DeepPAMM: Deep Piecewise Exponential Additive Mixed Models for Complex Hazard Structures in Survival Analysis

Philipp Kopper[(✉)], Simon Wiegrebe, Bernd Bischl, Andreas Bender, and David Rügamer

Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany
`philipp.kopper@stat.uni-muenchen.de`

Abstract. Survival analysis (SA) is an active field of research that is concerned with time-to-event outcomes and is prevalent in many domains, particularly biomedical applications. Despite its importance, SA remains challenging due to small-scale data sets and complex outcome distributions, concealed by truncation and censoring processes. The piecewise exponential additive mixed model (PAMM) is a model class addressing many of these challenges, yet PAMMs are not applicable in high-dimensional feature settings or in the case of unstructured or multimodal data. We unify existing approaches by proposing DeepPAMM, a versatile deep learning framework that is well-founded from a statistical point of view, yet with enough flexibility for modeling complex hazard structures. We illustrate that DeepPAMM is competitive with other machine learning approaches with respect to predictive performance while maintaining interpretability through benchmark experiments and an extended case study.

Keywords: Deep learning · Time-to-event data · Survival analysis · Interpretability · Random effects · Mixed models

1 Introduction

Deep learning (DL) excels in many different areas of application through flexible and versatile network architectures. This has also been demonstrated in survival analysis (SA) [27,33], where it is often not straightforward to apply off-the-shelf machine learning models. Apart from medical applications such as the prediction of time-to-death or the time to disease onset, time-to-event models are also applied in a variety of other domains. Among other fields, SA is successfully employed for predictive maintenance, credit scoring, and customer churn prediction. In practice, time-to-event outcomes are not necessarily observed fully but might be censored, truncated or stem from a competing risks, or a multi-state process. While these aspects relate to the nature of the observation of event times, SA is also challenging due to the typically small amount of observations as well as complex feature effects and dependencies between observations. Medical survival data for instance potentially includes patient data of certain cohorts

(such as patients from different hospitals with varying levels of patient care), longitudinal data with recurrent events or includes time-varying features such as a patient's vital status. Additionally, data can be multimodal (e.g., tabular patient information paired with medical images).

Our Contribution. In this paper, we introduce a novel method called *DeepPAMM* for continuous time-to-event data that enables the hazard-based learning of survival models via neural networks and supports 1) many common survival tasks, including right-censored, left-truncated, competing risks, or multi-state data as well as recurrent events; 2) the estimation of inherently interpretable feature effects; 3) learning from multiple data sources (e.g., tabular and imaging data); 4) time-varying effects and time-varying features; 5) the modeling of repeated or correlated data using random effects.

2   Related Literature

Various models have been brought forward in SA. We will distinguish between models developed from a statistical point of view (Sect. 2.1), machine learning approaches (Sect. 2.2) and recently proposed deep learning frameworks (Sect. 2.3).

2.1   Piecewise Exponential Additive Models and Cox Proportional Hazard Models

The Cox proportional hazard model (CPH) [11] is the most widely used survival model. Under certain assumptions [42] the Cox PH model is equivalent to the piecewise exponential model (PEM). The original formulation of the PEM, a parametric, linear effects, PH model, goes back to [14]. The general idea is to partition the follow-up time into J intervals and to assume piecewise constant hazards in each interval. The originally proposed PEM requires a careful choice of the number and placement of interval cutpoints. The piecewise exponential additive model (PAM) [2,3,9] is an extension of the PEM. PAMs estimate the baseline hazard and other time-dependent effects as smooth functions over time via penalized splines. This leads to more plausible and robust hazard estimates and (indirectly) lower computational cost. PAMs can be further generalized to piecewise exponential additive mixed models (PAMMs) by adding frailty terms (random effects). While PEMs and PAMMs can deal with many types of survival data (see, e.g., [4,6]), they are limited w.r.t. the complexity of feature effects that they can estimate, especially in the case of high-dimensional features and interactions and cannot handle unstructured data.

2.2   Machine Learning Approaches

In recent years a large number of machine learning methods for SA have been put forward. Random forest (RF) based methods include the random survival forest (RSF) [20] and more recently the oblique random survival forests (ORSF) [21]. In contrast to conventional RFs [8], these adaptions make the models applicable to survival data by adjusting the splitting criterion. Next to trees and forests, several boosting methods

exist, such as XGBoost [10] or component-wise boosting for accelerated-failure time models [36] and non-parametric hazard boosting [28]. More recently and closest to our work, [4] have proposed a general machine learning approach for various survival tasks based on PEMs and demonstrated its application using the standard XGBoost implementation.

2.3 Deep Learning Approaches

Various deep learning approaches have been proposed for SA, with the first approaches dating back to the mid-1990s (see, e.g., [12]). More recent approaches include both discrete-time methods like DeepHit [27] or Nnet-survival [15] and continuous-time methods such as DeepSurv [22] or CoxTime [24]. DeepHit parametrizes the probability mass function by a neural network and specifically targets competing risks, but is only able to predict survival probabilities for a given set of discrete follow-up time points due to its time-discretization approach. Nnet-survival, by contrast, models discrete hazards and provides flexibility in terms of architecture choice, but it also relies on discretization of event times. DeepSurv is a Cox PH model with the linear predictor replaced by a deep feed-forward neural network. CoxTime further improves upon DeepSurv by allowing for time-varying effects, thereby overcoming the proportional hazards assumption. A deep Gaussian process to predict competing risks is proposed in [1]. While all previous methods focus on tabular data, a few multimodal networks such as [17, 23, 30, 40] have also been proposed as well as survival tasks combined with a generative appraoch [41]. The first combination of PEMs with a NN was proposed by [29]. [7] discussed the estimation of PEM by representing generalized linear models via feed-forward NNs, and [13] proposed the estimation of the shape of the hazard rate with NNs. [25] also discussed the parametrization of the PEM via NNs with application to tabular data. As for PEMs, the choice of cut-points in their framework is crucial for performance and computational complexity. Our framework eliminates this problem.

3 Piecewise Exponential Additive Models

Survival analysis aims to estimate the survival function $S(t) = P(T > t)$. Instead of directly estimating $S(t)$, the hazard function

$$h(t) := \lim_{\Delta t \to 0^+} \frac{P(t < T < t + \Delta t | T \geq t)}{\Delta t} \tag{1}$$

is modeled. The survival function can be derived from $h(t)$ via $S(t) = \exp(-\int_0^t h(s)\,ds)$. A hazard for time point $t \in \mathcal{T}$, conditional on a potentially time-varying feature vector $\boldsymbol{x}(t) \in \mathbb{R}^P$, can be defined by

$$h(t|\boldsymbol{x}(t), k) = \exp\left(\rho(\boldsymbol{x}(t), t, k)\right), k = 1, \ldots, K. \tag{2}$$

The function $\rho(\cdot)$ represents the effect of (time-dependent) features $\boldsymbol{x}(t)$ on the hazard and can itself be potentially time- and transition-specific. k indicates a transition, e.g., from status 0 to status k in competing risks or the transition between two states in the multi-state setting. In the following, we will set K to 1 for better readability and only address the single risk application if not stated otherwise. Further omitting the dependence on t, (2) reduces to the familiar PH form known from the Cox model.

3.1   Data Transformation

PEMs and PAMs approximate (2) via piecewise constant hazards, which requires a specific data transformation, creating one row in the data set for each interval a subject was at risk. Assume observations (subjects) $i = 1, \ldots, n$, for which the tuple $(t_i, \delta_i, \boldsymbol{x}_i)$ with event time t_i, event indicator $\delta_i \in \{0, 1\}$ (1=event, 0=censoring) and feature vector $\boldsymbol{x}_i$ is observed. PAMs partition the follow up into J intervals $(\kappa_{j-1}, \kappa_j]$, $j = 1, \ldots, J$. This implies a new status variable $\delta_{ij} = 1$ if $t_i \in (\kappa_{j-1}, \kappa_j] \wedge \delta_i = 1$, and 0 otherwise, indicating the status of subject i in interval j. Further, we create a variable t_{ij}, the time subject i was at risk in interval j, which will enter the analysis as an offset. Lastly, the variable t_j, (e.g., $t_j := \kappa_j$) is a representation of time in interval j and the feature based on which the model estimates the baseline hazard and time-varying effects. In order to transform the data to the piecewise exponential data format (PED), time-constant features $\boldsymbol{x}_i$ are repeated for each of J_i rows, where J_i, denotes the number of intervals in which subject i was at risk. This data augmentation step transforms a survival task into a standard Poisson regression task. Depending on the setting, e.g., right-censoring, recurrent events, left truncation, etc., the specifics of the data transformation vary, but the general principles remain the same. For more details we refer to [4,5,32].

3.2   Model Estimation

Given the transformed data, PAMs approximate (2) by $h(t|\boldsymbol{x}_i(t)) = \exp(\rho(\boldsymbol{x}_{ij}, t_j)) := h_{ij}, \forall t \in (\kappa_{j-1}, \kappa_j]$, where $\boldsymbol{x}_{ij}$ is the feature vector of subject i in interval j. Assuming $\delta_{ij} \sim \text{Poisson}(\mu_{ij} = h_{ij} t_{ij})$, the log-likelihood contribution of subject i is given by $\ell_i = \sum_{j=1}^{J_i} (\delta_{ij} \log(h_{ij}) - h_{ij} t_{ij})$, where

$$\log(h_{ij}) = \beta_0 + f_0(t_j) + \sum_{p=1}^{P} x_{ij,p} \beta_p + \sum_{l=1}^{L} f_l(x_{ij,l}),$$

with log-baseline hazard $\beta_0 + f_0(t_j)$, linear feature effects β_p of features $x_{ij,p} \subseteq \boldsymbol{x}_{ij}$ and univariate, non-linear feature effects $f_l(x_{ij,l})$ of features $x_{ij,l} \subseteq \boldsymbol{x}_{ij}$. Both f_0 and f_l are defined via a basis representation, i.e., $f_l(x_{ij,l}) = \sum_{m=1}^{M_l} \theta_{l,m} B_{l,m}(x_{ij,l})$ with basis functions $B_{\cdot,m}(\cdot)$ (such as B-spline bases) and basis coefficients $\theta_{\cdot,m}$. To avoid underfitting, the basis dimensions M_0 (for f_0) and M_l (for f_l) are set relatively high. To avoid overfitting, the basis coefficients are estimated by optimizing an objective function that penalizes differences between neighboring coefficients. Let $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_P)^\top$ and $\boldsymbol{\theta}_l = (\theta_{l,1}, \ldots, \theta_{l,M_l})^\top$, $l = 0, \ldots, L$. The objective function minimized to estimate PAMs is the penalized negative log-likelihood given by $-\log \mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_L) + \sum_{l=0}^{L} \psi_l \Psi(\boldsymbol{\theta}_l)$, where the first term is the standard negative logarithmic Poisson likelihood, comprised of likelihood contributions ℓ_i, and the second term $\Psi(\boldsymbol{\theta}_l)$ is a quadratic penalty with smoothing parameter $\psi_l \geq 0$ for the respective spline f_l. Larger ψ_l lead to smoother f_l estimates (see [6,43] for details).

4   Deep Piecewise Exponential Additive Mixed Models

DeepPAMMs extend PAM(M)s with hazard as defined in (2) by allowing for deep neural networks (NN) in the additive predictor. Instead of combining PAMMs with (deep)
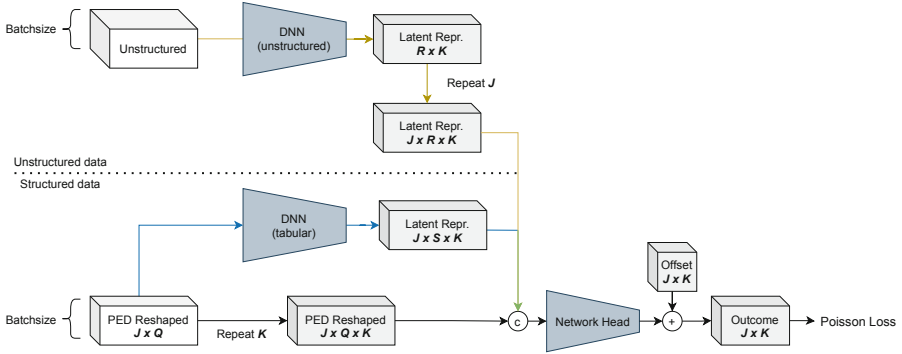
Fig. 1. Exemplary architecture of a DeepPAMM. A DeepPAMM comprises a PAMM (black path) and additionally either a deep neural network (DNN) for unstructured data (yellow path), a DNN for tabular features (blue path), or both. The unstructured data, e.g., images, are summarized to latent representations of size R, repeated J times, and concatenated (c) with the tabular data's latent representation of size S, as well as raw tabular data of size Q. Finally, the offset is added to the output and the network is trained using the Poisson loss for each of the K competing risks.

NNs in a two-stage approach, we embed the PAMM into the NN similar to [35] and train the network based on the (penalized) likelihood in an end-to-end manner.

Network Definition. While PAMMs restrict ρ to structured additive effects, the hypothesis space of DeepPAMMs can also be modeled using a deep NN. Assume that the NN $d(\cdot)$ is used to process a potentially time-varying (unstructured) data source $\boldsymbol{z}(t)$. We first assume a time-constant effect of $\boldsymbol{z}(t)$ and extend the PAMM's definition to

$$h(t|\boldsymbol{x}(t), \boldsymbol{z}(t)) = \exp\{\rho(\boldsymbol{x}(t), t) + d(\boldsymbol{z}(t))\}, \tag{3}$$

by adding one (or several) NN predictor(s) to the structured predictor.

The predictor $d(\boldsymbol{z}(t))$ can be modeled using an arbitrary NN. For example, a Deep-PAMM can combine a PAM with an additional NN to explore non-linearities and interactions in tabular features (beyond the ones specified in the structured part). Alternatively, a DeepPAMM can combine different data modalities, e.g., tabular patient data and corresponding medical scans using a convolutional NN for d. By (3), DeepPAMM learns a piecewise constant hazard rate

$$h_{ij} = \exp\Big\{\boldsymbol{B}_{ij}\boldsymbol{w} + \sum_{u=1}^{U} \zeta_{ij,u}\gamma_u\Big\}, \tag{4}$$

for each observation i and each discrete interval j, where $\boldsymbol{B}_{ij}$ subsumes all Q structured features (linear and basis evaluated features) with weights $\boldsymbol{w}$. $\zeta_{ij,1}, \ldots, \zeta_{ij,U}$ are $U = R + S$ latent representations learned from the deep network part that processes tabular data (into S latent features) and unstructured data (into R latent features). The network then combines these U latent representations to learn the effect $\gamma_1, \ldots, \gamma_U$ of each of these feature effects. Due to the additive structure in predictor (4), the structured terms with linear effects $\boldsymbol{w}$ preserve their interpretability inherited from PAMMs.

PED and Latent Representations. $d(\boldsymbol{z}(t))$ can be viewed as linear effects of U latent representations derived from inputs $\boldsymbol{z}(t)$. In (3) this representation is combined with the structured features in a last layer summing up the two predictors. If $\boldsymbol{z}$ is constant over time, i.e., $\boldsymbol{z}(t) \equiv \boldsymbol{z}$, it is not straightforward to combine these latent representations with the PED format properly. A naive approach would be to repeat the original data source $\boldsymbol{z}$ over all J intervals. This, however, leads to significant computational overhead and storage of redundant information. Instead, we resort to weight-sharing and reshaping within the network that allows learning a single latent representation per observation for all J intervals (cf. Fig. 1). First, the original tabular data is transformed to the PED format prior to the network training. Subsequently, the reshaped three-dimensional PED tensor batches with the same sampling dimension as the unstructured data source $\boldsymbol{z}$ are passed through the network. $\boldsymbol{z}$ itself is transformed into R latent representations and then repeated J times for each interval. This avoids repeating the original unstructured data source multiple times. Finally, we combine these representations with the original tabular data and the S non-linear representations of the structured data part into a joint set of features. While we here focus on time-constant unstructured data, our framework can be extended to allow for time-varying unstructured features by simply also supplying the time t to the deep NN d explicitly, i.e., extending $d(\boldsymbol{z}(t))$ in (3) to $d(\boldsymbol{z}(t), t)$.

Learning Non-proportional Hazards. PAMMs allow for non-proportional hazards via an interaction of features $\boldsymbol{x}$ with a feature that represents time in each of the J intervals. In practice, however, the accompanying computational complexity and manual definition of these interactions are often infeasible. In DeepPAMM, such interactions can be modeled using an appropriate multilayer NN architecture. In particular, interactions between features $\boldsymbol{z}(t)$ and the follow-up t can be expressed by $h(t|\boldsymbol{x}(t), \boldsymbol{z}(t)) = \exp\{\rho(\boldsymbol{x}(t), d(\boldsymbol{z}(t)), t)\}$, where ρ now also depends on the the specified NN to model a non-proportional hazard in $\boldsymbol{z}(t)$. As the PH assumption is a helpful inductive bias for applications with small sample sizes, we recommend this extension for larger data sets or in applications where the PH assumption is clearly violated.

Learning Competing Risks Hazards. When modeling competing risks data with K different risks that determine the time-to-event, one is interested in retrieving the cumulative incidence functions of each risk (CIFs). Our architecture allows for a holistic way of modeling the hazard of subject i in interval j and cause k in a joint NN: $h_{ijk} = \exp\{\boldsymbol{B}_{ijk}\boldsymbol{w}_k + \sum_{u=1}^{U} \zeta_{ijk,u}\gamma_{k,u}\}$, where $\boldsymbol{B}_{ijk}$ is equivalent to the input $\boldsymbol{B}_{ij}$, i.e., we repeat $\boldsymbol{B}_{ij}$ K times so that cause-specific weights $\boldsymbol{w}_k$ share the same inputs. Similarly, the latent representations $\zeta_{ijk,u}$ now also depend on the risk $k = 1, \ldots, K$ to yield cause-specific effects $\gamma_{k,u}$ for each latent feature. Figure 1 illustrates the CR case for an exemplary network architecture. Training the network is based on a joint loss summing up all K loss contributions for each CR and weighted by binary interval weights if the observation is still at risk in the jth interval and 0 if not.

Learning Mixed Effects and Recurrent Events. In many SA settings, data comes in clusters. For example, the survival of patients has been observed at different locations. This is typically the case for multi-center studies for which survival may substantially vary between clusters while being more homogeneous within each cluster. A random

effect (RE), i.e., a linear effect for each cluster with a normal prior, can account for this within-cluster correlation. REs can also be used to account for repeated measurements and recurrent events. Optimization of NNs with random or mixed effects can be done using an EM-type optimization routine (see, e.g., [45]), by training a Bayesian NN (see, e.g., [19]), or by tuning the prior variance based on the equivalence of a random normal prior and a ridge-penalized effect (see, e.g., [43]). While learning the RE prior variance explicitly is desirable, a carefully chosen ridge penalization should yield similar results (due to their mathematical equivalence) while being more straightforward to incorporate in most NNs.

5   Numerical Experiments

We first explore DeepPAMM by investigating some of the proposed model properties in a simulation study. Additionally, we compare DeepPAMM with state-of-the-art algorithms on various benchmark data sets including real-world medical applications. We examine model performance via the integrated Brier score (IBS) [16], which measures both, discrimination and calibration of predicted survival probabilities. Instead of integrating over the whole time domain, we evaluate the IBS at the first three quartiles (Q25, Q50, Q75) of the observed event times in the test set, in order to assess the performance at different time points. While DL-based approaches usually require large data sets for training, DeepPAMM also works well in small data set regimes. In the worst case, if there is not enough data to train the deep part of our network, the structured network part will dominate the predictions. DeepPAMM will then effectively fall back to estimate a PAMM, which in turn is well suited for small data sets. This property is especially important in SA where most data sets are relatively small.

5.1   Simulation and Ablation Study

The goal of our simulation study is to investigate the performance of DeepPAMM under various controlled settings with a focus on 1) mixed effects, 2) competing risks, 3) multimodal data. For all simulations, the data generating process incorporates both, linear effects and non-linear interactions. For every setting, we repeat this procedure 25 times to account for variance in data generation and model fitting. In the spirit of an ablation study, we compare DeepPAMM with its corresponding PAM(M) to investigate the attribution of performance gains as well as the relation to an ideal model (Optimal).

For **competing risks**, we simulate two competing risks based on two different hazards structures. While cause 1 is based on 5 features and multiple non-linear interaction effects, cause 2 relates to 3 features and a more moderate level of interactions as well as non-linearities.

For **mixed effects**, we simulate repeated measurements by defining 60 clusters and drawing a random effect for each cluster unit from a normal distribution with zero mean and a standard deviation of 1.5. Before training DeepPAMM, we pre-train the random effects of the DeepPAMM with the corresponding PAMM and use the associated ridge penalty as a warm start for tuning.

Table 1. Comparison of the average IBS (with standard deviation in brackets) across the three quartiles Q25, Q50, Q75 (rows) for different methods (columns) in different study settings. The †-symbol indicates methods that can only take tabular data information into account.

	CR (cause 1)			CR (cause 2)			Mixed effects			Multimodal		
	Q25	Q50	Q75	Q25	Q50	Q75	Q25	Q50	Q75	Q25	Q50	Q75
KM	5.1	10.8	16.2	4.3	9.1	13.6	7.0	13.5	18.2	4.1†	8.0†	12.1†
	(0.43)	(0.49)	(0.57)	(0.31))	(0.52)	(0.60)	(0.47)	(0.57)	(0.68)	(0.44)	(0.49)	(0.66)
PAMM	3.3	6.0	8.9	2.5	4.5	7.0	3.9	6.9	9.1	3.7†	6.3†	8.6†
	(0.32)	(0.52)	(0.52)	(0.21)	(0.38)	(0.61)	(0.49)	(0.71)	(0.89)	(0.43)	(0.63)	(0.71)
Ours	**2.9**	**5.4**	**8.1**	**2.4**	**4.4**	**6.8**	**3.2**	**5.7**	**7.5**	**3.6**	**6.1**	**8.4**
	(0.41)	(0.40)	(0.43)	(0.38)	(0.46)	(0.68)	(0.41)	(0.61)	(0.69)	(0.43)	(0.52)	(0.65)
Optimal	2.9	5.4	8.0	2.1	4.1	6.5	2.9	5.2	6.8	3.6	6.1	8.3
	(0.72)	(0.80)	(0.79)	(0.22)	(0.39)	(0.63)	(0.36)	(0.57)	(0.68)	(0.42)	(0.58)	(0.68)

For the **multimodal data** scenario, we simulate log-hazards based on linear latent effects from point clouds (PC) based on the data set from ModelNet10 [44]. Each of the PC labels is associated with a different latent coefficient ranging from -0.5 to 0.75. The hazard is defined to depend on these latent coefficients as well as on tabular features. A reduced PointNet [31] is used to model the PCs. This set up has been adapted from [23].

Results. Model comparisons are provided in Table 1. In summary, our proposed model is the best performing method across all three settings and in most cases yields performance values close to the optimal error in terms of the IBS. While performance gains in absolute terms seem small, the decrease in IBS relative to the optimal error is especially noteworthy for CR (cause 1) and the mixed effects setting. Results confirm that DeepPAMM works well in various of the proposed data situations. The ablation study further justifies the deep part of DeepPAMM by its improved performance in comparison to PAMM.

5.2   Benchmark Analysis

We compare our approach with various state-of-the-art methods (Table 2). Comparisons include a tree-based method (ORSF; [21]), a boosting approach (PEMXGB; [4]), as well as (DeepHit; [27]), a well-established deep NN for SA. As baseline models we use a Kaplan-Meier estimator (KM; [11]) and a Cox PH model (CPH; [11]). We restrict our comparison to directly and publicly available SA data sets that have been used in the benchmarks of methods listed above, namely *tumor* [5], *gbsg2* [37], *metabric* (cf. [27]), *breast* [39], *mgus2* [26], and *icu* (cf. [18]). For each method, we perform a random search with 50 configurations and compare the aggregated (mean and std. deviation) test set performances on 25 distinct train-test-splits. The data sets impose different challenges, including CR (icu, mgus2), high-dimensional data (breast), and mixed effects (icu). For these, DeepPAMM is consistently among the best-performing survival models. The main point here is that DeepPAMM is competitive compared to other state-of-the-art methods while maintaining interpretability as illustrated in Sect. 5.3.

Table 2. Performance comparison based on the IBS ($\downarrow$) at the three quartiles (Q25, Q50, Q75) across different data sets (rows) and models (columns) with best models per row highlighted in bold. Missing entries are due to missing support for CRs.

Data set		KM	Cox PH	ORSF	PEMXGB	DeepHit	DeepPAMM
tumor	Q25	6.6 (0.59)	6.0 (0.58)	**5.5** (0.56)	5.7 (0.63)	5.6 (0.55)	5.7 (0.59)
	Q50	12.3 (0.86)	11.2 (0.82)	**10.8** (0.91)	10.9 (1.05)	11.0 (0.96)	10.9 (0.86)
	Q75	17.6 (0.79)	16.3 (0.77)	16.3 (0.85)	**16.2** (0.92)	16.4 (0.95)	**16.2** (0.81)
gbsg2	Q25	3.1 (0.49)	3.1 (0.45)	**3.0** (0.45)	**3.0** (0.46)	3.1 (0.49)	3.1 (0.41)
	Q50	6.8 (0.80)	6.5 (0.72)	**6.2** (0.70)	6.3 (0.68)	6.6 (0.8)	6.5 (0.69)
	Q75	12.5 (1.04)	11.4 (0.94)	**11.1** (0.95)	11.3 (1.01)	11.9 (0.99)	11.5 (0.95)
metabric	Q25	4.0 (0.22)	4.0 (0.26)	4.1 (0.25)	**3.8** (0.22)	4.0 (0.22)	3.9 (0.27)
	Q50	8.6 (0.51)	8.2 (0.54)	8.9 (0.46)	**7.8** (0.49)	8.4 (0.45)	7.9 (0.45)
	Q75	14.0 (0.38)	12.9 (0.47)	14.7 (1.19)	**12.3** (0.51)	13.5 (0.40)	12.6 (0.44)
breast	Q25	**1.9** (0.61)	–	2.0 (0.59)	2.0 (0.57)	2.1 (0.54)	**1.9** (0.60)
	Q50	4.1 (0.89)	–	**4.0** (0.80)	**4.0** (0.83)	4.2 (0.81)	**4.0** (0.90)
	Q75	7.1 (1.13)	–	**6.7** (0.96)	**6.7** (1.10)	7.1 (1.02)	6.8 (1.33)
mgus2 (cause 1)	Q25	**1.1** (0.21)	–	–	1.9 (0.34)	**1.1** (0.21)	**1.1** (0.21)
	Q50	**2.2** (0.34)	–	–	4.1 (0.55)	**2.2** (0.34)	**2.2** (0.34)
	Q75	**3.4** (0.48)	–	–	6.9 (0.69)	3.5 (0.51)	**3.4** (0.49)
mgus2 (cause 2)	Q25	8.7 (0.52)	–	–	8.6 (0.65)	**8.1** (0.55)	8.3 (0.49)
	Q50	14.4 (0.61)	–	–	13.9 (0.84)	**12.9** (0.66)	13.1 (0.65)
	Q75	18.4 (0.60)	–	–	17.9 (1.04)	**15.8** (0.67)	16.0 (0.67)
icu (cause 1)	Q25	**1.3** (0.06)	–	–	1.4 (0.66)	**1.3** (0.06)	**1.3** (0.06)
	Q50	3.6 (0.14)	–	–	3.6 (0.13)	**3.5** (0.13)	**3.5** (0.13)
	Q75	6.7 (0.19)	–	–	6.7 (0.19)	6.5 (0.20)	**6.4** (0.19)
icu (cause 2)	Q25	3.5 (0.15)	–	–	3.5 (0.14)	**3.4** (0.14)	**3.4** (0.14)
	Q50	7.6 (0.17)	–	–	7.6 (0.17)	**7.3** (0.17)	**7.3** (0.16)
	Q75	12.0 (0.15)	–	–	12.1 (0.17)	11.5 (0.20)	**11.3** (0.17)

5.3 Extended Case Study

In this extended case study, we show how DeepPAMM can be used to obtain interpretable feature effects and at the same time incorporate potentially high-dimensional interactions. To illustrate this, we apply DeepPAMM to spatio-temporal data where the outcome is response times (time-to-arrival) of the London fire brigade to fire-related emergency calls [38]. Additionally, the data includes geographic coordinates of the site of the fire as well as information about the ward from which the truck was deployed and the time of day of the incident. We expect a non-linear effect of the time of day that varies with day and night times as well as traffic hours and a bivariate spatial effect of the location with different hazards in different regions of the city. Therefore, we model the hazard for arrival at time t given time of day t_d, spatial coordinates (c_1 and c_2) and ward $v = 1, \ldots, V$ as

Table 3. Performance comparison based on the IBS ($\downarrow$) at the three quartiles (Q25, Q50, Q75) across different models (columns) for the data set of [38] with best models per row highlighted in bold. The performance has been assessed using 25 train-test splits.

Quantile	KM	PAMM	DeepPAMM
Q25	12.8 (0.28)	12.3 (0.30)	**12.2** (0.32)
Q50	18.1 (0.18)	16.9 (0.21)	**16.7** (0.23)
Q75	19.9 (0.11)	18.4 (0.14)	**18.2** (0.14)

$$\log(h(t|t_d, c_1, c_2, v)) = \underbrace{\beta_0 + f_0(t) + f_1(t_d) + f_2(c_1, c_2) + b_v}_{\text{structured}} + \underbrace{d(t, t_d, c_1, c_2, v)}_{\text{unstructured}}$$

where $f_1(t_d)$ is estimated as a cyclic spline that enforces equal values of the function at 0 and 24 h, $f_2(c_1, c_2)$ is a bivariate tensor product spline and b_v are random effects for the individual wards. In the unstructured part, we additionally allow for high-dimensional interactions between the features from the structured part. This way, we can investigate whether the predictive performance can be improved beyond the structured part. Structured effects are given in Fig. 2. For interpretation, note that higher hazards imply shorter response times, thus response times are on average longer during night hours and between 12 and 18 p.m. as well as in the periphery of the city. The results w.r.t. the predictive performance are shown in Table 3, where we compare our model with a KM baseline and the respective PAMM. In addition to the PAMM specification, our model includes a NN with three layers (64, 32, 8 neurons) to model feature interactions. The results indicate that on average the performance improves slightly when the unstructured part is added. Given the resulting standard deviations, we conclude that the structured part is sufficient. Further, DeepPAMM's structured effects are in line with results presented in [38]. This shows the strength of DeepPAMM: maintaining interpretability of covariate effects as illustrated in Fig. 2, while also allowing the investigation of additional effects in the unstructured part.
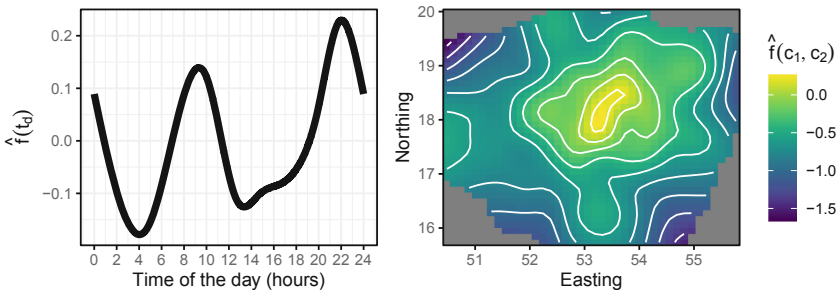

Fig. 2. Smooth cyclic (left) and spatial (right) effect of a DeepPAMM. Effects are from a single of the 25 runs.

6   Concluding Remarks

We present DeepPAMM, a novel semi-structured deep learning approach to survival analysis. Our experiments demonstrate that our model has high predictive capacity and is capable of modeling diverse complex data associations. DeepPAMM allows to include non-linear and feature interaction effects in the model, can be used to model non-proportional hazards, time-varying effects and competing risks, while also accounting for correlation in the data using mixed effects. The deep part of the model further makes estimation in high-dimensional settings possible and can be used to include unstructured data into the survival analysis. The additive predictor in our approach allows for straightforward interpretability and to recover the PAM(M) when no additional deep predictors are necessary. Our method can be fit using existing software solutions (e.g., deepregression [34]).

References

1. Alaa, A.M., van der Schaar, M.: Deep multi-task gaussian processes for survival analysis with competing risks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 2326–2334. Curran Associates Inc. (2017)
2. Argyropoulos, C., Unruh, M.L.: Analysis of time to event outcomes in randomized controlled trials by generalized additive models. PLoS ONE **10**(4), e0123784 (2015)
3. Bender, A., Groll, A., Scheipl, F.: A generalized additive model approach to time-to-event analysis. Statist. Model. **18**(3–4), 299–321 (2018)
4. Bender, A., Rügamer, D., Scheipl, F., Bischl, B.: A general machine learning framework for survival analysis. In: Hutter, F., Kersting, K., Lijffijt, J., Valera, I. (eds.) ECML PKDD 2020. LNCS (LNAI), vol. 12459, pp. 158–173. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67664-3_10
5. Bender, A., Scheipl, F.: pammtools: piece-wise exponential additive mixed modeling tools. arXiv preprint arXiv:1806.01042 (2018)
6. 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)
7. Biganzoli, E., Boracchi, P., Marubini, E.: A general framework for neural network models on censored survival data. Neural Netw. **15**(2), 209–218 (2002)
8. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
9. Cai, T., Hyndman, R.J., Wand, M.P.: Mixed model-based hazard estimation. J. Comput. Graph. Statist. **11**(4), 784–798 (2002)
10. 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 - KDD 2016, pp. 785–794 (2016)
11. Cox, D.R.: Regression models and life-tables. J. Roy. Statist. Soc. Ser. B (Methodological) **34**(2), 187–220 (1972)
12. Faraggi, D., Simon, R.: A neural network model for survival data. Statist. Med. **14**(1), 73–82 (1995)

13. Fornili, M., Ambrogi, F., Boracchi, P., Biganzoli, E.: Piecewise exponential artificial neural networks (PEANN) for modeling Hazard function with right censored data. In: Formenti, E., Tagliaferri, R., Wit, E. (eds.) CIBB 2013 2013. LNCS, vol. 8452, pp. 125–136. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09042-9_9

14. Friedman, M.: Piecewise exponential models for survival data with covariates. Ann. Statist. **10**(1), 101–113 (1982)

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

16. Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Statist. Med. **18**(17–18), 2529–2545 (1999)

17. Haarburger, C., Weitz, P., Rippel, O., Merhof, D.: Image-based survival prediction for lung cancer patients using CNNS. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 1197–1201. IEEE (2019)

18. Hartl, W.H., Bender, A., Scheipl, F., Kuppinger, D., Day, A.G., Küchenhoff, H.: Calorie intake and short-term survival of critically ill patients. Clin. Nutr. **38**(2), 660–667 (2019)

19. Hernández-Lobato, J.M., Adams, R.: Probabilistic backpropagation for scalable learning of Bayesian neural networks. In: International Conference on Machine Learning, pp. 1861–1869. PMLR (2015)

20. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S.: Random survival forests. Ann. Appl. Statist. **2**(3), 841–860 (2008)

21. Jaeger, B.C., et al.: Oblique random survival forests. Ann. Appl. Statist. **13**(3), 1847–1883 (2019)

22. 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. **18**(1), 1–12 (2018)

23. Kopper, P., Pölsterl, S., Wachinger, C., Bischl, B., Bender, A., Rügamer, D.: Semi-structured deep piecewise exponential models. In: Survival Prediction-Algorithms, Challenges and Applications, pp. 40–53. PMLR (2021)

24. Kvamme, H., Borgan, Ø., Scheel, I.: Time-to-event prediction with neural networks and cox regression. arXiv preprint arXiv:1907.00825 (2019)

25. Kvamme, H., Borgan, Ø.: Continuous and discrete-time survival prediction with neural networks. arXiv preprint arXiv:1910.06724 (2019)

26. Kyle, R.A., et al.: A long-term study of prognosis in monoclonal gammopathy of undetermined significance. New Engl. J. Med. **346**(8), 564–569 (2002)

27. Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: DeepHit: a deep learning approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

28. Lee, D., et al.: Theory and software for boosted nonparametric hazard estimation. In: Survival Prediction - Algorithms, Challenges and Applications, pp. 149–158. PMLR (2021)

29. Liestøl, K., Andersen, P.K., Andersen, U.: Survival analysis and neural nets. Statist. Med. **13**(12), 1189–1200 (1994)

30. Pölsterl, S., Sarasua, I., Gutiérrez-Becker, B., Wachinger, C.: A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. In: Cellier, P., Driessens, K. (eds.) ECML PKDD 2019. CCIS, vol. 1167, pp. 453–464. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_37

31. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)

32. Ramjith, J., Bender, A., Roes, K.C., Jonker, M.A.: Recurrent Events Analysis with Piecewise exponential Additive Mixed Models. Preprint at Research Square (2021)

33. Ranganath, R., Perotte, A., Elhadad, N., Blei, D.: Deep survival analysis. In: Proceedings of the 1st Machine Learning for Healthcare Conference, vol. 59, pp. 101–114 (2016)
34. Rügamer, D., et al.: Deepregression: a flexible neural network framework for semi-structured deep distributional regression. arXiv preprint arXiv:2104.02705 (2021)
35. Rügamer, D., Kolb, C., Klein, N.: Semi-structured deep distributional regression: combining structured additive models and deep learning. arXiv preprint arXiv:2002.05777 (2021)
36. Schmid, M., Hothorn, T.: Flexible boosting of accelerated failure time models. BMC Bioinformatics **9**(1), 1–13 (2008)
37. Schumacher, M., et al.: Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German breast cancer study group. J. Clin. Oncol. **12**(10), 2086–2093 (1994)
38. Taylor, B.M.: Spatial modelling of emergency service response times. J. Roy. Statist. Soc. Ser. A (Statist. Soc.) **180**(2), 433–453 (2017)
39. Ternès, N., Rotolo, F., Heinze, G., Michiels, S.: Identification of biomarker-by-treatment interactions in randomized clinical trials with survival outcomes and high-dimensional spaces. Biometric. J. **59**(4), 685–701 (2017)
40. Vale-Silva, L.A., Rohr, K.: Long-term cancer survival prediction using multimodal deep learning. Sci. Rep. **11**(1), 1–12 (2021)
41. Weber, T., Ingrisch, M., Bischl, B., Rügamer, D.: Towards modelling hazard factors in unstructured data spaces using gradient-based latent interpolation. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
42. Whitehead, J.: Fitting cox's regression model to survival data using glim. J. Roy. Statist. Soc. Ser. C (Appl. Statist.) **29**(3), 268–275 (1980)
43. Wood, S.N.: Generalized Additive Models: An Introduction with R, 2 rev edn. Chapman & Hall/CRC Texts in Statistical Science, Boca Raton (2017)
44. Wu, Z., et al.: 3d shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
45. Xiong, Y., Kim, H.J., Singh, V.: Mixed effects neural networks (Menets) with applications to gaze estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7743–7752 (2019)