Journal of Data Science 0 (0), 1–14
??? 2025
DOI: 10.6339/25-JDS1173
Statistical Data Science

Neural Network for Correlated Survival Outcomes Using Frailty Model

Ruiwen Zhou[1], Kevin He[2], Di Wang[2], Lili Liu[1], Shujie Ma[3], Annie Qu[4], J. Philip Miller[1], and Lei Liu[1,*]

[1]*Division of Biostatistics, Washington University in St. Louis, St. Louis, Missouri, USA*
[2]*Department of Biostatistics, University of Michigan, Ann Arbor, Michigan, USA*
[3]*Department of Statistics, University of California, Riverside, California, USA*
[4]*Department of Statistics, University of California, Irvine, California, USA*

Abstract

Extensive literature has been proposed for the analysis of correlated survival data. Subjects within a cluster share some common characteristics, e.g., genetic and environmental factors, so their time-to-event outcomes are correlated. The frailty model under proportional hazards assumption has been widely applied for the analysis of clustered survival outcomes. However, the prediction performance of this method can be less satisfactory when the risk factors have complicated effects, e.g., nonlinear and interactive. To deal with these issues, we propose a neural network frailty Cox model that replaces the linear risk function with the output of a feed-forward neural network. The estimation is based on quasi-likelihood using Laplace approximation. A simulation study suggests that the proposed method has the best performance compared with existing methods. The method is applied to the clustered time-to-failure prediction within the kidney transplantation facility using the national kidney transplant registry data from the U.S. Organ Procurement and Transplantation Network. All computer programs are available at https://github.com/rivenzhou/deep_learning_clustered.

Keywords *correlated survival outcomes; deep learning; prediction; random effect*

1 Introduction

Survival models have been extensively developed in medical research to make inferences and predictions on failure times. The Cox proportional hazards model is the most commonly used regression model for survival outcomes. In the conventional Cox model, the survival outcomes from different observational units are assumed to be independent, given observed covariates.

However, dependence among survival outcomes is likely to occur. To account for within-cluster dependency, extensive literature has been published on frailty models, where the survival outcomes are assumed to be independent conditional on an unobserved frailty (random effect). In the Cox proportional hazards frailty model, the frailty or random effect is assumed to follow a probability distribution (Balan and Putter (2020)). To illustrate, Paik et al. (1994), Shih and Louis (1995), and Hens et al. (2009) assumed the frailty follows a gamma distribution, while Ripatti and Palmgren (2000) considered the log-normal frailty distribution. Other examples include the power-variance-function (PVF) family, where the marginal distribution of survival

*Corresponding author. Email: lei.liu@wustl.edu.

Received September 5, 2024; Accepted March 1, 2025

outcomes can be obtained in a closed form. Besides the frailty models, the stratified Cox model is a popular tool for clustered survival outcomes because of its simplicity in computation and interpretation. However, according to Glidden and Vittinghoff (2004), the stratified Cox model discards between-cluster comparison information, leading to inefficient estimation. This issue becomes particularly pronounced when dealing with a large number of strata or clusters, such as in the correlated survival outcomes observed in the motivating kidney transplant study.

Our motivating example is the kidney transplant registry data from the U.S. Organ Procurement and Transplantation Network (OPTN: https://optn.transplant.hrsa.gov/data/). The dataset includes the incidence of graft failure or death following transplantation for each patient across multiple kidney transplant centers. Patients from the same transplant center may receive treatments under the same protocol, adhere to uniform center policies, or be influenced by the same local environmental factors. Such commonalities may result in similar health outcomes for patients at the same transplant center. Ignoring such associations leads to inefficiency and bias in predicting the time-to-event. Furthermore, the impact of myriad of variables from both donors and recipients on the survival outcome may be complicated. It is of great interest to predict the patients' survival outcome based on these predictors while accounting for the correlation structure within each cluster.

Recently, deep learning methods have surged as effective tools for prediction. Fan et al. (2021) discussed the theoretical foundations of deep neural networks and explained their practical and theoretical benefits over traditional statistical methods by applying depth, overparameterization, and training techniques (e.g., stochastic gradient descent and batch normalization). Neural networks improve estimation and prediction performance because they are highly flexible and can model non-linear relationships. Besides, with multiple hidden layers structure, neural networks can learn hierarchical representations of data, potentially capturing intricate patterns and interactions. In addition, modern training techniques of neural networks offer various regularization techniques (e.g., dropout and L_2 regularization) that can help prevent overfitting. These methods, when applied to survival models, have demonstrated superior predictive abilities, especially when dealing with complex nonlinear and interactive risk effects. Works by Liao and Ahn (2016), Martinsson (2017), and Ranganath et al. (2016) introduced deep learning algorithms assuming survival outcomes adhered to the Weibull distribution. Under the semiparametric Cox proportional hazards model framework, Faraggi and Simon (1995) first adopted a feed-forward neural network. Later, Katzman et al. (2018) introduced "Deepsurv", an algorithm that harnesses advanced deep learning methodologies while minimizing the loss function derived from the partial likelihood function. Zhong et al. (2021) proposed deep extensions of the extended hazard model, named as DeepEH, which encompassed the Cox and accelerate failure time (AFT) models. Ching et al. (2018) suggested Cox-nnet for high-throughput RNA sequencing data. Hao et al. (2018) illustrated Cox-PASNet method, which integrates high-dimensional gene expression data and clinical data on a simple neural network architecture for survival analysis to improve the biological interpretation of genes and pathways. A review on deep learning methods for survival analysis was provided by Wiegrebe et al. (2023).

Despite the considerable exploration of deep learning for survival outcomes, correlated survival outcomes remain relatively untouched. Lee et al. (2023) proposed a deep neural network based on the Gamma frailty model using the H-likelihood framework. Later, Wu et al. (2024) proposed the neural frailty machine, using frailty terms for modeling crossing hazards and injecting a domain-specific inductive bias for nonparametric hazard regression. However, they did not consider within-cluster prediction for correlated survival outcomes. In our study, we introduce a neural network aimed at predicting correlated survival times under the Cox proportional

hazards model with a normal frailty. This model predicts the risk score based on covariates using a feed-forward neural network. To address computational challenges, we employ a penalized partial likelihood formulation with the Laplace approximation to define the loss function.

The rest of the paper is organized as follows. Section 2 describes the proposed deep-learning method for correlated survival outcomes. In Section 3, we undertake simulation studies to evaluate the predictive performance of our proposed method against alternative methods. Section 4 presents a data analysis of kidney post-transplant graft failure or death prediction for patients from the same transplant center using our proposed method. We summarize our method and present future directions in Section 5.

2 Model

2.1 Problem Formulation

Let T_{ij} denote the event time for the j-th unit within the i-th cluster, where $i = 1, \ldots, s$ and $j = 1, \ldots, n_i$. The sample size $n = \sum_{i=1}^{s} n_i$. We denote C_{ij} as the censoring time, $U_{ij} = \min\left(T_{ij}, C_{ij}\right)$ as the observed time, and $\Delta_{ij} = I\left\{T_{ij} \leqslant C_{ij}\right\}$ as the right-censored indicator. Given frailty (or random effect) b_i, the event times are assumed independent with the conditional hazard function

$$\lambda_{ij}(t \mid b_i) = \lambda_0(t) \exp\left(\boldsymbol{X}_{ij}^T \boldsymbol{\beta} + b_i\right),$$

where $\boldsymbol{X}_{ij}$ is the vector of explanatory variables, $\lambda_0(t)$ represents the baseline hazard function. In this frailty model, only the random intercept is considered, which follows a normal distribution with mean 0 and variance θ. We can also consider more complicated forms of random effects, e.g., replacing b_i by $\boldsymbol{Z}_{ij}^T \boldsymbol{u_i}$, where $\boldsymbol{u_i} \sim N(0, \Sigma_u)$ is a vector of random effects and $\boldsymbol{Z}_{ij}$ is the associated covariate vector.

To better describe the covariate effects, we consider a feed-forward artificial neural network (FNN) with L hidden layers. We adapt the classical FNN under Cox proportional hazards model to a deep learning method within the frailty model framework, which may lead to more accurate hazard function estimates and improved survival predictions. The covariate $\boldsymbol{X}_{ij}$ has p variables, and $\boldsymbol{X}_{ij}^T \boldsymbol{\beta}$ can be replaced by a nonlinear function of the predictors $\boldsymbol{X}_{ij}$ with network weights $\boldsymbol{\omega}^{(l)}$ and bias $\boldsymbol{\delta}^{(l)}$ through a series of nested activation function $g_l(\cdot)$ for layers $l = 0, \ldots, L$. Weights and biases are also called slope coefficients and intercepts, respectively, in statistical terms. To be specific, the k_0 nodes of the first hidden layer can be calculated through

$$\boldsymbol{\alpha}_{ij}^{(0)} = g_0\left\{\boldsymbol{\omega}^{(0)} \boldsymbol{X}_{ij} + \boldsymbol{\delta}^{(0)}\right\},$$

where $\boldsymbol{\omega}^{(0)}$ is a $k_0 \times p$ weight matrix, $\boldsymbol{\delta}^{(0)}$ is a bias vector of length k_0, and the activation function $g_0(\cdot)$ is applied element-wise to its input vector. For the l-th hidden layer ($l = 1, \ldots, L-1$) with k_l nodes, the layer's output is

$$\boldsymbol{\alpha}_{ij}^{(l)} = g_l\left\{\boldsymbol{\omega}^{(l)} \boldsymbol{\alpha}_{ij}^{(l-1)} + \boldsymbol{\delta}^{(l)}\right\},$$

where $\boldsymbol{\omega}^{(l)}$ is a $k_l \times k_{l-1}$ matrix and $\boldsymbol{\delta}^{(l)}$ is of length k_l. Finally, when only random intercept is considered, the univariate output from the neural network is related to the proportional hazards function by

$$\lambda_{ij}^{NN}(t \mid b_i) = \lambda_0(t) \exp(\alpha_{ij}^{(L)} + b_i), \tag{1}$$

where $\alpha_{ij}^{(L)} = g_L(\boldsymbol{\omega}^{(L)} \boldsymbol{\alpha}_{ij}^{(L-1)})$, $\boldsymbol{\alpha}_{ij}^{(L-1)}$ is the second to the last layer's output, and $\boldsymbol{\omega}^{(L)}$ is a $1 \times k_{L-1}$ vector.

2.2 Penalized Partial Likelihood

The marginal likelihood for cluster i in model (1) is

$$L_i^{NN}\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\big) = \int \prod_{j=1}^{n_i} \exp\Big[l_{ij}^{NN}\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}|b_i\big)\Big] p(b_i;\theta) db_i,$$

where

$$l_{ij}^{NN}\left(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}|b_i\right) = \Delta_{ij}\Big[\log\big(\lambda_0(t)\big) + \alpha_{ij}^{(L)} + b_i\Big] - \Lambda_0(t)\exp(\alpha_{ij}^{(L)} + b_i),$$

and

$$p(b_i;\theta) = \theta^{-1/2}(2\pi)^{-1/2}\exp(-\frac{1}{2}b_i{}'\theta^{-1}b_i).$$

The function $\Lambda_0(t) = \int_0^t \lambda_0(u)du$ is the baseline cumulative hazard function, $l_{ij}^{NN}(\cdot|b_i)$ denotes the log likelihood function for subject j in the i-th cluster given random effect b_i. The parameter $\boldsymbol{\omega}$ represents the combined vectorization of $\boldsymbol{\omega}^{(0)}, \ldots, \boldsymbol{\omega}^{(L)}$ into a single column vector, $\boldsymbol{\delta}$ represents the concatenation of $\boldsymbol{\delta}^{(0)}, \ldots, \boldsymbol{\delta}^{(L-1)}$ into a column vector. To avoid overfitting, following Mandel et al. (2023), we add L_2 penalization to the neural network parameters $\boldsymbol{\omega}$ and $\boldsymbol{\delta}$, with regularization parameter $\boldsymbol{\gamma}$.

The likelihood function for model (1) in cluster i with parameter regularization then becomes

$$\tilde{L}_i^{NN}\left(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\right) = \theta^{-1/2}(2\pi)^{-1/2}\int \exp\Bigg[\sum_{j=1}^{n_i}\Big\{\Delta_{ij}\Big[\log\big(\lambda_0(t)\big) + \alpha_{ij}^{(L)} + b_i\Big] - \Lambda_0(t)\times$$
$$\exp(\alpha_{ij}^{(L)} + b_i) - \frac{1}{2}b_i{}'\theta^{-1}b_i - \boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta})\Big\}\Bigg]db_i. \tag{2}$$

Under the normal distribution assumption for the frailty term, equation (2) is difficult to maximize with an integral. Following Ripatti and Palmgren (2000), we use a Laplace approximation for the integral in $\tilde{L}_i^{NN}\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\big)$. This leads to the approximated marginal log-likelihood for cluster i,

$$\log(\tilde{L}_i^{NN}) = l_i\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\big) \approx -\frac{1}{2}\log(\theta) - \frac{1}{2}\log\left|K_i''(\tilde{b}_i)\right| - K_i(\tilde{b}_i) - n_i\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}),$$

where

$$K_i(\tilde{b}_i) = -\sum_{j=1}^{n_i}\Big[\Delta_{ij}\Big[\log\big(\lambda_0(t)\big) + \alpha_{ij}^{(L)} + \tilde{b}_i\Big] + \Lambda_0(t)\exp(\alpha_{ij}^{(L)} + \tilde{b}_i) + \frac{1}{2}\tilde{b}_i'\theta^{-1}\tilde{b}_i\Big] \tag{3}$$

and

$$K_i''(\tilde{b}_i) = \frac{\partial^2 K(\tilde{b}_i)}{\partial^2 \tilde{b}_i} = \sum_{j=1}^{n_i}\Big[\Lambda_0(t)\exp(\alpha_{ij}^{(L)} + \tilde{b}_i) + \theta^{-1}\Big]. \tag{4}$$

The parameter $\tilde{b}_i = \tilde{b}_i(\boldsymbol{\omega}, \boldsymbol{\delta})$ denotes the solution to the partial derivatives of $K_i(b_i)$ with respect to b_i.

According to Lin et al. (2008); Yu and Liu (2011); Yu et al. (2013, 2014), omitting the complicated term $\log|K_i''(\tilde{b}_i)|$ in $\log(\tilde{L}_i^{NN})$ has a negligible effect on the parameter estimation. Their

simulation studies demonstrate that this simplification does not significantly affect the accuracy of the estimated parameters. Consequently, we have excluded this term from our likelihood approximation to streamline computation without compromising model performance. Further, for right-censored data, to avoid estimating the baseline hazard function, replacing the full likelihood in $K_i(\tilde{b}_i)$ with a partial likelihood leads to the following penalized approximated partial log-likelihood

$$pl = \sum_{i=1}^{s} pl_i = \sum_{i=1}^{s}\sum_{j=1}^{n_i}\left\{\Delta_{ij}\left[(\alpha_{ij}^{(L)} + b_i) - \log \sum_{d,q\in R(t_{ij})} \exp(\alpha_{dq}^{(L)} + b_q)\right] - \frac{1}{2} b_i' \theta^{-1} b_i\right\} - n\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}), \tag{5}$$

where $R\left(t_{ij}\right)$ denotes indexes for subjects who are at risk at time t_{ij}.

Ripatti and Palmgren's (2000) method estimates the fixed effects and random effects b_i using an iterative approach, alternating between estimating equations. To be specific, in the iterative algorithm, given θ, we can estimate $(\boldsymbol{\omega}, \boldsymbol{\delta})$ by solving $\frac{\partial pl}{\partial \boldsymbol{\omega}} = 0$ and $\frac{\partial pl}{\partial \boldsymbol{\delta}} = 0$. Then, given the updated $(\boldsymbol{\omega}, \boldsymbol{\delta})$, the random effect b_i is updated by solving $\frac{\partial pl}{\partial b_i} = 0$. These steps are repeated until convergence. This approach, however, complicates backpropagation and imposes a substantial computational burden, as it requires multiple nested loops to iteratively update fixed and random effects. To address the computational complexity of this iterative algorithm, we propose an alternative loss function for estimating $(\boldsymbol{\omega}, \boldsymbol{\delta})$ and the random effect b_i,

$$plnn = \sum_{i=1}^{s}\sum_{j=1}^{n_i}\left\{\Delta_{ij}\left[(\eta_{ij}^{(x)}\alpha_{ij}^{(L)} + \eta_i^{(b)}) - \log \sum_{d,q\in R(t_{ij})} \exp(\eta_{dq}^{(x)}\alpha_{dq}^{(L)} + \eta_q^{(b)})\right] - \frac{1}{2}\eta_i^{(b)\prime}\theta^{-1}\eta_i^{(b)}\right\} - n\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}), \tag{6}$$

where $\alpha_{ij}^{(L)} = g_L(\boldsymbol{\omega}^{(L)}\boldsymbol{\alpha}_{ij}^{(L-1)})$, $\boldsymbol{\eta}^{(x)} = (\eta_{11}^{(x)}, \ldots, \eta_{sc}^{(x)})$, and $\boldsymbol{\eta}^{(b)} = (\eta_1^{(b)}, \ldots \eta_c^{(b)})$ are weights for the final output layer. This allows the parameters $\boldsymbol{\omega}, \boldsymbol{\delta}, \eta_{ij}^{(x)}, \hat{b}_i = \hat{\eta}_i^{(b)}$ to be updated simultaneously during neural network backpropagation, bypassing the need for iterative updates and streamlining the estimation process. Consequently, rather than estimating the neural network parameters and the random effect b_i iteratively, we can estimate and update b_i in a single step using $\hat{\eta}_i{}^{(b)}$. Figure 1 illustrates our proposed neural network structure. In this structure, $\hat{\boldsymbol{\eta}}^{(b)}$ is updated simultaneously with other model parameters $(\hat{\boldsymbol{\omega}} = \{\hat{\boldsymbol{\omega}}^{(0)}, \ldots, \hat{\boldsymbol{\omega}}^{(L)}\}, \hat{\boldsymbol{\delta}} = \{\hat{\boldsymbol{\delta}}^{(0)}, \ldots, \hat{\boldsymbol{\delta}}^{(L-1)}\}, \hat{\boldsymbol{\eta}}^{(x)})$. For a single-layer network, differentiation of the approximated partial likelihood with respect to $\boldsymbol{\eta}^{(x)}$, $\boldsymbol{\eta}^{(b)}$, $\boldsymbol{\omega}$, $\boldsymbol{\delta}$ leads to the following quasi-score equations with $\alpha_{ij}^{(1)} = g_1(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})$:

$$\frac{\partial plnn}{\partial \eta_{ij}^{(x)}} = \Delta_{ij}\left(\alpha_{ij}^{(1)} - \frac{\alpha_{ij}^{(1)}\exp(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)})}{\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)})}\right), \tag{7}$$

$$\frac{\partial plnn}{\partial \eta_i^{(b)}} = \sum_{j=1}^{n_i}\Delta_{ij}\left(1 - \frac{\exp(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)})}{\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)})}\right) - \eta_i^{(b)}\theta^{-1}, \tag{8}$$

$$\frac{\partial plnn}{\partial \omega_{k1}^{(1)}} = \sum_{i=1}^{s}\sum_{j=1}^{n_i}\Delta_{ij}\left(\eta_{ij}^{(x)} - \frac{\eta_{ij}^{(x)}\exp(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)})}{\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)})}\right)\cdot g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)}) \, g_0(\omega_{.k}^{(0)}x_{ij} + \delta_k^{(0)}) - 2n\boldsymbol{\gamma}\omega_{k1}^{(1)}, \tag{9}$$

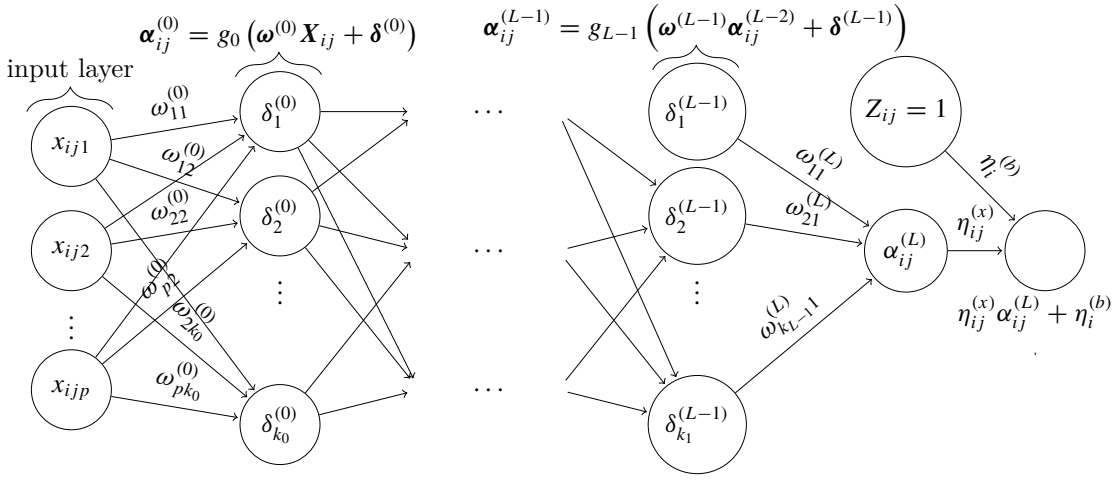

Figure 1: Network graph of a $(L+1)$-layer perceptron with p input units. The random effect intercept b_i and indicator covariates $Z_{ij} = 1$ are included in the final layer of the network.

$$\frac{\partial plnn}{\partial \omega_{lk}^{(0)}} = \sum_{i=1}^{s}\sum_{j=1}^{n_i}\Delta_{ij}\left(\eta_{ij}^{(x)} - \frac{\eta_{ij}^{(x)}\exp\left(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)}\right)}{\sum_{d,q\in R(t_{ij})}\exp\left(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)}\right)}\right) g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})$$
$$\omega_{k1}^{(1)} g_0'(\omega_{.k}^{(0)}x_{ij} + \delta_k^{(0)})x_{ijl} - 2n\boldsymbol{\gamma}\omega_{lk}^{(0)}, \tag{10}$$

$$\frac{\partial plnn}{\partial \delta_k^{(0)}} = \sum_{i=1}^{s}\sum_{j=1}^{n_i}\Delta_{ij}\left(\eta_{ij}^{(x)} - \frac{\eta_{ij}^{(x)}\exp\left(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_{ij}^{(b)}\right)}{\sum_{d,q\in R(t_{ij})}\exp\left(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)}\right)}\right)\cdot$$
$$g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})\omega_{k1}^{(1)} g_0'(\omega_{.k}^{(0)}x_{ij} + \delta_k^{(0)}) - 2n\boldsymbol{\gamma}\delta_k^{(0)}, \tag{11}$$

where $\boldsymbol{\eta}^{(x)}$ and $\boldsymbol{\eta}^{(b)}$ are the weights for the last layer, $\omega_{k1}^{(1)}$ is the weight connecting the k-th hidden node to the univariate output $\boldsymbol{\alpha}_{ij}^{(1)}$, $\omega_{lk}^{(0)}$ is the weight connecting the l-th input to the k-th hidden node in the hidden layer, $\delta_k^{(0)}$ is the bias of the k-th hidden node in the hidden layer, and $\omega_{.k}^{(0)}$ is the k-th entry of the vector $\omega^{(0)}$.

To train the neural network, we develop our code along the lines of the Deepsurv method (Katzman et al. (2018), Kvamme et al. (2019)): standardization of the continuous input, Adaptive Moment Estimation (Adam) for the gradient descent algorithm, Nesterov momentum, and learning rate schedule. We tune the hyperparameter exponential learning rate decay constant and apply inverse time decay to the learning rate at each epoch. Since the goal is prediction, we will focus on the estimation of $(\boldsymbol{\omega}, \boldsymbol{\delta}, \boldsymbol{\eta}^{(x)}, \boldsymbol{\eta}^{(b)})$. The parameter θ is estimated by solving the estimating equation derived from penalized partial likelihood function as in Ripatti and Palmgren (2000). The baseline hazard function can be estimated with a Breslow-type estimator:

$$\widehat{\Lambda}_0(t) = \sum_{i,j:x_{ij}\leqslant \mathrm{t}} \frac{\Delta_{ij}}{\Sigma_{d,q\in R(x_{ij})}\exp\left(\hat{\eta}_{dq}^{(x)}g_1(\hat{\boldsymbol{\omega}}^{(1)}\hat{\boldsymbol{\alpha}}_{dq}^{(0)}) + \hat{\eta}_q^{(b)}\right)}.$$

3 Simulation Study

We generate the data under the Cox model with shared frailty and nonlinear effects (true model). Then we compare the proposed method to (i) Deepsurv, (ii) the Cox model with only linear

effects, (iii) the Cox model with linear effects and all two-way interactions, (iv) the Cox model with frailty and linear effects, (v) the Cox model with frailty, linear and all two-way interaction effects, and (vi) the Cox with fixed clustering effects. As in the real data analysis, we are interested in the within-cluster prediction; so for subjects within a cluster, we randomly assign 50% subjects to the training dataset and the other 50% subjects to the test dataset. The ReLU (Rectified Linear Unit) activation function is selected in the neural network prediction.

To evaluate the performance of the models in terms of discrimination, we adopt the concordance index (C-index), a measure of the rank correlation between predicted risk scores and observed time points (Harrell Jr et al. (1984)). If C-index = 0.5, the method is the same as a random guess. If C-index = 1, the ranking of predicted risk scores perfectly matches that of the observed death times.

The data are generated from a proportional hazards model,

$$\lambda_{ij}\,(t \mid b_i) = \lambda_0(t)\exp(r_{ij}),$$

where $i = 1, \ldots, s$; $j = 1, \ldots, c$; $\lambda_0(t) = 1$ is an exponential baseline hazard; and r_{ij} is the risk score. To mimic the observations in the motivating example of kidney transplant data, we have $s = 200$ clusters, and the sizes of these clusters are randomly drawn from a distribution defined by $n_i \sim \text{Uniform}(20, 100)$. Four values for the frailty variance are used, i.e., $\theta = 0, 1.5, 2.5$, and 3.5. We consider two scenarios for covariates $\boldsymbol{X}_{ij}$. In scenario 1, we first generate five independent variables $\boldsymbol{M}_{ij} = \left(M_{ij1}, M_{ij2}, M_{ij3}, M_{ij4}, M_{ij5}\right)^T$ from normal distributions with mean 0 and variance 1, we then set $r_{ij} = \boldsymbol{X}_{ij}^T\boldsymbol{\beta} - 3 + b_i$. Following the setup in Katzman et al. (2018), the covariates are calculated by $\boldsymbol{X}_{ij} = \left(M_{ij1}^2, M_{ij2}^2, M_{ij3}^2, M_{ij4}^2, M_{ij5}^2\right)^T$, where no interactive covariate effects are considered here, and the parameters are $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^T = (0.5, 0.5, 0.5, 0.5, 0.5)^T$. The censoring times are generated from Uniform(0, 0.5) with around 70% of the event times independently right-censored. In scenario 2, we first generate 15 independent variables $\boldsymbol{M}_{ij} = (M_{ij1}, M_{ij2}, M_{ij3}, M_{ij4}, M_{ij5}, M_{ij6}, M_{ij7}, M_{ij8}, M_{ij9}, M_{ij10}, M_{ij11}, M_{ij12}, M_{ij13}, M_{ij14}, M_{ij15})^T$ from normal distributions with mean (1, 1, 1, 2, 2, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0) and variance 1, and then set $M_{ij8} = I(M_{ij8} < 1)$ to generate a binary covariate. We generate more complicated nonlinear effects inspired by case 3 in Zhong and Wang (2023): $\boldsymbol{X}_{ij} = \{\boldsymbol{X}_{ij1}, \boldsymbol{X}_{ij2}, \boldsymbol{X}_{ij3}, \boldsymbol{X}_{ij4}\} = \{0.1\exp(M_{ij1}(1 + M_{ij2} - M_{ij3}\ M_{ij4}M_{ij5})/2)|M_{ij5} + 0.2 - 0.01M_{ij9}M_{ij10}|,\ M_{ij5}(M_{ij3}M_{ij4} - 0.3)\ /(|2M_{ij3}M_{ij4}M_{ij6} - 1 + 0.01M_{ij11}M_{ij12}| + 1),\ 2\sin(M_{ij1}M_{ij2}M_{ij5})|M_{ij2}M_{ij5}M_{ij6} - 0.6 - 0.01M_{ij13}\ M_{ij14}|,\ \log(|M_{ij1}M_{ij2}M_{ij6}| + |M_{ij5}M_{ij7}M_{ij8} + 0.01M_{ij15}|)\}$. The risk scores are generated by $r_{ij} = \boldsymbol{X}_{ij}^T\boldsymbol{\beta} - 4 + b_i$, and $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)^T = (1, 1, 1, 1)^T$. The censoring times are generated from Uniform(0, 15) with around 70% of the right censoring rate.

Tables 1 and 2 show the results under different frailty variances for scenarios 1 and 2, respectively. The proposed method and Deepsurv are fitted under a two-layer neural network with (64, 64) hidden nodes. Following Kvamme et al. (2019), hyperparameters in the simulation study and real data analysis are selected through a random search on the validation set over the relevant parameters in Table 5 in the Appendix. We train the model for 100 epochs with a batch size of 128, a dropout rate of 0.2, and a weight decay of 0.001. The learning rate is dynamically adjusted based on the optimal rate identified by the learning rate finder. In terms of training stability, we implement early stopping with a patience threshold to halt training if no improvement is observed, ensuring stability across different runs. We ensure model stability by setting early stopping criteria and observing training consistency across multiple runs.

Table 1: C-index for 100 simulated test datasets in scenario 1 with simple quadratic nonlinear fixed effects (standard deviations in brackets). Abbreviations: Prop. (Proposed model), DS (DeepSurv), CF-Lin (Cox frailty with linear effects), CF-Int (Cox frailty with interactions), Cox (Cox proportional hazards model), Cox-Int (Cox model with interactions), Cox-FC (Cox model with fixed clustering effects), TM (True model).

Frailty Variance	Prop.	DS	CF-Lin	CF-Int	Cox	Cox-Int	Cox-FC	TM
0	72.06	79.07	50.04	50.54	50.05	50.55	50.01	83.92
	(0.10)	(0.08)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
1.5	78.05	78.21	50.06	50.26	50.05	50.18	67.96	87.24
	(0.09)	(0.05)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
2.5	76.56	76.35	50.08	50.20	50.06	50.14	72.86	88.39
	(0.09)	(0.04)	(0.02)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
3.5	78.45	72.88	50.10	50.17	50.05	50.05	76.13	89.31
	(0.04)	(0.05)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)

Table 2: C-index for 100 simulated test datasets in scenario 2 with complex nonlinear fixed effects (standard deviations in brackets). Abbreviations: Prop. (Proposed model), DS (DeepSurv), CF-Lin (Cox frailty with linear effects), CF-Int (Cox frailty with interactions), Cox (Cox proportional hazards model), Cox-Int (Cox model with interactions), Cox-FC (Cox model with fixed clustering effects), TM (True model).

Frailty Variance	Prop.	DS	CF-Lin	CF-Int	Cox	Cox-Int	Cox-FC	TM
0	64.42	66.90	73.82	71.89	73.82	71.93	70.66	77.29
	(0.03)	(0.03)	(0.24)	(0.30)	(0.24)	(0.40)	(0.25)	(0.39)
1.5	71.03	62.96	68.07	69.11	68.51	68.36	67.28	79.12
	(0.05)	(0.03)	(0.25)	(0.27)	(0.25)	(0.30)	(0.27)	(0.38)
2.5	74.43	60.85	69.67	63.92	70.61	67.47	62.75	80.99
	(0.05)	(0.04)	(0.24)	(0.32)	(0.24)	(0.30)	(0.30)	(0.37)
3.5	74.67	59.61	69.35	63.78	70.77	67.35	62.39	75.98
	(0.08)	(0.07)	(0.24)	(0.30)	(0.24)	(0.30)	(0.30)	(0.41)

These settings contribute to reliable training performance while maintaining computational efficiency.

Under both scenarios, our method yields the best AUC among all the methods when the random effect variance is large, e.g., $\theta = 2.5$ or 3.5. This demonstrates the advantage of our model in capturing the heterogeneity across clusters and characterizing the complex nonlinear and interactive covariate effects simultaneously. Specifically, the distinction between our proposed method and Deepsurv grows more pronounced as the value of θ escalates, primarily because our method integrates a frailty term to account for cluster effects. In Tables 1 and 2, we have reported the standard deviations for the C-index. The proposed method and the DeepSurv method exhibit consistently low standard deviations under both scenarios, demonstrating the stable performance of these deep learning approaches.

We have included the average computational time for each method in the simulation studies to provide insights into computational efficiency for potential users. As shown in Table 3, the

Table 3: Average computational time in seconds for simulation studies. Abbreviations: Prop. (Proposed model), DS (DeepSurv), CF-Lin (Cox frailty with linear effects), CF-Int (Cox frailty with interactions), Cox (Cox proportional hazards model), Cox-Int (Cox model with interactions), Cox-FC (Cox model with fixed clustering effect), TM (True model).

Prop.	DS	CF-Lin	CF-Int	Cox	Cox-Int	Cox-FC	TM
3.05	3.67	0.44	3.90	0.28	0.53	2.69	0.39

proposed method and DeepSurv require more time compared to standard Cox models, but their computational times are similar to the Cox model with fixed clustering effects and are less than the Cox frailty model with interactions. This result demonstrates that while the proposed and DeepSurv methods involve greater computational costs than simpler Cox models, they are comparable to more complex Cox models with frailty terms.

4 Kidney Transplant Data Analysis

We compare the accuracy of the proposed method with six other competing methods in predicting the time-to-graft-failure or death after kidney transplantation using the national kidney transplant registry data obtained from U.S. Organ Procurement and Transplantation Network (https://optn.transplant.hrsa.gov/data/). OPTN aims to improve the U.S. donation and transplantation system so that more life-saving organs are available for transplant. Following Liu et al. (2023), our study focuses on a cohort of 8,378 adult individuals (those aged 18 or older) who underwent kidney transplant between January 1st, 2007, and December 31st, 2007. These individuals were treated at 154 medical facilities, with the number of patients treated in each facility ranging from 20 to 205. Out of the 8,378 patients, 2,280 encountered either death or graft failure after the kidney transplant. The remaining patients were censored after a five-year post-transplant follow up, with a censoring rate of 72.78%.

In the analysis, we include 15 baseline factors: time on end-stage renal disease (ESRD), donor age, donor gender (male = 1, female = 0), donor body mass index (BMI), donor race (reference: white), donor history of hypertension (yes = 1, no = 0), donor meeting expanded criteria (yes = 1, no = 0), recipient gender (male = 1, female = 0), recipient race (reference: white), recipient insulin-dependent diabetes (yes = 1, no = 0), recipient non-insulin dependent diabetes (yes = 1, no = 0), recipient age at transplant, recipient BMI, whether the recipient received a previous kidney transplant (yes = 1, no = 0), recipient total cold ischemia time.

We apply our proposed method to predict the post-transplant graft failure or death for patients within each facility, which is regarded as a cluster. The goal of our analysis is to use the time-to-graft-failure or death of subjects in the training dataset to predict that of the subjects in the test dataset from the same facility. The ReLU activation function is selected for its faster convergence rate and better performance (Maas et al. (2013)).

Table 4 reports the C-indexes in five-fold cross-validation (CV) for performance comparison. Within each facility, we randomly assign 80% of its patients as the training set and the remaining 20% as the testing set. The proposed method has the highest average C-index among all the methods, indicating the advantages of incorporating non-linearity, interaction, and clustering effects in the risk function. Accurate predictions within the same facility are pivotal for identifying patients with a high risk of graft failure or subsequent death post-

Table 4: C-index on the kidney transplant data. Abbreviations: Prop. (Proposed model), DS (DeepSurv), CF-Lin (Cox frailty with linear effects), CF-Int (Cox frailty with interactions), Cox (Cox proportional hazards model), Cox-Int (Cox model with interactions), Cox-FC (Cox model with fixed clustering effects).

Prop.	DS	CF-Lin	CF-Int	Cox	Cox-Int	Cox-FC
59.24	55.18	53.46	55.22	53.48	55.23	53.46

transplantation. This identification aids in averting excessive treatments or suboptimal resource distribution.

To assess the calibration ability of the models, we also obtain the time-dependent Brier score (Gerds and Schumacher (2006), Sun et al. (2020)). The time-dependent Brier score is an extension of the Brier score, which takes into account the predicted survival probabilities at different time points and compares them to the actual survival probabilities over time. The time-dependent Brier score measures the mean square error between the observed status $Y_i(t) = I(U_i > t)$ and the predicted survival probability $S(t|X_i, Z_i)$ for subject i at time t. The Brier score, ranging between 0 and 1, reflects the accuracy of probabilistic predictions. A score of 0 signifies perfect prediction, where the predicted probabilities align precisely with the actual outcomes. A lower Brier score indicates enhanced calibration performance and greater accuracy of the model's probabilistic predictions at a given time point.

We estimate the time-dependent Brier score on the test dataset by the inverse probability weighting method (Sun et al. (2020)):

$$\widehat{BS}(t, \hat{S}) = \frac{1}{M} \sum_{i \in D_M} \hat{W}_i(t) \left\{ Y_i(t) - \hat{S}\left(t \mid X_i, Z_i\right) \right\}^2,$$

where D_M is the test dataset with size M, and $\hat{W}_i(t) = \frac{(1-Y_i(t))\delta_i}{\hat{G}(Y_i-)} + \frac{Y_i(t)}{\hat{G}(t)}$ is the inverse probability of censoring weights with $\hat{G}(t) = \hat{P}(C > t)$.

Similar to Sun et al. (2020), we report time-dependent Brier scores at different time points. Figure 2 presents the average time-dependent Brier scores on the five-fold cross-validation datasets under each prediction model. As shown in Figure 2, the Brier scores from our model are consistently lower than all the other models across all time points, indicating its superior performance in both discrimination and calibration. For a comprehensive view of the Brier scores across all seven methods at various time points, refer to Table 6 in the Appendix.

5 Discussion

We propose a neural network for correlated survival outcomes. The proposed method extends the classical neural network framework to include a random effect (frailty) accounting for within-cluster correlation. The model uses a feed-forward neural network for nonlinear and interactive fixed effects and estimates random effects in the last layer of the neural network to avoid iterative computation. The neural network is trained over a loss function derived from the penalized partial likelihood with a Laplace approximation. Through both simulation studies and real-world data evaluations, the advantages of our method are evident over conventional survival

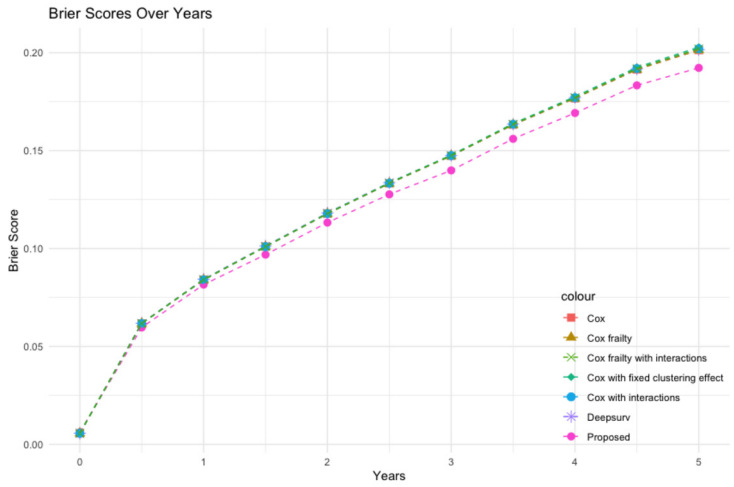

Figure 2: The average time-dependent Brier scores for the cross-validation datasets from seven prediction models (Proposed, Deepsurv, Cox frailty, Cox frailty with interactions, Cox, Cox with interactions, Cox with fixed clustering effects).

regression techniques and Deepsurv. In summary, our method stands out as an effective instrument for predicting correlated survival outcomes, particularly in modeling intricate covariate impacts.

There are several future directions in this research. First, while this study focuses on correlated survival outcomes in clusters, exploring methods for recurrent event data represents a captivating progression. This would delve deeper into another aspect of correlated survival outcomes (Cook et al. (2007)). Second, it would be of interest to develop deep learning prediction methods for correlated survival outcomes using e.g., the additive hazards model (Aalen (1989)) or the linear transformation model (Fine et al. (1998)). Third, this paper primarily focuses on predicting correlated survival endpoints, treating the random effects as nuisance parameters. In contrast, in provider profiling, the individual effect b_i is of central importance (Normand and Shahian (2007); He et al. (2013); Liu et al. (2023)). Extending our method to address provider profiling represents a promising direction for future research. Finally, we only consider time-independent covariates (at baseline) for predicting time to event. It is of importance to consider longitudinal biomarkers for dynamic prediction in the joint model and landmark model frameworks (Rizopoulos et al. (2017); Tanner et al. (2021); Lin and Luo (2022)).

Supplementary Material

To enhance reproducibility, we make all computer programs and sample data used for implementations available on https://github.com/rivenzhou/deep_learning_clustered. Restrictions apply to the availability of the real data, which were used under license for this study.

A Appendix

Table 5: Hyperparameter search space for simulation study and real data analysis.

Hyperparameter	Values
Layers	{1, 2, 3}
Nodes per layer	{16, 32, 64, 128}
Weight decay	{0.001, 0.01, 0.1, 0.2, 0.4}
Batch size	{64, 128, 256}
Dropout rate	{0.1, 0.2}
Epoch	{100, 200}

Table 6: Time-dependent Brier scores on the kidney transplant data at years 1, 2, 3, 4, and 5 after the kidney transplantation. Abbreviations: Prop. (Proposed model), DS (DeepSurv), CF-Lin (Cox frailty with linear effects), CF-Int (Cox frailty with interactions), Cox (Cox proportional hazards model), Cox-Int (Cox model with interactions), Cox-FC (Cox model with fixed clustering effects).

Time	Prop.	DS	CF-Lin	CF-Int	Cox	Cox-Int	Cox-FC
1	0.0815	0.0843	0.0840	0.0841	0.0840	0.0840	0.0841
2	0.1132	0.1181	0.1178	0.1178	0.1177	0.1178	0.1180
3	0.1398	0.1475	0.1473	0.1475	0.1473	0.1475	0.1477
4	0.1691	0.1770	0.1770	0.1770	0.1767	0.1770	0.1775
5	0.1922	0.2015	0.2013	0.2020	0.2013	0.2020	0.2028

Funding

This research is partly supported by NIH grants R21 EY031884, R21 EY033518, UL1 TR002345, R01 DK129539.

References

Aalen OO (1989). A linear regression model for the analysis of life times. *Statistics in Medicine*, 8(8): 907–925. https://doi.org/10.1002/sim.4780080803

Balan TA, Putter H (2020). A tutorial on frailty models. *Statistical Methods in Medical Research*, 29(11): 3424–3454. https://doi.org/10.1177/0962280220921889

Ching T, Zhu X, Garmire LX (2018). Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Computational Biology*, 14(4): e1006076. https://doi.org/10.1371/journal.pcbi.1006076

Cook RJ, Lawless JF, et al. (2007). *The Statistical Analysis of Recurrent Events*. Springer.

Fan J, Ma C, Zhong Y (2021). A selective overview of deep learning. *Statistical Science: A Review Journal of the Institute of Mathematical Statistics*, 36(2): 264.

Faraggi D, Simon R (1995). A neural network model for survival data. *Statistics in Medicine*, 14(1): 73–82. https://doi.org/10.1002/sim.4780140108

Fine JP, Ying Z, Wei L (1998). On the linear transformation model for censored data. *Biometrika*, 85(4): 980–986. https://doi.org/10.1093/biomet/85.4.980

Gerds TA, Schumacher M (2006). Consistent estimation of the expected Brier score in general survival models with right-censored event times. *Biometrical Journal*, 48(6): 1029–1040. https://doi.org/10.1002/bimj.200610301

Glidden DV, Vittinghoff E (2004). Modelling clustered survival data from multicentre clinical trials. *Statistics in Medicine*, 23(3): 369–388. https://doi.org/10.1002/sim.1599

Hao J, Kim Y, Mallavarapu T, Oh JH, Kang M (2018). Cox-pasnet: Pathway-based sparse deep neural network for survival analysis. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (H. Zheng, X. Hu, Z. Callejas, H. Schmidt, D. Griol, J. Baumbach, J. Dickerson, and L. Zhang, editors), 381–386. IEEE.

Harrell Jr FE, Lee KL, Califf RM, Pryor DB, Rosati RA (1984). Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2): 143–152. https://doi.org/10.1002/sim.4780030207

He K, Kalbfleisch JD, Li Y, Li Y (2013). Evaluating hospital readmission rates in dialysis facilities; adjusting for hospital effects. *Lifetime Data Analysis*, 19: 490–512. https://doi.org/10.1007/s10985-013-9264-6

Hens N, Wienke A, Aerts M, Molenberghs G (2009). The correlated and shared gamma frailty model for bivariate current status data: An illustration for cross-sectional serological data. *Statistics in Medicine*, 28(22): 2785–2800. https://doi.org/10.1002/sim.3660

Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y (2018). Deepsurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1): 1–12. https://doi.org/10.1186/s12874-017-0458-6

Kvamme H, Borgan Ø, Scheel I (2019). Time-to-event prediction with neural networks and Cox regression. *Journal of Machine Learning Research*, 20(129): 1–30.

Lee H, Ha I, Lee Y (2023). Deep neural networks for semiparametric frailty models via h-likelihood. arXiv preprint: https://arxiv.org/2307.06581/.

Liao L, Ahn Hi (2016). Combining deep learning and survival analysis for asset health management. *International Journal of Prognostics and Health Management*, 7(4), 1–10.

Lin J, Luo S (2022). Deep learning for the dynamic prediction of multivariate longitudinal and survival data. *Statistics in Medicine*, 41(15): 2894–2907. https://doi.org/10.1002/sim.9392

Lin X, Taylor JM, Ye W (2008). A penalized likelihood approach to joint modeling of longitudinal measurements and time-to-event data. *Statistics and its Interface*, 1(1): 33–45. https://doi.org/10.4310/SII.2008.v1.n1.a4

Liu L, He K, Wang D, Ma S, Qu A, Lin L, et al. (2023). Healthcare center clustering for Cox's proportional hazards model by fusion penalty. *Statistics in Medicine*, 42(20), 3685–3698.

Maas AL, Hannun AY, Ng AY, et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In: *Proc. Icml, Volume 30, Atlanta, GA* (S. Dasgupta, D. McAllester, editors), 3.

Mandel F, Ghosh RP, Barnett I (2023). Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics*, 79(2): 711–721. https://doi.org/10.1111/biom.13615

Martinsson E (2017). *Wtte-rnn: Weibull time to event recurrent neural network a model for sequential prediction of time-to-event in the case of discrete or continuous censored data, recurrent events or time-varying covariates*. Doctoral dissertation, Chalmers University of Technology and University of Gothenburg.

Normand SLT, Shahian DM (2007). Statistical and clinical aspects of hospital outcomes profiling.

Statistical Science, 22(2), 206–226.

Paik MC, Tsai WY, Ottman R (1994). Multivariate survival analysis using piecewise gamma frailty. *Biometrics*, 50(4), 975–988. https://doi.org/10.2307/2533437

Ranganath R, Perotte A, Elhadad N, Blei D (2016). Deep survival analysis. In: *Machine Learning for Healthcare Conference* (F. Doshi-Velez, J. Fackler, D. Kale, B. Wallace, and J. Wiens, editors), 101–114. PMLR.

Ripatti S, Palmgren J (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, 56(4): 1016–1022. https://doi.org/10.1111/j.0006-341X.2000.01016.x

Rizopoulos D, Molenberghs G, Lesaffre EM (2017). Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. *Biometrical Journal*, 59(6): 1261–1276. https://doi.org/10.1002/bimj.201600238

Shih JH, Louis TA (1995). Assessing gamma frailty models for clustered failure time data. *Lifetime Data Analysis*, 1: 205–220. https://doi.org/10.1007/BF00985771

Sun T, Wei Y, Chen W, Ding Y (2020). Genome-wide association study-based deep learning for survival prediction. *Statistics in Medicine*, 39(30): 4605–4620. https://doi.org/10.1002/sim.8743

Tanner KT, Sharples LD, Daniel RM, Keogh RH (2021). Dynamic survival prediction combining landmarking with a machine learning ensemble: Methodology and empirical comparison. *Journal of the Royal Statistical Society. Series A. Statistics in Society*, 184(1): 3–30. https://doi.org/10.1111/rssa.12611

Wiegrebe S, Kopper P, Sonabend R, Bender A (2023). Deep learning for survival analysis: A review. arXiv preprint: https://arxiv.org/2305.14961.

Wu R, Qiao J, Wu M, Yu W, Zheng M, Liu T, et al. (2024). Neural frailty machine: Beyond proportional hazard assumption in neural survival regressions. In: *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors), 36.

Yu Z, Liu L (2011). A joint model of recurrent events and a terminal event with a nonparametric covariate function. *Statistics in Medicine*, 30(22): 2683–2695. https://doi.org/10.1002/sim.4297

Yu Z, Liu L, Bravata DM, Williams LS (2014). Joint model of recurrent events and a terminal event with time-varying coefficients. *Biometrical Journal*, 56(2): 183–197. https://doi.org/10.1002/bimj.201200160

Yu Z, Liu L, Bravata DM, Williams LS, Tepper RS (2013). A semiparametric recurrent events model with time-varying coefficients. *Statistics in Medicine*, 32(6): 1016–1026. https://doi.org/10.1002/sim.5575

Zhong Q, Mueller JW, Wang JL (2021). Deep extended hazard models for survival analysis. *Advances in Neural Information Processing Systems*, 34: 15111–15124.

Zhong Q, Wang JL (2023). Neural networks for partially linear quantile regression. *Journal of Business & Economic Statistics*, 42(2), 603–614.