An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes

Other titles in Foundations and Trends® in Machine Learning

Automated Deep Learning: Neural Architecture Search Is Not the End
Xuanyi Dong, David Jacob Kedziora, Katarzyna Musial and Bogdan Gabrys
ISBN: 978-1-63828-318-8

AutonoML: Towards an Integrated Framework for Autonomous Machine Learning
David Jacob Kedziora, Katarzyna Musial and Bogdan Gabrys
ISBN: 978-1-63828-316-4

Causal Fairness Analysis: A Causal Toolkit for Fair Machine Learning
Drago Plečko and Elias Bareinboim
ISBN: 978-1-63828-330-0

User-friendly Introduction to PAC-Bayes Bounds
Pierre Alquier
ISBN: 978-1-63828-326-3

A Friendly Tutorial on Mean-Field Spin Glass Techniques for Non-Physicists
Andrea Montanari and Subhabrata Sen
ISBN: 978-1-63828-212-9

Reinforcement Learning, Bit by Bit
Xiuyuan Lu, Benjamin Van Roy, Vikranth Dwaracherla, Morteza Ibrahimi, Ian Osband and Zheng Wen
ISBN: 978-1-63828-254-9

An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes

George H. Chen
Carnegie Mellon University
georgechen@cmu.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

G. H. Chen. *An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes*. Foundations and Trends® in Machine Learning, vol. 17, no. 6, pp. 921–1100, 2024.

ISBN: 978-1-63828-455-0

Foundations and Trends® in Machine Learning
Volume 17, Issue 6, 2024

Editorial Scope

Foundations and Trends® welcomes monographs that touch on fundamental problems in machine learning from theoretical, methodological, and/or computational perspectives. We are particularly interested in monographs that seek to bridge such problems and perspectives with those from related fields, including (but not limited to) statistics, economics, and optimization.

Information for Librarians

Foundations and Trends® in Machine Learning, 2024, Volume 17, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Contents

An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes

George H. Chen

Carnegie Mellon University, USA; georgechen@cmu.edu

ABSTRACT

Many applications involve reasoning about time durations before a critical event happens—also called *time-to-event outcomes*. When will a customer cancel a subscription, a coma patient wake up, or a convicted criminal reoffend? Accurate predictions of such time durations could help downstream decision-making tasks. A key challenge is *censoring*: commonly, when we collect training data, we do not get to observe the time-to-event outcome for every data point. For example, a coma patient has not woken up yet, so we do not know the patient's time until awakening. However, these data points should not be excluded from analysis as they could have characteristics that explain why they have yet to or might never experience the event.

Time-to-event outcomes have been studied extensively within the field of *survival analysis* primarily by the statistical, medical, and reliability engineering communities, with textbooks already available in the 1970s and '80s. Recently, the machine learning community has made significant methodological advances in survival analysis that take advantage of the representation learning ability of deep neural

---

George H. Chen (2024), "An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes", Foundations and Trends® in Machine Learning: Vol. 17, No. 6, pp 921–1100. DOI: 10.1561/2200000114.

networks. At this point, there is a proliferation of deep survival analysis models. How do these models work? Why? What are the overarching principles in how these models are generally developed? How are different models related?

This monograph aims to provide a reasonably self-contained modern introduction to survival analysis. We focus on predicting time-to-event outcomes at the individual data point level with the help of neural networks. Our goal is to provide the reader with a working understanding of precisely what the basic time-to-event prediction problem is, how it differs from standard regression and classification, and how key "design patterns" have been used time after time to derive new time-to-event prediction models, from classical methods like the Cox proportional hazards model to modern deep learning approaches such as deep kernel Kaplan-Meier estimators and neural ordinary differential equation models. We further delve into two extensions of the basic time-to-event prediction setup: predicting which of several critical events will happen first along with the time until this earliest event happens (the competing risks setting), and predicting time-to-event outcomes given a time series that grows in length over time (the dynamic setting). We conclude with a discussion of a variety of topics such as fairness, causal reasoning, interpretability, and statistical guarantees.

Our monograph comes with an accompanying code repository that implements every model and evaluation metric that we cover in detail: https://github.com/georgehc/survival-intro

1

Introduction

Predicting time durations before a critical event happens arises in numerous applications. These durations are called *time-to-event outcomes*. For example, an e-commerce company may be interested in predicting a user's time until making a purchase (*e.g.*, Chapelle 2014). A video streaming service may be interested in predicting when a customer will stop watching a show (*e.g.*, Hubbard *et al.* 2021). In healthcare, hospitals may be interested in predicting when a patient's disease will relapse (*e.g.*, Zupan *et al.* 2000). In criminology, courts may be interested in predicting the time until a convicted criminal reoffends (*e.g.*, Chung *et al.* 1991). Accurate predictions for these time-to-event outcomes could help in decision-making tasks such as showing targeted advertisements or promotions in the e-commerce or video streaming examples, planning treatments to reduce a patient's risk of disease relapse in the healthcare example, and making bail decisions in the criminology example.

A defining feature of time-to-event prediction problems is that commonly, when we collect training data to learn a model from, we do not get to see the true time-to-event outcome for every data point, *i.e.*, their time-to-event outcome is *censored*. As an example, some training points (*e.g.*, a coma patient) might not have experienced the critical event of interest yet (*e.g.*, waking up). Discarding the points that have not experienced the event would be unwise: they could have characteristics that make them much less likely to experience the event (*e.g.*, the patient's brain activity is highly abnormal).

1.1 Survival Analysis and Time-to-Event Outcomes: Some History and Commentary on Naming

Time-to-event outcomes have been studied for hundreds of years if not longer, where the initial focus was on predicting time until death. Early analyses introduced the use of "life tables", which in a nutshell contain counts such as numbers of births and deaths over time. Graunt (1662) published what might be the first life table and looked at the chance of survival for different age groups.[1] This particular dataset from London was challenging since the survival times (age at the time of death) were not actually recorded, corresponding to a censoring problem. Instead, Graunt largely guessed survival times based on causes of death, which were recorded (albeit they were not necessarily accurate). A few decades later, Halley (1693) analyzed a life table collected from modern day Wrocław and computed the probability of dying within the next year. Halley used these probabilities to determine how to price an annuity (roughly, an expected payout over a person's remaining lifetime). For a historical account of life tables and more generally reasoning about survival times, see for instance the book by Bacaër (2011) and the bibliographical notes accompanying the different chapters of Namboodiri and Suchindran (2013)—these readings altogether walk through highlights from hundreds of years of research on survival times leading up to modern time.

So much of the pioneering research on time-to-event outcomes was on time until death that the enterprise of modeling time-to-event outcomes is now commonly called *survival analysis*, with textbooks already available decades ago (*e.g.*, Mann *et al.* 1974; Kalbfleisch and Prentice

[1] As noted by Glass (1963) and Bacaër (2011) among others, there has been some debate as to whether Graunt or his friend William Petty wrote the book but regardless, Graunt's book had five editions published between 1662 and 1676 (for which our citation just uses the earliest year).

1980; Cox and Oakes 1984; Fleming and Harrington 1991). Countless other (text)books on survival analysis have since been written and have mainly originated from statistical, medical, and reliability engineering communities (*e.g.*, Klein and Moeschberger 2003; Machin *et al.* 2006; Selvin 2008; Kleinbaum and Klein 2012; Li and Ma 2013; Harrell 2015; Klein *et al.* 2016; Ebeling 2019; Prentice and Zhao 2019; Gerds and Kattan 2021; Collett 2023), and at this point, there is also a book tailored to social scientists (Box-Steffensmeier and Jones, 2004).

We want to emphasize though that as the examples we opened the monograph with showed, the critical event need not be death, meaning that we might not be reasoning about "survival" literally. In fact, some researchers work on survival analysis but in titling their papers choose to opt for more general phrasing such as "time-to-event modeling" (*e.g.*, Chapfuwa *et al.* 2018). We further emphasize that the "time" in "time-to-event outcome" does not literally have to measure time. For example, a survival analysis model could be used to predict how many units of an inventory item (*e.g.*, a newspaper) to stock the next day given past days' sales counts, so that the "time-to-event outcome" here measures an integer number of items (Huh *et al.*, 2011). Ultimately, "survival" analysis or "time-to-event" models have been broadly applied to numerous applications far beyond reasoning about either "survival" or "time-to-event" outcomes in a literal sense.

1.2 Machine Learning Models for Survival Analysis

The phrase "machine learning" was only coined in 1959 (Samuel, 1959), the year after the highly influential paper by Kaplan and Meier (1958) came out that analyzed a survival model based on life tables using what is called the "product-limit" estimator (Böhmer, 1912). (Kaplan and Meier's estimator remains one of the major workhorses of modern time-to-event data analysis; we will see it and deep learning versions of it later in this monograph.) Suffice it to say, machine learning as a field is young compared to survival analysis. Precisely when the first machine learning survival analysis model came about is perhaps not entirely straightforward to trace, in part because nowadays, what is considered a "machine learning model" depends on who one asks. While we may consider k nearest neighbor and kernel survival analysis (Beran, 1981) and survival trees (Ciampi *et al.*, 1981; Gordon and Olshen, 1985) to be machine learning models, would the authors of these original papers?

Fast-forwarding to present time, there is now an explosion in the number of machine learning survival analysis models available. For much larger lists of models than what we cover in this monograph, see the excellent surveys by Wang *et al.* (2019) and Wiegrebe *et al.* (2023). As part of their survey, Wiegrebe *et al.* (2023) provide an online catalog of over 60 deep-learning-based survival models (which we will just abbreviate throughout this monograph as *deep survival models*).[2] This catalog is not exhaustive!

With so many machine learning models for survival analysis, what exactly are the major innovations? When and why do different models work? How do they relate to each other? What are overarching patterns in model development? In answering these questions, we think that it is extremely important to distinguish between innovations that are specific to time-to-event prediction vs ones that are not. For the purposes of this monograph, we want to focus on the former as they could help us better understand what is special about time-to-event prediction that helps us build better models.

1.3 The Motivation for This Monograph

We set out to write this monograph for two key reasons:

- First, we wanted to provide a reasonably self-contained introductory text that covers the key concepts of survival analysis with a focus on time-to-event prediction *at the individual data point level* and that also exploits the availability of now standard neural network software. We focus on neural network survival models (*i.e.*, deep survival models) because these models are easy to modify (*e.g.*, to accommodate different data modalities, add loss terms, set a custom learning rate schedule, *etc*). Note that every model that we present in detail has publicly available source code (we discuss software shortly in Section 1.6.4). For readers who are new

[2] https://survival-org.github.io/DL4Survival/

to survival analysis but are already very comfortable working with standard neural network software at the level of writing custom models and loss functions, we hope that our monograph provides enough survival analysis background to make implementing deep survival models from "scratch" using standard neural network software fairly straightforward.

- Second, we wanted to clearly convey how several major categories of deep survival models are related, and how in deriving these different survival models, we use some of the same key design patterns or derivation techniques over and over again. We hope that by leading the reader through many examples, these patterns will become apparent.

To the best of our knowledge, no existing text provides the sort of introduction to survival analysis that our monograph aims to be. The surveys of machine learning survival models (Wang *et al.*, 2019; Wiegrebe *et al.*, 2023) are not written nor intended for the purpose of giving the reader a working knowledge of how to actually derive survival models from first principles. Meanwhile, the vast majority of survival analysis (text)books do not cover neural networks or deep learning due to how new these are (an example of a textbook that covers neural networks for survival analysis can be found in Chapter 11 of Dybowski and Gant (2001), but this book pre-dates the invention of nearly all the deep survival models we cover).

Overall, we hope that our introduction to survival analysis provides the reader with a solid understanding of what precisely the time-to-event problem setup is, why it is different from standard regression and classification, and how to build survival models with the help of neural networks. We also hope that the reader learns a little bit about where the state-of-the-art is in terms of a variety of other topics that we mention but do not discuss in detail, such as how fairness, causal reasoning, and interpretability play into survival models, and what progress has been made on theoretically analyzing some of these models.

1.4 Monograph Overview and Outline

Our coverage is not meant to remotely be exhaustive in showcasing how deep survival models have been used for time-to-event prediction. We specifically cover the following:

- **Basic Time-to-Event Prediction Setup (Section 2).** We first go over the standard time-to-event prediction problem setup. We state its statistical framework, its prediction task, common ways of writing a likelihood function to be maximized (maximum likelihood is the standard way of learning time-to-event prediction models), and how to evaluate prediction accuracy. Along the way, we lead the reader through various example models to help solidify concepts, all of which could be related to maximizing likelihood functions: exponential and Weibull time-to-event prediction models, DeepHit (Lee *et al.*, 2018), Nnet-survival (Gensheimer and Narasimhan, 2019), the Kaplan-Meier estimator (Kaplan and Meier, 1958), and the Nelson-Aalen estimator (Nelson, 1969; Aalen, 1978). Importantly, we distinguish between modeling time as continuous vs discrete since the math involved is a bit different. This section also discusses how time-to-event prediction relates to classification and regression.

- **Deep Proportional Hazards Models (Section 3).** We next cover perhaps the most widely used family of time-to-event prediction models in practice, which are called *proportional hazards models*. We define proportional hazards models in a general manner in terms of neural networks. Special cases include the exponential and Weibull models from Section 2, the classical Cox model (Cox, 1972), and DeepSurv (Faraggi and Simon, 1995; Katzman *et al.*, 2018). Proportional hazards models make a strong assumption that, in some sense, decouples how time contributes to a prediction and how a data point's features contribute to a prediction. This assumption often does not hold in practice. We present a generalization of the DeepSurv model called Cox-Time (Kvamme *et al.*, 2019) that removes this proportional hazards assumption.

- **Deep Conditional Kaplan-Meier Estimators (Section 4).** One of the standard models we encounter in Section 2 is the Kaplan-Meier estimator, which is extremely popular in practice and also different from deep proportional hazards models because it is *nonparametric* (*i.e.*, it does not assume the time-to-event outcome's distribution has a parametric form). However, it only works to describe a population and does not provide predictions for individual data points. We present deep learning versions of the Kaplan-Meier estimator that can make predictions at the individual level. Namely, we cover deep kernel survival analysis (Chen, 2020) and its generalization called survival kernets (Chen, 2024); the latter can scale to large datasets, can in some sense be interpreted in terms of clusters, and has a statistical guarantee on accuracy for a special case of the model.

- **Neural Ordinary Differential Equation Formulation of Time-to-Event Prediction (Section 5).** We then present a model that can encode all the models we presented in preceding sections, where we phrase the standard time-to-event prediction problem in terms of a *neural ordinary differential equation* model. We specifically go over the neural ODE time-to-event prediction model by Tang *et al.* (2022b) called SODEN. In presenting SODEN, we also mention some model classes that we did not previously point out, such as deep accelerated failure time models and deep extended hazard models (Zhong *et al.*, 2021).

- **Beyond the Basic Time-to-Event Prediction Setup: Multiple Critical Events and Time Series as Raw Inputs (Section 6).** Whereas all the previous sections used the basic time-to-event prediction setup from Section 2, we now consider two generalizations. First we consider the so-called *competing risks setting* where there are multiple critical events of interest (*e.g.*, for a coma patient, we consider the patient waking up and the patient dying as two different critical events; note that censoring could still happen but is not considered as one of the critical events). We aim to predict *which critical event will happen first* and also the *time until this earliest critical event happens*. The example model we use here is DeepHit (Lee *et al.*, 2018). Note that the special case of there being one critical event reduces the problem to the one from Section 2. We then generalize the competing risks setting further by considering what happens when we want to make predictions as we see more and more of a given a time series (the dynamic setting). The example model we use here is Dynamic-DeepHit (Lee *et al.*, 2019).

- **Discussion (Section 7).** We end the monograph by discussing a variety of topics that we either only briefly glossed over or that we did not mention at all. For example, we discuss different kinds of censoring, ways to encourage a survival model to be "fair", causal reasoning with survival models, interpretability of deep survival models, issues of statistical guarantees, and more.

Specifically for the example models we cover in Sections 2 to 5, we show how these models relate in Figure 1.1. When one model is a child of another in this figure, it means that the child model could be represented (possibly with a known approximation) by the parent model. Note that in the figure, just because two models do not overlap does *not* mean that they cannot represent the same underlying true time-to-event outcome distribution. For example, even though deep extended hazard models (Zhong *et al.*, 2021) and survival kernets (Chen, 2024) do not overlap in the figure, they can represent many of the same time-to-event outcome distributions.

We emphasize that just because SODEN (Tang *et al.*, 2022b) can in principle represent all the other models we cover in Sections 2 to 4 (possibly with an approximation), that does not mean that one is best off just using SODEN. An important point is that many of these models are trained in different ways. SODEN's training procedure may not work the best for some of the simpler model classes that it can represent. In particular, it invokes calls to an ordinary differential equation (ODE) solver, which could be overkill if we just want to use one of the simpler models (that has its own simpler training procedure which typically is faster to run). By relying on an ODE solver, we could also run into numerical stability issues that occasionally arise with ODE solvers.

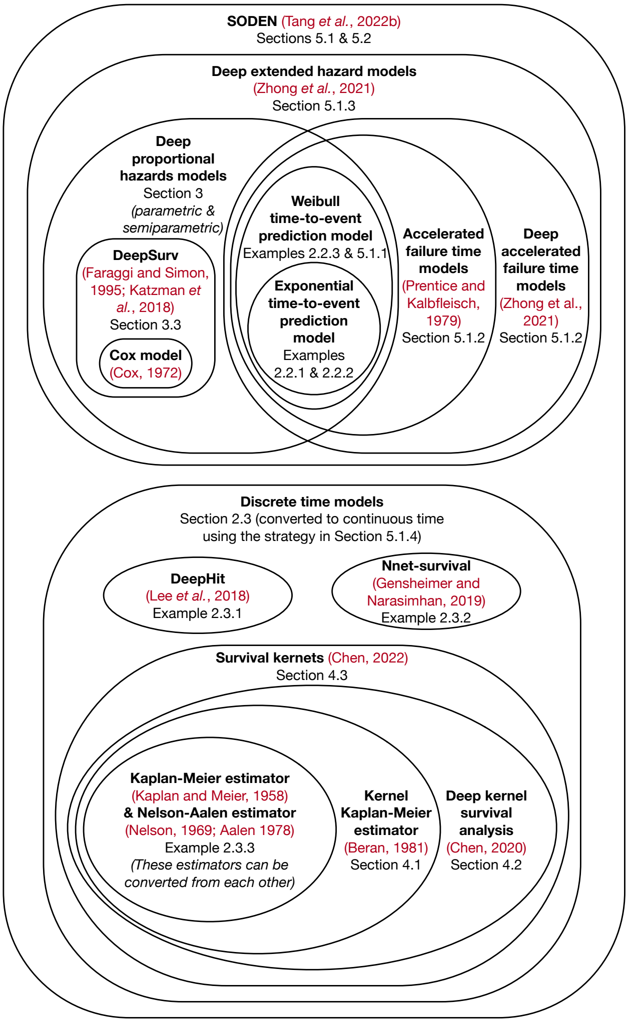

Figure 1.1: An overview of the models we cover in detail in Sections 2 to 5. One model being the child of another means that the child model could be represented (possibly with a known approximation) by the parent model. Note that when interpreting this diagram, two non-overlapping models could still possibly represent the same underlying time-to-event outcome distribution. For example, deep extended hazard models (Zhong *et al.*, 2021) and survival kernets (Chen, 2024) are capable of modeling many of the same time-to-event outcome distributions. Note that we also cover Cox-Time (Kvamme *et al.*, 2019), which does not easily fit in the diagram; Cox-Time is a generalization of the semiparametric model called DeepSurv (Faraggi and Simon, 1995; Katzman *et al.*, 2018), but Cox-Time can also represent models that are not deep extended hazard models.

Separately, it is good to keep in mind that deep survival models that we cover allow the modeler to flexibly choose a "base" neural network to use with the model. For example, when working with tabular data, the modeler could choose the base neural network to be a multilayer perceptron. When working with images, the modeler could choose the base neural network to be a convolutional neural network or a vision transformer. And so forth. Such base neural networks could be chosen to be arbitrarily complicated (*e.g.*, we could use as many layers and as many hidden nodes as we would like). Consequently, many deep survival models could in theory be considered equally expressive in what sorts of time-to-event outcomes they can model. However, in practice, training these deep models often requires using standard neural network tricks such as using early stopping, weight decay, dropout, *etc.* Roughly, we would train these models with some regularization to prevent overfitting, and how this regularization impacts different models then depends on their different modeling assumptions.

We remark that the example models we chose to present in this monograph are not necessarily the "best". Instead, they were chosen largely for pedagogical considerations and also to showcase some classes of models that are quite different from one another. There are plenty of other time-to-event prediction models (deep-learning-based or not) that work well! By understanding the fundamental concepts in our monograph, the reader should be well-equipped to understand many of these other models. As a reminder, the surveys by Wang *et al.* (2019) and Wiegrebe *et al.* (2023) provide fairly extensive listings of many existing machine learning models for time-to-event prediction.

1.5 Examples of Topics Beyond the Scope of Our Monograph

To elaborate a bit more on our scope of coverage, our monograph focuses on survival models for *prediction* that are estimated via *maximum likelihood estimation* in a *neural network framework*. We acknowledge that many survival analysis methods were originally derived for the purpose of statistical inference (*e.g.*, reasoning about population-level quantities and constructing confidence intervals for these quantities) rather than for prediction, including the classical Kaplan-Meier estimator (Kaplan and Meier, 1958) and the Cox model (Cox, 1972). Our emphasis in this monograph, however, is on learning survival models for prediction, as this is what neural survival models currently are well-suited for. As such, we typically will not cover how to address questions of statistical inference. For readers interested in learning more about statistical inference with survival models, we point out that the various (text)books mentioned in Section 1.1 cover statistical inference results for classical models.[3]

Next, many survival models are learned in a manner that is fundamentally not based on maximum likelihood estimation. For example, countdown regression (Avati *et al.*, 2020) defines a score function to optimize that does not correspond to the usual survival likelihood used in the literature. Meanwhile, Chapfuwa *et al.* (2018) train a survival model using adversarial learning (with a generative adversarial network, Goodfellow *et al.*, 2014) rather than maximum likelihood. As another example, random survival forests (Ishwaran *et al.*, 2008) are trained in a greedy manner, where one cannot easily write down a global objective function that is being optimized.

In fact, many decision tree survival models are not optimized in a neural network framework at all, such as the aforementioned random survival forests (Ishwaran *et al.*, 2008) as well as XGBoost (Chen and Guestrin, 2016) (note that the official implementation of XGBoost supports survival analysis), optimal survival trees (Bertsimas *et al.*, 2022), or optimal sparse survival trees (Zhang *et al.*, 2024). While it is possible to set up learning a decision tree survival model in a neural network framework (Sun and Qiu, 2023), at the time of writing, this line of research appears to still be in early development.

For ease of exposition, we do not cover latent variable models for survival analysis (*e.g.*, Nagpal *et al.* 2021a; Nagpal *et al.* 2021b; Manduchi *et al.* 2022; Moon *et al.* 2022; Chen *et al.* 2024). These particular models build on the ideas we present in this monograph and further use tools not covered in this monograph, notably that of

[3]As a concrete example, the textbook by Klein and Moeschberger (2003) routinely explains how to construct confidence intervals for various estimated quantities, such as confidence intervals for survival functions obtained from the Kaplan-Meier estimator (see Section 4.3 of their book) and the Cox model (see Section 8.8 of their book).

variational inference (*e.g.*, Blei *et al.* 2017). We think that a reader who understands the fundamentals of our monograph and of variational inference should be well-versed in understanding latent variable models for survival analysis.

1.6 Preliminaries

Before we move onto other sections, we go over some prerequisite knowledge that we will assume that the reader is familiar with. We then explain how we view neural networks and the notation that we use throughout the rest of the monograph. At the end of this section, we provide links to some available software packages and to our companion code repository for this monograph.

1.6.1 Prerequisites

We assume that the reader knows calculus, introductory probability and statistics, and the basics of machine learning, especially neural networks, including how to code them up and optimize them in standard neural network software (*e.g.*, PyTorch, Paszke *et al.*, 2019, TensorFlow, Abadi *et al.*, 2015, JAX, Bradbury *et al.*, 2018). For example, we assume that the reader knows how to run minibatch gradient descent using a standard neural network optimizer (*e.g.*, Adam, Kingma and Ba, 2015). For a primer on neural networks, see, for instance, the interactive textbook *Dive into Deep Learning* by Zhang *et al.* (2023).[4]

In terms of neural network architectures that the reader should already know to understand our monograph, we have intentionally tried to keep this listing short:

- (Sections 2 to 5 and the first half of Section 6) The reader should know multilayer perceptrons for classification and regression (corresponding to the case where raw input data are fixed-length feature vectors). For example, the reader should know that softmax activation yields a probability distribution, and that the function defined by an inner product $\mathbf{f}(x;\theta) := x^\top\theta$ for $x, \theta \in \mathbb{R}^d$

[4] https://D2L.ai

is a special case of a multilayer perceptron. (Note that we present the material in a general manner where the raw inputs need not be fixed-length feature vectors.)

- (Second half of Section 6) In the latter half of Section 6, in addition to multilayer perceptrons, the reader should also know recurrent neural networks (RNNs). RNNs enable us to work with variable-length time series as raw inputs.[5]

1.6.2 How We View Neural Networks

As we mentioned in Section 1.4, deep survival models that we cover all depend on a base neural network. By analogy, if we were tackling a classification problem with k classes using deep learning, then the standard strategy is to specify a base neural network (such as a multilayer perceptron) and then we feed the output of the base neural network to a linear layer (also called a full-connected layer or a dense layer) with k output nodes and softmax activation (so that the output of the overall network consists of predicted probabilities of the k classes).[6] Then when we learn the network, we use a classification loss function (*e.g.*, cross entropy loss). The final linear layer added with k output nodes and softmax activation is referred to as a "prediction head". If instead of classification, we were looking at a regression problem (predicting a single real number), then we could set the prediction head to be a linear layer with 1 output node and no nonlinear activation.

When working with deep survival models for time-to-event prediction, the idea is the same. We first specify a base neural network. Afterward, to get the overall network to predict a time-to-event outcome, it is as simple as choosing a "survival layer" at the end (to serve as the prediction head) and using an appropriate survival loss. Depending on the survival layer chosen, there are restrictions on what the output of the base neural network is. For example, when we cover the Cox

[5]While we do not explicitly cover nor assume that the reader knows transformers, we point out that transformers can also handle variable-length inputs (so that in our coverage, RNNs can actually be replaced by transformers).

[6]This strategy would require the base neural network to output some number of nodes that should be at least k (if it is less than k, then we would have trouble representing all k classes).

proportional hazards model, we will see that the base neural network should be set to output a single real number (which could be interpreted as a risk score), and there is actually no additional layer to add. The loss function is then specified a particular way for model training using these "risk scores".

Every survival model we cover could be thought of as a different possible survival layer to use as the prediction head. Each model comes with a loss function. For the models we cover, the loss function will always be a negative log likelihood loss with possibly some other loss terms added, depending on the model.

Extremely importantly, we will typically not spell out details of how to set the base neural network aside from what we require of its output, meaning that we usually intentionally leave the specific architecture choices up to the modeler. We do this precisely because standard tricks can be used for how to choose the base neural network (as we mentioned above, we could choose a multilayer perceptron when working with tabular data, a convolutional neural network or a vision transformer when working with images, *etc*). This also means that advances in neural network technology that are not specific to time-to-event prediction could also trivially be incorporated. For example, if we were to work with multimodal data such as the raw inputs being both images and text, then we could choose the base neural network to be based off a model such as CLIP (Radford *et al.*, 2021). *An important implication is that when we cover an existing deep survival model in detail, even if the original authors of the model provided architecture details in their paper, we omit the architecture details that are not essential to understanding the design of their overall model.*

Another reason why we do not state very specific neural network architectures to use is because the technology has rapidly been changing! The latest trends in neural network architectures today might be out of fashion tomorrow. To complicate matters, depending on the dataset used in a time-to-event prediction task, which specific architecture works the best might vary, and also which neural network optimizer we should use and with what learning schedule might also vary. Our monograph does not dwell on these engineering details, which are important in practice but are not needed in understanding the core high-level concepts.

1.6.3 Notation

We typically use uppercase letters (*e.g.*, X) to denote random variables and lowercase letters (*e.g.*, x) to denote deterministic quantities such as constants or specific realized values of random variables. Functions could either be uppercase or lowercase, where we have tried to stick to common conventions used in survival analysis literature (*e.g.*, the so-called conditional survival function is represented by uppercase S). Bold letters (*e.g.*, $\mathbf{f}$) are usually used to represent parametric functions such as neural networks. We also frequently use the notation $[m] := \{1, 2, \ldots, m\}$, where m is a positive integer. When we use the "log" function, we always mean natural log.

Optimization problems regularly appear in the monograph. When we write $\widehat{\theta} := \arg\min_\theta \mathbf{L}(\theta)$, where $\mathbf{L}$ is a loss function with parameter variable θ, this minimization would be carried out using (some variant of) minibatch gradient descent and, technically, we are usually not finding a solution that achieves the global minimum.

1.6.4 Software Packages and Datasets

As our exposition assumes that the reader is familiar with standard neural network software that have developer communities that primarily work in Python, we list some Python survival analysis packages in Table 1.1. This list is not exhaustive. We list packages for both deep and non-deep survival models since we think that trying both is important in practice. Per package, we list some (not all) of the models and evaluation metrics implemented. We anticipate that over time, many of these packages will add functionality. Overall, the current state of software packages that support deep survival models is a bit scattered: no single package is—in our opinion—sufficiently comprehensive, and at the time of writing, some packages have not been regularly maintained.

Currently, the packages in Table 1.1 do not implement all the models that we cover in detail. SODEN (Tang *et al.*, 2022b), deep kernel survival analysis (Chen, 2020), survival kernets (Chen, 2024), and Dynamic-DeepHit (Lee *et al.*, 2019) are not currently included in the software packages in Table 1.1, but their code is available from the original authors; see the links in Table 1.2.

Table 1.1: Some software packages used for survival analysis/time-to-event prediction. Models in blue and evaluation metrics in red are ones that we cover in detail in this monograph.

Package	Link	Some supported methods (not exhaustive)
`scikit-survival` (Pölsterl, 2020)	https://github.com/sebp/scikit-survival	Kaplan-Meier estimator[1], Nelson-Aalen estimator[2], Cox model[3], various survival tree ensemble methods including random survival forests[4], concordance index[5], time-dependent concordance index (truncated)[6], time-dependent AUC[7], Brier score[8]
`lifelines` (Davidson-Pilon, 2019)	https://github.com/CamDavidsonPilon/lifelines	Kaplan-Meier estimator[1], Nelson-Aalen estimator[2], Cox model[3] and regularized variants, accelerated failure time (AFT) models[9], concordance index[5]
`xgboost` (Chen and Guestrin, 2016)	https://github.com/dmlc/xgboost	XGBoost supports using Cox and accelerated failure time loss functions
`glmnet_python` (Simon *et al.*, 2011)	https://github.com/bbalasub1/glmnet_python	Cox model[3] and regularized variants; this is the official port of `glmnet` from R
`pycox` (Kvamme *et al.*, 2019)	https://github.com/havakv/pycox	unified PyTorch implementations of DeepSurv[10], Cox-Time[11], Nnet-survival[12], DeepHit[13], N-MTLR[14], time-dependent concordance index (not truncated)[15], Brier score[8]
`pysurvival` (Fotso *et al.*, 2019)	https://github.com/square/pysurvival	N-MTLR implementation by original author[14], random survival forests[4]
`auton-survival` (Nagpal *et al.*, 2022b)	https://github.com/autonlab/auton-survival	DeepSurv[10], Deep Survival Machines[16], Deep Cox Mixtures[17]
`SurvivalEVAL` (Qi *et al.*, 2024a)	https://github.com/shi-ang/SurvivalEVAL	concordance index[5], Brier score[8], D-calibration[18], margin[18] and pseudo-observation[19] MAE scores
`torchsurv` (Monod *et al.*, 2024)	https://github.com/Novartis/torchsurv	Cox model[3], Weibull AFT model[9], concordance index[5], time-dependent AUC[7], Brier score[8]

[1]Kaplan and Meier (1958) [2]Nelson (1969) and Aalen (1978) [3]Cox (1972) [4]Ishwaran *et al.* (2008)
[5]Harrell *et al.* (1982) [6]Uno *et al.* (2011) [7]Uno *et al.* (2007) and Hung and Chiang (2010)
[8]Graf *et al.* (1999) [9]Prentice and Kalbfleisch (1979) [10]Faraggi and Simon (1995) and Katzman *et al.* (2018)
[11]Kvamme *et al.* (2019) [12]Gensheimer and Narasimhan (2019) [13]Lee *et al.* (2018) [14]Fotso (2018)
[15]Antolini *et al.* (2005) [16]Nagpal *et al.* (2021a) [17]Nagpal *et al.* (2021b) [18]Haider *et al.* (2020)
[19]Qi *et al.* (2023)

Table 1.2: Some models that we cover that are not currently implemented in the packages in Table 1.1.

Model	Link
Deep kernel survival analysis (Chen, 2020)	`https://github.com/georgehc/dksa`
Survival kernets (Chen, 2024)	`https://github.com/georgehc/survival-kernets`
SODEN (Tang *et al.*, 2022b)	`https://github.com/jiaqima/SODEN`
Dynamic-DeepHit (Lee *et al.*, 2019)	`https://github.com/chl8856/Dynamic-DeepHit`

In terms of publicly available survival datasets, the `pycox` software package comes with datasets that are all sufficiently large for learning neural network models (mostly in the thousands of data points along with one dataset with roughly 3 million points). The `scikit-survival` and `lifelines` packages also come with datasets; some are a bit small though (a few hundred or fewer points).

Companion code repository. To help readers with starting to work with deep survival analysis models in Python, we provide Python code that accompanies our monograph in the following code repository:

`https://github.com/georgehc/survival-intro`

This repository includes sample code for every model and every evaluation metric that we discuss in detail. Our code shows how to train different deep survival models, use them to predict time-to-event outcomes, and evaluate the quality of the predictions using some standard evaluation metrics. Our code is primarily in the form of Jupyter notebooks, which include a mix of code cells and explanations for different parts of the code. As we progress through the monograph, we point to specific Jupyter notebooks in our code repository for readers interested in seeing how concepts we cover get translated into code.

Our code has been written with pedagogy in mind. We stick to using standard PyTorch conventions, and we have written our notebooks at a level that exposes the main neural net optimization loop (minibatch gradient descent) and highlights where base neural networks appear in various deep survival models. Our code aims to make various preprocessing and model training steps more transparent, so that if one wants to modify any part of these, doing so should be straightforward.

Moreover, for ease of exposition, our notebooks that accompany Sections 2 through 5 all use the same standard dataset SUPPORT (Knaus *et al.*, 1995), for which we predict the time until death of severely ill hospitalized patients with various diseases.[7] Our notebooks that accompany Section 6 use the PBC dataset (Fleming and Harrington, 1991), which is on predicting the time until death and the time until transplantation of patients with primary biliary cirrhosis of the liver; here, death and transplantation are viewed as competing events where we only observe whichever one happens first for a training patient (or alternatively, if neither has happened for a training patient, then we at least know the last check-up time with the patient).

Importantly, in our Jupyter notebooks, we do *not* extensively optimize hyperparameters for any particular deep survival model to try to push the prediction performance of the model to be as good as possible. Thus, the final evaluation scores obtained in our notebooks should not be interpreted as the best possible scores achievable by the different models we implement. Furthermore, our code is not written to be "production-grade" with, for instance, extensive sanity checks or unit tests.

Lastly, we anticipate occasionally updating our code notebooks to accommodate updates to software packages, to improve exposition or clarity, or to fix bugs that are discovered. The latest version will be available at the GitHub link provided above.

[7]For these particular code notebooks, we also provide an example of how to modify the code to work with different data, with the concrete example being training on the Rotterdam tumor bank dataset (Foekens *et al.*, 2000) and then testing on the German Breast Cancer Study Group dataset (Schumacher *et al.*, 1994); these two datasets are on predicting survival times of breast cancer patients.

References

Aalen, O. O. and S. Johansen. (1978). "An empirical transition matrix for non-homogeneous Markov chains based on censored observations". *Scandinavian Journal of Statistics*: 141–150.

Aalen, O. O. (1978). "Nonparametric inference for a family of counting processes". *The Annals of Statistics*. 6(4): 701–726.

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. URL: https://www.tensorflow.org/.

Allison, P. D. (1982). "Discrete-time methods for the analysis of event histories". *Sociological Methodology*. 13: 61–98.

Angelopoulos, A. N. and S. Bates. (2023). "A gentle introduction to conformal prediction and distribution-free uncertainty quantification". *Foundations and Trends® in Machine Learning*. 16(4): 494–591.

Antolini, L., P. Boracchi, and E. Biganzoli. (2005). "A time-dependent discrimination index for survival data". *Statistics in Medicine*. 24(24): 3927–3944.

Avati, A., T. Duan, S. Zhou, K. Jung, N. H. Shah, and A. Y. Ng. (2020). "Countdown regression: Sharp and calibrated survival predictions". In: *Uncertainty in Artificial Intelligence*. PMLR. 145–155.

Bacaër, N. (2011). *A Short History of Mathematical Population Dynamics*. Vol. 618. Springer.

Beran, R. (1981). "Nonparametric regression with randomly censored survival data". *Technical report, University of California, Berkeley*.

Bertsimas, D., J. Dunn, E. Gibson, and A. Orfanoudaki. (2022). "Optimal survival trees". *Machine Learning*. 111(8): 2951–3023.

Blanche, P., M. W. Kattan, and T. A. Gerds. (2019). "The c-index is not proper for the evaluation of-year predicted risks". *Biostatistics*. 20(2): 347–357.

Blanche, P., A. Latouche, and V. Viallon. (2013). "Time-dependent AUC with right-censored data: a survey". *Risk Assessment and Evaluation of Predictions*: 239–251.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe. (2017). "Variational inference: A review for statisticians". *Journal of the American Statistical Association*. 112(518): 859–877.

Boag, J. W. (1949). "Maximum likelihood estimates of the proportion of patients cured by cancer therapy". *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 11(1): 15–53.

Böhmer, P. E. (1912). "Theorie der unabhängigen Wahrscheinlichkeiten". In: *Rapports Memoires et Proces verbaux de Septieme Congres International dActuaires Amsterdam*. Vol. 2. 327–343.

Bommasani, R. *et al.* (2021). "On the opportunities and risks of foundation models". *arXiv preprint arXiv:2108.07258*.

Box-Steffensmeier, J. M. and B. S. Jones. (2004). *Event History Modeling: A Guide for Social Scientists*. Cambridge University Press.

Bradbury, J., R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: http://github.com/jax-ml/jax.

Breslow, N. (1972). Discussion of the paper by D R Cox (1972). *Journal of the Royal Statistical Society, Series B*. 34(2):216–217.

Brown, C. C. (1975). “On the use of indicator variables for studying the time-dependence of parameters in a response-time model”. *Biometrics*. 31(4): 863–872.

Buolamwini, J. and T. Gebru. (2018). “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on Fairness, Accountability and Transparency*. PMLR. 77–91.

Candès, E., L. Lei, and Z. Ren. (2023). “Conformalized survival analysis”. *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 85(1): 24–45.

Chagny, G. and A. Roche. (2014). “Adaptive and minimax estimation of the cumulative distribution function given a functional covariate”. *Electronic Journal of Statistics*. 8(2): 2352–2404.

Chapelle, O. (2014). “Modeling delayed feedback in display advertising”. In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1097–1105.

Chapfuwa, P., S. Assaad, S. Zeng, M. J. Pencina, L. Carin, and R. Henao. (2021). “Enabling counterfactual survival analysis with balanced representations”. In: *Proceedings of the Conference on Health, Inference, and Learning*. 133–145.

Chapfuwa, P., C. Li, N. Mehta, L. Carin, and R. Henao. (2020). “Survival cluster analysis”. In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. 60–68.

Chapfuwa, P., C. Tao, C. Li, C. Page, B. Goldstein, L. C. Duke, and R. Henao. (2018). “Adversarial time-to-event modeling”. In: *International Conference on Machine Learning*. PMLR. 735–744.

Chen, G. H. (2019). “Nearest neighbor and kernel survival analysis: Nonasymptotic error bounds and strong consistency rates”. In: *International Conference on Machine Learning*. PMLR. 1001–1010.

Chen, G. H. (2020). “Deep kernel survival analysis and subject-specific survival time prediction intervals”. In: *Machine Learning for Healthcare Conference*. PMLR. 537–565.

Chen, G. H. (2023). “A General Framework for Visualizing Embedding Spaces of Neural Survival Analysis Models Based on Angular Information”. In: *Conference on Health, Inference, and Learning*. PMLR. 440–476.

Chen, G. H. (2024). “Survival Kernets: Scalable and Interpretable Deep Kernel Survival Analysis with an Accuracy Guarantee”. *Journal of Machine Learning Research*. 25(40): 1–78.

Chen, G. H., L. Li, R. Zuo, A. Coston, and J. C. Weiss. (2024). “Neural topic models with survival supervision: Jointly predicting time-to-event outcomes and learning how clinical features relate”. *Artificial Intelligence in Medicine*.

Chen, G. H. and D. Shah. (2018). “Explaining the Success of Nearest Neighbor Methods in Prediction”. *Foundations and Trends® in Machine Learning*. 10(5-6): 337–588.

Chen, R. T., Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. (2018). “Neural ordinary differential equations”. In: *Advances in Neural Information Processing Systems*.

Chen, T. and C. Guestrin. (2016). “XGBoost: A scalable tree boosting system”. In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

Chilinski, P. and R. Silva. (2020). “Neural likelihoods via cumulative distribution functions”. In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 420–429.

Chung, C.-F., P. Schmidt, and A. D. Witte. (1991). “Survival analysis: A survey”. *Journal of Quantitative Criminology*. 7: 59–98.

Ciampi, A., R. S. Bush, M. Gospodarowicz, and J. E. Till. (1981). “An approach to classifying prognostic factors related to survival experience for non-Hodgkin’s lymphoma patients: Based on a series of 982 patients: 1967–1975”. *Cancer*. 47(3): 621–627.

Collett, D. (2023). *Modelling Survival Data in Medical Research, Fourth Edition*. Chapman and Hall/CRC.

Cover, T. and P. Hart. (1967). “Nearest neighbor pattern classification”. *IEEE Transactions on Information Theory*. 13(1): 21–27.

Cox, D. R. (1972). “Regression models and life-tables”. *Journal of the Royal Statistical Society: Series B*. 34(2): 187–202.

Cox, D. R. and D. Oakes. (1984). *Analysis of Survival Data*. CRC press.

Craig, E., C. Zhong, and R. Tibshirani. (2021). “Survival stacking: casting survival analysis as a classification problem”. *arXiv preprint arXiv:2107.13480*.

Cui, Y., M. R. Kosorok, E. Sverdrup, S. Wager, and R. Zhu. (2023). "Estimating heterogeneous treatment effects with right-censored data via causal survival forests". *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 85(2): 179–211.

Curth, A., C. Lee, and M. van der Schaar. (2021). "SurvITE: Learning heterogeneous treatment effects from time-to-event data". In: *Advances in Neural Information Processing Systems*.

Daley, D. J. and D. Vere-Jones. (2003). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer.

Daley, D. J. and D. Vere-Jones. (2008). *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Springer.

Damera Venkata, N. and C. Bhattacharyya. (2022). "When to Intervene: Learning Optimal Intervention Policies for Critical Events". In: *Advances in Neural Information Processing Systems*.

Danks, D. and C. Yau. (2022). "Derivative-based neural modelling of cumulative distribution functions for survival analysis". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 7240–7256.

Davidson-Pilon, C. (2019). "lifelines: survival analysis in Python". *Journal of Open Source Software*. 4(40): 1317.

Do, H., Y. Chang, Y. S. Cho, P. Smyth, and J. Zhong. (2023). "Fair Survival Time Prediction via Mutual Information Minimization". In: *Machine Learning for Healthcare Conference*.

Downey, A. B. (2011). *Think stats*. O'Reilly Media, Inc.

Du, N., H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. (2016). "Recurrent marked temporal point processes: Embedding event history to vector". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1555–1564.

Duchi, J., T. Hashimoto, and H. Namkoong. (2022). "Distributionally robust losses for latent covariate mixtures". *Operations Research*.

Duchi, J. C. and H. Namkoong. (2021). "Learning models with uniform performance via distributionally robust optimization". *The Annals of Statistics*. 49(3): 1378–1406.

Dybowski, R. and V. Gant. (2001). *Clinical Applications of Artificial Neural Networks*. Cambridge University Press.

Ebeling, C. E. (2019). *An Introduction to Reliability and Maintainability Engineering*. Waveland Press.

Ezquerro, A., B. Cancela, and A. López-Cheda. (2023). "On the Reliability of Machine Learning Models for Survival Analysis When Cure Is a Possibility". *Mathematics*. 11(19): 4150.

Faraggi, D. and R. Simon. (1995). "A neural network model for survival data". *Statistics in Medicine*. 14: 73–82.

Fine, J. P. and R. J. Gray. (1999). "A proportional hazards model for the subdistribution of a competing risk". *Journal of the American Statistical Association*. 94(446): 496–509.

Fleming, T. R. and D. P. Harrington. (1991). *Counting Processes and Survival Analysis*. John Wiley & Sons.

Foekens, J. A., H. A. Peters, M. P. Look, H. Portengen, M. Schmitt, M. D. Kramer, N. Brünner, F. Jänicke, M. E. Meijer-van Gelder, S. C. Henzen-Logmans, W. L. J. van Putten, and J. G. M. Klijn. (2000). "The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients". *Cancer Research*. 60(3): 636–643.

Földes, A. and L. Rejtö. (1981). "Strong uniform consistency for nonparametric survival curve estimators from randomly censored data". *The Annals of Statistics*. 9(1): 122–129.

Fornili, M., F. Ambrogi, P. Boracchi, and E. Biganzoli. (2014). "Piecewise exponential artificial neural networks (PEANN) for modeling hazard function with right censored data". In: *Computational Intelligence Methods for Bioinformatics and Biostatistics: 10th International Meeting, CIBB 2013, Nice, France, June 20-22, 2013, Revised Selected Papers 10*. Springer. 125–136.

Fotso, S. (2018). "Deep neural networks for survival analysis based on a multi-task framework". *arXiv preprint arXiv:1801.05512*.

Fotso, S. *et al.* (2019). PySurvival: Open source package for Survival Analysis modeling. URL: https://www.pysurvival.io/.

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

Gerds, T. A. and M. W. Kattan. (2021). *Medical Risk Prediction Models: With Ties to Machine Learning*. Chapman and Hall/CRC.

Glass, D. V. (1963). "John Graunt and his Natural and political observations". *Proceedings of the Royal Society of London. Series B, Biological Sciences*. 159(974): 2–37.

Gneiting, T. and A. E. Raftery. (2007). "Strictly proper scoring rules, prediction, and estimation". *Journal of the American Statistical Association*. 102(477): 359–378.

Goldstein, M., X. Han, A. Puli, A. Perotte, and R. Ranganath. (2020). "X-CAL: Explicit calibration for survival analysis". In: *Advances in Neural Information Processing Systems*.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems*.

Gordon, L. and R. A. Olshen. (1985). "Tree-structured survival analysis". *Cancer Treatment Reports*. 69(10): 1065–1069.

Graf, E. (1998). "Explained variation measures for survival data". *PhD thesis*.

Graf, E., C. Schmoor, W. Sauerbrei, and M. Schumacher. (1999). "Assessment and comparison of prognostic classification schemes for survival data". *Statistics in Medicine*. 18(17-18): 2529–2545.

Graunt, J. (1662). Natural and Political Observations Mentioned in a Following Index and Made upon the Bills of Mortality.

Gray, R. J. (1988). "A class of K-sample tests for comparing the cumulative incidence of a competing risk". *The Annals of Statistics*: 1141–1154.

Groha, S., S. M. Schmon, and A. Gusev. (2020). "A general framework for survival analysis and multi-state modelling". *arXiv preprint arXiv:2006.04893*.

Haider, H., B. Hoehn, S. Davis, and R. Greiner. (2020). "Effective Ways to Build and Evaluate Individual Survival Distributions". *Journal of Machine Learning Research*. 21(85): 1–63.

Halley, E. (1693). "An estimate of the degrees of the mortality of mankind; drawn from curious tables of the births and funerals at the city of Breslaw; with an attempt to ascertain the price of annuities upon lives". *Philosophical Transactions of the Royal Society of London*. 17: 596–610.

Harrell, F. E. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal regression, and Survival Analysis*. Spinger.

Harrell, F. E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. (1982). "Evaluating the yield of medical tests". *Journal of the American Medical Association*. 247(18): 2543–2546.

Harrell Jr, F. E., K. L. Lee, and D. B. Mark. (1996). "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors". *Statistics in Medicine*. 15(4): 361–387.

Hashimoto, T., M. Srivastava, H. Namkoong, and P. Liang. (2018). "Fairness without demographics in repeated loss minimization". In: *International Conference on Machine Learning*. PMLR. 1929–1938.

Hawkes, A. G. (1971). "Spectra of some self-exciting and mutually exciting point processes". *Biometrika*. 58(1): 83–90.

He, K., X. Zhang, S. Ren, and J. Sun. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 1026–1034.

Hochreiter, S. and J. Schmidhuber. (1997). "Long short-term memory". *Neural Computation*. 9(8): 1735–1780.

Hu, S. and G. H. Chen. (2024). "Fairness in Survival Analysis with Distributionally Robust Optimization". *Journal of Machine Learning Research*. 25(246): 1–85.

Hubbard, D., B. Rostykus, Y. Raimond, and T. Jebara. (2021). "Beta survival models". In: *Survival Prediction - Algorithms, Challenges and Applications*. PMLR. 22–39.

Huh, W. T., R. Levi, P. Rusmevichientong, and J. B. Orlin. (2011). "Adaptive data-driven inventory control with censored demand based on Kaplan-Meier estimator". *Operations Research*. 59(4): 929–941.

Hung, H. and C.-T. Chiang. (2010). "Estimation methods for time-dependent AUC models with survival data". *Canadian Journal of Statistics*. 38(1): 8–26.

Ishwaran, H., U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. (2008). "Random survival forests". *The Annals of Applied Statistics*. 2(3): 841–860.

Jeanselme, V., N. Agarwal, and C. Wang. (2024). "Review of Language Models for Survival Analysis". In: *AAAI 2024 Spring Symposium on Clinical Foundation Models*.

Jeanselme, V., B. Tom, and J. Barrett. (2022). "Neural Survival Clustering: Non-parametric mixture of neural networks for survival clustering". In: *Conference on Health, Inference, and Learning*. PMLR. 92–102.

Jeanselme, V., C. H. Yoon, B. Tom, and J. Barrett. (2023). "Neural Fine-Gray: Monotonic neural networks for competing risks". In: *Conference on Health, Inference, and Learning*. PMLR. 379–392.

Kalbfleisch, J. D. and R. L. Prentice. (1980). *The Statistical Analysis of Failure Time Data*. John Wiley & Sons.

Kaplan, E. L. and P. Meier. (1958). "Nonparametric estimation from incomplete observations". *Journal of the American Statistical Association*. 53(282): 457–481.

Katzman, J. L., U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger. (2018). "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network". *BMC Medical Research Methodology*. 18(24).

Keya, K. N., R. Islam, S. Pan, I. Stockwell, and J. Foulds. (2021). "Equitable allocation of healthcare resources with fair survival models". In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM. 190–198.

Kingma, D. P. and J. Ba. (2015). "Adam: A method for stochastic optimization". In: *International Conference for Learning Representations*.

Klein, J. P. and M. L. Moeschberger. (2003). *Survival Analysis: Techniques for Censored and Truncated Data, Second Edition*. Springer.

Klein, J. P., H. C. Van Houwelingen, J. G. Ibrahim, and T. H. Scheike. (2016). *Handbook of Survival Analysis*. CRC Press.

Kleinbaum, D. G. and M. Klein. (2012). *Survival Analysis: A Self-Learning Text, Third Edition*. Springer.

Knaus, W. A., F. E. Harrell, J. Lynn, L. Goldman, R. S. Phillips, A. F. Connors, N. V. Dawson, W. J. Fulkerson, R. M. Califf, N. Desbiens, P. Layde, R. K. Oye, P. E. Bellamy, R. B. Hakim, and D. P. Wagner. (1995). "The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults". *Annals of Internal Medicine*. 122(3): 191–203.

Kovalev, M. S., L. V. Utkin, and E. M. Kasimov. (2020). "SurvLIME: A method for explaining machine learning survival models". *Knowledge-Based Systems*.

Koziol, J. A. and Z. Jia. (2009). "The concordance index C and the Mann–Whitney parameter $\Pr(X > Y)$ with randomly censored data". *Biometrical Journal: Journal of Mathematical Methods in Biosciences*. 51(3): 467–474.

Kpotufe, S. and N. Verma. (2017). "Time-accuracy tradeoffs in kernel prediction: controlling prediction quality". *Journal of Machine Learning Research*.

Krzyziński, M., M. Spytek, H. Baniecki, and P. Biecek. (2023). "SurvSHAP(t): Time-dependent explanations of machine learning survival models". *Knowledge-Based Systems*.

Kvamme, H., Ø. Borgan, and I. Scheel. (2019). "Time-to-Event Prediction with Neural Networks and Cox Regression". *Journal of Machine Learning Research*. 20(129): 1–30.

Kvamme, H. and Ø. Borgan. (2021). "Continuous and discrete-time survival prediction with neural networks". *Lifetime Data Analysis*. 27(4): 710–736.

Lambert, J. and S. Chevret. (2016). "Summary measure of discrimination in survival models based on cumulative/dynamic time-dependent ROC curves". *Statistical Methods in Medical Research*. 25(5): 2088–2102.

Lee, C., J. Yoon, and M. Van Der Schaar. (2019). "Dynamic-DeepHit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data". *IEEE Transactions on Biomedical Engineering*. 67(1): 122–133.

Lee, C., W. Zame, J. Yoon, and M. Van Der Schaar. (2018). "DeepHit: A deep learning approach to survival analysis with competing risks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Lei, J., M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. (2018). "Distribution-free predictive inference for regression". *Journal of the American Statistical Association*. 113(523): 1094–1111.

Lei, J., A. Rinaldo, and L. Wasserman. (2015). "A conformal prediction approach to explore functional data". *Annals of Mathematics and Artificial Intelligence*. 74(1-2): 29–43.

Li, J. and S. Ma. (2013). *Survival Analysis in Medicine and Genetics*. CRC Press.

Li, M., H. Namkoong, and S. Xia. (2021). "Evaluating model performance under worst-case subpopulations". In: *Advances in Neural Information Processing Systems*.

Liu, W., R. Lin, Z. Liu, L. Xiong, B. Schölkopf, and A. Weller. (2021). "Learning with hyperspherical uniformity". In: *International Conference On Artificial Intelligence and Statistics*. PMLR. 1180–1188.

Lundberg, S. M. and S.-I. Lee. (2017). "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*.

Machin, D., Y. B. Cheung, and M. Parmar. (2006). *Survival Analysis: A Practical Approach*. John Wiley & Sons.

Manduchi, L., R. Marcinkevičs, M. C. Massi, T. Weikert, A. Sauter, V. Gotta, T. Müller, F. Vasella, M. C. Neidert, M. Pfister, B. Stieltjes, and J. E. Vogt. (2022). "A deep variational approach to clustering survival data". In: *International Conference on Learning Representations*.

Mann, N. R., R. E. Schafer, and N. D. Singpurwalla. (1974). *Methods for Statistical Analysis of Reliability and Life Data*. John Wiley & Sons.

Mantel, N. (1966). "Evaluation of survival data and two new rank order statistics arising in its consideration". *Cancer Chemotherapy Reports*. 50(3): 163–170.

Molnar, C. (2022). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. URL: https://christophm.github.io/interpretable-ml-book.

Monod, M., P. Krusche, Q. Cao, B. Sahiner, N. Petrick, D. Ohlssen, and T. Coroller. (2024). TorchSurv: A Lightweight Package for Deep Survival Analysis. DOI: https://doi.org/10.48550/arXiv.2404.10761.

Moon, I., S. Groha, and A. Gusev. (2022). "SurvLatent ODE: A Neural ODE based time-to-event model with competing risks for longitudinal data improves cancer-associated Venous Thromboembolism (VTE) prediction". In: *Machine Learning for Healthcare Conference*.

Nagpal, C., M. Goswami, K. Dufendach, and A. Dubrawski. (2022a). "Counterfactual phenotyping with censored time-to-events". In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3634–3644.

Nagpal, C., X. Li, and A. Dubrawski. (2021a). "Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks". *IEEE Journal of Biomedical and Health Informatics*. 25(8): 3163–3175.

Nagpal, C., W. Potosnak, and A. Dubrawski. (2022b). "auton-survival: An open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data". In: *Machine Learning for Healthcare Conference*. PMLR. 585–608.

Nagpal, C., S. Yadlowsky, N. Rostamzadeh, and K. Heller. (2021b). "Deep Cox mixtures for survival regression". In: *Machine Learning for Healthcare Conference*. PMLR. 674–708.

Namboodiri, K. and C. M. Suchindran. (2013). *Life Table Techniques and Their Applications*. Academic Press.

Nelson, W. (1969). "Hazard plotting for incomplete failure data". *Journal of Quality Technology*. 1: 27–52.

Papadopoulos, H., K. Proedrou, V. Vovk, and A. Gammerman. (2002). "Inductive confidence machines for regression". In: *European Conference on Machine Learning*. Springer. 345–356.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*.

Peng, Y. and B. Yu. (2021). *Cure Models: Methods, Applications, and Implementation*. CRC Press.

Pölsterl, S. (2020). "scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn". *Journal of Machine Learning Research*. 21(1): 8747–8752.

Prentice, R. L. and J. D. Kalbfleisch. (1979). "Hazard rate models with covariates". *Biometrics*: 25–39.

Prentice, R. L. and S. Zhao. (2019). *The Statistical Analysis of Multivariate Failure Time Data: A Marginal Modeling Approach*. CRC Press.

Putzel, P., H. Do, A. Boyd, H. Zhong, and P. Smyth. (2021). "Dynamic survival analysis for EHR data with personalized parametric distributions". In: *Machine Learning for Healthcare Conference*. PMLR. 648–673.

Qi, S.-A., N. Kumar, M. Farrokh, W. Sun, L.-H. Kuan, R. Ranganath, R. Henao, and R. Greiner. (2023). "An Effective Meaningful Way to Evaluate Survival Models". In: *International Conference on Machine Learning*. Vol. 202. PMLR. 28244–28276.

Qi, S.-a., W. Sun, and R. Greiner. (2024a). "SurvivalEVAL: A Comprehensive Open-Source Python Package for Evaluating Individual Survival Distributions". In: *Proceedings of the 2023 AAAI Fall Symposia*. Vol. 2. No. 1.

Qi, S.-a., Y. Yu, and R. Greiner. (2024b). "Conformalized Survival Distributions: A Generic Post-Process to Increase Calibration". In: *International Conference on Machine Learning*. Vol. 235. *Proceedings of Machine Learning Research*. PMLR. 41303–41339.

R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. (2021). "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 8748–8763.

Rahman, M. M. and S. Purushotham. (2022). "Fair and Interpretable Models for Survival Analysis". In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1452–1462.

Raykar, V. C., H. Steck, B. Krishnapuram, C. Dehing-oberije, and P. Lambin. (2007). "On Ranking in Survival Analysis: Bounds on the Concordance Index". In: *Advances in Neural Information Processing Systems*.

Ribeiro, M. T., S. Singh, and C. Guestrin. (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.

Rindt, D., R. Hu, D. Steinsaltz, and D. Sejdinovic. (2022). "Survival regression with proper scoring rules and monotonic neural networks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 1190–1205.

Samuel, A. L. (1959). "Some studies in machine learning using the game of checkers". *IBM Journal of Research and Development*.

Schumacher, M., G. Bastert, H. Bojar, K. Huebner, M. Olschewski, W. Sauerbrei, C. Schmoor, C. Beyerle, R. L. Neumann, and H. F. Rauschecker. (1994). "Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group." *Journal of Clinical Oncology*. 12(10): 2086–2093.

Selvin, S. (2008). *Survival Analysis for Epidemiologic and Medical Research*. Cambridge University Press.

Shchur, O., M. Biloš, and S. Günnemann. (2020). "Intensity-Free Learning of Temporal Point Processes". In: *International Conference on Learning Representations*.

Shen, X., J. Elmer, and G. H. Chen. (2023). "Neurological Prognostication of Post-Cardiac-Arrest Coma Patients Using EEG Data: A Dynamic Survival Analysis Framework with Competing Risks". In: *Machine Learning for Healthcare Conference*. PMLR. 667–690.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani. (2011). "Regularization paths for Cox's proportional hazards model via coordinate descent". *Journal of Statistical Software*. 39(5): 1.

Steinberg, E., J. A. Fries, Y. Xu, and N. Shah. (2024). "MOTOR: A Time-To-Event Foundation Model For Structured Medical Records". In: *International Conference on Learning Representations*.

Sun, X. and P. Qiu. (2023). "NSOTree: Neural Survival Oblique Tree". *arXiv preprint arXiv:2309.13825*.

Tang, W., K. He, G. Xu, and J. Zhu. (2022a). "Survival Analysis via Ordinary Differential Equations". *Journal of the American Statistical Association*.

Tang, W., J. Ma, Q. Mei, and J. Zhu. (2022b). "SODEN: A Scalable Continuous-Time Survival Model through Ordinary Differential Equation Networks". *Journal of Machine Learning Research*. 23(34): 1–29.

Tibshirani, R. (1997). "The lasso method for variable selection in the Cox model". *Statistics in Medicine*. 16(4): 385–395.

Tutz, G. and M. Schmid. (2016). *Modeling Discrete Time-to-Event Data*. Springer.

Uno, H., T. Cai, M. J. Pencina, R. B. D'Agostino, and L.-J. Wei. (2011). "On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data". *Statistics in Medicine*. 30(10): 1105–1117.

Uno, H., T. Cai, L. Tian, and L.-J. Wei. (2007). "Evaluating prediction rules for t-year survivors with censored regression models". *Journal of the American Statistical Association*. 102(478): 527–537.

Van der Maaten, L. and G. Hinton. (2008). "Visualizing data using t-SNE". *Journal of Machine Learning Research*. 9(11).

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods*. 17: 261–272. DOI: 10.1038/s41592-019-0686-2.

Vovk, V., A. Gammerman, and G. Shafer. (2005). *Algorithmic Learning in a Random World*. Springer Science & Business Media.

Wang, P., Y. Li, and C. K. Reddy. (2019). "Machine learning for survival analysis: A survey". *ACM Computing Surveys (CSUR)*. 51(6): 1–36.

Wang, T. and P. Isola. (2020). "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *International Conference on Machine Learning*.

Wiegrebe, S., P. Kopper, R. Sonabend, and A. Bender. (2023). "Deep Learning for Survival Analysis: A Review". *arXiv preprint arXiv:2305.14961*.

Xu, J. and Y. Peng. (2014). "Nonparametric cure rate estimation with covariates". *Canadian Journal of Statistics*. 42(1): 1–17.

Xu, Y., N. Ignatiadis, E. Sverdrup, S. Fleming, S. Wager, and N. Shah. (2023). "Treatment heterogeneity with survival outcomes". In: *Handbook of Matching and Weighting Adjustments for Causal Inference*. Chapman and Hall/CRC. 445–482.

Yanagisawa, H., K. Miyaguchi, and T. Katsuki. (2022). "Hierarchical lattice layer for partially monotone neural networks". In: *Advances in Neural Information Processing Systems*.

Yu, C.-N., R. Greiner, H.-C. Lin, and V. Baracos. (2011). "Learning patient-specific cancer survival distributions as a sequence of dependent regressors". In: *Advances in Neural Information Processing Systems*.

Zhang, A., Z. C. Lipton, M. Li, and A. J. Smola. (2023). *Dive into Deep Learning*. Cambridge University Press.

Zhang, Q., A. Lipani, O. Kirnap, and E. Yilmaz. (2020). "Self-attentive Hawkes process". In: *International Conference on Machine Learning*. PMLR. 11183–11193.

Zhang, R., R. Xin, M. Seltzer, and C. Rudin. (2024). "Optimal Sparse Survival Trees". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 352–360.

Zhang, W. and J. C. Weiss. (2022). "Longitudinal fairness with censorship". In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhong, Q., J. Mueller, and J.-L. Wang. (2022). "Deep learning for the partially linear Cox model". *The Annals of Statistics*. 50(3): 1348–1375.

Zhong, Q., J. W. Mueller, and J.-L. Wang. (2021). "Deep extended hazard models for survival analysis". In: *Advances in Neural Information Processing Systems*.

Zou, H. and T. Hastie. (2005). "Regularization and variable selection via the elastic net". *Journal of the Royal Statistical Society Series B*. 67(2): 301–320.

Zuo, S., H. Jiang, Z. Li, T. Zhao, and H. Zha. (2020). "Transformer Hawkes process". In: *International Conference on Machine Learning*. PMLR. 11692–11702.

Zupan, B., J. Demšar, M. W. Kattan, J. R. Beck, and I. Bratko. (2000). "Machine learning for survival analysis: a case study on recurrence of prostate cancer". *Artificial Intelligence in Medicine*. 20(1): 59–75.