CrossMark

DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network

Jared L. Katzman[1], Uri Shaham[2,5,9], Alexander Cloninger[3,8], Jonathan Bates[3,4,5], Tingting Jiang[6] and Yuval Kluger[3,6,7*] (ID)

Abstract

Background: Medical practitioners use survival models to explore and understand the relationships between patients' covariates (e.g. clinical and genetic features) and the effectiveness of various treatment options. Standard survival models like the linear Cox proportional hazards model require extensive feature engineering or prior medical knowledge to model treatment interaction at an individual level. While nonlinear survival methods, such as neural networks and survival forests, can inherently model these high-level interaction terms, they have yet to be shown as effective treatment recommender systems.

Methods: We introduce DeepSurv, a Cox proportional hazards deep neural network and state-of-the-art survival method for modeling interactions between a patient's covariates and treatment effectiveness in order to provide personalized treatment recommendations.

Results: We perform a number of experiments training DeepSurv on simulated and real survival data. We demonstrate that DeepSurv performs as well as or better than other state-of-the-art survival models and validate that DeepSurv successfully models increasingly complex relationships between a patient's covariates and their risk of failure. We then show how DeepSurv models the relationship between a patient's features and effectiveness of different treatment options to show how DeepSurv can be used to provide individual treatment recommendations. Finally, we train DeepSurv on real clinical studies to demonstrate how it's personalized treatment recommendations would increase the survival time of a set of patients.

Conclusions: The predictive and modeling capabilities of DeepSurv will enable medical researchers to use deep neural networks as a tool in their exploration, understanding, and prediction of the effects of a patient's characteristics on their risk of failure.

Keywords: Deep learning, Survival analysis, Treatment recommendations

Background

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 [1–4]. One standard survival model is the Cox proportional hazards model (CPH) [5]. The CPH is a semiparametric model that calculates the effects of observed covariates on the risk of an event occurring (e.g. 'death'). The model assumes that a patient's log-risk of failure is a linear combination of the patient's covariates. This assumption is referred to as the *linear proportional hazards* condition. However, in many applications, such as providing personalized treatment recommendations, it may be too simplistic to assume that the log-risk function is linear. As such, a richer family of survival models is needed to better fit survival data with nonlinear log-risk functions.

*Correspondence: yuval.kluger@yale.edu
Majority of work was done while at Yale University
[3]Applied Mathematics Program, Yale University, 51 Prospect Street, 06511 New Haven, CT, USA
[6]Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, 06511 New Haven, CT, USA
Full list of author information is available at the end of the article

Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 2 of 12

To model nonlinear survival data, researchers have applied three main types of neural networks to the problem of survival analysis. These include variants of: (i) classification methods (see details in [6, 7]), (ii) time-encoded methods (see details in [8, 9]), (iii) and risk-predicting methods (see details in [10]). This third type is a feed-forward neural network (NN) that estimates an individual's risk of failure. In fact, Faraggi-Simon's network is seen as a nonlinear extension of the Cox proportional hazards model.

Risk neural networks learn highly complex and nonlinear relationships between prognostic features and an individual's risk of failure. In application, for example, when the success of a treatment option is affected by an individual's features, the NN learns the relationship without prior feature selection or domain expertise. The network is then able to provide a personalized recommendation based on the computed risk of a treatment.

However, previous studies have demonstrated mixed results on NNs ability to predict risk. For instance, researchers have attempted to apply the Faraggi-Simon network with various extensions, but they have failed to demonstrate improvements beyond the linear Cox model, see [11–13]. One possible explanation is that the practice of NNs was not as developed as it is today. To the best of our knowledge, NNs have not outperformed standard methods for survival analysis (e.g. CPH). Our manuscript shows that this is no longer the case; with modern techniques, risk NNs have state-of-the-art performance and can be used for a variety of medical applications.

The goals of this paper are: (i) to show that the application of deep learning to survival analysis performs as well as or better than other survival methods in predicting risk; and (ii) to demonstrate that the deep neural network can be used as a personalized treatment recommender system and a useful framework for further medical research.

We propose a modern Cox proportional hazards deep neural network, henceforth referred to as DeepSurv, as the basis for a treatment recommender system. We make the following contributions. First, we show that DeepSurv performs as well as or better than other survival analysis methods on survival data with both linear and nonlinear effects from covariates. Second, we include an additional categorical variable representing a patient's treatment group to illustrate how the network can learn complex relationships between an individual's covariates and the effect of a treatment. Our experiments validate that the network successfully models the treatment's risk within a population. Third, we use DeepSurv to provide treatment recommendations tailored to a patient's observed features. We confirm our results on real clinical studies, which further demonstrates the power of DeepSurv. Finally, we show that the recommender system

supports medical practitioners in providing personalized treatment recommendations that potentially could increase the median survival time for a set of patients.

The organization of the manuscript is as follows: in "Background" section, we provide a brief background on survival analysis. In "Methods" section, we present our contributions, including an explanation of our implementation of DeepSurv and our proposed recommender system. In "Results" section, we describe the experimental design and results. "Conclusion" and "Discussion" sections conclude the manuscript.

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

Survival data

Survival data is comprised of three elements: a patient's baseline data x, a failure 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*. If one opts to use standard regression methods, the right-censored data is considered to be a type of missing data. This is typically discarded which can introduce a bias in the model. Therefore, modeling right-censored data requires special consideration or the use of a survival model.

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' beyond time t. 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 the probability an individual will not survive an extra infinitesimal amount of time δ, given they have already survived up to time t. Thus, a greater hazard signifies a greater risk of death.

Linear survival models

The Cox proportional hazards model is a common method for modeling an individual's survival given their baseline data x. In accordance with the standard R survival package *coxph*, we use notation from [14] to describe the Cox model. The model assumes that the hazard function is composed of two non-negative functions: a baseline

Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 3 of 12

hazard function, $\lambda_0(t)$, and a risk score, $r(x) = e^{h(x)}$, defined as the effect of an individual's observed covariates on the baseline hazard [14]. We denote $h(x)$ as the log-risk function. The hazard function is assumed to have the form

$$\lambda(t|x) = \lambda_0(t) \cdot e^{h(x)}. \tag{2}$$

The CPH is a proportional hazards model that estimates the log-risk function, $h(x)$, by a linear function $\hat{h}_\beta(x) = \beta^T x$ [or equivalently $\hat{r}_\beta(x) = e^{\beta^T x}$]. 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

$$L_c(\beta) = \prod_{i:E_i=1} \frac{\hat{r}_\beta(x_i)}{\sum_{j\in\Re(T_i)} \hat{r}_\beta(x_j)} = \prod_{i:E_i=1} \frac{\exp(\hat{h}_\beta(x_i))}{\sum_{j\in\Re(T_i)} \exp(\hat{h}_\beta(x_j))}, \tag{3}$$

where the values T_i, E_i, and x_i are the respective event time, event indicator, and baseline data for the ith observation. The product is defined over the set of patients with an observable event $E_i = 1$. The risk set $\Re(t) = \{i : T_i \geq t\}$ is the set of patients still at risk of failure at time t.

In many applications, for example modeling nonlinear gene interactions, we cannot assume the data satisfies the linear proportional hazards condition. In this case, the CPH model would require computing high-level interaction terms. This becomes prohibitively expensive as the number of features and interactions increases. Therefore, a more complex nonlinear model is needed.

Nonlinear survival models

The Faraggi-Simon method is a feed-forward neural network that provides the basis for a nonlinear proportional hazards model. [10] experimented with a single hidden layer network with two or three nodes. Their model requires no prior assumption of the log-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 log-risk function. Similar to Cox regression, the network optimizes a modified Cox partial likelihood. They replace the linear combination of features $\hat{h}_\beta(x)$ in Eq. 3 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 [10, 12, 13]. Furthermore, to the best of our knowledge, we were the first to attempt applying modern deep learning techniques to the Cox proportional hazards loss function.

Another popular machine learning approach to modeling patients' hazard function is the random survival forest (RSF) [15, 16]. The random survival forest is a tree method that produces an ensemble estimate for the cumulative hazard function.

A more recent deep learning approach models the event time according to a Weibull distribution with parameters given by latent variables generated by a deep exponential family [17].

Methods

In this section, we describe our methodology for providing personalized treatment recommendations using DeepSurv. First, we describe the architecture and training details of DeepSurv, an open source Python module that applies recent deep learning techniques to a nonlinear Cox proportional hazards network. Second, we define DeepSurv as a prognostic model and show how to use the network's predicted log-risk function to provide personalized treatment recommendations.
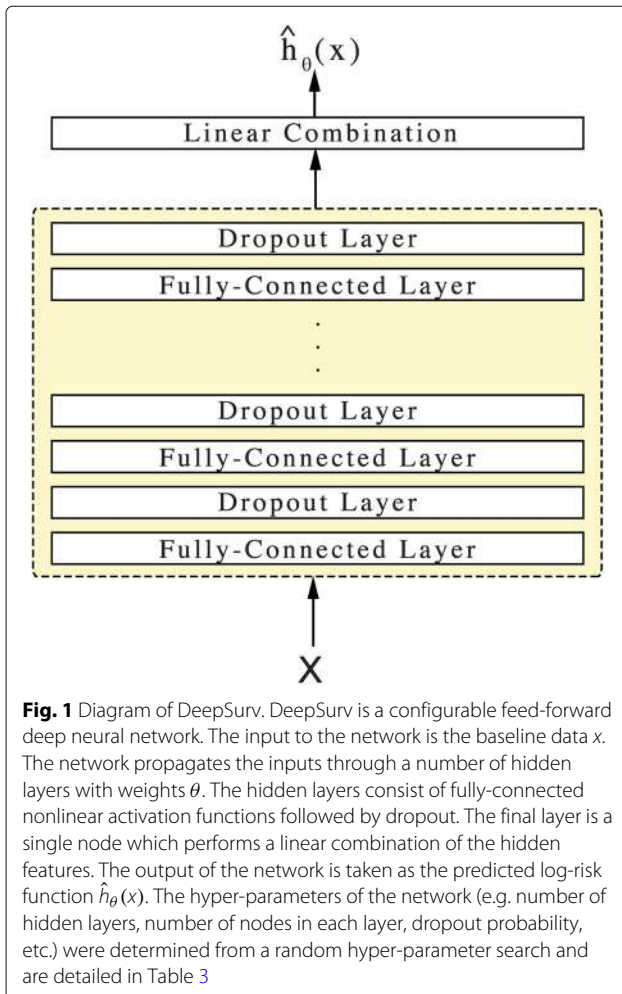
DeepSurv

DeepSurv is a deep feed-forward neural network which predicts the effects of a patient's covariates on their hazard rate parameterized by the weights of the network θ. Figure 1 illustrates the basic components of DeepSurv. The input to the network is a patient's baseline data x. The hidden layers of the network consist of a fully-connected layer of nodes, followed by a dropout layer [18]. The output of the network $\hat{h}_\theta(x)$ is a single node with a linear activation which estimates the log-risk function in the Cox model (Eq. 2). We train the network by setting the objective function to be the average negative log partial likelihood of Eq. 3 with regularization:

$$l(\theta) := -\frac{1}{N_{E=1}} \sum_{i:E_i=1} \left(\hat{h}_\theta(x_i) - \log \sum_{j\in\Re(T_i)} e^{\hat{h}_\theta(x_j)} \right) + \lambda \cdot ||\theta||_2^2, \tag{4}$$

where $N_{E=1}$ is the number of patients with an observable event and λ is the ℓ_2 regularization parameter. We then use gradient descent optimization to find the weights of the network which minimize Eq. 4.

We use modern deep learning techniques to optimize the training of the network. These include: standardizing the input, Scaled Exponential Linear Units (SELU) [19] as the activation function, Adaptive Moment Estimation (Adam) [20] for the gradient descent algorithm, Nesterov momentum [21], and learning rate scheduling [22]. To tune the network's hyper-parameters, we perform a Random hyper-parameter optimization search [23]. For more technical details, see Appendix A.

Katzman *et al. BMC Medical Research Methodology*   (2018) 18:24

Page 4 of 12

Fig. 1 Diagram of DeepSurv. DeepSurv is a configurable feed-forward deep neural network. The input to the network is the baseline data x. The network propagates the inputs through a number of hidden layers with weights θ. The hidden layers consist of fully-connected nonlinear activation functions followed by dropout. The final layer is a single node which performs a linear combination of the hidden features. The output of the network is taken as the predicted log-risk function $\hat{h}_\theta(x)$. The hyper-parameters of the network (e.g. number of hidden layers, number of nodes in each layer, dropout probability, etc.) were determined from a random hyper-parameter search and are detailed in Table 3

Treatment recommender system

In a clinical study, patients are subject to different levels of risk based on their relevant prognostic features and which treatment they undergo. We generalize this assumption as follows. Let all patients in a given study be assigned to one of n treatment groups $\tau \in \{0, 1, \ldots, n-1\}$. We assume each treatment i to have an independent risk function $e^{h_i(x)}$. Collectively, the hazard function becomes:

$$\lambda(t; x | \tau = i) = \lambda_0(t) \cdot e^{h_i(x)}. \tag{5}$$

For any patient, the network should be able to accurately predict the log-risk $h_i(x)$ of being prescribed a given treatment i. Then, based on the assumption that each individual has the same baseline hazard function $\lambda_0(t)$, we can take the log of the hazards ratio to calculate the personal risk-ratio of prescribing one treatment option over another. We define this difference of log hazards as the *recommender* function or $\text{rec}_{ij}(x)$:

$$\text{rec}_{ij}(x) = \log\left(\frac{\lambda(t; x | \tau = i)}{\lambda(t; x | \tau = j)}\right) = \log\left(\frac{\lambda_0(t) \cdot e^{h_i(x)}}{\lambda_0(t) \cdot e^{h_j(x)}}\right) \tag{6}$$
$$= h_i(x) - h_j(x).$$

The recommender function can be used to provide personalized treatment recommendations. We first pass a patient through the network once in treatment group i and again in treatment group j and take the difference. When a patient receives a positive recommendation $\text{rec}_{ij}(x)$, treatment i leads to a higher risk of death than treatment j. Hence, the patient should be prescribed treatment j. Conversely, a negative recommendation indicates that treatment i is more effective and leads to a lower risk of death than treatment j, and we recommend treatment i.

DeepSurv's architecture holds an advantage over the CPH because it calculates the recommender function without an a priori specification of treatment interaction terms. In contrast, the CPH model computes a constant recommender function unless treatment interaction terms are added to the model, see Appendix B for more details. Discovering relevant interaction terms is expensive because it requires extensive experimentation or prior biological knowledge of treatment outcomes. Therefore, DeepSurv is more cost-effective compared to CPH.

Results

We perform four sets of experiments: (i) simulated survival data, (ii) real survival data, (iii) simulated treatment data, and (iv) real treatment data. First, we use simulated data to show how DeepSurv successfully learns the true log-risk function of a population. Second, we validate the network's predictive ability by training DeepSurv on real survival data. Third, we simulate treatment data to verify that the network models multiple risk functions in a population based on the specific treatment a patient undergoes. Fourth, we demonstrate how DeepSurv provides treatment recommendations and show that DeepSurv's recommendations improve a population's survival rate. For more technical details on the experiments, see Appendix A.

In addition to training DeepSurv on each dataset, we run a linear CPH regression for a baseline comparison. We also fit a RSF to compare DeepSurv against a state-of-the-art nonlinear survival model. Even though we can compare the RSF's predictive accuracy to DeepSurv's, we do not measure the RSF's performance on modeling a simulated dataset's true log-risk function $h(x)$. This is due to the fact that the the RSF calculates the cumulative hazard function $\Lambda(t) = \int_0^t \lambda(\tau) d\tau$ rather than the hazard function $\lambda(t)$.

Katzman *et al. BMC Medical Research Methodology*   (2018) 18:24

Page 5 of 12

Evaluation

Survival data

To evaluate the models' predictive accuracy on the survival data, we measure the concordance-index (C-index) c as outlined by [24]. The C-index is the most common metric used in survival analysis and reflects a measure of how well a model predicts the ordering of patients' death times. For context, a $c = 0.5$ is the average C-index of a random model, whereas $c = 1$ is a perfect ranking of death times. We perform bootstrapping [25] and sample the test set with replacement to obtain confidence intervals.

Treatment recommendations

We determine the recommended treatment for each patient in the test set using DeepSurv and the RSF. We do not calculate the recommended treatment for CPH; without preselected treatment-interaction terms, the CPH model will compute a constant recommender function and recommend the same treatment option for all patients. This would effectively be comparing the survival rates between the control and experimental groups. DeepSurv and the RSF are capable of predicting an individual's hazard per treatment because each computes relevant interaction terms. For DeepSurv, we choose the recommended treatment by calculating the recommender function (Eq. 11). Because the RSF predicts a cumulative hazard for each patient, we choose the treatment with the minimum cumulative hazard.

Once we determine the recommended treatment, we identify two subsets of patients: those whose treatment group aligns with the model's recommended treatment (Recommendation) and those who do not undergo the recommended treatment (Anti-Recommendation). We calculate the median survival time of each subset to determine if a model's treatment recommendations increase the survival rate of the patients. We then perform a log-rank test to validate whether the difference between the two subsets is significant.

Simulated survival data

In this section, we perform two experiments with simulated survival data: one with a linear log-risk function and one with a nonlinear (Gaussian) log-risk function. The advantage of using simulated datasets is that we can ascertain whether DeepSurv can successfully model the true log-risk function instead of overfitting random noise.

For each experiment, we generate a training, validation, and testing set of $N = 4000, 1000, 1000$ observations respectively. Each observation x represents a patient vector with $d = 10$ covariates. The ten variables are each drawn from a uniform distribution on $[-1, 1)$. We then generate a patient's death time T as a function of their covariates by using the exponential Cox model [26]:

$$T \sim \text{Exp}(\lambda(t; x)) = \text{Exp}\left(\lambda_0 \cdot e^{h(x)}\right). \tag{7}$$

In both experiments, the log-risk function $h(x)$ only depends on two of the ten covariates. This allows us to verify that DeepSurv discerns the relevant covariates from the noise. Next, we choose a censoring time to represent the 'end of study' such that 50 percent of the patients have an observed event, $E = 1$, in the dataset. Further details of the simulated data generation are found in Appendix C.

Linear experiment

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

$$h(x) = x_0 + 2x_1. \tag{8}$$

Because the linear proportional hazards assumption holds true, we expect the linear CPH to accurately model the log-risk function in Eq. 8.

Our results (see Table 1) demonstrate that DeepSurv performs as well as the standard linear Cox regression and better than RSF in predictive ability.

Figure 2 demonstrates how DeepSurv more accurately models the log-risk function compared to the linear CPH. Figure 2a plots the true log-risk function $h(x)$ for all patients in the test set. As shown in Fig. 2b, the CPH's estimated log-risk function $\hat{h}_\beta(x)$ does not perfectly model the true log-risk for a patient. In contrast, as shown in Fig. 2c, DeepSurv better estimates the true log-risk function.

To quantify these differences, Fig. 2d and e show that the CPH's estimated log-risk has a significantly larger absolute error than that of DeepSurv, specifically for patients with a high positive log-risk. We calculate the mean-squared-error (MSE) between a model's predicted log-risk and the true log-risk values. The MSEs of CPH and DeepSurv are 20.528 057 878 872 541 and 0.192 683 15, respectively. Even though DeepSurv and CPH have similar predictive abilities, this demonstrates that DeepSurv is superior than the CPH at modeling the true risk function of the population.

Nonlinear experiment

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

$$h(x) = \log(\lambda_{\max}) \exp\left(-\frac{x_0^2 + x_1^2}{2r^2}\right). \tag{9}$$

The surface of the log-risk function is depicted in Fig. 3a. Because this log-risk function is nonlinear, we do not expect the CPH to predict the log-risk function properly without adding quadratic terms of the covariates to the model. We expect DeepSurv to reconstruct the Gaussian log-risk function and successfully predict a patient's risk.

Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 6 of 12

Table 1 Experimental results for all experiments C-index (95% confidence interval)

Experiment	CPH	DeepSurv	RSF
Simulated Linear	**0.779239 (0.777,0.781)**	0.778065 (0.776,0.780)	0.757863 (0.756,0.760)
Simulated Nonlinear	0.486728 (0.484,0.489)	**0.652434 (0.650, 0.655)**	0.626552 (0.624,0.629)
WHAS	0.816025 (0.813, 0.819)	0.866723 (0.863,0.870)	**0.892884 (0.890,0.895)**
SUPPORT	0.583076 (0.581,0.585)	0.618907 (0.617,0.621)	**0.619302 (0.618,0.621)**
METABRIC	0.631674 (0.627,0.636)	**0.654452 (0.650,0.659)**	0.619517 (0.615,0.624)
Simulated Treatment	0.516620 (0.514,0.519)	**0.575400 (0.573,0.578)**	0.550298 (0.548,0.553)
Rotterdam & GBSG	0.658773 (0.655, 0.662)	**0.676349 (0.673,0.679)**	0.647924 (0.644, 0.651)

The bold faced numbers signify the best performing algorithm

Lastly, we expect the RSF and DeepSurv to accurately rank the order of patient's deaths.

The CI results in Table 1 shows that DeepSurv outperforms the linear CPH and predicts as well as the RSF. In addition, DeepSurv correctly learns nonlinear relationships between a patient's covariates and their log-risk. As shown in Fig. 3, DeepSurv is more successful than the linear CPH in modeling the true log-risk function. Figure 3b demonstrates that the linear CPH regression fails to determine the first two covariates as significant. The CPH has a C-index of 0.486728, which is equivalent to the performance of randomly ranking death times. Meanwhile, Fig. 3c demonstrates that DeepSurv reconstructs the Gaussian relationship between the first two covariates and a patient's log-risk.

Real survival data experiments

We compare the performance of the CPH and DeepSurv on three datasets from real studies: the Worcester Heart Attack Study (WHAS), the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT), and The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC). Because previous research shows that neural networks do not outperform the CPH, our goal is to demonstrate that DeepSurv does indeed have state-of-the-art predictive ability in practice on real survival datasets.

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 [27]. The dataset consists of 1638 observations and 5 features: age, sex, body-mass-index (BMI), left heart failure complications (CHF), and order of MI (MIORD). We reserve 20 percent of the dataset as a testing set. A total of 42.12 percent of patients died during the survey with a median death time of 516.0 days.
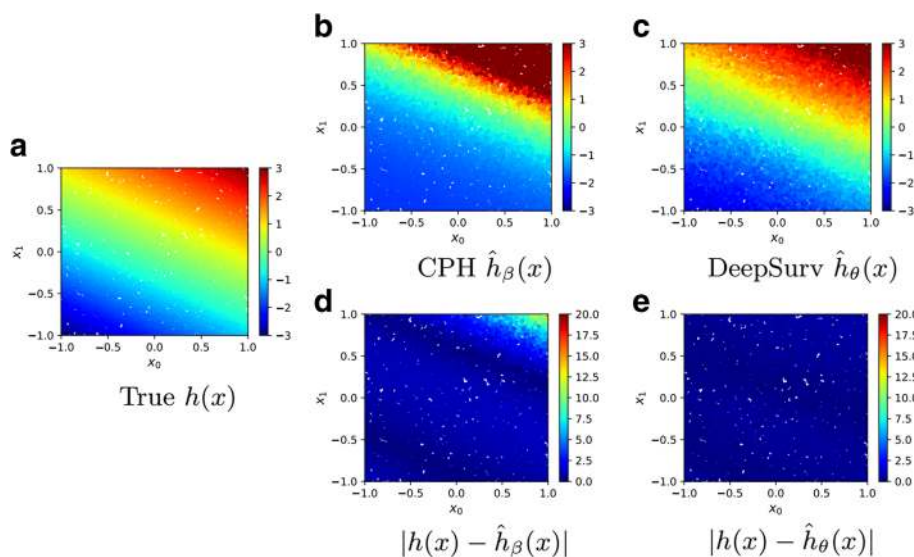

Fig. 2 Simulated Linear Experimental Log-Risk Surfaces. Predicted log-risk surfaces and errors for the simulated survival data with linear log-risk function with respect to a patient's covariates x_0 and x_1. **a** The true log-risk $h(x) = x_0 + 2x_1$ for each patient. **b** The predicted log-risk surface of $\hat{h}_\beta(x)$ from the linear CPH model parameterized by β. **c** The output of DeepSurv $\hat{h}_\theta(x)$ predicts a patient's log-risk. **d** The absolute error between true log-risk $h(x)$ and CPH's predicted log-risk $\hat{h}_\beta(x)$. **e** The absolute error between true log-risk $h(x)$ and DeepSurv's predicted log-risk $\hat{h}_\theta(x)$
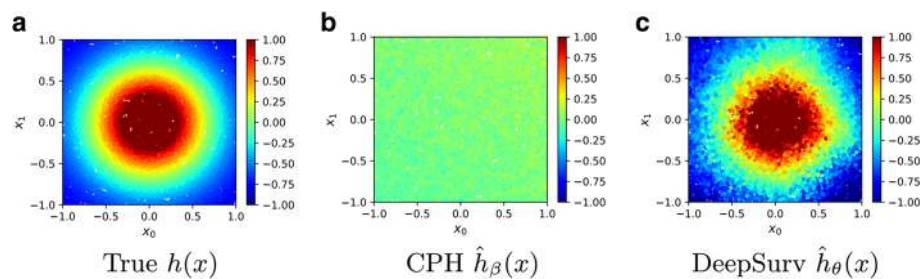
Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 7 of 12

Fig. 3 Simulated Nonlinear Experimental Log-Risk Surfaces. Log-risk surfaces of the nonlinear test set with respect to patient's covariates x_0 and x_1. **a** The calculated true log-risk $h(x)$ (Eq. 9) for each patient. **b** The predicted log-risk surface of $\hat{h}_\beta(x)$ from the linear CPH model parameterized on β. The linear CPH predicts a constant log-risk. **c** The output of DeepSurv $\hat{h}_\theta(x)$ is the estimated log-risk function

As shown in Table 1, DeepSurv outperforms the CPH; however, the RSF outperforms DeepSurv.

Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT)

The Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) is a larger study that researches the survival time of seriously ill hospitalized adults [28]. The dataset consists of 9,105 patients and 14 features for which almost all patients have observed entries (age, sex, race, number of comorbidities, presence of diabetes, presence of dementia, presence of cancer, mean arterial blood pressure, heart rate, respiration rate, temperature, white blood cell count, serum's sodium, and serum's creatinine). We drop patients with any missing features and reserve 20 percent of the dataset as a testing set. A total of 68.10 percent of patients died during the survey with a median death time of 58 days.

As shown in Table 1, DeepSurv performs as well as the RSF and better than the CPH with a larger study. This validates DeepSurv's ability to predict the ranking of patient's risks on real survival data.

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 1,980 patients, and 57.72 percent have an observed death due to breast cancer with a median survival time of 116 months [29]. We prepare the dataset in line with the Immunohistochemical 4 plus Clinical (IHC4+C) test, which is a common prognostic tool for evaluating treatment options for breast cancer patients [30]. We join the 4 gene indicators (*MKI67, EGFR, PGR, and ERBB2*) with the a patient's clinical features (hormone treatment indicator, radiotherapy

indicator, chemotherapy indicator, ER-positive indicator, age at diagnosis). We then reserved 20 percent of the patients as the test set.

Table 1 shows that DeepSurv performs better than both the CPH and RSF. This result demonstrates not only DeepSurv's ability to model the risk effects of gene expression data but also shows the potential for future research of DeepSurv as a comparable prognostic tool to common medical tests such as the IHC4+C.

Treatment recommender system experiments

In this section, we perform two experiments to demonstrate the effectiveness of DeepSurv's treatment recommender system. First, we simulate treatment data by including an additional covariate to the simulated data from "Nonlinear experiment" section. Second, after demonstrating DeepSurv's modeling and recommendation capabilities, we apply the recommender system to a real dataset used to study the effects of hormone treatment on breast cancer patients. We show that DeepSurv can successfully provide personalized treatment recommendations. We conclude that if all patients follow the network's recommended treatment options, we would gain a significant increase in patients' lifespans.

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 log-risk function $h_0(x)$. The other group $\tau = 1$ is prescribed a treatment with Gaussian effects (Eq. 9) and has a log-risk function $h_1(x)$ with $\lambda_{\max} = 10$ and $r = 0.5$.

Figure 4 illustrates the network's success in modeling both treatments' log-risk functions for patients. Figure 4a plots the true log-risk distribution $h(x)$. As expected, Fig. 4b shows that the network models a constant log-risk for a patient in treatment $\tau = 0$, independent of a patient's covariates. Figure 4c shows how DeepSurv models the Gaussian effects of a patient's covariates on
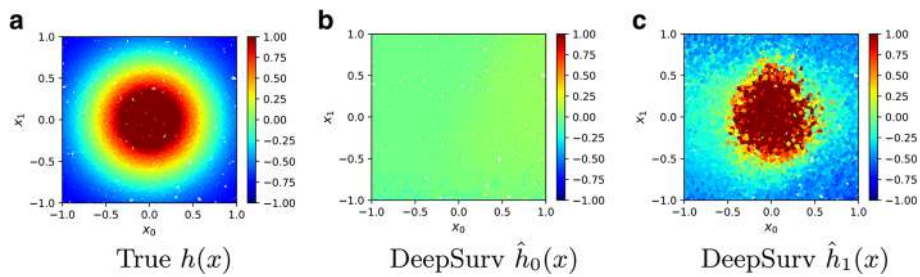
Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 8 of 12

Fig. 4 Simulated Treatment Log-Risk Surface. Treatment Log-Risk Surfaces as a function of a patient's relevant covariates x_0 and x_1. **a** The true log-risk $h_1(x)$ if all patients in the test set were given treatment $\tau = 1$. We then manually set all treatment groups to either $\tau = 0$ or $\tau = 1$. **b** The predicted log-risk $\hat{h}_0(x)$ for patients with treatment group $\tau = 0$. **c** The network's predicted log-risk $\hat{h}_1(x)$ for patients in treatment group $\tau = 1$

their treatment log-risk. To further quantify these results, Table 1 shows that DeepSurv has the largest concordance index. Because the network accurately reconstructs the risk function, we expect that it will provide accurate treatment recommendations for new patients.

In Fig. 5, we plot the Kaplan-Meier survival curves for both the Recommendation and Anti-Recommendation subset for each method. Figure 5a shows that 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. This is further quantified by the median survival times summarized in Table 2. The *p*-value of DeepSurv's recommendations is less than 0.000090, and we can reject the null hypothesis that DeepSurv's recommendations would not affect the population's survival time. As shown in Table 2, the subset of patients that follow RSF's recommendations have a shorter survival time than those who do not follow RSF's recommended treatment. Therefore, we could take the RSF's recommendations and provide the

patients with the opposite treatment option to increase median survival time; however, Fig. **??** shows that that improvement would not be statistically valid. While both methods of DeepSurv and RSF are able to compute treatment interaction terms, DeepSurv is more successful in recommending personalized treatments.

Rotterdam & German Breast Cancer Study Group (GBSG)

We first train DeepSurv on breast cancer data from the Rotterdam tumor bank [31]. and construct a recommender system to provide treatment recommendations to patients from a study by the German Breast Cancer Study Group (GBSG) [32]. The Rotterdam tumor bank dataset contains records for 1546 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 [33].
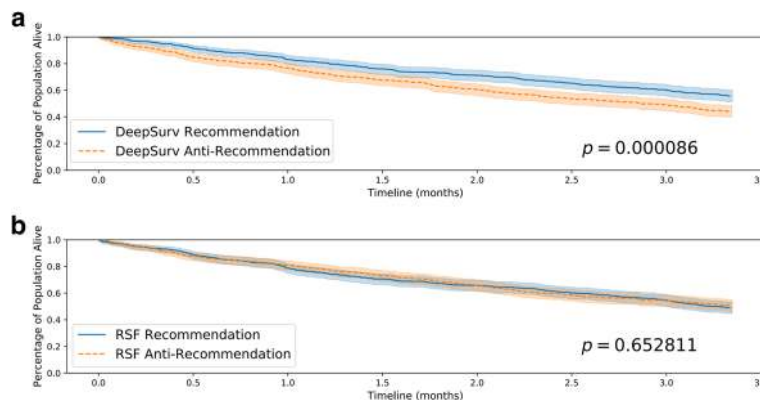

Fig. 5 Simulated Treatment Survival Curves. Kaplan-Meier estimated survival curves with confidence intervals ($\alpha = .05$) for the patients who were given the treatment concordant with a method's recommended treatment (Recommendation) and the subset of patients who were not (Anti-Recommendation). We perform a log-rank test to validate the significance between each set of survival curves. **a** Effect of DeepSurv's Treatment Recommendations (Simulated Data), **b** Effect of RSF's Treatment Recommendations (Simulated Data)

Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 9 of 12

Table 2 Experimental results for treatment recommendations: median survival time (months)

Experiment	DeepSurv		RSF	
	Rec	Anti-Rec	Rec	Anti-Rec
Simulated	**3.334**	**2.867**	3.270	3.334
Rotterdam & GBSG	**40.099**	**31.770**	39.014	30.752

The bold faced numbers signify the best performing algorithm

We first validate DeepSurv's performance against the RSF and CPH baseline. 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.

Table 1 shows that DeepSurv provides an improved predictive ability relative to the CPH and RSF. In Fig. 6, we plot the Kaplan-Meier survival curves for both the Recommendation subset and the Anti-Recommendation subset for each method. Figure 6a shows that the survival curve for DeepSurv's Recommendation subset is statistically significant from the Anti-recommendation subset, and Table 2 shows that DeepSurv's recommendations increase the median survival time of the population. Figure 6b demonstrates that RSF is unable to provide significant treatment recommendations, despite an increase in median survival times (see Table 2). The results of this experiment demonstrate not only DeepSurv's superior modeling capabilities but also validate DeepSurv's ability in providing personalized treatment recommendations on real clinical data. Moreover, we can train DeepSurv on survival data from one clinical study and transfer the learnings to provide personalized treatment recommendations to a different population of breast cancer patients.

Conclusion

In conclusion, we demonstrated that the use of deep learning in survival analysis allows for: (i) higher performance due to the flexibility of the model, and (ii) effective treatment recommendations based on the predicted effect of treatment options on an individual's risk. We validated that DeepSurv predicts patients' risk mostly as well as or better than other linear and nonlinear survival methods. We experimented on increasingly complex survival datasets and demonstrated that DeepSurv computes complex and nonlinear features without a priori selection or domain expertise. We then demonstrated that DeepSurv is superior in predicting personalized treatment recommendations compared to the state-of-the-art survival method of random survival forests. We also released a Python module that implements DeepSurv and scripts for running reproducible experiments in Docker, see [34] for more details.

Discussion

The success of DeepSurv's predictive, modeling, and recommending abilities paves the way for future research in deep neural networks and survival analysis. DeepSurv can lead to various extensions, such as the use of convolution neural networks to predict risk with medical imaging. With more research at scale, DeepSurv has the potential to supplement traditional survival analysis methods and become a standard method for medical practitioners to study and recommend personalized treatment options.

Appendix A
Experimental details

We run all linear CPH regression, Kaplan-Meier estimations, c-index statistics, and log-rank tests using the Lifelines Python package. DeepSurv is implemented in
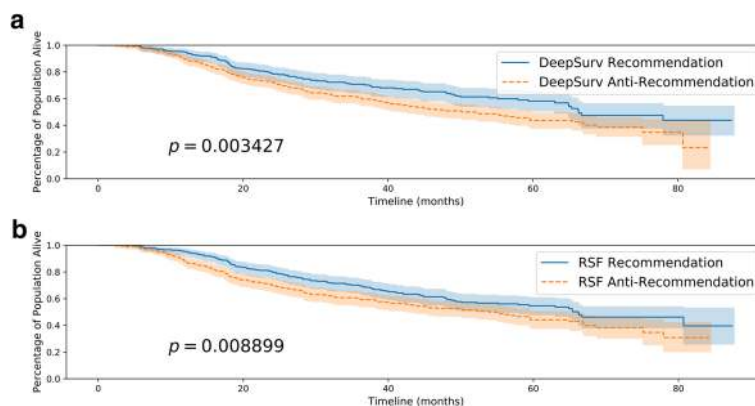

Fig. 6 Rotterdam & German Breast Cancer Study Group (GBSG) Survival Curves. Kaplan-Meier estimated survival curves with confidence intervals ($\alpha = .05$) for the patients who were given the treatment concordant with a method's recommended treatment (Recommendation) and the subset of patients who were not (Anti-Recommendation). We perform a log-rank test to validate the significance between each set of survival curves. **a** Effect of DeepSurv's Treatment Recommendations (GBSG), **b** Effect of RSF's Treatment Recommendations (GBSG)

Katzman *et al. BMC Medical Research Methodology* (2018) 18:24

Page 10 of 12

Theano with the Python package Lasagne. We use the R package randomForestSRC to fit RSFs. All experiments are run using Docker containers such that the experiments are easily reproducible. We use the FloydHub base image for the DeepSurv docker container.

The hyper-parameters of the network include: the depth and size of the network, learning rate, ℓ_2 regularization coefficient, dropout rate, exponential learning rate decay constant , and momentum. We run the Random hyper-parameter optimization search as proposed in [23] using the Python package Optunity. We use the Sobol solver [35, 36] to sample each hyper-parameter from a predefined range and evaluate the performance of the configuration using k-means cross validation ($k = 3$). We then choose the configuration with the largest validation C-index to avoid models that overfit. The hyper-parameters we use in all experiments are summarized in the next "Model Hyper-parameters" section.

Model Hyper-parameters

As described in "Experimental details" section, we tune DeepSurv's hyper-parameters by running a random hyper-parameter search using the Python package Optunity. The table below summarizes the hyper-parameters we use for each experiment's DeepSurv network.

We applied inverse time decay to the learning rate at each epoch:

$$decayed\_LR := \frac{LR}{1 + epoch \cdot lr\_decay\_rate}. \tag{10}$$

Appendix B

CPH recommender function

Let each patient in the dataset have a set of n features x_n, in which one feature is a treatment variable $x_0 = \tau$. The CPH model estimates the log-risk function as a linear combination of the patient's features $\hat{h}_\beta(x) = \beta^T x = \beta_0 \tau + \beta_1 x_1 + \ldots + \beta_n x_n$. When we calculate the recommender function for the CPH model, we show that the model returns a constant function independent of the patient's features:

$$
\begin{aligned}
\text{rec}_{ij}(x) &= \log\left(\frac{\lambda(t; x|\tau = i)}{\lambda(t; x|\tau = j)}\right) \\
&= \log\left(\frac{\lambda_0(t) \cdot e^{\beta_0 i + \beta_1 x_1 + \ldots + \beta_n x_n}}{\lambda_0(t) \cdot e^{\beta_0 j + \beta_1 x_1 + \ldots + \beta_n x_n}}\right) \\
&= \log\left(e^{\beta_0 i + \beta_1 x_1 + \ldots + \beta_n x_n - (\beta_0 j + \beta_1 x_1 + \ldots + \beta_n x_n)}\right) \\
&= \beta_0 i - \beta_0 j \\
&= \beta_0 (i - j).
\end{aligned}
\tag{11}
$$

The CPH will recommend all patients to choose the same treatment option based on whether the model calculates the weight β_0 to be positive or negative. Thus, the CPH would not be providing personalized treatment recommendations. Instead, the CPH determines whether the treatment is effective and, if so, then recommending it to all patients. In an experiment, when we calculate which patients took the CPH's recommendation, the Recommendation and Anti-Recommendation subgroups will be equal to the control and treatment groups. Therefore, calculating treatment recommendations using the CPH provides little value to the experiments in terms of comparing the models' recommendations.

Appendix C

Simulated data generation

Each patient's baseline information x is drawn from a uniform distribution on $[-1, 1)^d$. For datasets that also involve treatment, the patient's treatment status τ_x is drawn from a Bernoulli distribution with $p = 0.5$.

The Cox proportional hazard model assumes that the baseline hazard function $\lambda_0(t)$ is shared across all patients. The initial death time is generated according to an exponential random variable with a mean $\mu = 5$, which we denote $u \sim Exp(5)$. The individual death time is then generated by

Table 3 DeepSurv's experimental hyper-parameters

Hyper-parameter	Sim linear	Sim nonlinear	WHAS	SUPPORT	METABRIC	Sim treatment	GBSG
Optimizer	sgd	sgd	adam	adam	adam	adam	adam
Activation	SELU	ReLU	ReLU	SELU	SELU	SELU	SELU
# Dense layers	1	3	2	1	1	1	1
# Nodes / Layer	4	17	48	44	41	45	8
Learning rate (LR)	2.922e−4	3.194e−4	0.067	0.047	0.010	0.026	0.154
ℓ_2 Reg	1.999	4.425	16.094	8.120	10.891	9.722	6.551
Dropout	0.375	0.401	0.147	0.255	0.160	0.109	0.661
LR decay	3.579e−4	3.173e−4	6.494e−4	2.573e−3	4.169e−3	1.636e−4	5.667e−3
Momentum	0.906	0.936	0.863	0.859	0.844	0.845	0.887

$$T = \frac{u}{e^{h(x)}}, \text{ when there is no treatment variable,}$$

$$T = \frac{u}{e^{\tau_x h(x)}}, \text{ when there is a treatment variable.}$$

These times are then right censored at an end time to represent the end of a trial. The end time T_0 is chosen such that 90 percent of people have an observed death time.

Because we cannot observe any T beyond the end time threshold, we denote the final observed outcome time

$$Z = \min(T, T_0).$$

Abbreviations
BMI: Body-mass-index; C-index: Concordance-index; CHF: Left heart failure complications; CPH: Cox proportional hazards model; GBSG: The German breast cancer study group; IHC4+C: Immunohistochemical 4 plus clinical; METABRIC: Molecular taxonomy of breast cancer international consortium; MI: Acute myocardial infraction; MIOR: Order of MI; MSE: Mean squared error; NN: Neural network; ReLU: Rectified linear unit; RSF: Random survival forest; SELU: Scaled exponential linear unit; SUPPORT: Study to understand prognoses preferences outcomes and risks of treatment; WHAS: The worcester heart attack study

Availability of data and materials
Project Name: DeepSurv
Project home page: https://github.com/jaredleekatzman/DeepSurv
Archived version: https://doi.org/10.5281/zenodo.1134133
Operating system(s): Platform independent
Programming language: Python
Other requirements: Theano 0.8.2 or higher, Lasagne 0.2.dev1 or higher, and Lifelines 0.9.2 or higher
License: MIT
Any restrictions to use by non-academics: Licence needed
The data that support the findings of this study were published in earlier studies by others and are also available on DeepSurv's GitHub repository.

Authors' contributions
JLK, US, AC, JRB, and YK were responsible for the design of the project. TJ helped with data analysis in consultation with JLK and YK. JLK wrote an initial version of the manuscript and incorporated comments from US, AC, JRB, and YK. JLK wrote the software in consultation with US and AC. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Department of Computer Science, Yale University, 51 Prospect Street, 06511 New Haven, CT, USA. [2]Department of Statistics, Yale University, 24 Hillhouse Avenue, 06511 New Haven, CT, USA. [3]Applied Mathematics Program, Yale University, 51 Prospect Street, 06511 New Haven, CT, USA. [4]Yale School of Medicine, 333 Cedar Street, 06510 New Haven, CT, USA. [5]Center of Outcomes Research and Evaluation, Yale-New Haven Hospital, 06511 New Haven, CT, USA. [6]Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, 06511 New Haven, CT, USA. [7]Department of Pathology and Yale Cancer Center, Yale University School of Medicine, 06511 New Haven, CT, USA. [8]Department of Mathematics, University of California, San Diego, 92093 La Jolla, CA, USA. [9]Final Research, Herzliya, Israel.

References
1. RW Y, EA S, DJ K, 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. 2016;315(16):1735–49. https://doi.org/10.1001/jama.2016.3775.
2. Royston P, Altman DG. External validation of a cox prognostic model: principles and methods. BMC Med Res Methodol. 2013;13(1):1.
3. Bair E, Tibshirani R. Semi-supervised methods to predict patient survival from gene expression data. PLoS Biol. 2004;2(4):108.
4. Cheng W-Y, Yang T-HO, Anastassiou D. Development of a prognostic model for breast cancer survival in an open challenge environment. Sci Total Environ. 2013;5(181):181–5018150.
5. Cox DR. In: Kotz S, Johnson NL, editors. Regression Models and Life-Tables. New York: Springer; 1992, pp. 527–41. https://doi.org/10.1007/978-1-4612-4380-9.
6. Liestbl K, Andersen PK, Andersen U. Survival analysis and neural nets. Stat Med. 1994;13(12):1189–200.
7. Street WN. A neural network model for prognostic prediction. In: Kaufmann M, editor. Proceedings of the Fifteenth International Conference on Machine Learning. San Francisco; 1998. p. 540–46.
8. Jerez JM, Franco L, Alba E, Llombart-Cussac A, Lluch A, Ribelles N, Munárriz B, Martín M. Improvement of breast cancer relapse prediction in high risk intervals using artificial neural networks. Breast Cancer Res Treat. 2005;94(3):265–72. https://doi.org/10.1007/s10549-005-9013-y.
9. Biganzoli E, Boracchi P, Mariani L, Marubini E. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med. 1998;17(10):1169–86.
10. Faraggi D, Simon R. A neural network model for survival data. Stat Med. 1995;14(1):73–82.
11. Sargent DJ. Comparison of artificial neural networks with other statistical approaches. Cancer. 2001;91(S8):1636–42.
12. Xiang A, Lapuerta P, Ryutov A, Buckley J, Azen S. Comparison of the performance of neural network methods and cox regression for censored survival data. Comput Stat Data Anal. 2000;34(2):243–57.
13. Mariani L, Coradini D, Biganzoli E, Boracchi P, Marubini E, Pilotti S, Salvadori B, Silvestrini R, Veronesi U, Zucali R, 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 Res Treat. 1997;44(2):167–78.
14. Therneau T, Grambsch PM. Modeling Survival Data : Extending the Cox Model. New York: Springer; 2000.
15. Ishwaran H, Kogalur UB. Random survival forests for r. R News. 2007;7(2):25–31.
16. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. Ann Appl Statist. 2008;2(3):841–60.
17. Ranganath R, Perotte A, Elhadad N, Blei D. Deep survival analysis. In: Doshi-Velez F, Fackler J, Kale D, Wallace B, Weins J, editors. Proceedings of the 1st Machine Learning for Healthcare Conference. Proceedings of Machine Learning Research, vol 56. Northeastern University, Boston, MA, USA: PMLR; 2016. p. 101–14. http://proceedings.mlr.press/v56/Ranganath16.html.
18. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
19. Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-normalizing neural networks. In: Advances in Neural Information Processing Systems; 2017. p. 972–81. arXiv preprint. 1706.02515.
20. Kingma D, Ba J. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). 2015. arXiv preprint arXiv:1412.6980. https://dare.uva.nl/search?identifier=a20791d3-1aff-464a-8544-268383c33a75.

Katzman *et al. BMC Medical Research Methodology*   (2018) 18:24

Page 12 of 12

21. Nesterov Y. Gradient methods for minimizing composite functions. Math Program. 2013;140(1):125–61.
22. Senior A, Heigold G, Ranzato M, Yang K. 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 Acoustics, Speech and Signal Processing. IEEE; 2013. p. 6724–8.
23. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(1):281–305.
24. Harrell FE, Lee KL, Califf RM, Pryor DB, Rosati RA. Regression modeling strategies for improved prognostic prediction. Stat Med. 1984;3(2):143–52.
25. Efron B, Tibshirani RJ. An Introduction to the Bootstrap. New York: Chapman & Hall; 1993.
26. Austin PC. Generating survival times to simulate cox proportional hazards models with time-varying covariates. Stat Med. 2012;31(29):3946–58.
27. Hosmer DW, Lemeshow S, May S. Applied Survival Analysis: Regression Modeling of Time to Event Data. 2nd ed. New York: Wiley-Interscience; 2008.
28. Knaus WA, Harrell FE, Lynn J, Goldman L, Phillips RS, Connors AF, Dawson NV, Fulkerson WJ, Califf RM, Desbiens N, et al. The support prognostic model: objective estimates of survival for seriously ill hospitalized adults. Ann Intern Med. 1995;122(3):191–203.
29. Curtis C, Shah SP, Chin S-F, Turashvili G, Rueda OM, Dunning MJ, Speed D, Lynch AG, Samarajiwa S, Yuan Y, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature. 2012;486(7403):346–52.
30. Lakhanpal R, Sestak I, Shadbolt B, Bennett GM, Brown M, Phillips T, Zhang Y, Bullman A, Rezo A. Ihc4 score plus clinical treatment score predicts locoregional recurrence in early breast cancer. The Breast. 2016;29:147–52.
31. Foekens JA, Peters HA, Look MP, Portengen H, Schmitt M, Kramer MD, Brünner N, Jänicke F, Meijer-van Gelder ME, Henzen-Logmans SC, et al. The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients. Cancer Res. 2000;60(3):636–43.
32. Schumacher M, Bastert G, Bojar H, Huebner K, Olschewski M, Sauerbrei W, Schmoor C, Beyerle C, Neumann R, Rauschecker H. 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. 1994;12(10):2086–93.
33. Altman DG, Royston P. What do we mean by validating a prognostic model? Stat Med. 2000;19(4):453–73.
34. Katzman JL. DeepSurv. GitHub. 2017. https://doi.org/10.5281/zenodo.1134133. https://github.com/jaredleekatzman/DeepSurv.
35. Sobol IM. Uniformly distributed sequences with an additional uniform property. USSR Comput Math Math Phys. 1976;16(5):236–42.
36. Fox BL. Algorithm 647: Implementation and relative efficiency of quasirandom sequence generators. ACM Trans Math Softw. 1986;12(4): 362–76. https://doi.org/10.1145/22721.356187.