Imperial College London

IMPERIAL COLLEGE LONDON

SCHOOL OF PUBLIC HEALTH

A Survey of Machine Learning Methodologies for Survival Analysis

Author:
01210941

Submitted in partial fulfillment of the requirements for the MSc degree in Health Data Analytics and Machine Learning of Imperial College London

September 2021

Abstract

Background: Survival analysis in clinical research deals with the prediction of patient survival given their covariates. The semi-parametric Cox proportional hazards (PH) model and the fully parametric accelerated failure time models have been the statistical gold-standard for survival analysis. However, despite they perform well and are easily interpretable, these models have stringent assumptions that impede realistic modelling and do not allow the detection of complexities in the data.

Aims: This project uses the Cox PH model as the benchmark and surveys three other machine learning models - Random Survival Forest (RSF), eXtreme Gradient Boosting (XGBoost), and neural network based methods (DeepSurv and DeepHit). The models are compared to determine to what extent these more complex methods can improve from the classical method in terms of prediction performance, interpretability, and robustness to censoring.

Methods: This project uses R for survival analysis with the Cox PH model, RSF, and XGBoost, and uses Python for survival analysis using neural networks. The models are compared firstly for overall survival prediction performance measured by Harrell's concordance index (c-index), and secondly for variable importance according to each models' state-of-the-art variable selection method, both on the Primary Biliary Chirrhosis (PBC) dataset. Lastly, the methods are surveyed for robustness to right-censoring on simulated datasets with different censoring rates.

Results: XGBoost ranked first in survival prediction performance (c-index=0.90), and was followed by RSF (c-index=0.88), the Cox PH model (c-index=0.84), and finally the neural networks DeepHit (c-index=0.64) and DeepSurv (c-index=0.58). RSF predicted the most similar Kaplan-Meier curves to the Cox PH model, while XGBoost made extreme predictions of survival and death outcomes, and the neural network based methods made nearly linear predictions. It was found that Harrell's c-index increases with higher censoring rates for all models except for neural networks.

Conclusions: The survival analysis method chosen has relevant implications to patient prognosis and interventions. While XGBoost and RSF outperformed the Cox PH model, they predicted vastly different survival curves. Additionally, the c-index is biased for incomplete observations, producing a false improved performance at higher censoring rates. Therefore, caution should be exercised when choosing a predictive model for survival, and considerations regarding the required prediction and the amount of censoring should be made.

Contents

Chapter 1

List of tables and figures

List of Figures

List of Tables

Chapter 2

Introduction

Survival analysis, also referred to time-to-event analysis or reliability analysis, is utilised in economics, finance, actuarial science, and medicine to predict the time at which an event of interest, such as selling of a stock option, failure of a mechanical part, or death of a patient, occurs. Survival analysis is characterised by understanding the distribution of survival times as a function of covariates or features. Ideally we would observe the time at which each subject under study experiences the event, and that distribution of survival times would then be modeled against covariates or features to either understand how covariate values affect survival times or, as we will see in this study, how covariate information can improve prediction of survival times. In practice, however, for some fraction of the subjects, we will only observe that they have not yet experienced the event after some period of time, which would mean that we only observe a lower bound to their survival time; this is called *censoring*, or more specifically, *right censoring*. This can happen if we stop the study after a fixed time interval, or if subjects drop out of the study, for example. Survival analysis methods are developed to account for such censoring, which distinguishes these methods from more traditional regression or supervised learning methods. Survival outcomes are characterised by two variables rather than one: the last time at which a subject's information is recorded, and the subject's status (say, dead/alive) at that time. Types of censoring and how they are included in the solution to survival models will be explored in section 3.1. An advantage of using the dual response variable of time and status for survival analysis as opposed to classification lies in the fact that it allows the prediction of personalised full survival curves and the change in risk at all points in time, rather than a binary response that might be tied to a particular time [16]; instead of just predicting survival rates at, say, 5 years after the start of the study, we can model the entire survival experience both individually and in aggregate.

Survival times can be modeled using various positive-valued probability distributions, like the exponential, Weibull, and Gamma distributions; for instance, the Weibull distribution is known to describe the changing risk of death over time of a human lifetime, and additional distributions will be explored in section 3.2. It is also quite widely modeled semi-parametrically or nonparametrically under certain

assumptions. This project will look at survival analysis in a medical setting, which is utilised for instance to determine whether a treatment affects patient survival in clinical trials and what risk factors affect patient survival in a disease. This information aids public health decisions such as when to screen for cancer or how treatments might differ for different stages of a disease [14]. The methods that we discuss in this thesis are, of course, not restricted to medical or healthcare applications, but can be applied to various other settings including manufacturing, equipment maintenance, reliability of products, customer analytics and others.

This project will survey traditional statistical methods (Cox proportional hazards regression), traditional machine learning methods applied to survival analysis (random survival forests, boosting, and penalised regression), and deep learning methods for survival analysis. For each method, a background of the theory will be provided, examining the loss functions to be optimised, how censoring is incorporated, and what predictions can be provided. Additionally, for each method, datasets will be simulated to assess the robustness of performance of the methods at different levels of censoring, and how much data is needed as complexity of models increases.

> ”There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown.” - Leo Breiman, Statistical Modeling: The Two Cultures

The appeal of this topic lies in the extent to which predictive machine learning methods are able to handle the peculiarities of survival data such as right-censoring. Leo Breiman, a professor of statistics at Berkeley, made a career out of exploring the discrepancies and similarities between the two worlds of statistics and machine learning, which he collected in the pivotal 2001 paper ”Statistical Modeling: The Two Cultures” [1]. In statistics, or the ”data modelling culture”, the goal is to discover a model that better explains a given relationship, whereas in machine learning, the ”algorithmic modelling culture”, the goal is to better predict the output given new input data. Breiman argues that the use of algorithmic modelling is more scientific because it does not assume that there is a correct model, it uses success at prediction as model selection, does not avoid producing complex models for complex relationships, and is driven by problem-solving in practice. Particularly, Breiman argues against the use of p-values in favour of predictive tests. We will evaluate the performance of our portfolio of models using predictive performance metrics (Section 5.1), seeing how various models perform on a standard dataset as well as with simulated data.

Chapter 3

Background

3.1 Survival data and descriptive methods

Survival data is characterised by the presence of censoring, i.e. incomplete observations arising either from the event of interest not having occurred at the end of the follow-up period or due to dropout (*right censoring*; $[y, +\infty)$), or the event having occurred before the beginning of the follow-up period (*left censoring*), or finally the event having occurred in between follow-up times (*interval censoring*). Right censoring is the most common type of censoring in survival data, and will be considered in the text hereafter [8]. A crucial assumption is that censoring is **independent of survival time**, so the chance that an individual's data is censored is not affected by their risk of experiencing an event. There are situations where we may have *informative censoring* where the censoring may depend the subjects' survival risk (for example, dropping out of a study for being too sick); these situations need to be dealt with on a case-by-case basis. For this study, we will assume that we are only dealing with independent right censoring.

Descriptive analysis of survival data often involves nonparametric estimation of the survival and hazard functions, described below. These give an overall view of the patterns in the data, either marginally or stratified by some predictors of interest. For a univariate description of survival data, estimators such as the sample mean and variance are inappropriate due to the presence of censored observations. To describe this data we define its cumulative distribution function (CDF) for the random variable survival time *T*, which is the probability that a subject selected at random will have a survival time less than or equal to some stated value *t*.

$$F(t) = Pr(T \leq t)$$

However, survival data is better described by the chance of surviving at least for a particular length of time, which has some analytic advantages. Therefore, the survival function S(t) is preferably used for descriptive purposes:

$$S(t) = Pr(T > t) \tag{3.1}$$

Note that $F(t) + S(t) = 1$ for all times $t > 0$.

The hazard function $h(t)$ describes the *instantaneous* risk of an individual, i.e., the risk of dying (or experiencing the event being followed) at a time t, given the event has not yet occurred till just before time t. Conceptually, this answers the question of whether I'll die tomorrow given I've lived until today; the answer to this question obviously changes at different life stages, so this quantity is of interest for modeling, and is what is often modeled in survival analysis. The mathematical definition of a hazard function is

$$h(t) = \frac{f(t)}{S(t-)}, \tag{3.2}$$

where f(t) is the probability density function of the survival time random variable and S(t-) is the survival function at a time just before t, i.e., $S(t-) = \lim_{s \uparrow t} S(s)$. Therefore, the hazard function defines the conditional probability of the event occurring at time *t* given that it has not occurred until that time. Mathematically, for continuous survival times, the following relation holds.

$$h(t) = -\frac{d}{dt} \ln S(t)$$

The *cumulative hazard*, defined as

$$\begin{aligned} H(t) &= \int_0^t h(s)\, ds \\ &= -\ln S(t) \end{aligned}$$

is often modeled against covariates in survival analysis, since it has some nice analytic properties and can be estimated quite easily. The cumulative hazard function can be nonparametrically estimated from observed data with the Nelson-Aalen estimator

$$H(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}.$$

where d_i are the number of deaths (events) observed at time t_i, and n_i are the number of individuals at risk just prior to t_i.

The survival function $S(t)$ can also be estimated nonparametrically using the *Kaplan-Meier* or product-limit estimator, defined as

$$\hat{S}(t) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

where the notation is the same as that for the Nelson-Aalen estimtor above.

The Kaplan-Meier and Nelson-Aalen estimators give us consistent non-parametric estimators of the survival and cumulative hazard functions under the independent censoring assumption, and assuming that there is a common survival distribution for all the subjects in the study. However, we typically believe that survival times can be affected by different covariates or features, such as sex, age, family history, genetics,

environmental factors, medications and others. So we often *model* the survival times against various covariates using parametric and semi-parametric models, which also account for right-censoring. These regression or supervised learning models are the focus of this study.

3.2 Regression models for survival data

Regression models for survival analysis are specified by a hazard function $h(t)$ built of two components which accomplish two separate goals: the error component describes the underlying distribution of survival time, and the systematic component describes how the distribution changes as a function of the covariates. In parametric and semi-parametric models, described below, the systematic component is specified as a parametric regression function, generally of the form $\eta(x) = \sum \beta_i x_i$ where the x_i are (possibly functions of) covariates or features.

3.2.1 Semiparametric models

Semiparametric regression models do not specify the error component, i.e. the underlying distribution of survival time. Therefore, inferences are only based on the systematic component of the hazard function and estimate parameters defining relative risk between groups characterised by different covariates. Cox (1972) [4] was the first to propose such a model, defining the hazard function as

$$\lambda(t, x, \beta) = \lambda_0(t)e^{x\beta}$$

thus the hazard ratio between two values of the covariates can be written as

$$HR(t, x_1, x_0) = e^{\beta(x_1 - x_0)} = \frac{e^{\beta x_1}}{e^{\beta x_0}}$$

Note that the change in the hazard does not depend on time t, and so the change is the same regardless of the time of observation. This *proportional hazards* assumption is central to the Cox model.

The appeal of semiparametric models lies in the fact that the underlying distribution of the time random variable does not need to be known or specified. In addition to simplifying the model, it also frees it of the stringent assumptions of fully parametric models. Nonetheless, semiparametric models rely on the proportional hazards assumption, whose validity has to be verified by confirming that for different values of the covariate, the cumulative hazard functions are parallel over time [23].

Another appeal of the Cox proportional hazards model is that it can be solved using partial likelihood rather than full likelihood. In fact, Cox showed that when the Breslow estimate of the baseline hazard $\lambda_0(t)$ is substituted into the full likelihood equation the baseline hazards are cancelled out. In fact, the Cox model can only predict risk ratios to compare groups but not absolute risk predictions, for instance at certain timepoints. This is because absolute risk estimation also needs to account

for the baseline hazard, which the Cox model treats as a nuisance parameter and does not estimate. Having said that, we will see that survival can be estimated by the Cox model **along with** some estimate of the the baseline hazard function. This fact is also leveraged by different survival modeling strategies using both statistical and machine learning methods, to provide survival probability predictions.

A method for estimating *parsimonious* models for survival is using **penalized regression**, which optimizes the objective function subject to a penalty constraint on the parameters. This results in nullifying or attenuating parameters which do not improve the model much. If in the full Cox model the objective is to minimise log partial likelihood $\ell(\beta)$, as seen in section 3.2.1, then, when the goal is variable selection and shrinkage, the Lasso penalty can be applied via the method proposed by Tibshirani in the homonymous 1997 paper, which estimates β via

$$\widehat{\beta} = \text{argmin } \ell(\beta), \text{ subject to } \sum |\beta_j| \leq s$$

where $s > 0$ is an arbitrary parameter chosen by k-fold cross-validation to minimise log-likelihood deviance [21]. Other options include ridge regression (penalty is the L_2-norm of the parameters) and elastic net, which is a weighted combination of the ridge and lasso penalties. Penalized regression models are often useful when there is multicollinearity among the predictors, as is common with genomic predictors; the penalty will often suppress the slope estimates of predictors that are collinear with each other in favor of keeping one of them in the model.

3.2.2 Fully parametric models

Fully parametric regression models for survival specify both systematic and error components, i.e. the underlying distribution of the time random variable is defined upon previous knowledge of the data. Accelerated failure time (AFT) models are a class of parametric survival models that take their name from the fact the the covariate is multiplicative on the time scale, effectively accelerating survival time. Various distributions are used to describe the survival process in these models, as seen below.

Parametric model	Hazard function	Survival function
Exponential	γ	$exp(-\gamma t)$
Weibull	$\gamma\alpha t^{\alpha-1}$	$exp(-\gamma t^{\alpha})$
Gompertz	$\gamma exp(\alpha t)$	$exp(-(\gamma/\alpha)(e^{\alpha t}-1)$
Log-logistic	$\alpha\gamma^{\alpha}t^{\alpha-1}/(1+(\gamma t)^{\alpha})$	$1/(1+(\gamma t)^{\alpha})$

In an exponential regression model, the error component is distributed exponentially, hence the probability of the event is the same for every time point. The model is linearised by taking the natural log of both sides, resulting in the equation

$$ln(T) = \beta_0 + \beta_1 x + \epsilon^*$$

The Weibull regression model is a generalisation of the exponential regression model whereby the shape of the parameter describing the distribution of the error term can

be different than 1, resulting in the equation

$$ln(T) = \beta_0 + \beta_1 x + \sigma * \epsilon^*$$

AFT models are estimated by maximum likelihood estimation. Censored data points are accounted for in the likelihood using the survival function, while uncensored data points contribute the p.d.f. to the likelihood. If we write the AFT model as

$$Y_i = x_i^T \beta + \sigma W_i$$

then we can write the likelihood as

$$\begin{aligned} L(\beta, \sigma | y, d) &= \Pi_i \{\sigma^{-1} f(w_i)\}^{d_i} \{S(w_i)\}^{1-d_i} \\ &= \Pi_i \{\sigma^{-1} h(w_i)\}^{d_i} S(w_i) \end{aligned}$$

where f, h and S represent the density, hazard and survival functions for the error distribution, $w_i = (y_i - x_i^T \beta)/\sigma$, and d_i is the censoring indicator, which is 1 if the event occurs and 0 if the individual is censored at the ith time.

3.3 Machine learning models for survival data

Survival analysis can benefit from a machine learning approach in order to relax assumptions made in statistical models, particularly the proportional hazards assumption which assumes that covariates are multiplicatively related to the hazard. This allows for more realistic modelling and for more complex potentially non-linear relationships between covariates and the hazard [3]. This approach replaces the linear parametric form of the systematic component with a more flexible form based on machine learning models. Additionally, survival analysis may benefit from a machine learning approach due to its capacity to make predictions, useful in a healthcare setting for clinical prognoses [15]; the optimization of these algorithms is based on predictive performance of the models on test data, after training these models on independent training data. The validation of the performance of machine learning models on independent data sets is a central tenet of the machine learning method and allows for more robust predictive models as well as an evaluation of potential *overfitting* of models. Another aspect addressed by machine learning models is the *bias-variance tradeoff*, where we might obtain a slightly biased but more precise predictive model on evaluating our test set errors.

Classical machine learning methods for survival analysis build on decision trees: Random Survival Forests (RSFs) and gradient boosting. Additionally, neural networks, a.k.a. deep learning has also been adapted to analyse survival data.

3.3.1 Random Survival Forests

Random Survival Forest (RSF) is a specific random forest (RF) algorithm for survival data developed by Breiman (2003) [2], distinct from other forest approaches.

Random forests for survival consist of (a) binary survival trees with their associated Nelson-Aalen cumulative hazard function as base learners, and (b) a cumulative hazard function formed by the average of each tree's Nelson-Aalen cumulative hazard function as the ensemble [10]. Breiman's RSF algorithm additionally ensures that all aspects of growing the forest take into account the right-censored nature of the survival outcome, meaning that the splitting criterion to grow a tree takes into account both survival time and censoring information, and node impurity measures separation by survival difference [9]. RSFs therefore allow prediction of a new individual's survival probability via the ensemble's cumulative hazard function for the given event time [19].

Following is a description of the RSF algorithm based on Breiman's 2003 forest method:

1. Draw B boostrap samples from the original data, where each bootstrap sample leaves 37% out-of-bag (OOB) data
2. Grow a survival tree recursively for each bootstrap sample
    - Each node only considers p random candidate variables
    - The best split over variables x and split values c is the variable x^* at value c^* that maximises survival difference between daughter nodes. This is often done using the *log-rank statistic*.
    - As the tree becomes deeper, nodes become more homogeneous
3. Reach the constraint that a node should have no less than $d_0 > 0$ unique deaths, indicating that this is the terminal node.
    - Terminal nodes, denoted τ, are the final nodes of a saturated tree
    - In a terminal node $h \in \tau$, the survival time and censoring information for each individual is $(T_{1,h})(\delta_{1,h}), ..., (T_{n(h),h})(\delta_{n(h),h})$, where $\delta_{i,h} = 1$ if individual i experienced the event at time $T_{i,h}$, and $\delta_{i,h} = 0$ if individual i is right censored at time $T_{i,h}$.
4. Calculate the Nelson-Aalen estimate of the cumulative hazard function for each tree
$$H(t) = \sum_{t_{l,h} \leq t} \frac{d_{l,h}}{Y_{l,h}}$$
5. Calculate the ensemble CHF by averaging over B survival trees
$$H_e^{**}(t|x_i) = \frac{\sum_{b=1}^{B} I_{i,b} \hat{H}_b(t|x_i)}{\sum_{b=1}^{B} I_{i,b}}$$
where $I_{i,b}$ =1 if i is an OOB point for b, otherwise it is 0.
6. Using OOB data calculate prediction error for the ensemble CHF

Usually the *log-rank statistic* is used to determine splits for each survival tree in the forest, but other choices are available. For example, Schmid, et al [20] suggest that Harrell's C-statistic may be a better statistic to use for determining splits.

An advantage of RFs is the use of deep trees, i.e., trees with many layers of branches. Another advantage of RFs is the randomisation at two levels used to grow the trees: firstly, each tree is grown from a different bootstrap sample, secondly, a tree is grown by splitting nodes based on a random subset of features [11]. These features help reduce the effects of overfitting and allow the forest to capture multiple aspects of the data within the training process.

Random Survival Forests also have the useful intrinsic ability to compute variable importance (VIMP), which can be considered a machine learning alternative to p-values. The use of p-values to make statements about a scientific effect has been criticised in recent years, with a Nature article asserting that p-values are "not as reliable as many scientist assume". This is due to the fact that the p-value indicates significance of the variable within the model itself, therefore depends on the strong assumption that the model correctly represents the scientific effect studied, which factors can interfere with, such as unreliability of goodness of fit measures, unconsidered interacting effects, and instability to sample size [17]. Whilst there are multiple types of VIMP measures, the most common on is permutation, or Breiman-Cutler importance. To calculate a variable j's permutation importance, the variable's 37% OOB data is permuted, the new covariates are passed through the RSF, and OOB error is recalculated. The original OOB error is subtracted from the noised up OOB error such that a largely positive resulting value signifies importance of the variable because the addition of noise increased error [23].

3.3.2 Gradient Boosting

Gradient boosting is another method which uses decision trees as the building blocks to build more powerful prediction models with the premise that an ensemble of weak learners outperforms one strong learner, similarly to random forest (described in section 3.3.1). In boosting, however, decision trees are added in a step-wise fashion such that each successive tree is fit utilising the residuals of the previous tree as the response rather than the survival outcome, making the trees within the ensemble complementary to each other. In practice, this means that successive trees essentially correct the errors of their prior trees to maximise prediction performance, effectively reducing bias. Prediction performance is maximised because a single decision tree risks producing a high bias thus overfitting and low generalisability, whilst this sequential fashion allows for slow learning which can prevent reaching the point of overfitting [12].

Following is a description of the boosting algorithm [12]:

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set
2. For each bootstrap sample $b = 1, 2, ..., B$:

(a) Fit a tree to that bootstrap sample $\hat{f}^b$ with d splits to the training data (X, r), where the features are the covariates X and the response are the residuals of the previous tree r

(b) Update $\hat{f}$ by adding in a shrunken version of the new tree

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

(c) Update the residuals

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

3. This results in a large number of decision trees $\hat{f}^1, ..., \hat{f}^B$, whose sum produces the output of the boosted model

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

An important tuning parameter to mention in boosting is the shrinkage parameter λ, which sets the learning rate of boosting. With small λ values the algorithm learns more slowly thus requires more trees in the ensemble to reach a satisfactory performance, and vice versa [12]. This concept of learning rate is comparable to the learning rate of neural networks, which will be described in section 3.3.3. Similarly, another boosting tuning parameter is the number of boosting iterations M. Each boosting iteration reduces the loss, therefore a large M may seem desirable, however, this risks learning the training data too well, increasing bias, and reducing generalisability of the model. Therefore, the optimal number of iterations is found using the validation set. Again, this concept is comparable to early stopping in neural networks. The commonality between boosting and neural networks in tuning these two hyperparameters to reduce bias is a ubiquitous notion in machine learning, and is the reason why models that learn more slowly produce better results [7].

Whilst many non-linear tree-based machine learning algorithms exist, eXtreme Gradient Boosting (referred to hereafter as XGBoost) proposed by Chen et al. in 2016 [3] will be utilised in this project as it is the most widely recognised and flexible. It is a very successful Python and R implementation of gradient boosting and was in fact used in 17 of the 29 winning solutions in Kaggle's 2015 competition [3]. XGBoost is capable of analysing survival data as it can incorporate both the Cox proportional hazards model and AFT model as learning objectives. By using the Cox regression as learning objective, XGBoost minimises the partial likelihood loss function (described in section 3.2.1) and is able to predict risk scores.

The measure of variable importance chosen for the XGBoost model is Shapley additive explanation (SHAP) values. These, similarly to VIMP values used for the RSF model (described in section 3.3.1), describe the importance of a feature to making predictions, according to the model. SHAP dependence plots can be generated with the feature on the x-axis and the SHAP value on the y-axis, to indicate the importance of the feature conditional on the value of that feature. The possibility of generating

dependence plots gives an advantage to the use of SHAP values compared to VIMP, because they do not assume that the importance of a feature is constant. Instead, understanding a possible non-linear relationship between a feature's importance and its values can be informative in practice. For instance, clinicians may be able to make different decisions on which features to use as risk factors when examining a patient's risk, depending on the patient's values for that covariate. As exemplified by Li et al. (2020) [15], if an 80-year old individual with cancer has a SHAP value for age of 0.4, then an age of 80 years is a contributing factor to predicted mortality, on the other hand, if a 40-year old individual's SHAP value for age is -0.2, then an age of 40 years is slightly contributing in favour of predicted survival. Consequently, SHAP values can aid clinical decision-making by informing on the effect of a patient's current level of given measurements.

3.3.3 Neural network based methods, a.k.a deep learning

The goal of neural networks for survival analysis is to learn, without any previous knowledge provided by the data analyst, the relationship between the patient's features and the their outcome, utilising the relationship's complexities and non-linearities in the description of said relationship. This is different from previously described methods because the distribution of survival times is learned by the network rather than unspecified (as in the Cox regression) or specified arbirtarily (as in AFT regressions and boosting) [14]. The type of outcome that can be obtained depends on the types of neural network-based method - three exist for survival analysis: those based on classification networks, those based on time-encoded networks, and those based on risk networks. This project will examine the latter, which consists of feed-forwards neural networks outputting the risk of failure [13]. Their use was introduced by Faraggi and Simon in 1995, which proposed a single hidden layer network with two or three nodes capable of modelling a non-linear Cox regression. The Faraggi-Simon method replaces the linear combination of features βx_i with the output of the network $g(x_i, \theta)$ [5].

The Python package `pycox` runs survival analysis and makes predictions using the `PyTorch` neural network framework. Specifically, this project will examine the survival models DeepSurv[1] and DeepHit[2] (whose code is available on the respective authors' GitHub) that are incorporated in the `pycox` package. DeepSurv is a continuous-time model consisting of feed-forward neural network backend with a linear activation function to output a risk equivalent to a non-linear Cox proportional hazards model. More specifically, it is a deep neural network in which the deep layers are repetitions of one fully-connected layer and one dropout layer (Figure 3.1). In the fully connected layers, by definition, every input is connected to every node in the following layer, this is necessary to obtain a good representation of the data and perform feature extraction. A dropout layer, on the other hand, acts as regularisation by randomly removing one node's incoming and outgoing connections, and is required after each fully connected layer to undo the overfitting it produces as a side

[1] https://github.com/jaredleekatzman/DeepSurv
[2] https://github.com/chl8856/DeepHit

effect of learning information. Finally the output of the DeepSurv neural network is a single layer whose linear activation function outputs the log-risk function of the Cox proportional hazards model, which is a product of the baseline hazard function and the risk score, as below.

$$\lambda(t|x) = \lambda_{(0)}(t)e^{h(x)}$$

The objective function of the DeepSurv neural network is the negative log of the partial likelihood of the Cox regression averaged for the number of patients in which the event has occurred with an L2 norm regularisation, seen below. The objective function is minimised by gradient descent to obtain the weights θ of the neural network.

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

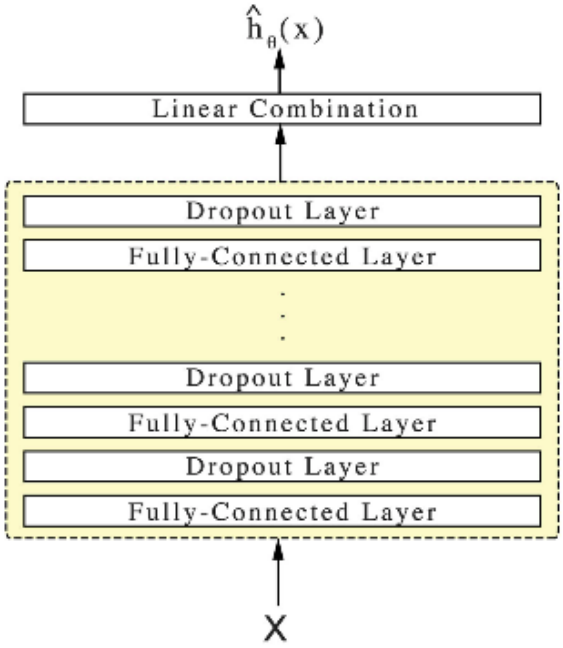

Figure 3.1: DeepSurv neural network architecture (taken from Katzman et al., 2018).

DeepHit is a discrete-time neural network model that incorporates competing risks in survival analysis. It consists of one initial shared multilayer perceptron network, followed by K cause-specific sub-networks for K competing events (Figure 3.2). The role of the common sub-network is to intake clinical features and output a feature matrix common to the K competing events. The role of the K cause-specific sub-networks is to intake both the commonalities given by the common sub-network and the individualities given by the raw covariates, and output the probability of the first hitting time of cause k. The K cause-specific sub-networks are therefore learning the distribution of the event for each cause in parallel. Given that the goal of DeepHit is to learn the joint distribution of the K competing events and not their marginal distribution, it uses a softmax classifier to predict the survival distribution which allows estimation of the probability of each event k at each time s. [14]. DeepHit can also be adapted to survival analysis with a single event type, in which case the response variable is an m-dimensional binary vector containing m event times and the patient status for each time. In this case the output of the neural network generated by the softmax activation function is the probability of death at each timepoint m+1.

The loss function of the DeepSurv neural network is a weighted sum of two loss functions: $\mathcal{L}_{total} = \mathcal{L}_1 + \mathcal{L}_2$, where $\mathcal{L}_1$ is the log-likelihood loss of the joint distribution of the first event time, and $\mathcal{L}_2$ a loss that incorporates ranking of the causes K.

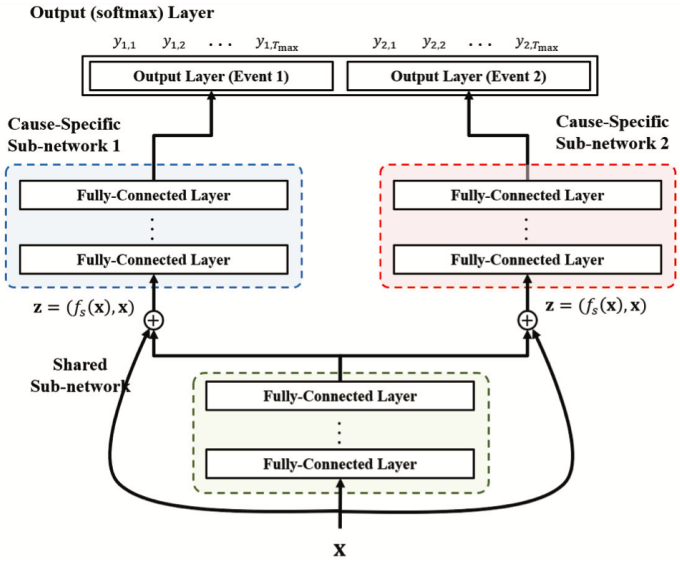

Figure 3.2: DeepHit neural network architecture (taken from Lee et al., 2018).

3.4 Model evaluation metrics

Standard regression error metrics for continuous outcomes such as R^2 and mean squared error (MSE) are unfit to assess model performance in survival analysis due to the presence of censored observations. Therefore, specialised metrics exist for survival scenarios, namely the C-index, Brier score, and mean absolute error (MAE) [24].

3.4.1 Harrell's concordance index

Harrell's concordance index (c-index) is a rank order statistic defined as the ratio of concordant pairs to total comparable pairs. Comparable pairs consist of either (1) two uncensored observations, or (2) one censored observation and one uncensored observation whose event time is smaller than the censored observation's censoring time. Given the pair of comparable observations (i, j) where t_i and t_j are the actual observed times and $S(t_i)$ and $S(t_j)$ are the predicted survival times, the pair (i, j) is concordant if $t_i > t_j$ and $S(t_i) > S(t_j)$, and is discordant if $t_i > t_j$ and $S(t_i) < S(t_j)$. The probability of concordance between rankings of actual and predicted values is therefore calculated for comparable pairs as:

$$c = Pr(\widehat{T}_i < \widehat{T}_j | T_i < T_j)$$

In practice, the c-index indicates the probability that a patient with a high survival time also has a lower risk, in any definition of risk, for instance the presence of a drug or a biomarker [18].

The c-index can consequently also be interpreted as the discriminatory power of a biomarker or combination of biomarkers on health and disease according to a medical study. In fact, the c-index of predictions produced by a machine learning algorithm trained on survival data is exactly the discrimination in the predictive power of the training features [18].

The c-index is affected by the level of censoring in the data. It has been shown to be biased upwards as the proportion of censoring increases [22].

Chapter 4

Methods

4.1 Datasets

4.1.1 Primary Biliary Cirrhosis (PBC) of the liver

The Primary Biliary Cirrhosis (PBC) dataset contains data from a randomized placebo controlled trial of the drug D-penicillamine on individuals affected by PBC, an autoimmune disease which progressively degenerated the liver's bile ducts. The trial was conducted between 1974 and 1984 by the Mayo Clinic, and included 424 cases, 312 of which are complete cases and 112 of which were not randomised but were followed, 6 of which were lost to follow-up [6]. Therefore, the 312 complete and randomised individuals are used for this survival analysis. The PBC dataset used in R is found in the `randomForestSRC` package and does not contain competing risks. However, the PBC dataset used in Python contains a third survival status "liver transplant" in addition to death and censoring which was manually recoded to censored. The PBC dataset in this project was used to compare the prediction performance via c-index and variable selection methods of the four machine learning approaches.

4.1.2 Simulated data

Five datasets of 1000 observations characterised by increasing censoring rates (1%, 25%, 50%, 75%, 99%) were simulated using using the `sim.survdata` function of the `coxed` package in R and the `SimStudyLinearPH` function of the `pycox.simulations` package in Python. The simulated datasets in this project were used to train the four machine learning models on, and obtain a c-index of prediction performance on the test set. This is a sensitivity analysis of censoring rate, showcasing to what extent the four machine learning methods handle censored data, and at what rate of censoring does prediction begin to break down, and ultimately which method is most robust to censoring.

4.2 Cox Proportional Hazards Model

Survival analysis using the Cox proportional hazards model was carried out in R. The Primary Biliary Cirrhosis (PBC) dataset was obtained from the `randomForestSRC` package. The dataset was cleaned as instructed in section 4.1.1 and the resulting observations were divided into 80% training and 20% testing. A Cox PH model model was fit on the training set using the `coxph` function of the `survival` package. Predictions were made on the test set using the `predictSurvProb` function from the `pec` package. The individuals with minimum, median, and maximum ages were identified, and their 10-year survival probabilities and full survival curves were printed using the `predictSurvProb` function. The c-index for prediction performance of the Cox PH model on the test set was computed via the `rcorr.cens` function of the `Hmisc` package.

A variable importance method for Cox regression analysis was Lasso regularisation, described in theory in section 3.2.1. Regularised Cox regression with Lasso penalty was fit on the PBC dataset via the `glmnet` package to determine which of the 17 covariates had the most effect on the survival outcome. The `cv.glmnet` function was used to carry out ten-fold cross-validation, and the optimal lambda value was chosen to be one standard deviation away from the minimum mean squared error. The regularised Cox regression was fit with the optimal lambda value using the `glmnet` function, and coefficients were printed.

To perform censoring rate sensitivity analysis, the simulated datasets with increasing censoring rates described in section 4.1.2 were used. The datasets were divided into 80:20 train-test split, a Cox PH model was fit to each training set, and predictions were made on each test set, with the functions previously described. The c-index was recorded for each censoring rate to examine the extent of prediction breakdown according to the extent of censoring in the dataset.

4.3 Random Survival Forest

The same PBC dataset and train-test split resulting from the previous section 4.2 were utilised. A RSF was trained on the training set using the `rfsrc` function of the `randomForestSRC` package, and, similarly to Cox regression analysis, predictions were made on the test set using the `predictSurvProb` function from the `pec` package. The 10-year survival probabilities and full survival curves were printed using the `predictSurvProb` function for the three individuals chosen during Cox regression analysis with approximately minimum, median, and maximum ages, for comparative purposes. The c-index for prediction performance of the RSF on the test set was computed via the `predict` function of base R.

The variable selection method for RSF was permutation VIMP, described in theory in section 3.3.1. This was extracted from the RSF fit on the PBC dataset via the `vimp` function, and plotted with the `plot` command in order to rank the 17 variables in their contribution to prediction of the survival outcome.

To perform censoring rate sensitivity analysis, the simulated datasets with increasing censoring rates described in section 4.1.2 were used. The datasets were divided into 80% training and 20% testing, a RSF was fit to each training set, and predictions were made on the test set, with the same methods described prior for RSF. The c-index was recorded for each censoring rate and reported.

4.4 Gradient boosting

The gradient boosting library of choice in this project is XGBoost, described theoretically in section 3.3.2, and implemented in R with the `xgboost` and `survXgboost` libraries. Recall that the Cox proportional hazards model upon which XGBoost is implemented on assumes hazard rates $h(t|X) = h_{()}(t)risk(X)$, therefore predicts only risk scores where $risk(X) = exp(X\beta)$. The `xgboost` package can be used to predict survival by setting the learning objective to Cox and the evaluation parameter to negative log likelihood, however its predict method returns only aforementioned risk scores $risk(X)$. The `survXgboost` package[1] provides a thin wrapper to compute full survival curve approximation.

Complete cases of the PBC dataset were split into 80% training and 20% testing, for each of which a matrix X contained the 17 covariates, and a structured array y contained a list of failure times, negative if right-censored. The XGBoost model was trained using the `xgboost` function, with . The model was initially trained for n rounds to determine point of early stopping, which was chosen to be at n. SHAP values were generated for the four most important variables with the `xgb.plot.shap` function.

4.5 Neural network based methods

The neural network based methods examined in this project are DeepSurv and DeepHit, described theoretically in section 3.3.3, and implemented with Python's pycox library[2]. For both neural networks, the PBC dataset was split into 60% training, to train the neural networks, 20% testing, to obtain the prediction performance metric, and 20% validation, to tune the learning rate hyperparamter.

All functions to build the DeepSurv model were obtained from CoxPH from the `pycox.models` package. Firstly, a neural network was constructed with the 17 PBC data covariates as the input features, 32 nodes, one output feature, and a dropout of 0.1 via the `MLPVanilla` function. The neural network was passed through a Cox PH model using the `CoxPH` function with an Adam optimiser to obtain the DeepSurv model. Once the DeepSurv model was fit, it was tuned for the optimal learning rate manually by visualising the smoothed curve of loss over learning rate via the `lr_finder` function, and identifying the optimal learning rate as the central point of the steepest downwards slope of loss, given that smaller learning rates train too

[1] https://github.com/IyarLin/survXgboost/
[2] https://pypi.org/project/pycox/

slowly and higher learning rates train too quickly. Figure 4.1 provides an example of such curve. The optimal learning rate was set to 0.001 and was incorporated into the model using the `set_lr` function. Model training was carried out with a batch size of 256 and 512 epochs. Early stopping was included by using the validation dataset to end model training when validation error starts to increase rather than when the global minimum is reached, in order to prevent overfitting [7], and found that 172 epochs are sufficient to train a generalisable model on the PBC dataset. Predictions on the test set using the DeepSurv model were carried out by computing the baseline hazard with the `compute_baseline_hazards` function followed by applying the `predict_surv_df` function to the results. The c-index was computed with the `pycox.evaluation` via the `concordance_td` command.

The second neural network architecture explored was DeepHit, which was also obtained from the `pycox.models` package. This neural network was built in the same way as its original paper but without the residual connections for simplicity. The same steps as described in the previous paragraph were taken in order to build the network, tune the learning rate hyperparameter, train it with early stopping, make predictions on the test set, and compute the c-index of prediction.

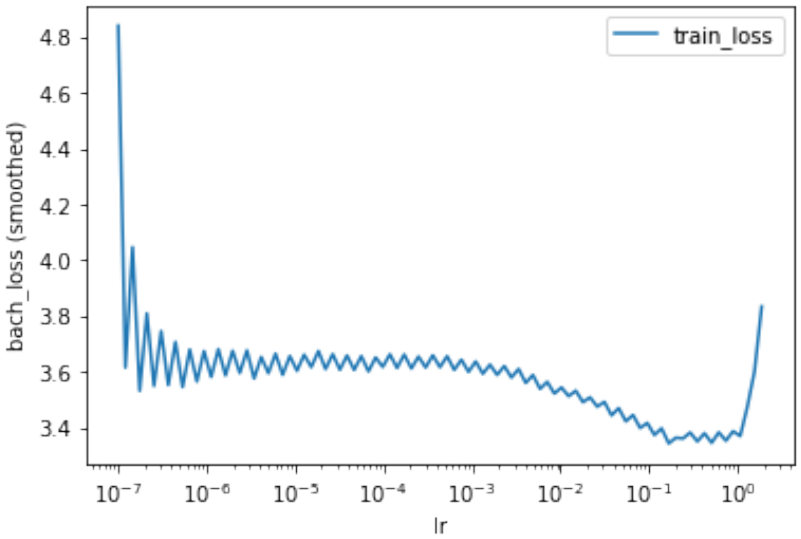

Figure 4.1: DeepSurv loss over learning rate.

Neural network training curves were used to ascertain that that with advancement of epochs the training loss decreases suggesting that the model is learning well, and the validation loss also decreases suggesting that the model is generalising well. This can be confirmed in Figure 4.2 for the DeepSurv model and in Figure 4.3 for the DeepHit model.

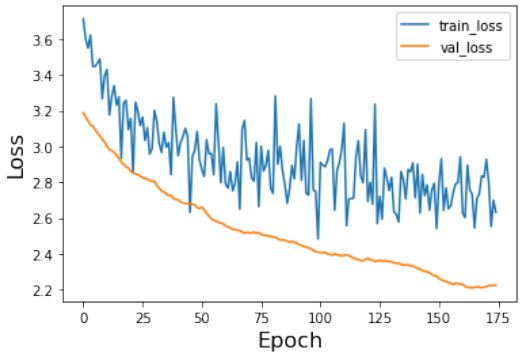

Figure 4.2: DeepSurv learning curve on PBC training set.

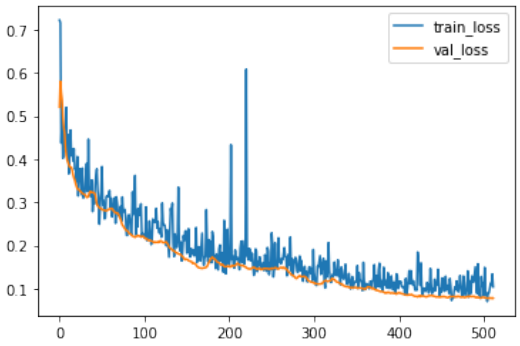

Figure 4.3: DeepHit learning curve on PBC training set.

Chapter 5

Results

5.1 Prediction performance

We split the pbc data into a training and test set, in a 4:1 ratio. We trained each model on the training set and evaluated predictive performance using Harrell's c-index on the test set (Table 5.1). We see that the classical methods (Cox regression, random survival forests, gradient boosting) do much better than the more complex neural network methods for this data. We do note here that the full `pbc` data comprised 276 complete cases, so the performance of the neural network models may be affected by the size of the data.

Model	Cox PH	RSF	XGBoost	DeepHit	DeepSurv
c-index	0.835	0.878	0.901	0.635	0.576

Table 5.1: Harrell's c-index from the PBC test data using various algorithms

For all machine learning methods applied to the PBC dataset, 10-year survival probabilities (Table 5.2) and full survival curve predictions (Figure 5.1 and 5.2) were made for individuals of three different ages: the minimum, median, and maximum present in the test set. The results show great discrepancies between the same individual's predicted survival probabilities among the model used to make such prediction. Random Survival Forest produces the most similar predictions to the Cox PH model, particularly for patients 2 and 3, and coherent with their similar c-indices (Table 5.1). However, RSF can be seen to underestimate survival at lower ages (patient 1) and overestimate survival at higher ages (patient 3) compared to the classical Cox regression. XGBoost makes very extreme predictions on either outcome compared to Cox regression, with an almost complete chance of survival after 10 years for patient 1 and an almost complete chance of death after 10 years for patient 3 – a critical result for the clinical use of prediction models for survivals. The neural network based models predict very similarly shaped curves for each individual in a very linear trend, incoherent with the known survival pattern for human life. DeepHit can also be seen to underestimate survival at younger ages (Figure 5.2). The possible

reasons for these discrepancies are crucial to choosing which model to employ and are examined in the discussion.

Patient ID	Age (days)	Cox	RSF	XGBoost	DeepSurv	DeepHit
1	10550	92.1	73.9	99.9	42.7	22.2
2	16941	17.8	18.8	< 0.0001	19.6	24.7
3	25006	11.1	26.8	1.49	24.3	24.7

Table 5.2: Predicted 10-year survival probabilities for three test set data to all models evaluated.

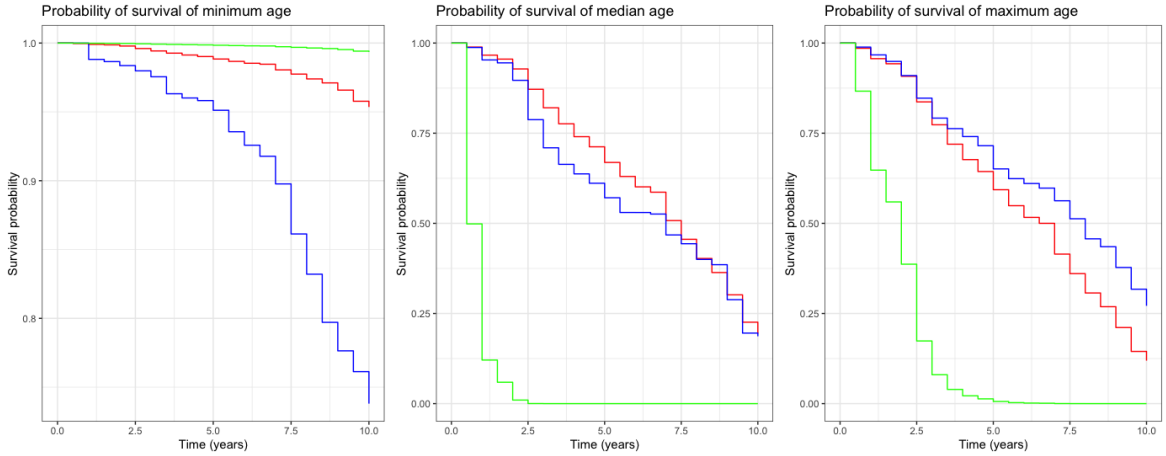

Figure 5.1: Full Kaplan-Meier survival curves for three individuals of increasing age predicted by Cox PH model (red), Random Survival Forest (blue), and eXtreme gradient boosting (green).

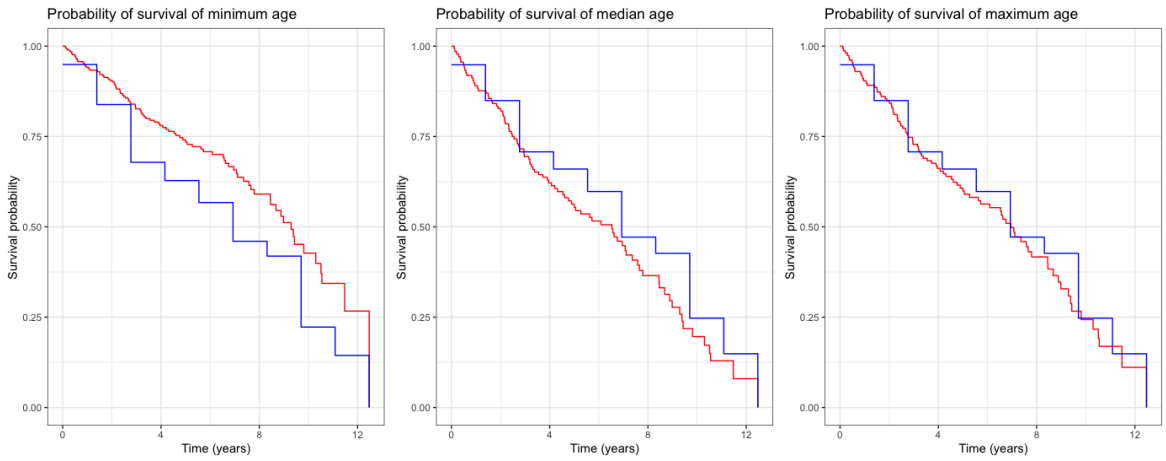

Figure 5.2: Full Kaplan-Meier survival curves for three individuals of increasing age predicted by DeepSurv (red) and DeepHit (blue).

5.2 Variable importance

The benchmark for variable importance measures according to different models is Cox regression p-values as they are the classical means of determining statistical significance of a variable in relation to a response, hence all following variable importance measures will be compared to this gold standard. According to Cox regression p-values, in the PBC dataset, presence of edema shows the most significant association to survival ($p<0.001$), followed by age, urine copper, and histologic stage ($p<0.01$). Serum bilirubin is also moderately significant ($p<0.05$). Secondly, a Lasso penalty was applied in order to keep only the most important variables to prediction of survival, and shrink the remaining's effect sizes to zero. Of the PBC dataset's 17 variables, eight coefficients were shrunk to zero, thereby selecting the remaining as a measure of their relative importance to the association with survival. According to regularised Cox regression effect sizes, in the PBC dataset, presence of edema, albumin, presence of ascites, and histologic stage have largest effect size in relation to survival (Table 5.3).

There is a moderate consistency between variables significant by p-value and variables selected by Lasso penalty. In fact, presence of edema shows the strongest selection according to both. The remaining moderately significant variables, such as age, serum bilirubin, urine copper and histologic stage, are also selected by both variable importance methods. The only major inconsistency is in the ascites variable, which has been selected by the regularised Cox PH model but has an non-significant p-value of 0.85.

Variable	Cox β	Cox p-value	Lasso Cox β
treatment	1.08e-03	0.997	0.0
age	9.58e-05	0.007 **	4.4e-06
sex	-3.05e-02	0.929	0.0
ascites	-8.45e-02	0.848	1.7e-01
hepatom	7.19e-02	0.786	0.0
spiders	-1.98e-01	0.491	0.0
edema	1.58	0.0002 ***	4.5e-01
bili	7.41e-02	0.019 *	7.9e-02
chol	9.94e-04	0.076	0.0
albumin	-6.36e-01	0.068	-3.2e-01
copper	4.14e-03	0.004 **	2.1e-03
alk	-2.17e-05	0.627	0.0
sgot	1.46e-03	0.528	0.0
trig	-1.26e-03	0.414	0.0
platelet	2.13e-04	0.883	0.0
prothrombin	1.84e-01	0.147	3.4e-02
stage	5.20e-01	0.007 **	1.5e-01

Table 5.3: Full Cox regression variable effect sizes and p-values, and Cox regression with Lasso regularisation variable effect sizes.

In the Random Survival Forest, permutation VIMP, explained in theory in section 4.3, was employed to determine variable importance to prediction of survival in the PBC dataset. According to permutation VIMP applied on the RSF model, the most important variable is serum bilirubin (bili) by a great extent, whose inclusion in the model improves survival prediction thus contributes to the c-index by 7%. The variable is followed in equal importance by urine copper, presence of ascites, and presence of edema (Figure 5.3). Interestingly, serum bilirubin was not the most important variable according to the Cox PH model variable importance methods, rather was only moderately significant ($p<0.05$, Lasso β=7.9e-02). In the same manner, the most important variable according to the Cox PH model variable importance methods was presence of edema which is only moderately important according to RSF VIMP (VIMP<0.02).

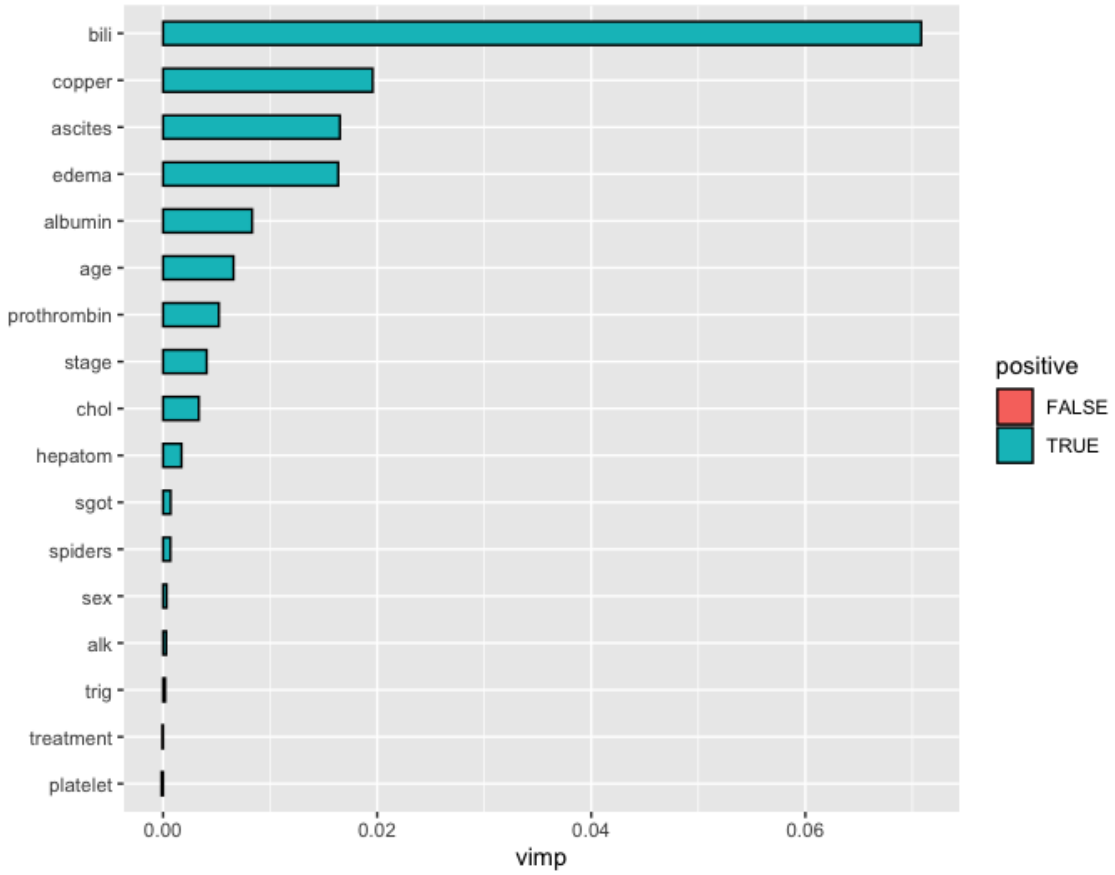

Figure 5.3: Variable importance (VIMP) plot obtained via the permutation method in Random Survival Forest (RSF).

In XGBoost, variable importance was determined by Shapley Additive Exlanation (SHAP) values which, similarly to VIMP, quantify the contribution of variables to the final survival prediction. The advantage of SHAP values is that they are non-linear with respect to a given covariate, meaning that the change in importance of the covariate over its values can be followed by plotting them against each other in dependency plots. According to the SHAP dependency plots (Figure 5.4) the most important variable in the PBC dataset for prediction of survival was serum bilirubin (bili), which positively affects probability of death consistently at values greater than 2.5 mg/dl. Next was age, which positively affects probability of death from ages greater than 48 years. Also important to survival prediction were albumin and serum cholesterol (chol).

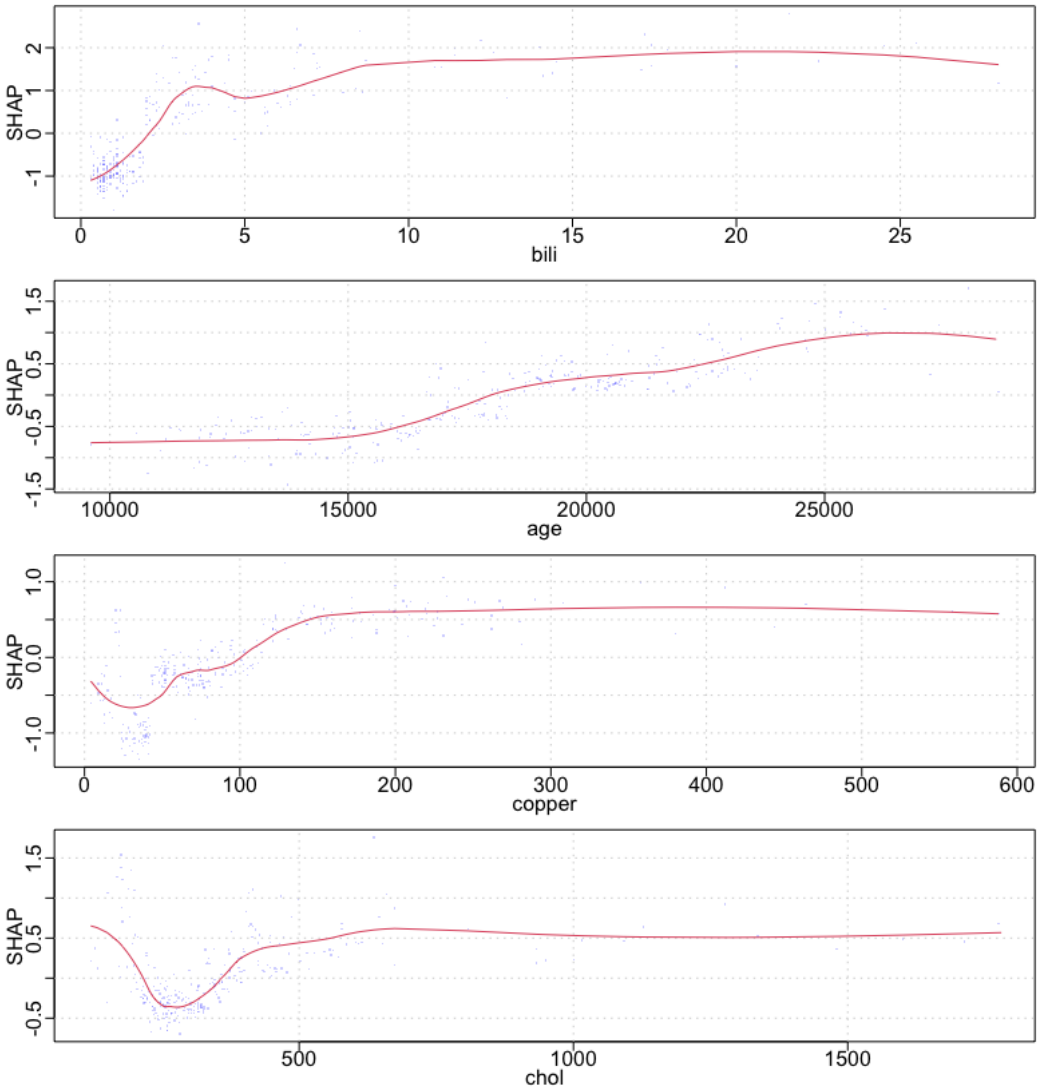

Figure 5.4: SHAP dependence plots for four most important variables in prediction of survival in PBC dataset.

5.3 Censoring rate sensitivity analysis

The benchmark model – Cox proportional hazards - deals with censoring in the minimisation problem of the partial likelihood equation. With increasing censoring rates, its estimated hazard ratio (HR) is increasingly overestimated compared to its true value, and confidence intervals widen increasingly with a large increase at 99%.

Censoring rate	1%	25%	50%	75%	99%
True HR	1.001	0.971	0.899	0.797	0.964
Cox HR	1.004	0.975	0.912	0.819	0.988
95% CI	0.992-1.017	0.961-0.989	0.896-0.928	0.799-0.840	0.874-1.117
HR diff.	0.003	-0.010	0.013	0.022	0.035
CI width	0.025	0.028	0.032	0.041	0.243

Table 5.4: Censoring rate sensitivity analysis for Cox regression.

With regards to the hypothesised breakdown of prediction performance with increasing censoring rates due to the nature of incomplete observations, the c-index was computed on survival predictions on the PBC dataset. The c-index sees a large increase when censoring rates reach 99% when predictions are computed both via the Cox PH model and RSF, implying that prediction performance by these models as summarised by the c-index improves when large proportion of observations are incomplete. However, this increase may be due to the known property that Harrell's c-index over-estimates the concordance in the presence of high levels of censoring, and this property is reflected in these results.

Censoring rate	1%	25%	50%	75%	99%
Cox PH	0.494	0.581	0.533	0.564	0.604
RSF	0.504	0.553	0.505	0.493	0.854
XGBoost	0.511	0.483	0.535	0.524	0.684
DeepSurv	0.991	0.990	0.990	0.984	0.986

Table 5.5: Censoring rate sensitivity analysis for Cox PH, RSF, XGBoost, and DeepSurv prediction performance.

5.4 Software used

All computations were carried out on R version 4.0.3 and Python version 2.7.16. All code is found at the GitHub repository:
https://github.com/camicallierotti/imperial-summer-project.

5.4.1 R packages

Package	Version
dplyr	1.0.7
rms	6.2
tidyverse	1.3.0
pec	2020.11.17
survival	3.2-12
coxed	0.3.3
glmnet	4.1-2
randomForestSRC	2.12.0
xgboost	1.4.1.1
survXgboost	1.0
ggplot2	3.3.5

Table 5.6: R packages used

5.4.2 Python packages

Package	Version
pandas	1.1.3
numpy	1.19.2
sklearn	0.23.2
matplotlib	3.3.2
pycox	0.2.2
pytorch	1.7.1
torchtuples	0.2.0

Table 5.7: Python packages used

Chapter 6

Discussion

6.1 Prediction performance

In section 5.1 we surveyed the performance of four machine learning models of increasing complexity on right-censored survival data: the classical Cox proportional hazards model, Ishwaran's Random Survival Forest, Extreme Gradient Boosting, and neural networks DeepSurv and DeepHit. When predicting survival using the models on the same three individuals from the test set, widely different results were obtained. For instance, a 68-year-old female individual with stage 1 primary biliary chirrhosis has a 10-year survival probability of 92% predicted by the Cox proportional hazards model, of 74% predicted by the Random Survival forest model, and of 99.9% predicted by the XGBoost model. Even more surprisingly, she has the same probability is predicted as low as 42.7% and 22.2% by the neural network models DeepSurv and DeepHit respectively. Understanding the cause of these differences and pinpointing how to determine the reliable most reliable choice is crucial to clinical decision-making.

The benchmark of survival prediction performance on unseen data was the Cox proportional hazards model as it is the classical method of predicting survival. The Cox PH model nonetheless ranked third for the best prediction performance out of the four machine learning methods when predicting survival of new individuals on the PBC dataset in terms of the c-index. Previous studies simulating linear and non-linear survival data found that the Cox proportional hazards model actually outperformed more complex models such as RSF and DeepSurv in terms of predicting survival in the case of simulated data being linear [13]. This suggests that the PBC dataset likely does not contain major non-linearities and complex interaction terms, thus a simple linear model with a stringent proportional hazards assumption is an appropriate choice.

XGBoost ranked first for prediction performance in terms of c-index, hence outperformed benchmark Cox PH model. This is coherent with results in the literature, as XGBoost is an ensemble algorithm, which is powerful due to the capacity of learning slowly thus preventing overfitting and correcting residual error. A drawback of XG-

Boost for survival was that it produced very extreme predictions for 10-year survival probabilities, with almost complete chance of survival in the youngest individual and almost complete chance of death in the older individual. These point predictions widely differed from the previous models' suggesting that XGBoost may not be a well-calibrated model. Interestingly, This discrepancy suggests that XGBoost does in fact outperform classical methods in terms of comparison between concordant pairs, but does not reliably predict the full Kaplan-Meier survival curve. The drastic difference in survival prediction by XGBoost is of clinical relevance because extreme prognoses may underpin potentially unsuitable treatments.

Random survival forest ranked second for prediction performance in terms of c-index, slightly outperforming the benchmark Cox PH model. The Kaplan-Meier survival curves estimated by RSF underestimated survival at low ages and overestimated survival at high ages, while the two methods perform comparably at median ages, a trend coherent with the findings of a previous study by Mogensen et al. (2012), which suggests the cause to be RSF utilising nearest-neighbours and Cox using extrapolation of the middle of age range [19].

Finally, the neural networks ranked last for prediction performance compared to classical survival methods, according to their poor c-indices of prediction and to their greatly incoherent point predictions and Kaplan-Meier survival curve predictions compared to previous methods. For example, for an individual for which previous methods predicted 10-year survival probabilities greater than 73%, the neural network methods predicted less than 43% (Figure 5.2), which makes a great difference clinically regarding a patient's prognosis. Additionally, the neural network methods seem not to accurately predict Kaplan-Meier survival curves, given that they produced similarly shaped curves despite the greatly different covariates of the individuals, and given their linear shape which is an uncommon survival distribution. The low prediction performance of the neural network based model can perhaps be attributed to overfitting: DeepSurv and DeepHit are both deep models, therefore when applied to a small dataset they tend to largely learn statistical noise, and although dropout layers are placed after each fully connected layer in order to prevent this, perhaps the choice of a neural network in the case of the PBC dataset is not ideal. Instead, previous studies predicting survival on a variety of medical datasets of higher dimensionality including WHAS, SUPPORT, METABRIC, and Rotterdam found that DeepSurv performed slighlty better than Cox regression and RSF, as they likely contained many features and observations to learn whilst propagating through its layers [8].

6.2 Variable importance

As introduced, the use of p-values to provide evidence for a scientific effect has become controversial in the last decade. At the same time, machine learning methods have emerged which have shown increased performance in situations characterised by non-linearities, interactions, and other complexities. However, with this advantage comes the pitfall of being considered a "black-box" as they perform very well

but are not as easily interpretable as classical statistical methods, such as the Cox proportional hazards model for survival analysis which produces easily interpretable hazard ratios and associated p-values. These concomitant evolutions in data analysis have allowed VIMP measures to be borrowed from machine learning to determine the effect of a variable in a model [17].

There are few discrepancies between the Cox proportional hazards model p-values and Random Survival Forest's VIMP values for the seventeen covariates of the PBC dataset, as outlined in section 5.2. It is important to note that VIMP values produced by RSF are calculated by change in prediction error therefore hold true regardless of whether the model and its assumptions do, contrary to p-values which are merely a measure of significance within the statistical model. In the case of the PBC dataset the results show that the variables selected both by p-value and by VIMP are largely overlapping, and the only discrepancies lie in the ranking of importance of such selected variables. This suggests that, for this dataset, a more complex model than the benchmark is not required to compute variable importances translatable to reality. Consequently, the result is also an indicator that the assumptions of the Cox PH model on the PBC dataset hold true.

As introduced in section 3.3.2, following the non-linearities of SHAP values can greatly inform the understanding of clinical predictors of survival in disease. In section 5.2 we applied variable importance methods to the four survival models surveyed and compared the most selected variables. Age, for instance, was found to be an important variable to survival prediction in the PBC dataset according to the Cox PH model ($p=0.007$), but SHAP values specify that it only has a predictive effect when age is greater than 48 years. The added information regarding non-linear effects of covariates provided by SHAP values is a valuable tool to inform clinicians on prognosis interventions.

6.3 Robustness to right-censoring

In section 5.3 we simulated datasets containing different rates of censoring and applied the four survival models in order to make predictions and survey their performance at increasing censoring rates. The benchmark Cox proportional hazards model showed proved fairly robust in hazard ratio predictions at different censoring rates, with the only pitfall being widening of the 95% confidence interval when censoring reaches a rate of 99%. With regards to the c-index, the classical models appear to perform increasingly better at predicting survival with a great improvement at 99% censoring rates. However, this can be attributed to Harrell's c-index's handling of incomplete cases rather than to a true improvement in performance: as less complete cases are present, less are used to compute the statistic and a bias is introduced. Therefore, it is essential to take into account a dataset's censoring rate when determining a model's predictive ability via the concordance index.

Despite the poor prediction performance of the neural networks, seen in the previous section 6.2, DeepSurv and DeepHit revealed themselves very robust to censoring,

producing differences in the c-index among different censoring rates in the order of thousandths. Robustness to censoring is a very important feature of neural networks, as it is expected for all models to perform more poorly as censoring increases due to information missingness thus higher bias and a more inaccurate estimated survival function.

Chapter 7

Conclusions and future work

This project found that XGBoost ranked first in predicting overall survival of individuals in the PBC dataset, however it made extreme predictions of the full Kaplan-Meier curve, suggesting that it is a good choice to classify a large number of patients but not to provide a prognosis for individual patients; it may not be a *well-calibrated* model. Deep learning methods ranked last in predicting survival and predicted unrealistic Kaplan-Meier survival curves. These models may have been handicapped by the small number of observations in the data, which may be insufficient for proper learning and optimization of the neural network models. Overall, Random Survival Forest was the optimal solution between these extremes, with a fair prediction performance and Kaplan-Meier survival curves similar to those of the Cox proportional hazards model.

Given that the PBC dataset is a classical clinical dataset in terms of features and observation size, the results of this project can be directly translated to the datasets used in clinical studies. However, given the growing dimensionality of data collected in clinical research in recent years thanks to high throughput methods, neural networks may prove very useful when feature interactions and non-linearities are too difficult or expensive to uncover manually. In these cases, the neural network's architecture will have the advantage of learning these complexities during training, while a simpler model such as the Cox regression would treat the features as independent terms unless explicitly specified. To build on the results of this project, I believe repeating the study with a more complex dataset would favour the performance of the more complex neural network based methods, in order to ascertain whether these are needed in the big data era, or if the Cox proportional hazards model suffices to predict survival without the expenditure of excessive computational power.

Chapter 8

References

[1] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16:199–231, 2001. pages 5

[2] Leo Breiman and Adele Cutler. Manual–setting up, using, and understanding random forests v4.0. 2003. *URL https://www. stat. berkeley. edu/˜ breiman/Using_random_forests_v4.0. pdf*, 2011. pages 10

[3] Yifei Chen, Zhenyu Jia, Dan Mercola, and Xiaohui Xie. A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index. *Computational and Mathematical Methods in Medicine*, pages 1–8, 2016. pages 10, 13

[4] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972. pages 8

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

[6] Thomas R. Fleming and David P. Harrington. *Counting Processes and Survival Analysis*. 1991. pages 18

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Texts in Statistics. Springer New York, 22 edition, 2017. pages 13, 21

[8] David W. Hosmer, Stanley Lemeshow, and Susanne May. *Applied Survival Analysis: Regression Modeling of Time-to-Event Data*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2008. pages 6, 31

[9] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3), September 2008. pages 11

[10] Hemant Ishwaran, Udaya B. Kogalur, Xi Chen, and Andy J. Minn. Random survival forests for high-dimensional data: Random Survival Forests for High-Dimensional Data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):115–132, February 2011. pages 11

[11] Hemant Ishwaran and Min Lu. Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Statistics in Medicine*, 38(4):558–582, February 2019. pages 12

[12] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. Springer New York, New York, NY, 2013. pages 12, 13

[13] Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):24, December 2018. pages 14, 30

[14] Changhee Lee, William Zame, and Jinsung Yoon. DeepHit: A Deep Learning Approach to Survival Analysis with Competing Risks. page 8, 2018. pages 5, 14, 16

[15] Richard Li, Ashwin Shinde, An Liu, Scott Glaser, Yung Lyou, Bertram Yuh, Jeffrey Wong, and Arya Amini. Machine Learning–Based Interpretation and Visualization of Nonlinear Interactions in Prostate Cancer Survival. *JCO Clinical Cancer Informatics*, (4):637–646, September 2020. pages 10, 14

[16] Pei Liu, Bo Fu, Simon X. Yang, Ling Deng, Xiaorong Zhong, and Hong Zheng. Optimizing Survival Analysis of XGBoost for Ties to Predict Disease Progression of Breast Cancer. *IEEE Transactions on Biomedical Engineering*, 68(1):148–160, January 2021. pages 4

[17] Min Lu and Hemant Ishwaran. A Machine Learning Alternative to P-values. *arXiv:1701.04944 [cs, stat]*, February 2017. arXiv: 1701.04944. pages 12, 32

[18] Andreas Mayr and Matthias Schmid. Boosting the Concordance Index for Survival Data – A Unified Framework To Derive and Evaluate Biomarker Combinations. *PLoS ONE*, 9(1):e84483, January 2014. pages 17

[19] Ulla B. Mogensen, Hemant Ishwaran, and Thomas A. Gerds. Evaluating Random Forests for Survival Analysis Using Prediction Error Curves. *Journal of Statistical Software*, 50(11), 2012. pages 11, 31

[20] Matthias Schmid, Marvin N. Wright, and Andreas Ziegler. On the use of Harrell's C for clinical risk prediction via random survival forests. *Expert Systems with Applications*, 63:450–459, November 2016. pages 12

[21] Robert Tibshirani. The Lasso Method for Variable Selection in the Cox Model. *Statistics in Medicine*, 16:385–295, 1997. pages 9

[22] Hajime Uno, Tianxi Cai, Michael J. Pencina, Ralph B. D'Agostino, and L. J. Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in Medicine*, 30(10):1105–1117, 2011. pages 17

[23] Hong Wang and Gang Li. A Selective Review on Random Survival Forests for High Dimensional Data. *Quantitative Bio-Science*, 36(2):85–96, November 2017. pages 8, 12

[24] Ping Wang, Yan Li, and Chandan K. Reddy. Machine Learning for Survival Analysis: A Survey. *arXiv:1708.04649 [cs, stat]*, August 2017. arXiv: 1708.04649. pages 17