Deep Survival: A Deep Cox Proportional Hazards Network

Jared L. Katzman^{1}, Uri Shaham^{2}, Alexander Cloninger^{3}, Jonathan Bates^{3,4,5}, Tingting Jiang^{6}, and Yuval Kluger^{3,6,7}

[1]Department of Computer Science, Yale University, 51 Prospect Steet, New Haven, CT 06511, USA
[2]Department of Statistics, Yale University, 24 Hillhouse Avenue, New Haven, CT 06511, USA
[3]Applied Mathematics Program, Yale University, 51 Prospect Steet, New Haven, CT 06511, USA
[4]Yale School of Medicine, 333 Cedar Street, New Haven CT 06510, USA
[5]Center for Outcomes Research and Evaluation, Yale-New Haven Hospital, New Haven, CT
[6]Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06511, USA
[7]Department of Pathology and Yale Cancer Center, Yale University School of Medicine, New Haven, CT, USA

Abstract

Previous research has shown that neural networks can model survival data in situations in which some patients' death times are unknown, e.g. right-censored. However, neural networks have rarely been shown to outperform their linear counterparts such as the Cox proportional hazards model. In this paper, we run simulated experiments and use real survival data to build upon the risk-regression architecture proposed by Faraggi and Simon. We demonstrate that our model, DeepSurv, not only works as well as the standard linear Cox proportional hazards model but actually outperforms it in predictive ability on survival data with linear and nonlinear risk functions. We then show that the neural network can also serve as a recommender system by including a categorical variable representing a treatment group. This can be used to provide personalized treatment recommendations based on an individual's calculated risk. We provide an open source Python module that implements these methods in order to advance research on deep learning and survival analysis.

1 Introduction

Medical researchers use survival models to evaluate the significance of prognostic variables in outcomes such as death or cancer recurrence and subsequently inform patients of their treatment options [1, 2, 3, 4]. One standard survival model is the Cox proportional hazards model (CPH) [5], a semiparametric model that calculates the effects of observed covariates on the risk of an event occurring (henceforth defined as 'death'). The CPH assumes that a patient's risk of death is a linear combination of their covariates. This assumption is referred to as the *linear proportional hazards* condition. In many real world datasets, the assumption that the risk function is linear may be too simplistic. As such, a richer family of survival models is needed to better fit survival data with nonlinear risk functions. Since neural networks (NNs) can learn highly complex and nonlinear functions, researchers have attempted to use NNs to model the nonlinear proportional hazards of real survival datasets. However, studies have demonstrated mixed results, for example, see [6] and [7]. To the best of our knowledge, NNs have not outperformed standard methods for survival analysis such as the CPH.

There are three main approaches in the field of neural networks and survival analysis. These include variants of: (i) classification methods [see details in 8, 9], (ii) time-encoded methods [see details in 10, 11], (iii) and the Faraggi-Simon network [12], which implements a feed-forward neural network that estimates an individual's risk of death. The Faraggi-Simon network is seen as a nonlinear extension of the Cox proportional hazards model. Researchers have attempted to apply the Faraggi-Simon network with various extensions. However, perhaps because the practice of NNs was not as developed as it is today, they failed to demonstrate improvements beyond the linear Cox model, see [7] and [13].

An advantage of the Faraggi-Simon network is its ability to provide prognosis based on multiple prognostic features without prior selection. However, Schwarzder et al. [14] and others have raised concerns about using NNs in prognostic applications due to their tendency to overfit implausible biological functions. Therefore, further validation is needed to evaluate the prognostic abilities of the Faraggi-Simon network.

The goals of this paper are: (i) to show that the application of deep learning to survival analysis often outperforms the standard CPH; and (ii) to demonstrate how the deep neural network can be viewed as a personalized treatment recommender system and a useful framework for medical applications.

We propose a modern deep learning generalization of the Faraggi-Simon network, henceforth referred to as DeepSurv. We make the following contributions. First, we show that DeepSurv outperforms the CPH on survival data with both linear and nonlinear risk functions. Second, we include an additional categorical variable representing a patient's treatment group to illustrate how to view the network as a treatment recommender system. This, in turn, provides personalized treatments tailored to a patient's observed features. Our experimental results demonstrate that the network accurately models the risk function of the population. We validate our results on real survival data, which further demonstrates the power of the DeepSurv model. Additionally, we show that the recommender system can guide us in making decisions on personalized treatment recommendations and can potentially increase the median survival time for a set of breast cancer patients.

The organization of the manuscript is as follows: in Section 2, we provide a brief background on survival analysis. In Section 3, we present our contributions and explain the implementation of DeepSurv and our proposed recommender system. In Section 4, we describe the experimental design and results. Section 5 concludes the manuscript.

2 Background

In this section, we define survival data and the approaches for modeling a population's survival and death rates. Additionally, we discuss linear and nonlinear survival models and their limitations.

2.1 Survival Data

Survival data is comprised of three elements: baseline data x, an event time T, and an event indicator E. If an event (e.g. death) is observed, the time interval T corresponds to the time elapsed between the time in which the baseline data was collected and the time of the event occurring, and the event indicator is $E = 1$. If an event is not observed, the time interval T corresponds to the time elapsed between the collection of the baseline data and the last contact with the patient (e.g. end of study), and the event indicator is $E = 0$. In this case, the patient is said to be *right-censored*. Modeling right-censored survival data requires special consideration; if one opts to use standard regression methods, the right-censored data must be discarded.

Survival and hazard functions are the two fundamental functions in survival analysis. The survival function is denoted by $S(t) = \Pr(T > t)$, which signifies the probability that an individual has 'survived' up to time t. The hazard function corresponds to the probability that an individual dies at time t given that he or she has survived up to that point. The hazard function $\lambda(t)$ is defined as:

$$\lambda(t) = \lim_{\delta \to 0} \frac{\Pr(t \leq T < t + \delta \mid T \geq t)}{\delta}. \tag{1}$$

The hazard function is a measure of risk at time t. A greater hazard signifies a greater risk of failure.

A proportional hazards model is a common method for modeling an individual's survival given their baseline data x. The model assumes that the hazard function is composed of two functions: a baseline hazard function, $\lambda_0(t)$, and a risk function, $h(x)$, denoting the effects of an individual's covariates. The hazard function is assumed to have the form $\lambda(t|x) = \lambda_0(t) \cdot e^{h(x)}$.

2.2 Linear Survival Models

The CPH is a proportional hazards model that estimates the risk function $h(x)$ by a linear function $\hat{h}_\beta(x) = \beta^T x$. To perform Cox regression, one tunes the weights β to optimize the Cox partial likelihood. The partial likelihood is the product of the probability at each event time T_i that the event has occurred to individual i, given the set of individuals still at risk at time T_i. The Cox partial likelihood is parameterized by β and defined as

$$L_c(\beta) = \prod_{i:E_i=1} \frac{\exp(\hat{h}_\beta(x_i))}{\sum\limits_{j \in \Re(T_i)} \exp(\hat{h}_\beta(x_j))}, \tag{2}$$

where the values T_i, E_i, and x_i are the respective event time, event indicator, and baseline data for the i^{th} observation. The risk set $\Re(t) = \{i : T_i \geq t\}$ is the set of patients still at risk of death at time t.

The CPH model may be too simplistic for fitting real world datasets since they commonly do not satisfy the linear proportional hazards condition. In this case, a more complex set of features that includes higher-level interaction terms between covariates is needed.

2.3 Nonlinear Survival Models

The Faraggi-Simon method [12] is a feed-forward neural network that provides the basis for a nonlinear proportional hazards model. They experimented with a single hidden layer network with two or three nodes. Their model requires no prior assumption of the risk function $h(x)$ other than continuity. Instead, the NN computes nonlinear features from the training data and calculates their linear combination to estimate the risk function. Similar to Cox regression, Faraggi and Simon optimize a modified Cox partial likelihood. They replace the linear combination of features $\hat{h}_\beta(x)$ in Equation 2 with the output of the network $\hat{h}_\theta(x)$.

As previous research suggests, the Faraggi-Simon network has not been shown to outperform the linear CPH [12, 7, 13]. This could be due to the fact that neural network practice at the time was not mature. Furthermore, to the best of our knowledge, there have been no attempts at applying modern deep learning techniques to the Faraggi-Simon network.

3 Deep Survival

In this section, we describe our contributions: first, we leverage recent deep learning techniques unavailable to Faraggi and Simon. We improve upon their network by adding additional hidden layers to the network such that the covariates of the uppermost hidden layer of the deep network are used as an input to a CPH model. Second, we introduce an open source Python module that provides an interface to train DeepSurv. Third, we demonstrate how to view the network as a prognostic model and how the network's predicted risk function can provide personalized treatment recommendations.

3.1 DeepSurv

DeepSurv is a multi-layer perceptron similar to the Faraggi-Simon network. However, we allow a deep architecture (i.e., more than one hidden layer) and apply modern techniques such as weight decay regularization, Rectified Linear Units (ReLU) [15], Batch Normalization [16], dropout [17], stochastic gradient descent with Nesterov momentum [18], gradient clipping [19], and learning rate scheduling [20]. The output of the network is a single node, which estimates the risk function $\hat{h}_\theta(x)$ parameterized by the weights of the network θ. We set the loss function to be the negative log partial likelihood of Equation 2:

$$l(\theta) := -\sum_{i:E_i=1} \Big(\hat{h}_\theta(x_i) - \log \sum_{j \in \Re(T_i)} e^{\hat{h}_\theta(x_j)} \Big). \tag{3}$$

In addition, we perform a Random hyper-parameter optimization search [21]; see Section 4.4 for more details.

3.2 Treatment Recommender System

It is a common practice in medical applications to determine the relationship between a patient's observable covariates and his or her risk of an event [13, 3, 22]. Survival models based on NNs are rarely used in clinical research because NNs tend to overfit implausible biological functions [14]. However, our results show that DeepSurv is able to accurately generalize biologically significant relationships between a patient's covariates and his or her risk of death. As a result, the network is able to provide guidance to physicians in terms of personalized treatment recommendations.

Let all patients in a given study be assigned to one of n treatment groups $\tau \in \{0, 1, ..., n-1\}$. We assume each treatment i to have an independent risk function $h_i(x)$. Collectively, the hazard function becomes:

$$\lambda(t; x|\tau = i) = \lambda_0(t) \cdot e^{h_i(x)}. \tag{4}$$

For any patient, the network should be able to accurately predict the risk of being prescribed a given treatment. In addition, the network can compare the risk of undergoing any two treatments. For example, if we pass a patient through the network once in treatment group i and again in treatment group j, we can take the log of their hazards ratio to calculate the personal risk of prescribing one treatment option over another:

$$\text{rec}_{ij}(x) = \log\left(\frac{\lambda(t; x|\tau = i)}{\lambda(t; x|\tau = j)}\right) = \log\left(\frac{\lambda_0(t) \cdot e^{h_i(x)}}{\lambda_0(t) \cdot e^{h_j(x)}}\right) = h_i(x) - h_j(x). \tag{5}$$

We define this difference of log hazards as the *recommender* function or $\text{rec}_{ij}(x)$. In practice, when a patient receives a positive recommendation $\text{rec}_{ij}(x)$, treatment j is more effective than treatment i and leads to a lower risk of death. Conversely, a negative recommendation indicates that treatment j leads to a higher risk of death than treatment i. Hence, the patient should be prescribed treatment i.

DeepSurv is a tool for researchers to easily investigate more complex interactions between covariates and treatment success. DeepSurv's architecture has an advantage over the CPH because it does not require an *a priori* specification of treatment interaction terms, but learns these adaptively. In contrast, the CPH model is only able to provide personalized treatment recommendations when treatment interaction terms are added to the model. This requires extensive experimentation or prior biological knowledge of treatment outcomes.

4 Experiments

We perform three sets of experiments on: (i) simulated survival data, (ii) real survival data, and (iii) clinical treatment data. For the first set of experiments, we simulate both a linear and nonlinear risk function and show DeepSurv's superior modeling capabilities. Then, we train DeepSurv on real survival data and demonstrate the network's improved predictive ability. In addition, we verify that the network can model multiple risk functions within a population. Lastly, we demonstrate how DeepSurv's treatment recommendations can improve a population's survival rate.

To evaluate the predictive accuracy of DeepSurv, we measure the concordance-index (C-index) c as outlined by Harrell et al. [23]. For all possible pairs of patients with comparable event times (a non-comparable event, for example, is two censored patients or one patient who is censored before another's death time), a pair is *concordant* with the true outcomes if the patient with a higher predicted risk dies first. The C-index is the ratio of the number of concordant predictions and the set of all possible pairs. For context, a $c = 0.5$ is the average C-index of a random model whereas $c = 1$ is a perfect ranking of event times. We perform bootstrapping [24] and sample the test set with replacement to obtain confidence intervals. We report the confidence intervals (CI) of the C-indices of the bootstrapped samples for each model.

4.1 Simulated Survival Data

In this section, we perform two experiments with simulated survival data: one with a linear risk function and one with a nonlinear (Gaussian) risk function. In addition to training DeepSurv on each dataset, we run a linear CPH regression for a baseline comparison. The advantage of the simulated datasets is that we can ascertain whether DeepSurv or CPH successfully models the true risk function.

For each experiment, we generate a training, validation, and testing set of $N = 10000$ observations, such that an observation represents a patient vector with $d = 10$ covariates, each drawn from a uniform distribution on $[-1, 1)$. We generate the death time T according to an exponential Cox model [25]:

$$T \sim \text{Exp}(\lambda(t;x)) = \text{Exp}(\lambda_0 \cdot e^{-h(x)}) \tag{6}$$

In both experiments, the risk function $h(x)$ only depends on two of the ten covariates, and we demonstrate that DeepSurv is able to discern the relevant covariates from the noise. We then choose a censoring time to represent the 'end of study,' such that an average of 30-40 percent of the patients have an observed event in the dataset.

4.1.1 Linear Risk Experiment

We first simulate patients to have a linear risk function for $x \in \mathbb{R}^d$ so that the linear proportional hazards assumption holds true:

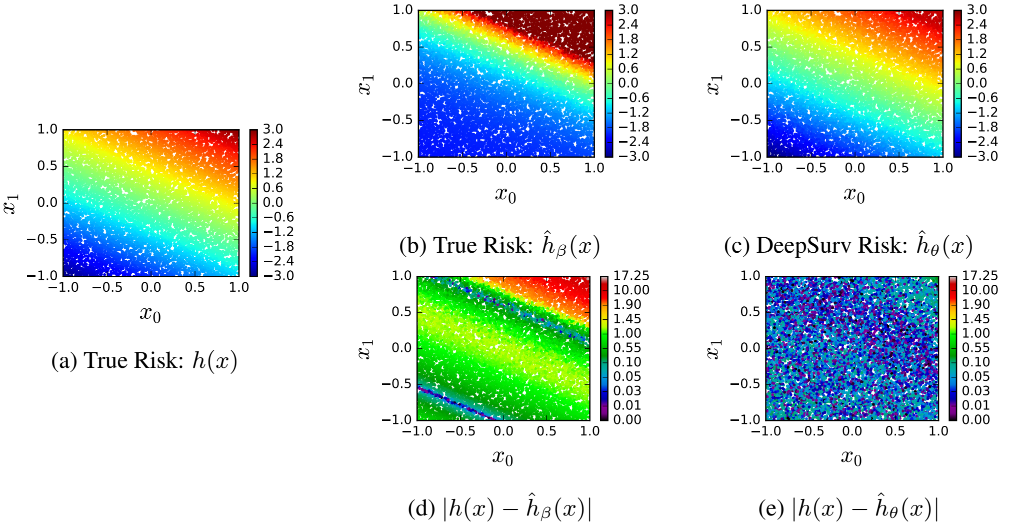
$$h(x) = x_0 + 2x_1. \tag{7}$$

Because the linear proportional hazards assumption holds true, we expect the linear CPH to accurately model the risk function in Equation 7.

Our results demonstrate that DeepSurv performs as well as the standard linear Cox regression in predictive ability. However, DeepSurv reconstructs the true risk function for all patients more accurately than the linear CPH.

The C-index of the linear CPH model is 0.79277 (95% CI: 0.79255 - 0.79298), and the linear CPH estimates the correct coefficients of x_0 and x_1 within ± 0.1 accuracy. The C-index of DeepSurv is 0.79272 (95% CI: 0.79251 - 0.79293), and DeepSurv reconstructs the surface of the risk function $h(x)$ within ± 0.1 accuracy. With comparable C-indices and successful coefficient estimations, DeepSurv and CPH have equivalent predictive performance.

Figure 1 demonstrates how DeepSurv more accurately models the risk function compared to the linear CPH. Figure 1(a) plots the true risk function $h(x)$ for all patients in the test set. As shown in Figure 1(b), the CPH's estimated risk function $\hat{h}_\beta(x)$ does not perfectly model the true risk for a patient. In contrast, as shown in Figure 1(c), DeepSurv estimates the true risk function. As depicted in Figures 1(d) and 1(e), the CPH's estimated risk has a significantly larger error than that of DeepSurv, especially for patients with a high positive risk. To quantify these differences, we calculate the



(a) True Risk: $h(x)$

(b) True Risk: $\hat{h}_\beta(x)$

(c) DeepSurv Risk: $\hat{h}_\theta(x)$

(d) $|h(x) - \hat{h}_\beta(x)|$

(e) $|h(x) - \hat{h}_\theta(x)|$

Figure 1: Predicted risk surfaces and errors for the simulated survival data with linear risk function with respect to a patient's covariates x_0 and x_1. 1(a) The true risk $h(x) = x_0 + 2x_1$ for each patient. 1(b) The predicted risk surface of $\hat{h}_\beta(x)$ from the linear CPH model parameterized by β. 1(c) The output of DeepSurv $\hat{h}_\theta(x)$ predicts a patient's risk. 1(d) The absolute error between true risk $h(x)$ and CPH's predicted risk $\hat{h}_\beta(x)$. 1(e) The absolute error between true risk $h(x)$ and DeepSurv's predicted risk $\hat{h}_\theta(x)$.

mean-squared-error (MSE) between a model's predicted risk and the true risk values. The MSEs of the linear CPH and DeepSurv are 4.264 and 0.003, respectively. This demonstrates the superior modeling capabilities of DeepSurv.

4.1.2 Nonlinear Risk Experiment

We set the risk function to be a Gaussian with $\lambda_{\max} = 5.0$ and a scale factor of $r = 0.5$:

$$h(x) = \log(\lambda_{\max})\ \exp\left(-\frac{x_0^2 + x_1^2}{2r^2}\right) \tag{8}$$

The surface of the risk function is depicted in 2(a). Because this risk function is nonlinear, we do not expect the CPH to predict the risk function properly without adding quadratic terms of the covariates to the model. We expect DeepSurv to be successful in reconstructing the Gaussian risk function and predicting a patient's risk.

As shown in Figure 2, DeepSurv is more successful than the linear CPH in modeling the true risk function. Figure 2(b) demonstrates that the linear CPH regression fails to determine the first two covariates as significant. The CPH has a C-index of 0.490 (95% CI: 0.490 - 0.491), which is equivalent to the performance of randomly ranking death times. DeepSurv has a higher predictive accuracy of 0.612 (95% CI: 0.611 - 0.612). Furthermore, Figure 2(c) shows that DeepSurv reconstructs the Gaussian relationship between the first two covariates and a patient's risk. DeepSurv clearly outperforms the linear CPH in predictive ability and is able to learn nonlinear relationships between a patient's covariates and their risk.

4.2 Real Survival Data Experiments

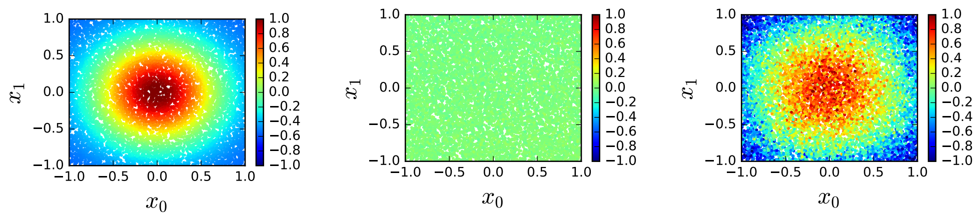
We compare the performance of the CPH and DeepSurv on two datasets from real clinical studies: the Worcester Heart Attack Study (WHAS) [26] and the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) [27]. Our goal is to demonstrate that DeepSurv has superior predictive ability in medical application and practice compared to the linear CPH.

4.2.1 Worcester Heart Attack Study (WHAS)

The Worcester Heart Attack Study (WHAS) investigates the effects of a patient's factors on acute myocardial infraction (MI) survival [28]. The dataset consists of 1,638 observations and 5 features: age, sex, body-mass-index (BMI), left heart failure complications (CHF), and order of MI (MIORD). A total of 42.12 percent of patients died during the survey with a median death time of 516.0 days.

DeepSurv outperforms the CPH even when the linear Cox regression successfully models the relationships between covariates and survival rate. DeepSurv has a C-index of 0.779 (95% CI: 0.777 - 0.782) on the test set. The C-index of the linear CPH is 0.669 (95% CI: 0.666 - 0.671).

To further explore the advantages of DeepSurv, we rerun the network on a reduced dataset consisting of the four factors (age, BMI, CHF, and MIORD) that the CPH found significant (p-value < 10^{-6}). We find that eliminating sex as an input feature decreases the C-index of DeepSurv to 0.748 (95%



(a) True Risk: $h(x)$ (b) CPH Risk: $\hat{h}_\beta(x)$ (c) DeepSurv Risk: $\hat{h}_\theta(x)$

Figure 2: Risk surfaces of the nonlinear test set with respect to patient's covariates x_0 and x_1. 2(a) The calculated true risk $h(x)$ (Equation 8) for each patient. 2(b) The predicted risk surface of $\hat{h}_\beta(x)$ from the linear CPH model parameterized on β. The linear CPH predicts a constant risk. 2(c) The output of DeepSurv $\hat{h}_\theta(x)$ is the estimated risk function.

CI: 0.745 - 0.750). This signifies that DeepSurv found sex to be significant in the calculation of a patient's risk. This result is expected, as Vaccarino et al. [29] have shown strong evidence that the interaction between age and sex affect MI survival.

4.2.2 Molecular Taxonomy of Breast Cancer International Consortium (METABRIC)

The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) uses gene and protein expression profiles to determine new breast cancer subgroups in order to help physicians provide better treatment recommendations.

The METABRIC dataset consists of gene expression data and clinical features for 1,981 patients, and 43.85 percent have an observed death due to breast cancer with a median survival time of 1,907 days [27]. We partition 20 percent of the observations for both the validation and test set.

Each gene expression profile includes 49,576 probes, representing the genes in the transcriptome. We reduce the dimension of the dataset to 14 using manual selection of features by human expert. The first four features (*ERBB2, MKI67, PGR, ESR1*) are probes corresponding to genes that are common indicators of breast cancer and are known to influence treatment outcomes. The other ten features are inspired by the winners of the Sage Bionetworks-DREAM Breast Cancer Prognosis Challenge (BCC), which was a competition to assess the accuracy of computational models trained on the METABRIC dataset. The competition winners found four metagene factors (*CIN, MES, LYM, FGD3-SUSD3*) to be high predictors of survival rates [30, 4]. A metagene is the average of a set of probes representing a particular biological-pathway. We also supplement the gene expression data with the clinical variables (age at diagnosis, number of positive nodes, tumor size, ER status, HER2 status, and treatment) that the winning model showed to improve predictive performance [4].

DeepSurv outperforms the CPH model on predicting a patient's risk. DeepSurv has a C-index of 0.695 (95% CI: 0.693 - 0.697). The linear CPH has a C-index of 0.688 (95% CI: 0.686 - 0.690). Although DeepSurv's C-index does not seem significant in absolute terms, these results have important medical implications. Studies have shown evidence that while different commercial assays provide equivalent prognostic information at the population level, these tests differ in risk stratification on the individual level, which has a direct impact on the quality of patient care [31]. Evident from the greater C-index, DeepSurv is able to model nonlinear interactions between a patient's covariates and his or her predicted risk. Therefore, while DeepSurv and CPH have similar prognostic abilities, we expect the two methods to differ in risk stratification. Thus, DeepSurv has significant implications for clinical application.

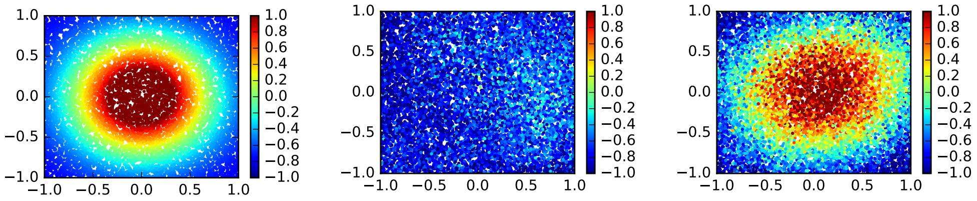
4.3 Treatment Recommender System Experiments

In this section, we perform two experiments to demonstrate the effectiveness of DeepSurv's treatment recommender system. First, we simulate clinical treatment data by including an additional covariate to the simulated data from Section 4.1.2. After demonstrating DeepSurv's modeling capabilities, we apply the recommender system to datasets from real clinical trials that study the effects of hormone treatment on breast cancer patients. We show that if all patients follow the network's recommended treatment option, we gain a significant increase in patient lifespan.

4.3.1 Simulated Treatment Data

We uniformly assign a treatment group $\tau \in \{0, 1\}$ to each simulated patient in the dataset. All of the patients in group $\tau = 0$ were 'unaffected' by the treatment (e.g. given a placebo) and have a constant risk function $h_0(x)$. The other group $\tau = 1$ is prescribed a treatment with Gaussian effects (Equation 8) and has a risk function $h_1(x)$ with $\lambda_{\max} = 10$ and $r = 0.5$.

Figure 3 illustrates the network's success in predicting the risk function for patients in the test set. Figure 3(a) plots the true risk distribution $h_1(x)$. As expected, Figure 3(b) shows that the network models a constant risk for a patient in treatment 0, independent of a patient's covariates. Figure 3(c) shows how DeepSurv models the Gaussian effects of a patient's covariates on their treatment risk. Because the network accurately reconstructs the risk function, we expect it will provide accurate treatment recommendations for new patients.



(a) True Risk: $h_1(x)$ (b) DeepSurv Risk: $\hat{h}_0(x)$ (c) DeepSurv Risk: $\hat{h}_1(x)$

Figure 3: Treatment Risk Surfaces versus a patient's relevant covariates x_0 and x_1. 3(a) plots the true risk $h_1(x)$ if all patients in the test set were given treatment 1. We then manually set all treatment groups to either $\tau = 0$ or $\tau = 1$. 3(b) shows the predicted risk $\hat{h}_0(x)$ for patients with treatment group $\tau = 0$. 3(c) depicts the network's predicted risk $\hat{h}_1(x)$ for patients in treatment group $\tau = 1$.

4.3.2 Hormone Treatment Recommendations for Breast Cancer

We first train DeepSurv on breast cancer data from the Rotterdam tumor bank [32] and construct a recommender system to provide treatment recommendations to patients from a study by the German Breast Cancer Study Group (GBSG) [33]. We then plot the two survival curves: the survival times of those who followed the recommended treatment and those who did not. If the recommender system is effective, we expect the population with the recommended treatments to survive longer than those who did not take the recommended treatment.

The Rotterdam tumour bank dataset contains records for 1,546 patients with node-positive breast cancer, and nearly 90 percent of the patients have an observed death time. The testing data from the GBSG contains complete data for 686 patients (56 percent are censored) in a randomized clinical trial that studied the effects of chemotherapy and hormone treatment on survival rate. We preprocess the data as outlined by Royston and Altman [34].

We then validate and compare the network against a linear CPH regression baseline. The C-indices of DeepSurv and linear CPH are 0.668 (95% CI 0.667 - 0.669) and 0.655 (95% CI: 0.654 - 0.656), respectively. Thus, DeepSurv provides an improvement relative to the CPH.

Next, we calculate the recommender function (Equation 5) for all patients in the GBSG test set and determine the recommended treatment for each patient. We then identify two subset of patients: those whose treatment group aligns with the network's recommended treatment (Recommendation) and those who did not undergo recommended treatment (Anti-Recommendation). In Figure 4, we plot the Kaplan-Meier survival curves for both the Recommendation subset and the Anti-Recommendation subset. The survival curve for the Recommendation subset is shifted to the right, which signifies an increase in survival time for the population following DeepSurv's recommendations. The median death time of the Recommendation population versus the Anti-Recommendation population is 40.099 and 31.770 months, respectively. The majority of patients were recommended to undergo hormone therapy (tamoxifen), which is in alignment with standard medical practice [35].

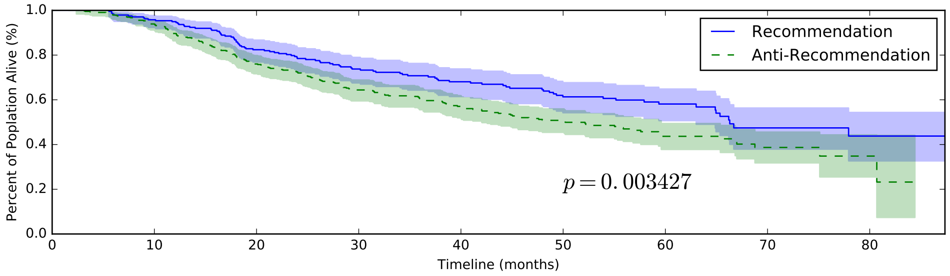


Figure 4: Kaplan-Meier estimated survival curves with confidence intervals ($\alpha = .05$) for the patients whom were given the treatment concordant with DeepSurv's recommended treatment (Recommendation) and the subset of patients who were not (Anti-Recommendation). We perform a log-rank test to validate the significance between the two curves ($p = 0.003427$).

4.4 Experimental Details

We run all linear CPH regression, Kaplan-Meier estimations, concordance-index statistics, and log-rank tests using the Lifelines Python package [36]. DeepSurv is implemented in Theano [37] with the Python package Lasagne [38].

The hyper-parameters of the network include: ℓ_2 regularization coefficient, learning rate, learning rate decay constant, dropout rate, momentum, and the size and depth of the network. We run the Random hyper-parameter optimization search as proposed in [21] using the Python package Optunity [39]. We perform random sampling on each hyper-parameter from a predefined range and evaluate the performance of the configuration on a validation set. We then choose the configuration with the largest validation C-index and with the smallest difference between validation C-index and training C-index to avoid models that overfit.

5 Summary

In conclusion, we demonstrated how deep learning can be applied to survival analysis and showed that DeepSurv is superior to the linear Cox proportional hazards model in predictive ability on survival datasets with linear and nonlinear risk functions. We illustrated that the network can provide personalized treatment recommendations for patients and can be used by physicians to guide their treatment decisions in order to improve patient lifespan. We also released a Python module that implements DeepSurv, see `https://github.com/jaredleekatzman/DeepSurv` for more details. With future research and development, this approach has the potential to replace traditional linear Cox regression and become a standard practice in biomedical applications.

Acknowledgements

This research was partially funded by NIH grant 1R01HG008383-01A1 (Y.K.) and supported by NSF Award DMS-1402254 (A.C.).

References

[1] Yeh RW, Secemsky EA, Kereiakes DJ, and et al. Development and validation of a prediction rule for benefit and harm of dual antiplatelet therapy beyond 1 year after percutaneous coronary intervention. *JAMA*, 315(16):1735–1749, 2016.

[2] Patrick Royston and Douglas G Altman. External validation of a cox prognostic model: principles and methods. *BMC medical research methodology*, 13(1):1, 2013.

[3] Eric Bair and Robert Tibshirani. Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol*, 2(4):e108, 2004.

[4] Wei-Yi Cheng, Tai-Hsien Ou Yang, and Dimitris Anastassiou. Development of a prognostic model for breast cancer survival in an open challenge environment. *Science translational medicine*, 5(181):181ra50–181ra50, 2013.

[5] David R Cox. Regression models and life-tables. In *Breakthroughs in statistics*. Springer, 1992.

[6] Daniel J Sargent. Comparison of artificial neural networks with other statistical approaches. *Cancer*, 91(S8):1636–1642, 2001.

[7] Anny Xiang, Pablo Lapuerta, Alex Ryutov, Jonathan Buckley, and Stanley Azen. Comparison of the performance of neural network methods and cox regression for censored survival data. *Computational statistics & data analysis*, 34(2):243–257, 2000.

[8] Knut Liestbl, Per Kragh Andersen, and Ulrich Andersen. Survival analysis and neural nets. *Statistics in medicine*, 13(12):1189–1200, 1994.

[9] W Nick Street. A neural network model for prognostic prediction. In *ICML*, pages 540–546, 1998.

[10] Leonardo Franco, José M Jerez, and Emilio Alba. Artificial neural networks and prognosis in medicine. survival analysis in breast cancer patients. In *ESANN*, pages 91–102, 2005.

[11] Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine*, 17(10):1169–1186, 1998.

[12] David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in medicine*, 14(1):73–82, 1995.

[13] L Mariani, D Coradini, E Biganzoli, P Boracchi, E Marubini, S Pilotti, B Salvadori, R Silvestrini, U Veronesi, R Zucali, et al. Prognostic factors for metachronous contralateral breast cancer: a comparison of the linear cox regression model and its artificial neural network extension. *Breast cancer research and treatment*, 44(2):167–178, 1997.

[14] Guido Schwarzer, Werner Vach, and Martin Schumacher. On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. *Statistics in medicine*, 19(4):541–561, 2000.

[15] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dr opout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[18] Yurii Nesterov et al. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

[19] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *Computing Research Repository (CoRR) abs/1211.5063*, 2012.

[20] Alan Senior, Georg Heigold, Marc'Aurelio Ranzato, and Ke Yang. An empirical study of learning rates in deep neural networks for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6724–6728. IEEE, 2013.

[21] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[22] Cancer Genome Atlas Research Network et al. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011.

[23] Frank E Harrell, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. Regression modeling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984.

[24] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[25] Peter C Austin. Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in medicine*, 31(29):3946–3958, 2012.

[26] Saczynski JS, Spencer FA, Gore JM, and et al. Twenty-year trends in the incidence of stroke complicating acute myocardial infarction: Worcester heart attack study. *Archives of Internal Medicine*, 168(19):2104–2110, 2008.

[27] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

[28] David W. Hosmer Jr., Stanley Lemeshow, and Susanne May. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. Wiley-Interscience, 2008.

[29] Viola Vaccarino, Ralph I Horwitz, Thomas P Meehan, Marcia K Petrillo, Martha J Radford, and Harlan M Krumholz. Sex differences in mortality after myocardial infarction: evidence for a sex-age interaction. *Archives of internal medicine*, 158(18):2054–2062, 1998.

[30] Wei-Yi Cheng, Tai-Hsien Ou Yang, and Dimitris Anastassiou. Biomolecular events in cancer revealed by attractor metagenes. *PLoS Comput Biol*, 9(2):e1002920, 2013.

[31] John MS Bartlett, Jane Bayani, Andrea Marshall, Janet A Dunn, Amy Campbell, Carrie Cunningham, Monika S Sobol, Peter S Hall, Christopher J Poole, David A Cameron, et al. Comparing breast cancer multiparameter tests in the optima prelim trial: No test is more equal than the others. *Journal of the National Cancer Institute*, 108(9):djw050, 2016.

[32] John A Foekens, Harry A Peters, Maxime P Look, Henk Portengen, Manfred Schmitt, Michael D Kramer, Nils Brünner, Fritz Jänicke, Marion E Meijer-van Gelder, Sonja C Henzen-Logmans, et al. The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients. *Cancer research*, 60(3):636–643, 2000.

[33] M Schumacher, G Bastert, H Bojar, K Huebner, M Olschewski, W Sauerbrei, C Schmoor, C Beyerle, RL Neumann, and HF Rauschecker. Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. german breast cancer study group. *Journal of Clinical Oncology*, 12(10):2086–2093, 1994.

[34] Douglas G Altman and Patrick Royston. What do we mean by validating a prognostic model? *Statistics in medicine*, 19(4):453–473, 2000.

[35] V Craig Jordan. Tamoxifen: a most unlikely pioneering medicine. *Nature Reviews Drug Discovery*, 2(3):205–213, 2003.

[36] Davidson-Pilon C. Lifelines. `https://github.com/camdavidsonpilon/lifelines`, 2016.

[37] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[38] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave. Lasagne: First release. `https://github.com/Lasagne/Lasagne`, August 2015.

[39] Marc Claesen, Jaak Simm, and Vilen Jumutc. Optunity. `https://github.com/claesenm/optunity`, 2014.