFT}^{\text{deep}}_{\text{SCRPS}}$	$\text{AFT}^{\text{deep}}_{\text{RISBS}}$	NP_{RISBS}
tumor	Q25	7.4 (1.58)	6.6 (1.43)	6.5 (1.23)	6.6 (1.52)	6.5 (2.03)	6.7 (1.80)	**6.4** (1.30)	6.5 (1.37)	6.6 (1.30)
$n = 776$	Q50	13.0 (1.46)	11.7 (1.46)	11.8 (1.40)	11.6 (1.51)	11.5 (1.84)	11.6 (1.91)	11.4 (1.21)	**11.3** (1.24)	**11.3** (1.32)
$p = 7$	Q75	17.8 (1.17)	16.3 (1.47)	16.4 (1.40)	16.3 (1.40)	16.2 (1.37)	16.2 (1.67)	16.2 (1.29)	**16.1** (1.30)	**16.1** (1.39)
gbsg2	Q25	4.9 (0.80)	4.7 (0.75)	**4.6** (0.73)	4.7 (0.80)	4.9 (1.03)	**4.6** (1.01)	4.7 (0.75)	4.7 (0.69)	**4.6** (0.80)
$n = 2232$	Q50	10.1 (0.80)	9.3 (0.69)	9.2 (0.73)	9.3 (0.82)	9.2 (0.95)	9.5 (0.83)	9.1 (0.69)	9.4 (0.68)	**9.1** (0.79)
$p = 7$	Q75	15.4 (0.56)	13.9 (0.47)	13.6 (0.60)	13.7 (0.62)	13.6 (0.82)	13.4 (0.71)	13.4 (0.61)	13.3 (0.54)	**13.2** (0.61)
metabric	Q25	5.1 (0.64)	5.0 (0.61)	5.1 (0.56)	5.0 (0.60)	5.5 (0.51)	5.2 (0.55)	**4.9** (0.61)	**4.9** (0.56)	5.0 (0.65)
$n = 1904$	Q50	11.4 (0.80)	10.8 (0.78)	10.9 (0.67)	10.7 (0.60)	10.9 (0.58)	10.9 (0.70)	**10.4** (0.73)	**10.4** (0.72)	**10.4** (0.79)
$p = 9$	Q75	16.5 (0.61)	15.2 (0.60)	15.8 (0.66)	15.1 (0.55)	15.1 (0.46)	15.4 (0.52)	15.0 (0.61)	**14.8** (0.55)	**14.8** (0.59)
breast	Q25	2.2 (1.14)	2.2 (1.14)	2.1 (1.11)	2.3 (1.13)	2.2 (1.01)	2.2 (1.02)	**2.0** (0.86)	2.1 (0.97)	2.4 (1.12)
$n = 614$	Q50	4.6 (1.51)	4.6 (1.51)	**4.3** (1.44)	4.8 (1.59)	4.4 (1.35)	4.5 (1.42)	4.4 (1.18)	4.4 (1.26)	4.4 (1.54)
$p = 1690$	Q75	7.8 (1.62)	7.8 (1.62)	**7.0** (1.61)	7.6 (1.71)	7.2 (1.60)	7.3 (1.64)	7.2 (1.50)	7.3 (1.52)	7.3 (1.75)
KKBox	Q25	1.02 (0.05)	0.92 (0.04)		0.87 (0.05)	0.95 (0.07)	0.93 (0.06)	**0.85** (0.06)	0.86 (0.05)	0.90 (0.07)
$n = 865$ K	Q50	1.66 (0.06)	1.41 (0.05)	–	1.25 (0.05)	1.31 (0.06)	1.35 (0.06)	1.27 (0.05)	1.21 (0.05)	**1.20** (0.07)
$p = 6$	Q75	2.72 (0.09)	2.19 (0.07)		**1.89** (0.06)	2.00 (0.05)	2.01 (0.06)	1.94 (0.06)	1.92 (0.06)	**1.89** (0.06)
synthetic	Q25	6.7 (0.70)	4.7 (0.51)	4.2 (0.66)	3.5 (0.63)	4.4 (0.66)	3.9 (0.61)	**3.3** (0.48)	3.4 (0.40)	3.5 (0.56)
$n = 1500$	Q50	13.9 (0.78)	9.2 (0.53)	8.8 (0.81)	6.7 (0.78)	8.5 (0.71)	8.0 (0.69)	**6.4** (0.36)	**6.4** (0.38)	**6.4** (0.51)
$p = 4$	Q75	19.7 (0.26)	12.8 (0.56)	10.0 (1.01)	9.2 (0.76)	9.6 (0.83)	9.5 (0.67)	8.6 (0.45)	**8.3** (0.46)	8.4 (0.50)

We selected common data sets in the survival analysis literature primarily related to various medical conditions with observations in the high hundreds or low thousands and a large churn data set: tumor [4], gbsgs2 [25], metabric [8], breast [27], and mgus2 [22]. KKBox [17] is a large churn data set obtained from Kaggle that we processed for SA. For KKBox, ORSF evaluations, however, failed due to the size of the data set. For similar computational reasons, Cox PH only uses one feature for *breast*, where the other methods use all $p > n$ features.

Results In summary, the results indicate that our proposed methods provide good predictive performance, competitive with established methods. We observe that for the AFT model, both scoring rules (SCRPS and RISBS) have very similar performances. This finding is in line with [26], who empirically study differences between proper and improper scoring rules and report only small differences. NP tends to perform worst on early quantiles, indicating potential overfitting for the early cut points.

Competing Risks For competing risk, we compare our approach against the baseline methods Aalen-Johannsen estimator [AJ; 1] and competing risks piecewise exponential additive model [CR PAMM; 15] as well as DeepHit, which is typically considered when dealing with competing risks. Next, the real-world data set *mgus2*, we also consider a synthetic data set with a risk with complex (cause 1) and simple (cause 2) feature associations.

Results DeepHit and our method perform similarly well in estimating survival probabilities in a competing risk setting. In some cases, both methods cannot

Table 3: Prediction accuracy of different methods (columns) for different competing risks data sets (rows) evaluated using the ISBS (smaller is better). We report the mean and standard deviation (in brackets) from 25 distinct train-test splits and highlight the best method in **bold**.

		AJ	CR PAMM	DeepHit	$\text{AFT}^{\text{deep}}_{\text{ISBS}}$
mgus2	Q25	**1.1** (0.39)	**1.1** (0.39)	**1.1** (0.42)	1.2 (0.58)
(cause 1)	Q50	2.1 (0.59)	**2.0** (0.57)	2.2 (0.59)	2.2 (0.58)
$n = 1384$	Q75	3.2 (0.84)	3.2 (0.80)	**3.1** (0.85)	3.3 (0.79)
mgus2	Q25	9.1 (1.18)	**8.6** (1.16)	8.7 (1.04)	8.7 (1.10)
(cause 2)	Q50	14.3 (1.03)	13.2 (1.17)	**12.9** (1.20)	13.0 (1.34)
$p = 6$	Q75	18.2 (0.74)	15.8 (0.99)	**15.6** (1.07)	15.7 (1.22)
synthetic	Q25	5.4 (0.66)	3.7 (0.49)	3.2 (0.56)	**3.1** (0.45)
(cause 1)	Q50	10.9 (0.96)	7.3 (0.67)	**5.7** (0.69)	**5.7** (0.60)
$n = 1500$	Q75	16.9 (0.83)	11.5 (0.55)	**8.3** (0.77)	8.4 (0.65)
synthetic	Q25	2.2 (0.72)	**2.0** (0.59)	**2.0** (0.62)	2.1 (0.53)
(cause 2)	Q50	5.8 (0.97)	**4.7** (0.78)	5.0 (0.79)	4.9 (0.73)
$p = 4$	Q75	9.7 (1.06)	7.9 (0.94)	7.9 (1.00)	**7.8** (0.79)

outperform a CR PAMM that assumes linear effects on the log hazards. However, this is most likely due to the differences between the empirical incidence of the two causes (only 12 % of observed events in *mgus2* are due to cause 1) and by construction (cause 2 of *synthetic* assumes only linear associations).

Alternative Implementations To test alternative implementations of our framework, we use two datasets: *simple* is the simulation introduced in Section 4.1 and reflects linear effects only, while *complex* uses four features that partially exhibit non-linearities and interactions. Both settings assume a log-normal distribution. As learners, we include KM and Cox PH for baseline comparisons and prototype implementations for an XGBoost ([6]), generalized additive model (GAM; [16]), and soft regression tree. Performance is reported quantile-wise and estimated with 25 times repeated subsampling (80-20). Methods are not tuned; our goal is only to show that these alternatives work in principle. We find that all methods perform better than the baseline (KM). However, a single tree does not outperform a Cox PH model. In the simple setting, GAM performs best. This is because the Cox PH model is slightly misspecified in the presence of log-normally distributed survival times. Unsurprisingly, the (untuned) XGBoost model overfits in this simple regime. For the complex setting, we allowed the GAM to capture non-linearities, yet no interactions. While this significantly boosts performance over Cox PH, the XGBoost approach achieves very good generalization despite being untuned. All in all, this suggests that the methods work as intended and provide reasonable results.

4.3 Ablation: Altering the Scoring Rule

As discussed in Section 3, our framework is scoring-rule agnostic. While we mainly focused on RISBS in the experiments, we also implemented all other

Table 4: Comparison of predictive performances of different learners (columns) for different datasets (rows). We report the mean (top) and standard deviation (below in brackets) from 25 distinct train-test splits. The best method is highlighted in **bold**.

		KM	Cox PH	$\text{AFT}^{\text{GAM}}_{\text{RISLL}}$	$\text{AFT}^{\text{tree}}_{\text{RISBS}}$	$\text{AFT}^{\text{XGB}}_{\text{RISLL}}$
simple	Q25	5.1	3.2	**3.0**	4.5	3.2
		(0.46)	(0.46)	(0.43)	(0.87)	(0.42)
$n = 1500$	Q50	10.7	6.3	**6.1**	8.9	6.3
		(0.87)	(0.43)	(0.40)	(0.69)	(0.43)
$p = 3$	Q75	15.5	8.7	**8.5**	13.3	8.9
		(0.42)	(0.51)	(0.44)	(0.60)	(0.51)
complex	Q25	6.7	4.7	3.8	6.2	**3.6**
		(0.71)	(0.52)	(0.54)	(0.59)	(0.47)
$n = 1500$	Q50	13.9	9.2	7.2	12.7	**6.6**
		(0.78)	(0.53)	(0.59)	(0.62)	(0.44)
$p = 4$	Q75	19.7	12.8	9.6	18.2	**8.6**
		(0.26)	(0.56)	(0.51)	(0.51)	(0.40)

scoring rules from Table 1. To investigate their influence, we study how the results from the benchmarking study qualitatively change when the optimized scoring rule is changed for the parametric sub-framework.

Table 5: Predictive performance of the ablation study. For two data sets from the benchmark study, we changed the training scoring rule to RISLL, RCLL, and ISBS, respectively. For RISLL and ISBS we only consider the 75 % percentile for τ^*. We also evaluate using the same scoring rules. RCLL is reported as -RCLL.

		$\text{AFT}^{\text{deep}}_{\text{RISBS}}$	$\text{AFT}^{\text{deep}}_{\text{RISLL}}$	$\text{AFT}^{\text{deep}}_{\text{RCLL}}$	$\text{AFT}^{\text{deep}}_{\text{ISBS}}$
gbsg2	RISBS	13.6 (0.54)	**13.4** (0.59)	13.5 (0.58)	13.6 (0.56)
	RISLL	40.8 (1.39)	**40.7** (1.35)	40.8 (1.47)	41.2 (1.55)
$n = 2232$	RCLL	2.67 (0.76)	2.65 (0.72)	**2.60** (0.70)	2.69 (0.79)
$p = 7$	ISBS	13.1 (0.53)	13.1 (0.59)	**13.0** (0.62)	13.2 (0.63)
synthetic	RISBS	**8.3** (0.46)	**8.3** (0.50)	8.4 (0.45)	8.6 (0.63)
	RISLL	26.7 (1.18)	**26.5** (1.17)	26.6 (1.28)	26.9 (1.20)
$n = 1500$	RCLL	1.58 (0.06)	**1.57** (0.06)	1.59 (0.06)	1.60 (0.07)
$p = 4$	ISBS	**8.5** (0.38)	**8.5** (0.35)	8.6 (0.34)	8.6 (0.58)

Results In Table 5, we observe that using different scoring rules only leads to minor changes for both training and evaluation. Choosing ISBS as the evaluation metric seems to give a slight advantage to the model, which is also trained on an ISBS loss. Among the proper scoring rules, we do not observe a similar pattern.

5 Discussion & Conclusion

We proposed a new method for estimating event time distributions from censored data, including competing risks, using scoring rules as a loss function.

Our framework can be seamlessly integrated into neural networks, but also into tree-based models, generalized additive models, and gradient boosting. Empirical results demonstrate that the proposed integration of scoring rules yields good predictive performance, and the proposed framework is on par with other state-of-the-art approaches. We particularly highlight the results from the recovery and ablation study in sections Section 4.1 and Section 4.3 that confirm that the framework can be used in a variety of settings and configurations. This validates theoretical propositions and claims and provides proof-of-concept evidence for the entire framework.

The use of scoring rules in survival model training can be viewed as another method for reducing survival problems into classification or regression problems through discretization. By extending the framework to any arbitrary model class, this work makes an important taxonomic contribution. We show that this specific discretization (in combination with a suitable scoring rule) is generally applicable (e.g. in single event and competing risks settings), similar to piecewise exponential models or discrete hazard models. This finding goes beyond previous attempts that assessed scoring-rule-based model training. In contrast to them, this work provides a rigorous separation of a learner into a loss (scoring rule in combination with discretization), a hypothesis space (model classes), and an optimization (dependent on model class). This point of view allows a very agnostic application of the framework and makes it easy to extend.

Limitations and Future Work While our approach works for right-censored data and competing risks, other SA use cases, such as interval-censored data, multi-state modeling, or recurrent events, are contemporary challenges that could be an interesting extension of our proposal for future research. The choice of a specific scoring rule and respective advantages and disadvantages for model optimization could also be explored further in the future.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Aalen, O.: Nonparametric inference for a family of counting processes. The Annals of Statistics pp. 701–726 (1978)

[2] Avati, A., Duan, T., Zhou, S., Jung, K., Shah, N.H., Ng, A.Y.: Countdown regression: sharp and calibrated survival predictions. In: Uncertainty in Artificial Intelligence. pp. 145–155. PMLR (2020)

[3] Bender, A., Rügamer, D., Scheipl, F., Bischl, B.: A general machine learning framework for survival analysis. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 158–173. Springer (2020)

[4] Bender, A., Scheipl, F.: pammtools: Piece-wise exponential Additive Mixed Modeling tools. arXiv:1806.01042 [stat] (2018)

[5] Bender, A., Scheipl, F., Hartl, W., Day, A.G., Küchenhoff, H.: Penalized estimation of complex, non-linear exposure-lag-response associations. Biostatistics **20**(2), 315–331 (2019)

[6] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)

[7] Cox, D.R.: Regression Models and Life-Tables. Journal of the Royal Statistical Society. Series B (Methodological) **34**(2), 187–220 (1972)

[8] Curtis, C., Shah, S.P., Chin, S.F., Turashvili, G., Rueda, O.M., et al.: The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature **486**(7403), 346–352 (2012)

[9] Dawid, A.P., Musio, M.: Theory and applications of proper scoring rules. Metron **72**(2), 169–183 (2014)

[10] Friedman, M.: Piecewise exponential models for survival data with covariates. The Annals of Statistics **10**(1), 101–113 (1982)

[11] Gensheimer, M.F., Narasimhan, B.: A scalable discrete-time survival model for neural networks. PeerJ **7**, e6257 (2019)

[12] Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association **102**(477), 359–378 (2007)

[13] Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Statistics in medicine **18**, 2529–2545 (1999)

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

[15] Hartl, W.H., Kopper, P., Bender, A., Scheipl, F., Day, A.G., Elke, G., Küchenhoff, H.: Protein intake and outcome of critically ill patients: analysis of a large international database using piece-wise exponential additive mixed models. Critical Care **26**(1), 1–12 (2022)

[16] Hastie, T., Tibshirani, R.: Generalized additive models. Statistical science **1**(3), 297–310 (1986)

[17] Howard, A., Chiu, A., McDonald, M., Kan, W., Yianchen: Wsdm - kkbox's music recommendation challenge (2017)
[18] Jaeger, B.C., Long, D.L., Long, D.M., Sims, M., Szychowski, J.M., Min, Y.I., Mcclure, L.A., Howard, G., Simon, N.: Oblique random survival forests. The Annals of Applied Statistics **13**(3), 1847–1883 (2019)
[19] Katzman, J., Shaham, U., Cloninger, A., Bates, J., et al.: Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC medical research methodology **18**(1), 1–12 (2018)
[20] Kopper, P., Wiegrebe, S., Bischl, B., Bender, A., Rügamer, D.: DeepPAMM: Deep piecewise exponential additive mixed models for complex hazard structures in survival analysis. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 249–261 (2022)
[21] Kvamme, H., Borgan, Ø.: Continuous and discrete-time survival prediction with neural networks. Lifetime data analysis **27**(4), 710–736 (2021)
[22] Kyle, R.A., Therneau, T.M., Rajkumar, S.V., Offord, J.R., Larson, D.R., Plevak, M.F., Melton, L.J.: A Long-Term Study of Prognosis in Monoclonal Gammopathy of Undetermined Significance. New England Journal of Medicine **346**(8), 564–569 (2002)
[23] Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. In: 23nd AAAI Conference on Artificial Intelligence (2018)
[24] Rindt, D., Hu, R., Steinsaltz, D., Sejdinovic, D.: Survival regression with proper scoring rules and monotonic neural networks. In: International conference on artificial intelligence and statistics. pp. 1190–1205. PMLR (2022)
[25] Schumacher, M., Bastert, G., Bojar, H., Hübner, K., Olschewski, M., et al.: Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. Journal of Clinical Oncology **12**(10), 2086–2093 (1994)
[26] Sonabend, R., Zobolas, J., De Bin, R., Kopper, P., Burk, L., Bender, A.: Examining properness in the external validation of survival models with squared and logarithmic losses. arXiv preprint arXiv:2212.05260 (2022)
[27] Ternes, N., Rotolo, F., Heinze, G., Michiels, S.: Identification of biomarker-by-treatment interactions in randomized clinical trials with survival outcomes and high-dimensional spaces. Biometrical Journal **59**(4), 685–701 (2017)
[28] Tutz, G., Schmid, M.: Modeling Discrete Time-to-Event Data. Springer Series in Statistics, Springer International Publishing, Cham (2016)
[29] Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: A survey. ACM Computing Surveys (CSUR) **51**(6), 1–36 (2019)
[30] Wei, L.J.: The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. Statistics in medicine **11**(14-15), 1871–1879 (1992)
[31] Wiegrebe, S., Kopper, P., Sonabend, R., Bischl, B., Bender, A.: Deep learning for survival analysis: a review. Artificial Intelligence Review **57**(3), 65 (2024)
[32] Yanagisawa, H.: Proper scoring rules for survival analysis. In: International Conference on Machine Learning. pp. 39165–39182. PMLR (2023)