entropy

MDPI

Article

Distance-Metric Learning for Personalized Survival Analysis

Wolfgang Galetzka [1,*], Bernd Kowall [1], Cynthia Jusi [2], Eva-Maria Huessler [1] and Andreas Stang [1]

[1] Institute of Medical Informatics, Biometrics and Epidemiology, University Hospital Essen, 45130 Essen, Germany
[2] Nisso Chemical Europe GmbH, 40212 Düsseldorf, Germany
* Correspondence: wolfgang.galetzka@uk-essen.de

Abstract: Personalized time-to-event or survival prediction with right-censored outcomes is a pervasive challenge in healthcare research. Although various supervised machine learning methods, such as random survival forests or neural networks, have been adapted to handle such outcomes effectively, they do not provide explanations for their predictions, lacking interpretability. In this paper, an alternative method for survival prediction by weighted nearest neighbors is proposed. Fitting this model to data entails optimizing the weights by learning a metric. An individual prediction of this method can be explained by providing the user with the most influential data points for this prediction, i.e., the closest data points and their weights. The strengths and weaknesses in terms of predictive performance are highlighted on simulated data and an application of the method on two different real-world datasets of breast cancer patients shows its competitiveness with established methods.

Keywords: survival analysis; machine learning; metric learning; kernel regression; personalized medicine

Citation: Galetzka, W.; Kowall, B.; Jusi, C.; Huessler, E.-M.; Stang, A. Distance-Metric Learning for Personalized Survival Analysis. *Entropy* **2023**, *25*, 1404. https://doi.org/10.3390/e25101404

Academic Editors: Leandro Pardo and Pedro Miranda

Received: 13 August 2023
Revised: 21 September 2023
Accepted: 26 September 2023
Published: 30 September 2023

1. Introduction

One important task in medical research and patient care is accurately predicting clinical outcomes for individual patients. Traditionally, (semi-)parametric approaches have been used, assuming a specific functional relationship, often linear, between the patient's clinical characteristics and the outcome. However, there is a growing trend towards utilizing supervised machine learning models for this task due to their flexibility and ability to overcome these limitations. Although the focus of this paper is on prediction, machine learning models may not only offer more accurate outcome predictions but also support causal inference. For instance, they can be used to estimate the average treatment effect in the presence of confounding, using targeted maximum likelihood estimation [1]. Another application is the estimation of heterogeneous treatment effects, where the focus is on exploring how the effect of a treatment is influenced by the patient's clinical characteristics [2,3].

The application of machine learning methods in medical sciences for classification or regression tasks is relatively straightforward. However, when it comes to time-to-event outcomes, which are of central importance in studying chronic diseases that progress over time (such as cancer or cardiovascular diseases), the challenges are more pronounced. This is because standard metrics used for training models, such as log-binomial loss or mean squared error, are not directly applicable due to the presence of (right) censored observations. Right censoring occurs when the true event time is not known for certain individuals, but only a lower bound of it. It can occur due to the loss of follow-up or if the event is not observed until the end of the study period. Despite these challenges, researchers have successfully adapted various machine learning methods [4], such as support vector machines [5], regression trees [6], forests [7], and more recently, deep neural networks [8,9], to handle time-to-event data. Although random survival forests and neural networks, in

particular, provide accurate prediction, these models lack interpretability; that is, they do not provide explicit explanations for their predictions. Therefore, they are also referred to as "black box" models [10]. Furthermore, models such as random survival forest still have difficulties in modeling non-proportional hazards.

Instance-based algorithms are a class of machine learning algorithms that provide explicit explanations for individual predictions [11] (Chapter 8). In instance-based learning, the prediction for a new instance (i.e., a data point) is made by comparing it to the instances used for training and calculating weights. These weights are determined using a kernel function and depend on the metric in the covariate space. Training instances that are close to the new instance are assigned higher weights, while those that are far away receive lower weights. The prediction for the new instance is then obtained by taking the weighted average of the outcomes of the training instances. To ensure precise predictions, it is essential to optimize the distance measure and ensure that instances close to each other have similar outcomes. While there are numerous metric learning methods available for classification [12,13] and regression tasks [14], the same is not true for time-to-event outcomes. Existing approaches to time-to-event analysis rely on a weighted Kaplan–Meier estimate for survival prediction, but they have shown mixed results in terms of predictive performance and were not competitive with the random survival forest. Some of these approaches utilize a pre-defined metric [15,16] for calculating the weights, while others discretize the time and learn the metric through maximum likelihood estimation [17].

In this paper, we propose an algorithm that predicts the survival probability based on weighted instances by estimating parameters for the piece-wise exponential distribution. Due to the use of the piece-wise exponential distribution, we are able to model non-proportional hazards. The log-likelihood for continuous distributions is utilized as the loss function to optimize the metric on the covariate space. To minimize the required number of intervals, for an accurate fit of the survival curve's shape, we employ a global transformation of time. Furthermore, we conduct a comparative analysis of the algorithm's performance on simulated and real-world data against two widely used approaches: the de facto standard, the Cox proportional hazards model, and the easily applicable and widely adopted random survival forest.

The structure of the remaining paper is as follows: In Section 2.1, we formally introduce the problem and provide a clear definition. We briefly sketch the well-established Cox proportional hazards and random survival forest approaches in Section 2.2. In Sections 2.3 and 2.4, we present our approach, the kernel survival prediction with piece-wise exponential distributions. Moving on to the experimental setup, in Section 3.1, we outline the conducted experiments and describe the performance measures utilized. We also discuss the distributions used for simulating data in Section 3.2. In Section 4, we present the results obtained from the conducted experiments. Finally, in the concluding Section 5, we summarize our findings and provide an outlook for future research directions.

2. Materials and Methods

2.1. Problem Setting

By capital letters X, Y, C, we denote the random variables for the covariates, the time until an event, and the time until censoring respectively. Of those, only realizations of X are observed. Instead of Y and C, the time $T = \min(Y, C)$ until event or censoring, whichever occurs first, and $\Delta = \mathbb{1}(Y \leq C)$, an indicator which is 1 if an event occurred before censoring and 0 otherwise, are observed. Our aim is to estimate the conditional survival probability

$$S(t|x) := P(Y > t|X = x),$$

given i.i.d. realizations $\mathcal{D} = (x_1, t_1, \delta_1), \ldots, (x_n, t_n, \delta_n) \in \mathbb{R}^m \times \mathbb{R}_{>0} \times \{0, 1\}$ of X, T and Δ. Equivalently, we can estimate the hazard rate $\lambda(t|x)$ or the cumulative hazard $\Lambda(t|x)$, which are given by

$$\lambda(t|x) := \lim_{\epsilon \to 0} \frac{S(t|x) - S(t+\epsilon|x)}{\epsilon S(t|x)}, \text{ and}$$
$$\Lambda(t|x) := \int_0^t \lambda(t'|x) dt'. \tag{1}$$

The survival function can be retrieved from the cumulative hazard through $S(t|x) = \exp(-\Lambda(t|x))$. The hazard rate at time t can be interpreted as the risk of an event in the next instant, given no event occurred until t. Oftentimes it is more convenient to model the hazard λ instead of modelling S directly. Throughout, we assume that C and Y are independent given X. Thus [18], the likelihood of an observation (x, t, δ) satisfies

$$f(T = t, \Delta = \delta | X = x) \propto \lambda(t|x)^\delta \exp(-\Lambda(t|x)). \tag{2}$$

2.2. Survival Prediction Based on Proportional Hazards

In this subsection, we quickly review the Cox regression and the idea behind the splitting rule for trees used to build the random survival forest. For the conventional Cox regression model, one assumes that the hazard rate $\lambda(t|x)$ can be factorized in one term only depending on t, the so-called baseline hazard λ_0, and one term only depending on x, i.e.,

$$\lambda(t|x) = \lambda_0(t) \exp(h(x)).$$

Furthermore, one assumes that h is linear in x, i.e., $h(x) = h_\beta(x) = \sum_{i=1}^m \beta_i x_i$ for some coefficients $\beta_i \in \mathbb{R}$. Under this assumption, we have

$$\frac{\lambda(t|x_1)}{\lambda(t|x_2)} = \frac{\exp(h(x_1))}{\exp(h(x_2))},$$

i.e., the hazard rates of x_1 and x_2 are proportional to each other independent of time t. One can show that the likelihood can be maximized with respect to β independently of $\lambda(t)$, by maximizing the partial likelihood

$$L_C = \prod_{i|\delta_i=1} \frac{\exp(h_\beta(x_i))}{\sum_{j|t_j \geq t_i} \exp(h_\beta(x_j))}.$$

To obtain survival predictions from the Cox regression model, it is then necessary to calculate the cumulative baseline hazard $\lambda(t)$ using the Breslow estimator. Instead of using a linear h it is also possible to use more complex structures like deep neural networks [8].

The partial likelihood L_C can also be used to derive a splitting criterion for random survival trees. To this end, each split gives rise to a binary covariate indicating which child node an observation is assigned to. Calculating the minimal partial likelihood L_C with respect to that covariate is a measure of the predictive quality of the split and, hence, can be used to pick the best split. This is, in fact, the basic idea of the splitting rule used for the most common random survival forests [7,19]. To estimate the cumulative hazard or survival probability within each leaf, one can utilize either the Nelson–Aalen estimate or the Kaplan–Meier estimate for the instances contained in the leaf.

One can expect that for both methods, the Cox regression model and the random survival forest, predictive performance decreases with increasing non-proportionality of the hazards.

2.3. Survival Prediction with Kernels

Predicting the survival probabilities for a patient with covariates x, based on the training data $(x_i, t_i, \delta_i)_{i=1,\dots,n}$, involves two steps. Firstly, we assign weights w_i to each instance in the training set, quantifying the similarity between x and x_i. Secondly, we utilize the weighted training data to estimate the parameters of a survival distribution for x.

The notion of similarity in the covariate space is formalized through a kernel function, denoted as $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$. A kernel function is a symmetric function which satisfies that the matrix $k(\tilde{x}_i, \tilde{x}_j)_{i,j=1,\dots,l}$ is positive semi-definite for any choice of $l \in \mathbb{N}$ and $\tilde{x}_1, \dots, \tilde{x}_l \in \mathbb{R}^m$. Although a variety of different kernel functions exists [20], we have for simplicity chosen to focus on the radial basis function. Consequently, the weights w_i in our case are calculated as

$$w_i = \exp(-d^2(x, x_i)),$$

where $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is a (pseudo)metric. For our algorithm, fitting entails learning d. We confine ourselves to the subset of metrics where the distance is induced by an inner product, meaning that we have

$$d^2(x, x_i) = (x - x_i)^\top A(x - x_i),$$

for a positive semi-definite, symmetric matrix A. Using the Cholesky decomposition, we have $A = U^\top U$ for an upper triangular matrix U. Hence, we can simply learn U instead of A and the expression for the weights w_i simplifies to $w_i(U, x) = \exp(-\|Ux - Ux_i\|^2)$, where $\|\cdot\|$ denotes the Euclidean norm. This first step is common in instance-based learning and used for regression tasks [14] as well as for survival prediction [15,17].

What mainly distinguishes our approach from existing methods, in regard to learning a metric for survival prediction, is the second step, i.e., the actual prediction of the survival probabilities using $(w_i(U, x), t_i, \delta_i)_{i=1,\dots,n}$, and the weighted outcomes of the given training data $\mathcal{D}$. While in existing approaches the survival predictions are derived from the weighted outcomes using the non-parametric weighted Kaplan–Meier estimate, we use the weighted outcomes to estimate the parameters of a piece-wise exponential distribution. To this end, it is necessary to split the time into $k \in \mathbb{N}$ disjoint intervals, i.e., $\mathbb{R}_{>0} = (\tau_0, \tau_1] \cup (\tau_1, \tau_2] \cup \dots \cup (\tau_{k-1}, \tau_k)$, with $\tau_0 = 0, \tau_k = \infty$ and k being a chosen number. Denote by $\delta_{ij} := \delta_i \mathbb{1}(t_i \in (\tau_{j-1}, \tau_j])$ if for the i-th training instance the event occurred in the j-th interval, by $r_j(t) := |(0, t] \cap (\tau_{j-1}, \tau_j]|$ the time at risk in interval j for an observed time t and, furthermore, by $j(t)$ the index of the interval containing t. The hazard rate in the j-th interval, as the maximum likelihood estimate of the weighted outcomes under the piece-wise exponential distribution, is then given by

$$\lambda_j(x) = \frac{\sum_{i=1}^n w_i(U, x)\delta_{ij}}{\sum_{i=1}^n w_i(U, x) r_j(t_i)}.$$

Its derivation can be found in Appendix A. Using (1), we obtain

$$\Lambda(t|x) = \sum_{j=1}^{k} \lambda_j(x) r_j(t) \text{ and}$$

$$S(t|x) = \exp(-\sum_{j=1}^{k} \lambda_j(x) r_j(t)),$$

for the cumulative hazard and the survival probability. While within the intervals the hazard rates, and therefore the hazard ratios, are constant, they will in general differ across intervals, i.e., $\lambda_j(x)/\lambda_j(\tilde{x}) \neq \lambda_l(x)/\lambda_l(\tilde{x})$, for $j \neq l$.

Let us now finally describe how to obtain the metric-inducing transformation U. We obtain the log-likelihood for an observation (x, t, δ) under the model by plugging the above equation in (2). Hence, we obtain

$$\log(f(T = t, \Delta = \delta | x)) = \delta \log\left(\frac{\sum_{i=1}^{n} w_i(U, x)\delta_{ij(t)}}{\sum_{i=1}^{n} w_i(U, x) r_{j(t)}(t_i)}\right) - \sum_{j=1}^{k} \frac{\sum_{i=1}^{n} w_i(U, x)\delta_{ij}}{\sum_{i=1}^{n} w_i(U, x) r_j(t_i)} r_j(t),$$

where we adopted the convention $0 \log 0 = 0$. However, instead of obtaining U by maximization of the log-likelihood on the training data, we add a regularization term $\eta \|U\|^2$, with $\eta > 0$ being another hyperparameter, and, as in Weinberger and Tesauro [14], set $w_i(U, x_i) = 0$, both to prevent overfitting. More specifically, by setting $w_i(U, x_i) = 0$ we make sure that, for learning U, the i-th instance is not used to predict its own outcome. Adding the term $\eta \|U\|^2$ results in a smaller U, which decreases the variance of the weights. Other regularization schemes, such as the more general elastic net regularization, could also be used [21]. All in all, U is then given by

$$U = \underset{\tilde{U}}{\operatorname{argmin}} \left(\eta \|\tilde{U}\|^2 + \frac{1}{n} \sum_{l=1}^{n} \left(\delta_l \log \frac{\sum_{i=1}^{n} w_i(\tilde{U}, x_l)\delta_{ij(t_l)}}{\sum_{i=1}^{n} w_i(\tilde{U}, x_l) r_{j(t_l)}(t_i)} - \sum_{j=1}^{k} \frac{\sum_{i=1}^{n} w_i(\tilde{U}, x_l)\delta_{ij}}{\sum_{i=1}^{n} w_i(\tilde{U}, x_l) r_j(t_i)} r_j(t_l) \right) \right)$$

We note that the computational complexity of the loss grows quadratically with the number of training samples, which might make our approach unpractical for large datasets and may require adaptations. Additionally, it is important to acknowledge the non-convex nature of the loss function, which may necessitate multiple random initializations of the matrix U when utilizing a gradient-based optimizer.

Furthermore, let us consider the available hyperparameters. In this paper, we solely focus on the Gaussian radial basis function as the kernel function, but other options exist. Moreover, the number and location of time intervals for the piece-wise exponential distribution have to be picked. Through our experiments, we will demonstrate that typically only a few ($\leq$5) intervals are needed for the algorithm to perform well when using the time transformation introduced in the next subsection. Lastly, the regularization parameter, denoted by η, has to be selected.

2.4. Transformation of Time

The shape of the survival curve might be very different from an exponential distribution, and thus to appropriately model the survival we would need a high number of time intervals. To avoid this issue, we use a time transformation introduced in LeBlanc and Crowley [22]. This transformation, denoted as $\varphi : t \to \tilde{t}$, is designed such that the transformed times for observations with events correspond to the Nelson–Aalen estimates of the cumulative hazard. For censored observations, the transformed times are linearly interpolated. This approach ensures that the survival prediction of the (unweighted) exponential model on the transformed times coincides with those of the Nelson–Aalen model.

By using this transformation, the need for a high number of time intervals is reduced, as the different time intervals are only required to model non-proportional hazards.

3. Experiments on Simulated and Real-World Data

3.1. Description of Data and Evaluation Criteria

We conducted a performance evaluation of the kernel prediction method by comparing it against two established approaches in survival analysis on simulated and real-world datasets: Cox regression, which serves as the de facto standard, and survival forest, which is widely used and easily applicable.

For the simulated data, we varied several parameters, such as the training data sizes, the degree of interaction, the nonlinear rescaling of covariates, the non-proportionality of hazards, and the proportion of censoring to evaluate the effect on the predictive perfor-

mance. Training and evaluation data were drawn from the same distributions. For each setting, we ran 100 simulations.

In addition to the simulated data, we also evaluated the performance of our method on two publicly available datasets: the METABRIC dataset [8,23] (1904 observations, 9 covariates, 42% censoring) and the Rotterdam/GBSG dataset [8,24,25] (2232 observations, 7 covariates, 43% censoring). Both datasets contain clinical attributes and information on survival of breast cancer patients. To obtain an unbiased assessment, we partitioned each dataset into randomly selected separate training and test sets. To investigate the influence of training data size on performance, we varied the size of the training set, including proportions of 30%, 50%, and 70% of the total data. We considered 100 different splits for each training data size to account for random variation and obtain reliable performance estimates. Unlike the random survival forest, the Cox regression and the introduced kernel prediction are not invariant under monotone transformations of the numerical covariates. Hence, to illustrate the importance of the choice of the appropriate scale on real-world data, we performed an additional analysis on the Rotterdam/GBSG dataset where we replaced the two covariates indicating levels of progesterone and estrogen, given in fmol/mg, by their logarithm, as this is a more natural scale for concentrations.

Antolini's concordance index [26] and the integrated Brier score [27] were employed as performance measures. The concordance index was utilized to evaluate the discriminative performance. To calculate Antolini's concordance index, one examines all pairs within the test set that can be ordered, meaning we can determine which event occurs first. This is the case if an event took place at the earlier event time, i.e., this set consists of all pairs $i, j = 1, \ldots, n$ which satisfy the conditions $t_i < t_j$ and $\delta_i = 1$. The concordance index is the proportion of those pairs which are ordered correctly by the model in the sense that $\hat{S}(t_i|x_i) < \hat{S}(t_i|x_j)$. Hence, the concordance index is an estimate for

$$\mathbb{P}(S(T_i|X_i) < S(T_i|X_j)|T_i < T_j \text{ and } \Delta_i = 1),$$

for a randomly chosen pair i, j of subjects. The higher the concordance index the better. A concordance index of 0.5 indicates that the predictions are not informative in terms of discrimination. The integrated Brier score on the other hand primarily assesses calibration. The Brier score at a specific time point is calculated as the weighted mean of squared differences between the predicted survival probability and the corresponding state (1 for alive, 0 for deceased) of an instance at that time. The inverse probability of censoring weighting is employed as a weighting scheme, ensuring that the Brier score for time t provides an unbiased estimate of

$$\mathbb{E}\Big[(\mathbb{1}(Y > t) - S(t|X))^2\Big].$$

To obtain the integrated Brier score, a single performance index for the whole follow-up, the Brier score is integrated numerically up to a specified time, which we chose to be the 95% quantile of the event times. As the final evaluation measure for a model M, we do not use the integrated Brier score $B_I(M)$ directly, but the improvement with respect to the Kaplan–Meier estimator, $R^2 := 1 - B_I(M)/B_I(KM)$, as suggested in Graf et al. [27] to measure the explained residual variation. While these two criteria are the most common to evaluate model performance [4], further model measures and diagnostics, such as graphical analysis of calibration [28], exist.

The experiments were carried out in Python 9.3.16. The considered hyperparameters for the methods were optimized using grid search and 10-fold cross-validation on the training data. The implementation in [19] of the random survival forest was used. The number of features used per split and the minimal number of observations per node were optimized. The classical Cox regression, with the linear dependency of the log hazard on the covariates, was carried out with [29]. For the kernel prediction, we only determined the regularization parameter η by cross-validation, the number of time intervals was set to four for all experiments. The quantiles of the observed event times were chosen as splitting

times to ensure a sufficient number of events in each time interval. Optimization of the log-likelihood was carried out with LBFGS [30] implemented in PyTorch 1.12.1 [31] on the standardized predictors with the identity matrix as the initial value. More details on the considered hyperparameters can be found in Appendix B.

3.2. Description of Simulation Settings

We conducted four series of simulations. In the first, series A, we varied the degree of interactions in a controlled way, in the second, series B, we varied the degree of non-proportionality of hazards, in the third, series C, we varied a monotonic transform on the covariates, and in the fourth, series D, we studied the behavior under ties. In all those series of simulations, the covariates X were drawn from an $m-$dimensional standard normal distribution. The parameter μ, specifying the location (series A, C and D) or log-scale (series B) of the survival distribution, depended in all simulations on the covariates X via $\mu(X) = \alpha_q\left(X^\top MX - \mathbb{E}_X[X^\top MX]\right) + \alpha_\ell X^\top b$. The matrix $M \in \mathrm{Sym}_m(\mathbb{R})$ and the vector $b \in \mathbb{R}^m$ were randomly chosen at each run of the simulation with b_i, $M_{ij} \sim \mathcal{N}(0,1)$ for $i = 1,\ldots,m, j \neq i$, and $M_{ii} = 0$. Thus, $X^\top MX$ specifies the dependency of μ on interaction terms of the covariates, while $X^\top b$ specifies the linear dependency. Noting that $\mathrm{Var}_X(\mu_X) = \alpha_q^2 2\mathrm{Tr}(M^2) + \alpha_\ell b^\top b$, we chose the parameters α_q and α_ℓ in a controlled way, such that (i) $\mathrm{Var}_X(\mu(X)) = 1$ and (ii) a specified percentage of the variance of μ stemmed from the interaction terms $X^\top MX$.

Simulation series A generated survival data by assuming a fixed shape of the survival curve, with the survival time Y following a lognormal distribution with $\log(Y) \sim \mathcal{N}(\mu, 1)$. The parameters α_q, α_ℓ were varied, such that either 0, 25, 50, 75, or 100% of the variance of μ stemmed from the interaction terms.

Simulation series B examined the performance of the models when one covariate breaks the proportional hazards assumption by influencing the shape of the survival curve. In this simulation, the event times were drawn from a Weibull distribution, i.e., $Y \sim \mathrm{Wei}(\mathrm{Scale}, \mathrm{Shape})$. To break the proportional hazards assumption, an additional binary covariate X_{m+1} with $\mathbb{P}(X_{m+1} = 1) = \mathbb{P}(X_{m+1} = 0) = 0.5$ specifying the shape of the curve was introduced. Specifically, we have

$$Y \sim \begin{cases} \mathrm{Wei}(\exp(\mu(X)), \theta_0), \text{for } X_{m+1} = 0, \\ \mathrm{Wei}(\exp(\mu(X) + \alpha_1), \theta_1), \text{for } X_{m+1} = 1. \end{cases}$$

The parameters α_1, θ_0, and θ_1 were varied such that (i) the hazard ratio between the survival curves $\mathrm{Wei}(1, \theta_1)$ and $\mathrm{Wei}(\exp(\alpha_1), \theta_1)$ was approximately 1 and (ii) the integrated squared difference between the two curves varied from 0 to 0.3 in approximately equidistant steps. In simulation series B, α_q and α_ℓ were chosen, such that 50% of the variance of μ was explained by the interaction terms. The exact values of α_1, θ_0, and θ_1 and a plot of the survival curves for $\mu = 0$ can be found in Appendix C.

For simulation series C and D, we used the same survival distribution as in A and again fixed α_q, α_ℓ, such that 50% of the variance of μ came from the interaction terms. However, for the simulations of series C, we did not use the realizations of X for fitting the prediction models but their values under the pointwise transformation $x \mapsto \mathrm{sgn}(x)|x|^p$ with $p \in \{0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$. For series D of simulations, we discretized the observed times of the training set by mapping them to the median of those quantiles of the event times that contained them. The different observed times in the training data varied from 15 over 30 and 60 up to 120.

For all four series of simulations, censoring times were generated from a Weibull distribution with a shape parameter of 1.2. The scale of this distribution was varied, such that either 0%, approximately 25%, or 50% of observations were censored. To reduce variance and to allow for a better understanding of performance loss due to the censoring of training data, the predictions of the models were evaluated against the uncensored test data. In all simulations, we set $m = 6$ and drew, for each run, 400 samples for the training

set and 4000 samples for the test set. We also conducted a sensitivity analysis to see if our results depended on the size of the training data and conducted the same series of analyses with a training set of 200 samples.

4. Results

4.1. Results of Simulation

The results of simulation A are shown in Figure 1a. We see that the concordance and the R^2 decrease for all models as the level of interaction increases. The decrease in performance is almost parallel for the random survival forest and the kernel prediction, with the kernel prediction being better. If there is no interaction at all, the Cox model performs slightly superior than the other models as its assumptions are approximately met. However, the performance declines rapidly with increasing levels of interactions, eventually becoming uninformative. While the performance of the Cox model could be improved by adding interaction terms, the simulation highlights an advantage of machine learning models as they do not require the specification of a certain functional form between covariates and outcomes. Moreover, as the censoring rate increases, the relative performance of the kernel prediction compared to the random survival forest increases slightly.

The results of simulation B are displayed in Figure 1b. For this simulation, the result strongly depends on the level of censoring, as the censoring influences the hazard ratio as well as the log-rank score [32]. Regarding concordance, we see that, except for the Cox proportional hazards model, the performance increases with the non-proportionality of hazards regardless of censoring. For the Cox model, the concordance decreases drastically when there is no censoring, but remains relatively constant when the censoring is 25 or 50%. The R^2 decreases for all models when censoring is absent or low. However, the kernel prediction shows by far the lowest decrease. When there is 50% censoring, only the Cox model shows a decrease in performance, while the performance of the other methods remains relatively constant.

The results of simulation C can be found in Figure 1c. The performance of the random forest is the same for all transformations, as for the algorithm only the ordering of the covariates is important. The kernel prediction, on the other hand, shows a very high dependence on the nonlinear scaling in terms of both, concordance and R^2. Depending on the transformation, its performance varies from being the best algorithm to a performance similar to the Cox model, below the random survival forest. The performance of the Cox model depends on the nonlinear scaling as well, although it is less sensitive than the kernel prediction. Also in this series of simulations, the relative performance of the kernel prediction compared to the random survival forest seems to increase with increasing censoring rate.

Lastly, the results of simulation D can be found in Figure 1d. Performance, in terms of concordance index and R^2, remains quite stable for kernel prediction and random survival forest. For the Cox regression model, we see that while the concordance index remains stable, the R^2 decreases when we only have a few different observed times, i.e., when the number of ties is high. The decrease is more pronounced when censoring is absent or small.

The results of sensitivity analysis for a smaller training set of size $n = 200$ showed the same patterns and can be found in Appendix D.

Overall we see similar trends in the performances of kernel prediction and random survival forest when the level of interaction increases. The kernel prediction handles non-proportional hazards better, while it is, unlike the random survival forest, very sensitive to nonlinear scaling of the covariates. If the scaling is suitable though, it consistently outperforms the random survival forest. Of the examined algorithms, the Cox model depends on most assumptions and its performance decreases fast in all considered scenarios when those assumptions are not met.

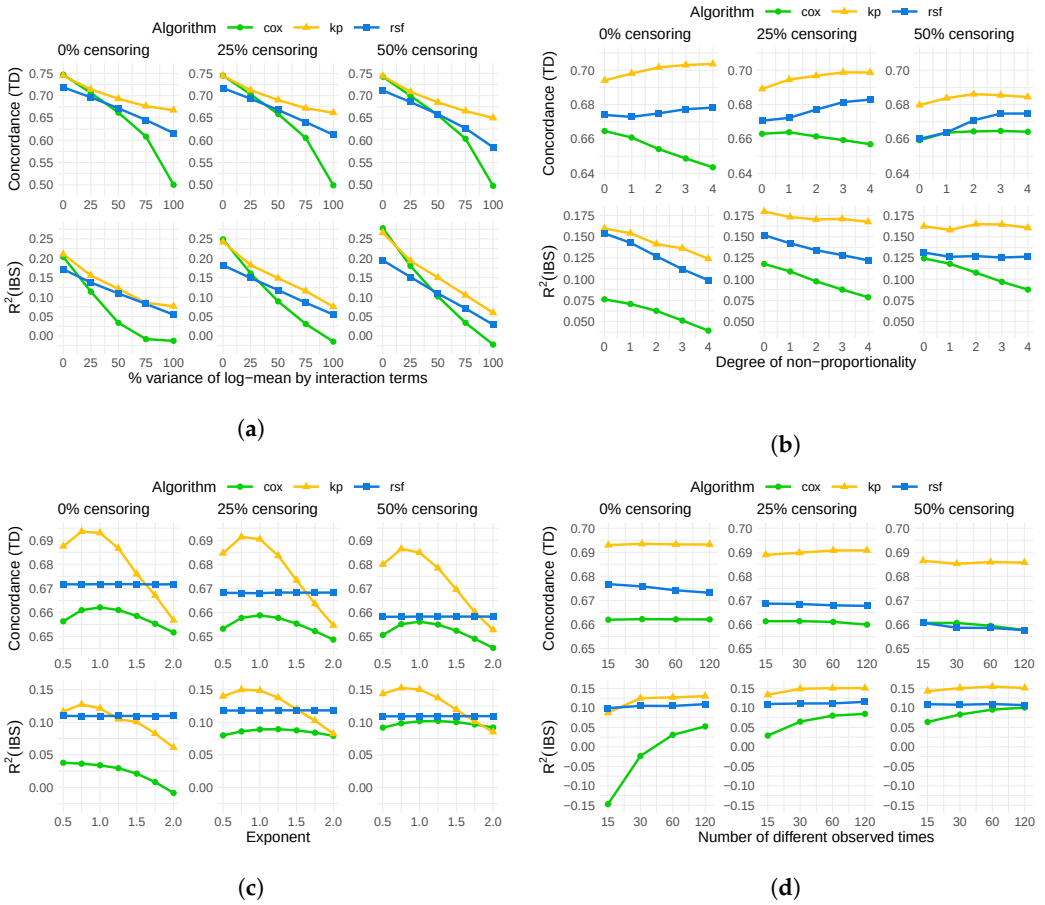
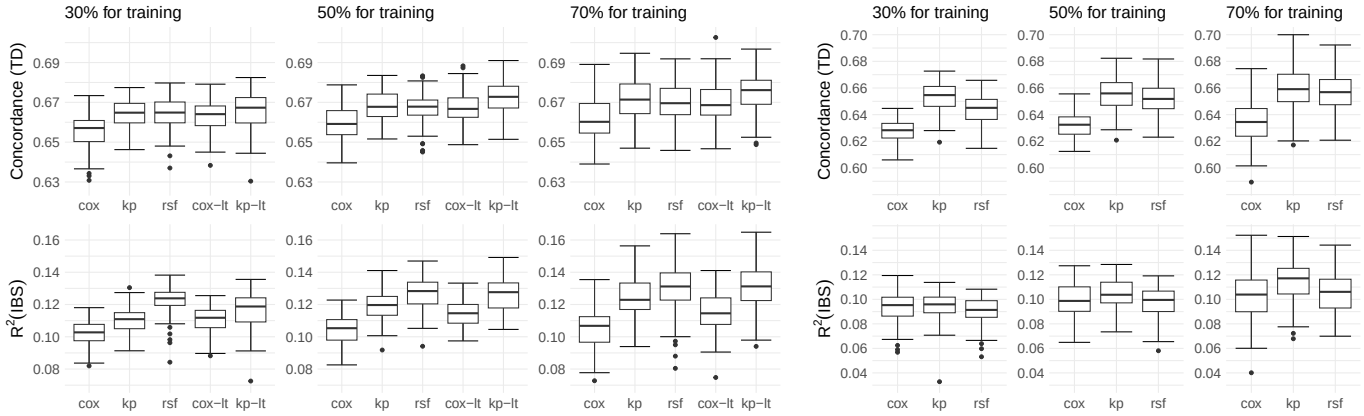


Figure 1. Mean concordance index and mean R^2 with respect to the integrated Brier score (IBS) of Cox proportional hazards model (Cox), kernel prediction (kp), and random survival forest (rsf) over 100 runs under different settings with a training size of $n = 400$. (**a**) Simulation A: By degrees of interaction. (**b**) Simulation B: By non-proportionality of hazards. (**c**) Simulation C: By exponent of covariate transformation. (**d**) Simulation D: By different numbers of observed times.

4.2. Results on Real-World Data

The box plots of the performance on real-world data are shown in Figure 2. In the case of the Rotterdam/GBSG dataset, kernel prediction and random survival forest show similar performance in terms of concordance across all training and test sizes. However, when it comes to R^2, random survival forest outperforms the other methods, while the Cox regression model performs the worst. It is worth noting that as the training size increases, the performance of kernel prediction catches up with the random survival forest. One reason for the comparable poor performance of the Cox regression model and the kernel prediction is the inappropriate scaling of the numerical variables indicating the concentration of progesterone and estrogen. For the rescaled data, we see that just like for simulated data and the METABRIC dataset, kernel prediction outperforms the random survival forest in terms of concordance. The R^2 is slightly better for the random survival forest for smaller training sizes, but almost identical when we use 70% of the data for training. Also, the Cox regression model performs substantially better after rescaling of the covariates. An example of how similar patients can be used to explain individual predictions on the Rotterdam/GBSG dataset is in Appendix F.



(**a**) (**b**)

Figure 2. Box plot of concordance index and R^2 with respect to the integrated Brier score (IBS) from 100 different train/test splits for the Cox proportional hazards model (Cox), kernel prediction (kp), and random survival forest (rsf). For Rotterdam/GBSG additionally with log-transform (lt) of two covariates. (**a**) Performance on Rotterdam/GBSG. (**b**) Performance on METABRIC.

The results for the METABRIC dataset are shown in Figure 2b. We see that when using a small training set, kernel prediction demonstrates superior performance in terms of concordance while the R^2 is similar for all methods. As the training size increases, the performances of kernel prediction and random survival forest become similar in terms of concordance but kernel prediction becomes superior in terms of R^2. We assume that part of the superior performance for the METABRIC dataset is due to the better ability of the kernel prediction to handle non-proportional hazards. Figure 3 illustrates this assumption by presenting the Kaplan–Meier estimates for two distinct subgroups within the dataset: the patients who received chemotherapy (n = 396) and those who did not (n = 1508). The Kaplan–Meier curves intersect with each other, indicating a violation of the proportional hazards assumption. Additionally, we depict the mean survival probability for each subgroup estimated by kernel prediction and random survival forest, which ideally should coincide with the Kaplan–Meier curves. The estimate of the Cox proportional hazard model, where it is not possible for the curves of the different treatment groups to intersect, can be found in Appendix E. It can be seen that both algorithms, the kernel prediction and the random survival forest, estimate the average survival probabilities of the patients without chemotherapy, the larger subgroup, well. For the treated patients, however, we see that the random survival forest produces highly biased survival estimates, which strongly overestimate the survival probabilities throughout the entire 30-year follow-up period. The kernel prediction, on the other hand, is quite accurate up to year 15, particularly when there is still a considerable number of patients at risk. For the remaining years, from year 15 to 30, it looks like a smoothed version of the Kaplan–Meier estimate, as it replaces the step-wise drops with a continuous decrease.

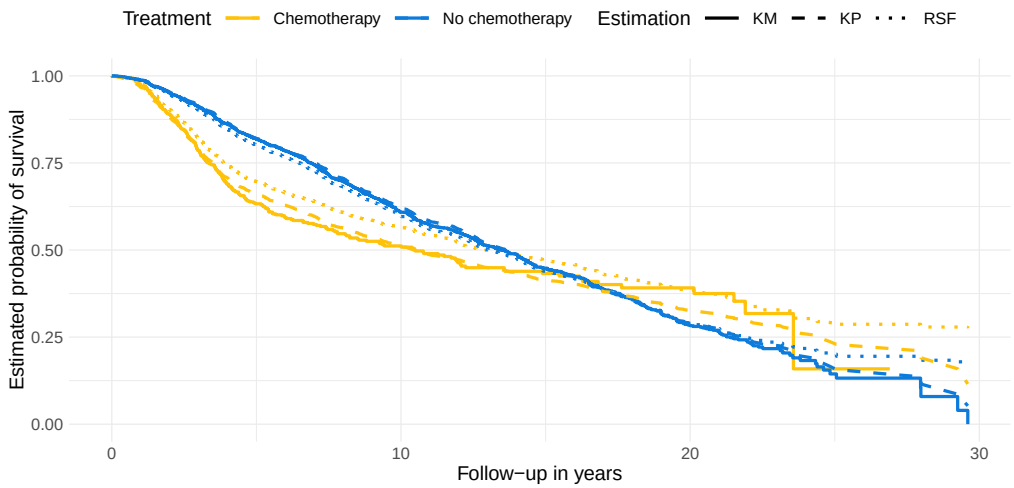


Figure 3. Survival probabilities of patients from the METABRIC dataset stratified by treatment with chemotherapy estimated with the Kaplan–Meier method (KM), kernel prediction (KP), and random survival forest (RSF).

5. Discussion

In this paper, we introduced a kernel learning approach for survival analysis that is easy to implement and provides direct interpretability of individual predictions through nearest neighbors. Our method proved to be competitive with other easily applicable techniques, such as random survival forest, on simulated and real-world data. However, there are still some limitations, open problems, and avenues for future research.

One drawback of our proposed kernel prediction compared to the random survival forest is that the latter can deal with missing values more effectively, e.g., via surrogate splits. Another limitation is the sensitivity of the proposed algorithm to the nonlinear scaling of covariates. It would be worth exploring the possibilities of learning a suitable scaling in a data-driven fashion to improve the predictive performance of the kernel prediction. When scaling of covariates is appropriate, kernel prediction performs better than random survival forest in our simulations and on the used real-world data.

Although the paper already demonstrates competitiveness in terms of predictive accuracy when considering only a single hyperparameter, namely the regularization, there are further hyperparameters that can be considered, such as the number and position of the time intervals. While for the experiments in this article, we did not vary the number of intervals, and used the quantiles to position them; this choice was made ad hoc and might not be ideal, in particular when event times are clustered. An investigation on how to best pick the intervals would also greatly benefit other methods using a piece-wise approach, such as neural networks [9]. In addition, the impact of the choice of the kernel function and the norm used in it can be explored. We solely focused on the Gaussian radial basis function; however, many other different possibilities exist. Apart from choosing one kernel, one might also consider employing algorithms for multiple kernel learning to best combine different kernels [33].

As already mentioned, the computational complexity of the presented algorithm scales quadratically in the number of observations, which limits the application to small- to medium-sized datasets. One ad hoc solution is to only use a subset of neighbors, e.g., the closest 1000 for each neighbor, for the calculation and optimization of the kernel, as suggested in [14]. Another possibility is to leverage kernel approximation methods, such as random features [34] or sparse grids for kernel learning [35]. These approaches could help reduce computational complexity while maintaining acceptable predictive performance.

Looking beyond the specific application of survival prediction, our kernel learning approach has a broader use. By deriving a meaningful metric between patients based on the learned kernel, our method enables clinicians to query for similar patients or facilitate subgroup discovery through metric-based clustering. This broader context opens up new avenues for utilizing our method in personalized medicine and healthcare applications.

Author Contributions: Conceptualization, W.G. and A.S.; methodology, W.G.; software, W.G.; validation, W.G.; formal analysis, W.G.; investigation, W.G.; data curation, W.G.; writing—original draft preparation, W.G.; writing—review and editing, W.G., B.K., C.J., E.-M.H. and A.S.; visualization, W.G.; supervision, B.K. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by a PhD grant from the DFG Research Training Group 2535 Knowledge- and data-based personalization of medicine at the point of care (WisPerMed), University of Duisburg-Essen, Germany. We acknowledge support by the Open Access Publication Fund of the University of Duisburg-Essen.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/jaredleekatzman/DeepSurv/tree/master/experiments/data. The python code to generate the simulated data is available upon request.

Acknowledgments: The authors thank Louise Bloch for the very helpful feedback on the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Given the weighted outcomes $(w_1, t_1, \delta_1), \ldots, (w_n, t_n, \delta_n)$, the log-likelihood of a single observation (w_i, t_i, δ_i), assuming a piece-wise constant hazard rate $\lambda(t) = \sum_{l=1}^{k} \lambda_l \mathbb{1}(t \in (\tau_{l-1}, \tau_l])$ is, using (2), given by

$$\begin{aligned}\log f(T = t_i, \Delta = \delta_i | \lambda_{l=1,\ldots,k}) &= \log\left(\lambda_{j(t_i)}^{\delta_i} \exp\left(-\int_0^{t_i} \sum_{l=1}^{k} \lambda_l \mathbb{1}(t' \in (\tau_{l-1}, \tau_l]) dt'\right)\right) + \text{Const.} \\ &= \delta_i \log \lambda_{j(t_i)} - \sum_{l=1}^{k} \lambda_l r_l(t_i) + \text{Const.} \\ &= \sum_{l=1}^{k} (\delta_{il} \log \lambda_l - \lambda_l r_l(t_i)) + \text{Const.},\end{aligned}$$

where the constant only depends on the censoring distribution. Hence, for the full-weighted log-likelihood, we obtain

$$\log f((w_i, T = t_i, \Delta = \delta_i)_{i=1,\ldots,n} | \lambda_{l=1,\ldots,k}) = \sum_{i=1}^{n} w_i \sum_{l=1}^{k} (\delta_{il} \log \lambda_l - \lambda_l r_l(t_i)) + \text{Const.}$$

For $j = 1, \ldots, l$, the maximum likelihood estimate satisfies

$$\begin{aligned}0 &= \frac{\partial}{\partial \lambda_j} \log f((w_i, T = t_i, \Delta = \delta_i)_{i=1,\ldots,n} | \lambda_{l=1,\ldots,k}) \\ &= \sum_{i=1}^{n} w_i \sum_{l=1}^{k} \left(\delta_{il} \frac{1}{\lambda_l} \mathbb{1}(l = j) - \mathbb{1}(l = j) r_l(t_i)\right) \\ &= \sum_{i=1}^{n} w_i \left(\delta_{ij} \frac{1}{\lambda_j} - r_j(t_i)\right) \\ \implies \lambda_j &= \frac{\sum_{i=1}^{n} w_i \delta_{ij}}{\sum_{i=1}^{n} w_i r_j(t_i)}.\end{aligned}$$

Appendix B

Table A1. Hyperparameters considered in cross-validation for random survival forest. The number of trees was set to 500.

Data	Number of Covariates Considered for Splits	Minimum Samples per Leaf
Simulation A, C	2, 4, 6	5, 10, 20, 40
Simulation B	2, 4, 7	5, 10, 20, 40
METABRIC	3, 4, 9	5, 10, 20, 40
Rotterdam/GBSG	2, 3, 7	5, 10, 20, 40

For the kernel prediction, we first considered a preliminary $\tilde{\eta}$ via cross-validation from $\{10^{-2}, 10^{-3}, 10^{-4}\}$. The final regularization parameter η was then selected via cross-validation from $\{\eta 2^{-3}, \eta 2^{-2}, \ldots, \eta 2^{3}\}$. We considered fewer hyperparameter combinations for the kernel prediction than for the random survival forest.

Appendix C

Table A2. Parameters of simulation B.

Degree of Non-Proportionality	θ_0	θ_1	α_1
0	1.22	1.22	0
1	0.79	1.90	0.24
2	0.64	2.35	0.33
3	0.57	2.70	0.39
4	0.50	3.00	0.43

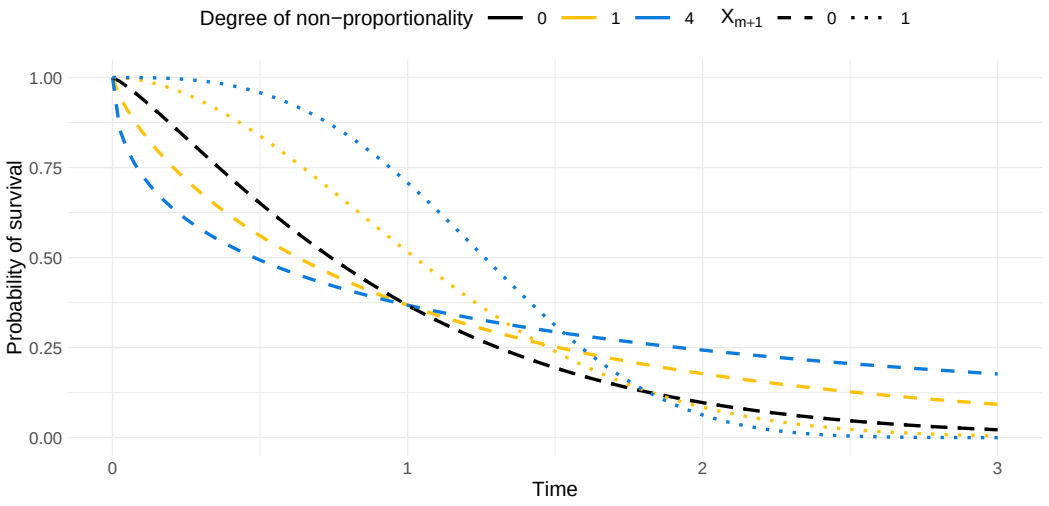


Figure A1. Survival probabilities in simulation B for $\mu = 0$ depending on X_{m+1} for the selected degrees of non-proportionality. If the degree of non-proportionality is 0, X_{m+1} does not have an effect on the shape of the curve.

Appendix D

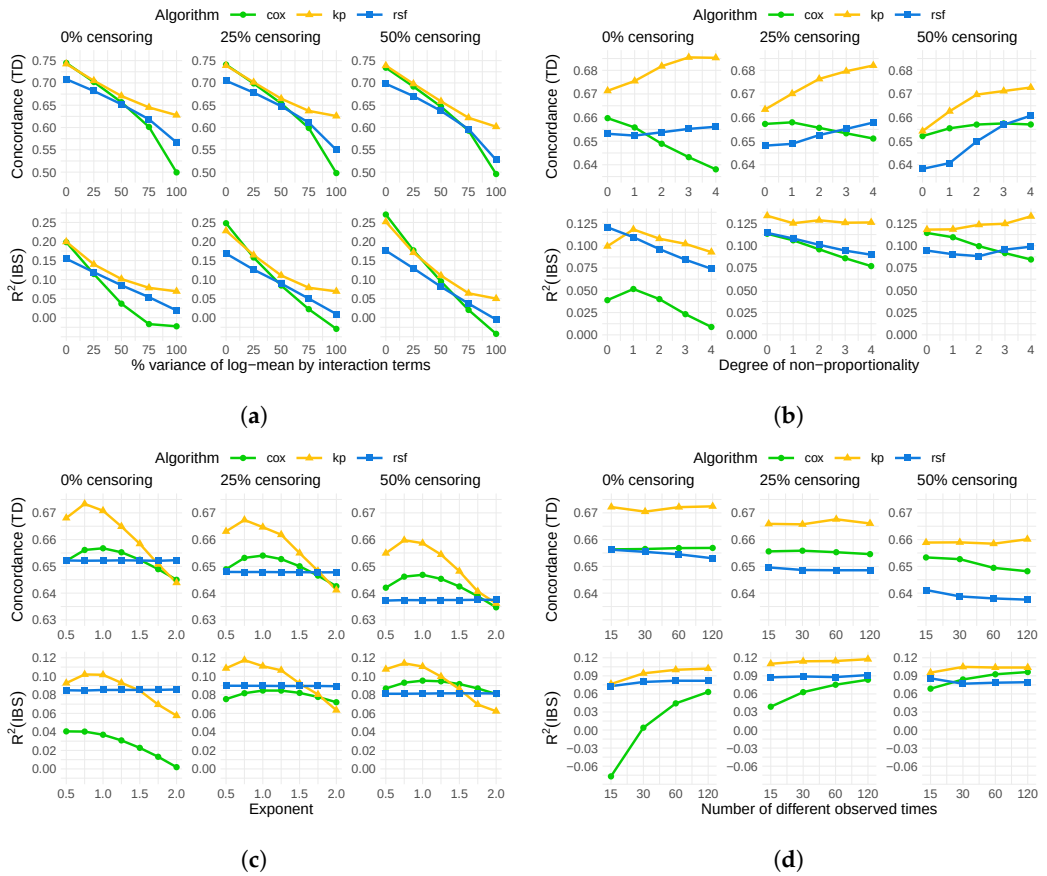


Figure A2. Mean concordance index and mean R^2 with respect to the integrated Brier score (IBS) of the Cox proportional hazards model (Cox), kernel prediction (kp), and random survival forest (rsf) under different settings and with a training size of $n = 200$. (**a**) Simulation A: By degrees of interaction. (**b**) Simulation B: By non-proportionality of hazards. (**c**) Simulation C: By exponent of covariate transformation. (**d**) Simulation D: By different numbers of observed times.

Appendix E

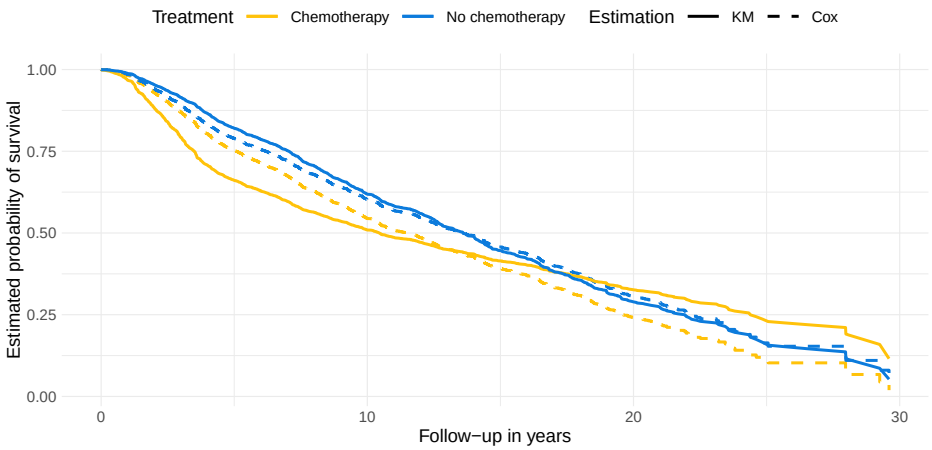
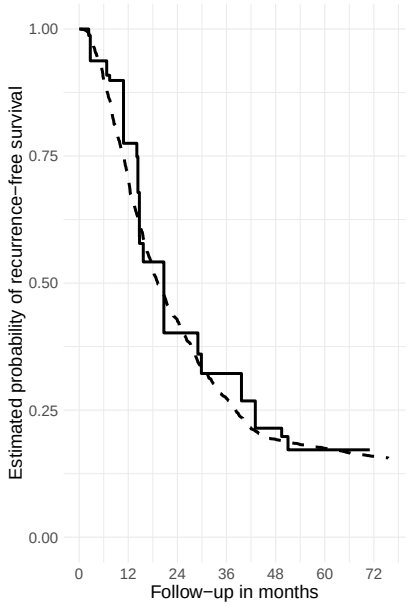


Figure A3. Survival probabilities of patients from the METABRIC dataset, stratified by treatment with chemotherapy, estimated with the Kaplan–Meier method (KM) and by the Cox proportional hazards model (Cox).

Appendix F



Hormonal Therapy	Tumor size, mm	Postmeno pausal	Age (years)	No. positive lymph nodes	log progesterone	log estrogene	Follow−Up (Months)	Recurrence	Weight (%)
0	>20 to <=50	0	38	24	0.00	0.00	NA	NA	NA
0	>20 to <=50	0	36	17	0.00	0.00	21	1	11.9
0	>20 to <=50	0	46	19	0.00	1.61	11	1	10.5
0	>20 to <=50	0	43	20	0.00	3.64	84	0	8.2
0	>50	0	32	24	0.00	2.64	15	1	6.3
0	>20 to <=50	0	37	18	3.09	0.00	14	1	5.7
0	>20 to <=50	0	32	15	0.00	2.08	3	1	4.0
0	>50	0	43	20	1.10	3.26	40	1	3.7
0	>50	0	50	23	0.00	1.95	43	1	3.7
0	>20 to <=50	0	38	13	0.00	0.00	16	1	2.9
0	>20 to <=50	0	47	20	3.43	2.20	29	1	2.8
0	>20 to <=50	0	38	19	0.00	5.71	30	1	2.6
0	>20 to <=50	0	46	16	2.56	2.20	21	0	2.3
0	>20 to <=50	0	42	16	3.33	1.61	7	1	2.3
0	>50	0	50	17	0.00	0.00	14	1	2.1
0	>50	0	33	14	0.00	0.00	51	1	1.7
0	>50	0	42	15	2.08	1.79	18	0	1.4
0	>50	0	35	13	0.00	0.00	59	0	1.2
0	>50	0	42	25	3.76	3.83	49	1	1.0
0	>50	0	32	27	4.09	2.56	2	1	1.0
0	>20 to <=50	0	50	23	3.81	3.81	15	1	1.0
0	>50	0	28	15	2.08	0.00	15	1	1.0
0	>50	0	44	19	3.37	3.47	1	0	1.0
1	>20 to <=50	0	44	17	1.10	1.39	7	1	0.8
0	>20 to <=50	0	43	23	4.70	4.23	84	0	0.8

Figure A4. Example of an explanation of a kernel prediction via nearest neighbors on the Rotterdam/GBSG dataset. The table shows the clinical characteristics of the patient for whom we predict recurrence-free survival (first-line, blue background) and of the most important patients used for prediction, ranked by the assigned weights (right column). Attributes that cause dissimilarity according to the learned metric are marked in red. The left side shows the weighted Kaplan–Meier curve (solid line) and the smooth kernel prediction (dashed line).

References

1. Smith, M.J.; Phillips, R.V.; Luque-Fernandez, M.A.; Maringe, C. Application of targeted maximum likelihood estimation in public health and epidemiological studies: A systematic review. *Ann. Epidemiol.* **2023**, *86*, 34–48.e28. [CrossRef]
2. Wager, S.; Athey, S. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *J. Am. Stat. Assoc.* **2018**, *113*, 1228–1242. [CrossRef]
3. Hu, L.; Ji, J.; Li, F. Estimating heterogeneous survival treatment effect in observational data using machine learning. *Stat. Med.* **2021**, *40*, 4691–4713. [CrossRef]
4. Wang, P.; Li, Y.; Reddy, C.K. Machine Learning for Survival Analysis: A Survey. *ACM Comput. Surv.* **2019**, *51*, 1–36. [CrossRef]
5. Pölsterl, S.; Navab, N.; Katouzian, A. Fast Training of Support Vector Machines for Survival Analysis. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*; Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 243–259. [CrossRef]
6. Segal, M.R. Regression Trees for Censored Data. *Biometrics* **1988**, *44*, 35–47. [CrossRef]
7. Ishwaran, H.; Kogalur, U.B.; Blackstone, E.H.; Lauer, M.S. Random survival forests. *Ann. Appl. Stat.* **2008**, *2*, 841–860. [CrossRef]
8. Katzman, J.L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; Kluger, Y. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med. Res. Methodol.* **2018**, *18*, 24. [CrossRef]
9. Kvamme, H.; Borgan, Ø. Continuous and discrete-time survival prediction with neural networks. *Lifetime Data Anal.* **2021**, *27*, 710–736. [CrossRef]
10. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: Review, opportunities and challenges. *Briefings Bioinform.* **2017**, *19*, 1236–1246. [CrossRef]
11. Mitchell, T.M. *Machine Learning*, 1st ed.; Series in Computer Science; McGraw-Hill: Chicago, IL, USA, 1997.
12. Weinberger, K.Q.; Saul, L.K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
13. Li, D.; Tian, Y. Survey and experimental study on metric learning methods. *Neural Netw.* **2018**, *105*, 447–462. [CrossRef]
14. Weinberger, K.Q.; Tesauro, G. Metric Learning for Kernel Regression. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, San Juan, PR, USA, 21–24 March 2007; Proceedings of Machine Learning Research; Meila, M., Shen, X., Eds.; PMLR: San Juan, PR, USA, 2007; Volume 2, pp. 612–619.

15. Chen, G.H. Nearest Neighbor and Kernel Survival Analysis: Nonasymptotic Error Bounds and Strong Consistency Rates. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Proceedings of Machine Learning Research (PMLR); Chaudhuri, K., Salakhutdinov, R., Eds.; PMLR: Long Beach, CA, USA, 2019; Volume 97, pp. 1001–1010.
16. Lowsky, D.; Ding, Y.; Lee, D.; McCulloch, C.; Ross, L.; Thistlethwaite, J.; Zenios, S. A K-nearest neighbors survival probability prediction method. *Stat. Med.* **2013**, *32*, 2062–2069. [CrossRef]
17. Chen, G.H. Deep Kernel Survival Analysis and Subject-Specific Survival Time Prediction Intervals. In Proceedings of the 5th Machine Learning for Healthcare Conference, Virtual, 7–8 August 2020; Proceedings of Machine Learning Research (PMLR); Doshi-Velez, F., Fackler, J., Jung, K., Kale, D., Ranganath, R., Wallace, B., Wiens, J., Eds.; PMLR: Virtual, 2020; Volume 126, pp. 537–565.
18. Klein, J.P.; Moeschberger, M.L. *Survival Analysis*, 2nd ed.; Statistics for Biology and Health; Springer: New York, NY, USA, 2003.
19. Pölsterl, S. scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn. *J. Mach. Learn. Res.* **2020**, *21*, 8747–8752.
20. Rasmussen, C.; Williams, C. Covariance functions. In *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2005. [CrossRef]
21. Zou, H.; Hastie, T. Regularization and Variable Selection Via the Elastic Net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]
22. LeBlanc, M.; Crowley, J. Relative Risk Trees for Censored Survival Data. *Biometrics* **1992**, *48*, 411–425. [CrossRef]
23. Curtis, C.; Shah, S.P.; Chin, S.F.; Turashvili, G.; Rueda, O.M.; Dunning, M.J.; Speed, D.; Lynch, A.G.; Samarajiwa, S.; Yuan, Y.; et al. The genomic and transcriptomic architecture of 2000 breast tumours reveals novel subgroups. *Nature* **2012**, *486*, 346–352. [CrossRef] [PubMed]
24. Foekens, J.A.; Peters, H.A.; Look, M.P.; Portengen, H.; Schmitt, M.; Kramer, M.D.; Brünner, N.; Jänicke, F.; Meijer-van Gelder, M.E.; Henzen-Logmans, S.C.; et al. The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients. *Cancer Res.* **2000**, *60*, 636–643.
25. Schumacher, M.; Bastert, G.; Bojar, H.; Hübner, K.; Olschewski, M.; Sauerbrei, W.; Schmoor, C.; Beyerle, C.; Neumann, R.L.; Rauschecker, H.F. Randomized 2 $\times$ 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group. *J. Clin. Oncol.* **1994**, *12*, 2086–2093. [CrossRef] [PubMed]
26. Antolini, L.; Boracchi, P.; Biganzoli, E. A time-dependent discrimination index for survival data. *Stat. Med.* **2005**, *24*, 3927–3944. [CrossRef] [PubMed]
27. Graf, E.; Schmoor, C.; Sauerbrei, W.; Schumacher, M. Assessment and comparison of prognostic classification schemes for survival data. *Stat. Med.* **1999**, *18*, 2529–2545. .:17/18<2529::AID-SIM274>3.0.CO;2-5. [CrossRef]
28. Austin, P.C.; Harrell, F.E.; van Klaveren, D. Graphical calibration curves and the integrated calibration index (ICI) for survival models. *Stat. Med.* **2020**, *39*, 2714–2742. [CrossRef]
29. Davidson-Pilon, C. Lifelines v0.27.7, Survival Analysis in Python. 2023. Available online: https://zenodo.org/record/7883870 (accessed on 17 May 2023).
30. Liu, D.C.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.* **1989**, *45*, 503–528. [CrossRef]
31. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; pp. 8024–8035.
32. Schemper, M.; Wakounig, S.; Heinze, G. The estimation of average hazard ratios by weighted Cox regression. *Stat. Med.* **2009**, *28*, 2473–2489. [CrossRef] [PubMed]
33. Gönen, M.; Alpaydin, E. Multiple Kernel Learning Algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2211–2268.
34. Liu, F.; Huang, X.; Chen, Y.; Suykens, J.A.K. Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7128–7148. [CrossRef] [PubMed]
35. Yadav, M.; Sheldon, D.R.; Musco, C. Kernel Interpolation with Sparse Grids. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: New York, NY, USA, 2022; Volume 35, pp. 22883–22894.